

A Survey of Large Language Model Empowered Agents for Recommendation and Search: Towards Next-Generation Information Retrieval

YU ZHANG*, Tsinghua University, China

SHUTONG QIAO*, University of Queensland, Australia

JIAQI ZHANG, University of Queensland, Australia

TZU-HENG LIN, Tsinghua University, China

CHEN GAO, Tsinghua University, China

YONG LI, Tsinghua University, China

Information technology has profoundly altered the way humans interact with information. The vast amount of content created, shared, and disseminated online has made it increasingly difficult to access relevant information. Over the past two decades, search and recommendation systems (collectively referred to as information retrieval systems) have evolved significantly to address these challenges. Recent advances in large language models (LLMs) have demonstrated capabilities that surpass human performance in various language-related tasks and exhibit general understanding, reasoning, and decision-making abilities. This paper explores the transformative potential of large language model agents in enhancing search and recommendation systems. We discuss the motivations and roles of LLM agents, and establish a classification framework to elaborate on the existing research. We highlight the immense potential of LLM agents in addressing current challenges in search and recommendation, providing insights into future research directions. This paper is the first to systematically review and classify the research on LLM agents in these domains, offering a novel perspective on leveraging this advanced AI technology for information retrieval. To help understand the existing works, we list the existing papers on agent-based simulation with large language models at this link: <https://github.com/tsinghua-fib-lab/LLM-Agent-for-Recommendation-and-Search>.

Additional Key Words and Phrases: Large language model agent; recommender system; search system; information system

ACM Reference Format:

Yu Zhang, Shutong Qiao, Jiaqi Zhang, Tzu-Heng Lin, Chen Gao, and Yong Li. 2025. A Survey of Large Language Model Empowered Agents for Recommendation and Search: Towards Next-Generation Information Retrieval. 1, 1 (March 2025), 30 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

1 INTRODUCTION

Information technology has fundamentally transformed human life, particularly in how we interact with information. The development of the information age has led to the creation, sharing, and dissemination of vast amounts of content on the internet, making it increasingly challenging for individuals to access relevant information.

*Both authors contributed equally to this research.

Authors' addresses: Yu Zhang, Tsinghua University, Beijing, China, zhangyuthu14@gmail.com; Shutong Qiao, University of Queensland, Brisbane, Australia, shutong.qiao@uq.edu.au; Jiaqi Zhang, University of Queensland, Brisbane, Australia, jqzhang927@gmail.com; Tzu-Heng Lin, Tsinghua University, Beijing, China, lzhbrian@gmail.com; Chen Gao, Tsinghua University, Beijing, China, chgao96@gmail.com; Yong Li, Tsinghua University, Beijing, China, liyong07@tsinghua.edu.cn.

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

© 2025 Association for Computing Machinery.

XXXX-XXXX/2025/3-ART \$15.00

<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

Against this backdrop, the past two decades have witnessed the evolution of information systems, including search systems [10, 125] and Recommender Systems (RSs) [44, 174]. Typically, search systems passively retrieve relevant content from vast amounts of information with users' query words, while recommender systems actively generate a list of candidates that users might be interested in based on collected user behavioral data or profile information [2, 71, 114]. Some researchers consider recommendation as a specialized form of search, where the understanding of potential user needs is modeled as search terms [155].

Recently, large language models (LLMs) have achieved remarkable success and are considered one of the most viable paths toward general artificial intelligence [1]. Large language models have demonstrated capabilities similar to or even surpassing human performance, not just in language-related tasks, exhibiting general understanding, reasoning, and decision-making abilities [179]. Researchers have leveraged the text processing and commonsense reasoning capabilities of large models to conduct a series of search and recommendation tasks [3, 145, 191]. Furthermore, to overcome the limitations of large language models, researchers build various large language model agents [151]. Specifically, large language model agents use large language model as the core and are equipped with functions such as memory management, workflow, input-output interfaces, external tool invocation, and large-small model collaboration, making their intelligence levels closer to human and capable of handling more complex tasks [95, 178, 185].

For the crucial and fundamental tasks of search and recommendation, researchers increasingly recognize the high value of large language model agents. In this paper, we make the first attempt to provide a systematic and comprehensive review of the research efforts on search and recommendation with large language model agents. We first introduce relevant background knowledge and further discuss the motivations for utilizing LLM agents, as well as how LLM agents address those critical challenges faced by search and recommendation. Next, we establish a taxonomy, categorizing these recent advances according to the role of large language model agents in the overall search or recommender system. We then elaborate on these works and analyze how they construct and use LLM agents. It is worth mentioning there have already been several works discussing the application of LLMs in the fields of IR and search, which either focus solely on LLMs [8, 13, 43, 148] or a specific IR application [98], but this survey paper significantly differs by focusing on the application of LLM agents across the entire spectrum of IR functionalities.

The contributions of this paper can be summarized as follows:

- First, this paper is the first to review and organize the research on large language model agents in the research of search and recommendation, which is a new yet fast-growing research field.
- Second, we construct a taxonomy that well organizes the existing work by answering the fundamental question of why LLM agents are needed and how LLM agents enhance recommendation and search.
- Finally, we conduct open discussions about the unresolved challenges and important future research directions, which can inspire the following works in this area.

The structure of this paper is as follows: In Section 2, we introduce the background knowledge of recommender systems, search systems, large language models, and large language model agents. In Section 3, we analyze why large language model agents are necessary for search and recommendation. In Section 4, we first establish a coherent and comprehensive taxonomy system and then elaborate on the existing work. In Section 6, we systematically analyze the current problems and significant future directions for search and recommendation based on large language model agents. Finally, we conclude the paper in Section 7.

2 BACKGROUND

2.1 Recommendation and Search

Recommendation and search are pivotal challenges in information retrieval and have been extensively studied for decades. Despite their distinct application scenarios, they share similar core components (namely *Interaction*

Interface, User/Query Modeling, Item Modeling, and Matching, Ranking, and Re-ranking). This section introduces the background and concepts of each core component.

2.1.1 Interaction Interface. The rapid development of smart devices and mobile Internet and has exposed users to numerous applications employing recommender systems and search engines. The traditional interaction interface between users and search engines usually consists of a keyword query submission from users (e.g., on a search engine website or a project management software) and a one-round result retrieval from the search engines [7]. Interacting with a recommender system is even simpler, where users are mostly unconsciously browsing contents (e.g., on an e-commerce website, or a video sharing platform) provided by the recommender system [109]. Conversational recommender systems and search engines have recently gained attention [176]. In this interface, users interact through a chat interface using natural language, iteratively providing additional information or feedback based on previous results, forming a multi-turn dialogue. This conversational interface is more closely aligned with human communication methods and achieves more precise and satisfying results.

2.1.2 User/Query Modeling. For recommender systems and search engines to retrieve satisfying results, they must understand the users' intentions. For example, identifying the products a user wants to buy on an e-commerce app or determining relevant websites based on a keyword query. This requires precise modeling of user characteristics and input queries. In recommender systems, this includes utilizing the user profile (gender, age, preferences, etc.), browsing/clicking/purchasing history, and more [84]. In search engines, this involves text segmentation, named entity recognition, part-of-speech tagging, relation extraction, etc [23]. Advanced systems like multi-modal and conversational search engines also incorporate image recognition, multi-turn dialogue comprehension, and other algorithms [62, 134].

2.1.3 Item Modeling. Apart from user/query modeling, item modeling also plays an essential role in efficient and extensive retrieval. It gathers information of the items, and preprocesses them into some representation (e.g., a feature vector) for later usage. In traditional search engines, this process is known as *Indexing* [23], where significant keywords or tags from web page content are identified and stored beforehand for faster retrieval. Advanced search engines may also use auxiliary knowledge bases specific to certain domains. In recommender systems, item modeling considers item descriptions, attributes, images, buyers, and other related information [82, 84].

2.1.4 Matching, Ranking, and Re-ranking. To effectively retrieve accurate items according to the users/queries from a huge-size candidate item set (usually millions or more), a recommender system or a search engine typically comprises of three consecutive stages: *Matching*, *Ranking*, and *Re-ranking* [55].

- **Matching:** The *Matching* (or *Recall*) [22] stage retrieves potentially relevant hundreds of items from the full millions of candidates. In this stage, the algorithm is usually coarse (e.g., vector inner product) and highly efficient due to the huge-size input item set.
- **Ranking:** The *Ranking* [16] stage ranks the output of the *Matching* stage and selects the top ten of the items. With a smaller item set, this stage employs more complex models (e.g., deep neural networks) for higher accuracy.
- **Re-ranking:** The *Re-ranking* [97] stage further adjusts the final item list according to some additional requirements such as diversity, fairness, freshness, and business goals, etc.

By leveraging these stages, recommender systems and search engines ensure precise and relevant results.

2.2 Large Language Model and LLM Agents

2.2.1 Large Language Models. Language models play a pivotal role in Natural Language Processing (NLP), primarily used for assessing sentence probabilities and generating grammatically correct text. In its early stages,

the development of language models relied on statistical and probability theories such as n-grams, maximum entropy, and Hidden Markov Models, but was limited by its ability to handle long-range dependencies, known as the "curse of dimensionality." With the rise of deep learning technologies, neural network language models gradually became mainstream due to their robust generalization capabilities and flexible architectural designs. Models like Recurrent Neural Networks (RNNs) [119], Long Short-Term Memory networks (LSTMs) [47], and Gated Recurrent Units (GRUs) [19] effectively mitigated the challenges of processing long sequences by utilizing internal memory units, thereby better capturing contextual dependencies within sequences and significantly enhancing language model performance.

The introduction of the Transformer model [131] revolutionized the NLP field by replacing recurrent structures with self-attention mechanisms, thereby improving training speed and parallelism. Concurrently inspired by the pre-training and fine-tuning paradigm from computer vision, language models began leveraging large amounts of unlabeled data for pre-training, followed by task-specific fine-tuning, which gave rise to high-performance models such as BERT [30]. With the rapid expansion of training data and significant growth in computational resources, LLMs have established absolute dominance in the NLP. This wave not only symbolizes the comprehensive transformation of NLP from focusing on single-task optimization to multimodal perception and generation but also heralds language models reaching new heights in generation, understanding, and transformation. In the latest research, LLMs such as GPT-4 [1] Claude [6], etc. not only perform well in tasks across various NLP tasks such as text generation, multilingual translation, and sentiment analysis but also demonstrate impressive performance in tasks traditionally considered to require deep cognitive abilities like solving mathematical problems, logical reasoning, and strategic planning. This comprehensive performance marks the maturity of artificial intelligence technology, steadily advancing in a direction closer to Artificial General Intelligence (AGI).

2.2.2 LLM Agents. The concept of agent originated in the 1960s, aiming to develop software entities that can simulate human intelligence. By the 1970s and 1980s, the rise of expert systems [142] represented the ability of intelligent programs to solve professional problems based on fixed rules and knowledge bases. Although these systems were powerful, they were limited by static knowledge and rigid logic. In the late 1980s, researchers such as Michael Wooldridge redefined agents, emphasizing their autonomy, responsiveness to external events, and interaction with the environment and other agents. Minsky elaborated on the concept of agents in his work [88], describing them as individuals that can participate in social interactions and exhibit intelligent behavior. Minsky's idea that many relatively simple components (i.e., agents) in a complex system can solve complex problems by collaborating with each other later became the basis for research in Multi-Agent Systems (MAS) [34] and distributed artificial intelligence [9]. In the 1990s, with the popularity of the Internet and distributed computing, the research focus turned to the cooperation of agents in an open, dynamic environment. This required them to have intelligent strategies and interaction capabilities, promoting complex intelligent ecosystems' design.

At the beginning of the 21st century, breakthroughs in machine learning, particularly represented by the rise of deep learning, enabled autonomous learning from massive amounts of data rather than relying solely on pre-set programming rules. This marked a shift in agent research from rule-based methods to data-driven approaches. Deep neural networks significantly enhance the capabilities of intelligent agents in areas such as image recognition, speech processing, and natural language understanding, thus ushering in a new era of Artificial Intelligence (AI) agents. Recently, DeepSeek [48] and GPT series [1, 94] and other major language models have made significant progress in NLP, greatly improving the performance of artificial intelligence agents and demonstrating their profound potential.

An agent is an autonomous computing entity capable of perceiving its environment, making decisions, and taking actions. Figure 1 shows the three core modules of the LLM agent, each detailed as follows:

- **Perception Module:** The perception module is equivalent to the agent's "**senses**", ensuring that the agent can obtain the information needed for decision-making on time. Unlike conventional single-channel

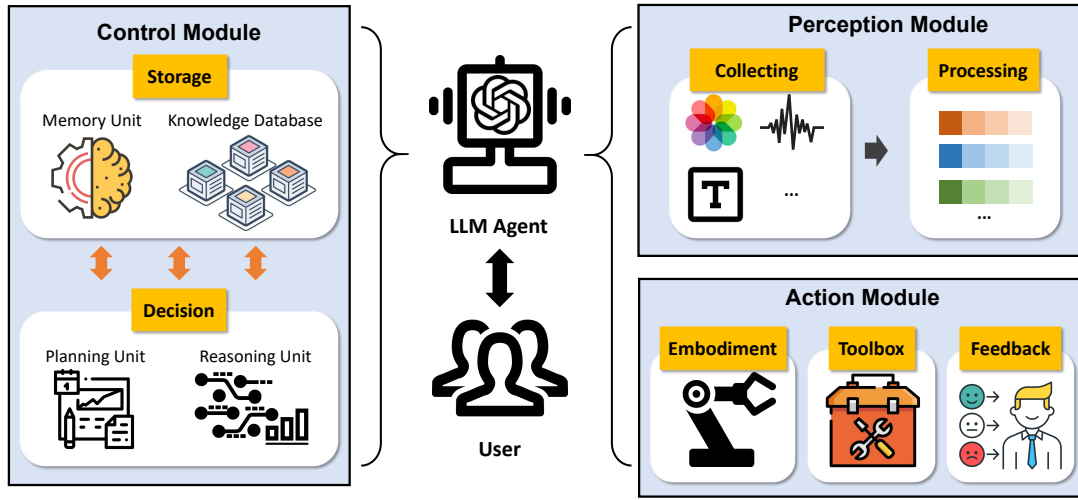


Fig. 1. Schematic diagram of the three core modules of agent

information processing systems, LLM agents demonstrate superior capability in seamlessly integrating multimodal information sources, including textual, auditory, visual, and even video data, thereby establishing a more robust and comprehensive perceptual framework [113].

- **Control Module:** The control module plays the role of a proxy "brain", analyzing the collected information, evaluating possible action plans, and selecting the best strategy. In LLM agents, an LLM serves as the core of the control module, enhancing the agent's understanding and generation ability of language contexts and strengthening its situational adaptability, creativity, and long-term memory functions.
- **Action Module:** The action module is equivalent to the "limbs" of the agent, ensuring that the agent's decisions are implemented, thereby changing the environmental state or achieving specific goals. In the LLM agent, this module is not limited to simple execution actions. Still, it is endowed with deeper functions, which can realize complex tasks such as language generation, tool use, and even concrete actions, greatly expanding the application scope and influence of the agent.

Currently, LLM agents have emerged in many fields. They can not only handle basic transactional work but also involve complex tasks that require deep understanding and creative thinking. For example, in game development, LLM agents can give non-player characters (NPCs) vivid and natural language capabilities to create a more immersive gaming environment [56, 95]; in the field of education, LLM agents can provide personalized tutoring for students as classroom assistants and help teachers correct homework [70]. In the field of scientific research, LLM agents such as CellAgent can perform complex tasks such as single-cell RNA sequencing data analysis that require the combination of specific domain knowledge [152]. LLM agents are constantly expanding the boundaries of human capabilities. As the technology matures, more innovative application scenarios will continue to emerge. LLM agents will drive human civilization in a more intelligent and sustainable direction.

3 WHY LLM AGENTS CAN BE USED FOR SEARCH AND RECOMMENDATION

As discussed in previous sections, LLM agents introduce new capabilities compared to simpler models and have diverse applications, such as game NPCs, education, RNA design, etc. In this section, we will demonstrate why these new capabilities provide LLM agents with distinct advantages in addressing IR problems.

3.1 LLM Agents Can Do Deep Thinking and Task Decomposition

In an era of information overload, user queries are often complex and diverse, making it challenging for traditional information retrieval systems to grasp the users' true needs accurately. However, LLM agents, with their advanced language comprehension abilities, can thoroughly analyze user queries, identifying key information, implicit intents, and contextual relationships. Two key capabilities underpin LLM agents' advances: deep thinking and extended information input. Firstly, Deep thinking enables the agent to perform a more in-depth analysis of a given issue, typically achieved through techniques of Chain-of-Thought (CoT) [38, 144]. Secondly, the use of extended information inputs enables the agent to remember more content, including user preferences, context information, and item profiles. It is usually facilitated by technologies such as a long context window that supports millions of token inputs. [32, 63]. Together, these capabilities enable the agent to decompose a complex IR task into multiple sub-tasks for more effective execution [5, 66]. For example, when a user plans a trip, an LLM agent can decompose the process into selecting a destination, planning the itinerary, booking flights and hotels, and calculating the budget, ultimately presenting the user with a complete report [153], which can be quite challenging for traditional IR systems.

3.2 LLM Agents can Interact with Environment and Integrate Information

In the field of information retrieval, another significant advantage of LLM agents is their ability to interact with environments and integrate results [53, 187]. IR often involves multiple data sources and complex retrieval scenarios, where traditional retrieval systems may struggle to effectively collect information from different sources. LLM agents with action modules can use various tools, browse web pages, operate mobile apps, and even independently search the internet to find needed information [102, 112]. Additionally, LLM agents can go a step further by filtering and summarizing the collected content, presenting users with a clear and concise report instead of delivering raw, unprocessed results like traditional IR systems. For example, when conducting academic literature searches, an LLM agent can interact with multiple academic databases to obtain research outcomes from different fields. It then analyzes and consolidates these results to provide users with a comprehensive literature review. Furthermore, LLM agents can interact with users, continuously adjusting retrieval strategies based on user feedback, thereby enhancing the quality of the retrieval results.

3.3 LLM Agents Can Serve as User Simulators to Generate Feedback to Information System

IR systems rely on real user feedback for improvement, but trial and error with actual users is costly and can negatively impact user experience. The various modules within LLM agents enable them to simulate human perception, memory, and actions, making them well-suited for collecting feedback in place of real users. LLM agents can simulate real user responses, making them invaluable for testing user experience when using a product and identifying potential improvements [169]. They can also be used to explore the evolution of users' mindsets and interests, aiding researchers in better understanding user behavior [95, 127]. LLM agents can simulate users' preferences and interact with recommender and search systems. These generated interaction data can be used to train recommender and search models, which can lead to more robust models [116, 177].

In summary, LLM agents bring several critical capabilities to the table that make them highly suitable for search and recommendation tasks. Their ability to think and plan, interface with various tools, and simulate user behavior makes a significant difference in the user experience of IR in the new era.

4 RECENT ADVANCES OF LLM AGENTS FOR RECOMMENDATION AND SEARCH

Recently, significant advancements have been made in the research on combining LLM agents with traditional search and recommendation algorithms. This section provides a comprehensive overview of these developments,

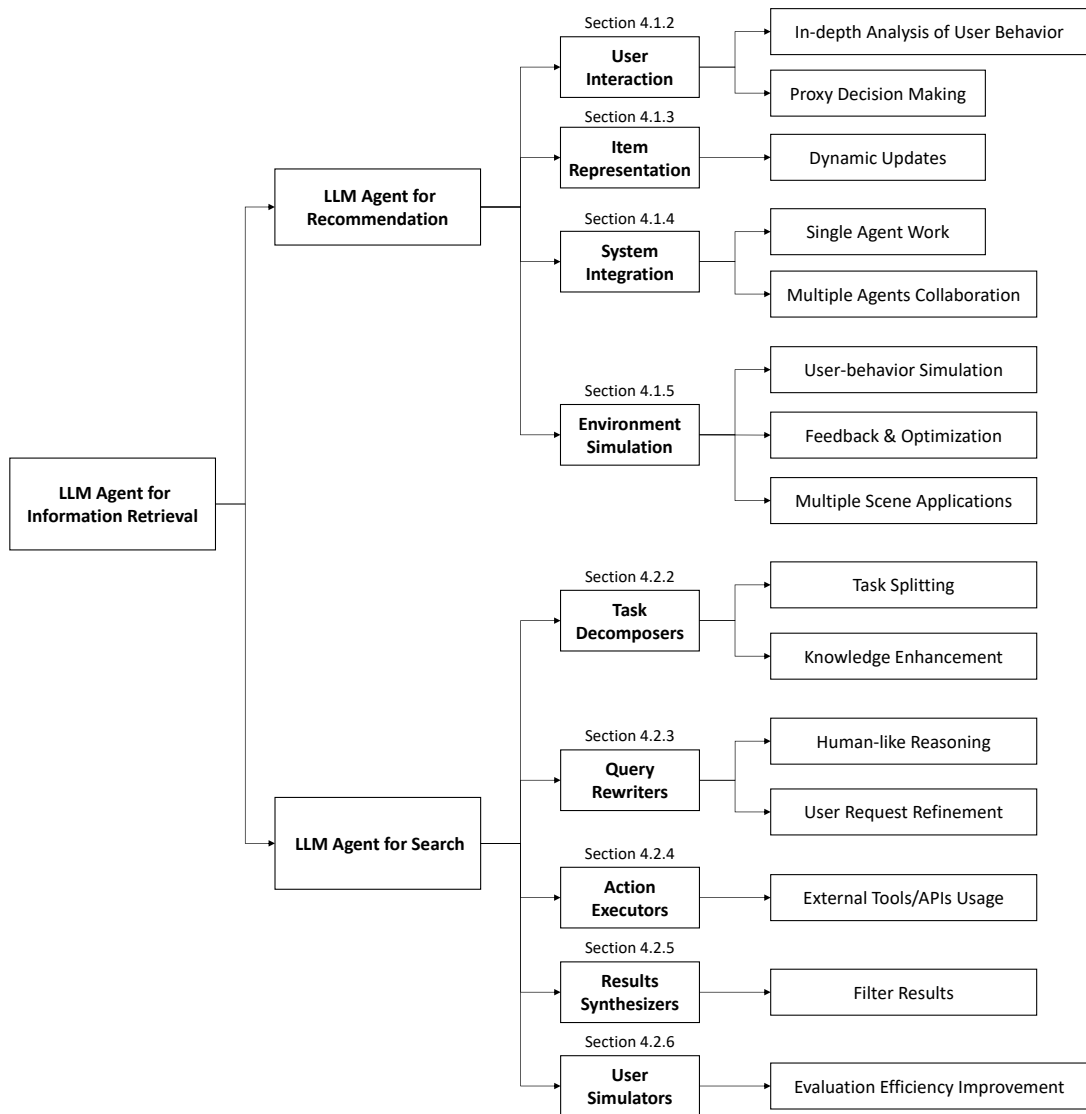


Fig. 2. Illustration of existing work of LLM agents for recommendation and search.

categorized into the contributions and innovations made in three primary areas: the overall domain and taxonomy, the role of agents, and recent significant papers. The classification of existing work in LLM agents for Recommendation and Search is shown in Figure 2.

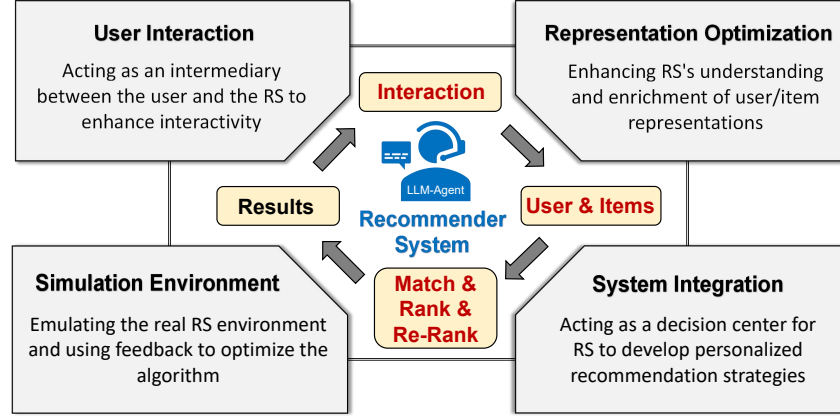


Fig. 3. Four domains of LLM agent's role in recommendation tasks

4.1 LLM Agents for Recommendation

4.1.1 Taxonomy Introduction. Deploying LLM agents in search and recommendation domains has given rise to new frameworks and methods. We redefine the taxonomy of these domains, identifying the key domains of LLM agents for RSs as user interaction, representation optimization, system integration, and environment simulation.

Figure 3 shows the four domains used by LLM agents for recommendation tasks, with detailed descriptions of each domain as follows:

- **User Interaction:** LLM agents serve as intermediaries between users and systems, enhancing interaction through natural language understanding and generation.
- **Representation Optimization:** The innovation focuses on better understanding and representation of users/items through LLM agents, making the recommendation process more precise.
- **System Integration:** LLM agent acts as the brain of RS, helping the RS analyze and make decisions for more effective results.
- **Environment Simulation:** LLM agent is used as a user simulator to build a simulation test environment for the RS and optimize feedback to achieve continuous improvement.

We categorize research work according to the descriptions of the domains as mentioned above and list typical works related to the four domains in Table 1.

4.1.2 User Interaction. Traditional RSs rely on static data (such as user historical behavior) to model user preferences, but this approach is difficult to cope with scenarios where user preferences change rapidly, new users or new products (cold start problem), and niche product recommendations (long tail effect). In addition, although the dialogue recommendation algorithm in the traditional recommendation algorithm can dynamically capture user intentions through natural language interaction, its implementation faces challenges such as how to efficiently understand complex human language, how to maintain coherence and goal orientation in the dialogue, and how to deal with ambiguity in the dialogue. LLM agent supports users to interact in natural language and provides highly personalized recommendations by analyzing user preferences, historical behaviors, conversation context, etc. The LLM agent can also flexibly call external tools to further optimize recommendation results according to specific needs, providing users with a richer and more personalized interactive experience.

Specifically, AutoConcierge [164] uses LLM to convert user questions into logical predicates, checks the consistency of information through the Answer Set Programming (ASP) system and updates the system status,

Table 1. A list of representative works of LLM agents for Recommendation.

Domain	Paper	What agents can do (ability)
Interaction	RAH [123]	Assists users in receiving customized recommendations and provide feedback
Interaction	ToolRec [180]	Uses tools for specific recommendation tasks
Interaction	RecAI [79]	Utilizes LLMs as an interface for traditional recommendation tools
Interaction	AutoConcierge [164]	Conducts real conversations with users
Interaction	FLOW [11]	Introduces a feedback loop to enable collaboration between the recommendation agent and the user agent
Representation	AgentCF [173]	Facilitates collaborative learning between user and item agents
Representation	Rec4Agentverse [170]	Controls the collaboration between the Intelligent Agent items and the Agent Recommenders.
Representation	KGLA [164]	Improves user agent memory
System	RecMind [139]	Introduces a self-inspiring algorithm for decision-making
System	InteRec [60]	Integrates LLMs and RSs for interactive recommendations
System	MACRec [141]	Develops a multi-agent collaboration framework for RSs
System	BiLLP [121]	Emphasizes long-term user retention using LLM-planned RL algorithms
System	MACRS [37]	Tackles dialog control and user feedback integration with multi-agent framework
System	PMS [129]	Uses multimodal, autonomous, multi-agent systems
System	CORE [64]	Combines conversational agents and RSs for better interaction
System	Hybrid-MACRS [90]	Combines LLM agent and search engine to optimize conversational recommendation
Simulation	Agent4Rec [168]	Trains LLM agents to simulate real users for evaluation
Simulation	RecAgent [135]	Simulates user behaviors related to the RS
Simulation	Yoon et al. [160]	Uses LLMs to simulate users for conversational recommendation tasks
Simulation	SUBER [21]	Develops an RL environment using LLM to simulate user feedback
Simulation	CSHI [189]	Proposes a framework for LLM-based user simulators in conversational RSs
Simulation	iEvaLM [137]	Suggests new evaluation methods using LLMs
Simulation	Zhu et al. [188]	Examines reliability and limitations of current LLM-based simulators
Simulation	OS-1 [157]	Develops an LLM-based eyewear system with conversational common ground
Simulation	CheatAgent [91]	Uses LLM agent to attack LLM-driven RSs
Simulation	Zhang et al. [177]	Improves the training efficiency and effectiveness of RSs based on reinforcement learning

and automatically generates questions to complete missing information when necessary. After obtaining all user preferences, the system searches for matching restaurants in its knowledge base and uses LLM to convert the results into natural language recommendations. If the user changes his mind or requests different suggestions, AutoConcierge can dynamically adjust its strategy, re-evaluate, and provide new recommendations. This process ensures efficient and accurate responses to user needs. ToolRec [180] framework uses LLM as a surrogate user to evaluate the degree of match between the user’s preferences and the current scenario, aiming to simulate the real user decision-making process. Then, external tools (ranking tools and retrieval tools) are called according to the user’s attribute instructions to explore different parts of the project pool. RecAI [79] is a practical toolkit that enhances or innovates RSs through the capabilities of LLMs. The LLM agent first develops a comprehensive execution plan based on the intent in the user conversation, then calls the tools, tracks the output of each tool, and ultimately generates a response for the user. In the FLOW [11] framework, the recommendation agent uses LLM, combined with the memory module and the recommendation module, to generate initial recommendations and optimize the recommendation results based on user feedback; the user agent is responsible for simulating user behavior and more accurately capturing the user’s potential interests by analyzing the interaction history with the recommendation agent. This iterative refinement process progressively enhances the cognitive capabilities of both the recommendation agent and the user simulation agent, consequently enabling more nuanced recommendation generation and higher-fidelity user behavior emulation. The RAH framework [123] combines RSs, assistants, and humans, using LLM agents to perceive, learn, act, criticize, and reflect. These agents collaborate through a learning-act-critic cycle to continuously improve their understanding of user personality. For example, when a user interacts or gives feedback, the learning agent extracts preliminary personality features, the action agent predicts

user behavior based on this, and the criticism agent evaluates the accuracy of the prediction. If the prediction is inaccurate, the criticism agent will analyze the reasons in depth and make suggestions for improvement, and the learning agent will adjust the personality features accordingly until the prediction is consistent with the user's actual behavior.

In summary, LLM agents can not only provide more accurate and personalized recommendation services but also provide users with a richer and more efficient interactive experience through continuous self-optimization and calling external resources. This marks the transformation of the RS from a simple information provider to an intelligent interactive partner, significantly improving the user experience and service quality.

4.1.3 Representation Optimization. In the traditional RS framework, the information records of each user and item are independent and static, including everything from basic information to detailed attributes, and most of these data updates rely on manual updates. However, user interests change dynamically, and different users have significantly different interests in the same item, and even the same user's interests change at different times. Traditional representation methods based on static data have difficulty capturing these personalized needs and changes over time, which poses a challenge to providing accurate and effective recommendation services. The LLM agent can generate detailed representations through deep semantic analysis, multimodal data fusion, and external world knowledge, and it can keenly perceive context changes and user feedback, promoting dynamic learning and updating of user and item representations.

Specifically, AgentCF[173] treats both users and items as agents, each with its memory module to maintain the preferences and tastes of potential adopters. In addition, the study introduces a new collaborative learning method for simultaneously optimizing user agents and item agents and designs a collaborative reflection mechanism that enables agents to adjust their memories based on differences in real-world interaction records to more accurately reflect user behavior. KGLA converts the path information in the knowledge graph into natural language descriptions, enhances the language agent's understanding of user preferences, and dynamically updates user memory based on the interaction between users and items during the simulation phase. At the same time, it uses information such as brands, categories, and product features in the knowledge graph to generate detailed item descriptions, and more accurately characterizes item features by analyzing the path relationship between users and items, thereby improving the personalization and accuracy of the recommendation. In the Rec4Agentverse [170] framework, items are converted into interactive, intelligent, and proactive LLM agents that can dynamically acquire user preferences and continuously update their own characteristics through multiple rounds of dialogue. Unlike traditional static items, LLM agents can provide multi-dimensional knowledge representation and enhance service content through collaboration with other agents. For example, a travel agent can not only recommend itineraries based on user interests but also cooperate with other agents to provide more comprehensive services. In addition, agent items can accurately model user preferences and improve the accuracy of personalized recommendations and user experience.

These works effectively enrich the representation of users and items by leveraging individual agent autonomy and inter-agent collaboration, giving them more initiative. This shift makes recommendation algorithms more situational and dynamic and can capture subtle changes in user preferences and the potential value of product attributes. The domain is still in its infancy, with less relevant work, and more excellent results are expected to emerge in the future.

4.1.4 System Integration. Traditional RSs often use a batch processing architecture to regularly update models and recommendation results. For example, models can be retrained every day or every week to reflect the latest user preferences. Data cleaning, feature engineering, model selection, and so on need to be considered separately during system integration. In contrast, LLM agents significantly reduce the need for manual intervention by learning and dynamically adjusting recommendation strategies in real time and supporting an instant feedback mechanism, which not only improves the efficiency of the recommendation system but also enhances the user

experience. In addition, LLM agents also support a team collaboration framework where each agent can focus on different data sources or tasks. Through multi-level and multi-dimensional collaboration and information sharing between these agents, the system can more comprehensively understand user preferences and provide highly personalized and context-related recommendation content. From the perspective of framework structure, these studies can be divided into single-agent work [60, 64, 121, 139], multi-agent collaboration [37, 90, 141].

In the **single-agent work category**, RecMind [139] introduces a self-inspiration algorithm for the LLM agent which is designed for RS to retain the status of all historical paths and use this historical information to optimize planning decisions, with the aim of addressing the limitations of existing RSs in generalizing their ability to perform new tasks and effectively utilizing external knowledge. InteRecAgent [60] takes LLM as its core, processing instruction understanding, common sense reasoning, and human-computer interaction, while the recommendation model serves as a tool for domain knowledge and user behavior patterns. Through memory components and dynamic examples to enhance task planning and reflection mechanisms, it upgrades traditional RSs, such as ID-based matrix decomposition to systems that support natural language interaction. The CORE framework [64] adopts an offline training and online checking model, where the RS model acts as an offline relevance score estimator, while the LLM conversational agent checks these scores online to reduce uncertainty by minimizing the sum of unchecked relevance scores. The LLM agent in BiLLP [121] is designed as a high-level planner that divides the learning process into two levels: macro-learning and micro-learning. Macro learning is responsible for acquiring high-level guiding principles, while micro-learning focuses on learning personalized recommendation strategies. This dual-level approach aims to achieve long-term recommendation strategies in RSs.

In the **multi-agent work category**, MACRS [37] is a multi-agent conversational RS that controls dialogue flow, generates responses through multiple LLM agents, learns from user feedback to refine dialogue strategies, prevents errors, and interprets implicit user semantics. MACRec [141] is a multi-agent framework that includes specialized agents like managers, analysts, reflectors, searchers, and task interpreters to address diverse recommendation tasks collaboratively. PAS [129] consists of three agents: The first agent recommends products suitable for answering a given question; the second agent asks follow-up questions based on images belonging to these recommended products; and the third agent then conducts the autonomous search. It also features real-time data extraction, recommendations based on user preferences, and adaptive learning. Hybrid-MACRS [90] consists of a central agent and a search agent. The central agent is driven by LLM and is responsible for user interaction, preference identification, search query generation, recommendation generation, etc. The search agent consists of a search engine and a relative search module, which is responsible for processing the search queries generated by the Central Agent, returning a list of matching products, and optimizing search results using personalized sorting based on user historical behavior.

These investigations collectively strive to transcend the constraints inherent in conventional recommendation systems by harnessing the advanced capabilities of LLM agents across multiple dimensions, including natural language comprehension, personalized preference modeling, and context-aware adaptive decision-making.

4.1.5 Environment Simulation. In the past, collecting user feedback to optimize recommendation algorithms was a time-consuming and resource-intensive process. However, with the advancement of technology, a LLM agent has emerged as an advanced simulator that can simulate user behaviors and preferences and support real-time response and large-scale concurrent testing. By leveraging the highly simulated environment provided by LLM agents, development teams can now test and optimize recommendation algorithms more efficiently. This simulation not only greatly shortens the testing cycle, but also reduces the reliance on real user data, while ensuring user privacy and data security. Most of the studies [135, 137, 160, 167, 188, 189] focus on using agents in conversational recommendation scenarios to simulate user behaviors and evaluate their effectiveness. In addition,

some studies [21, 91, 177] have explored in depth the new challenges and risks that LLM agents may bring in more practical applications.

Specifically, iEvaLM [137] represents an innovative interactive evaluation framework leveraging Large Language Models (LLMs), which incorporates an LLM-based user simulator to comprehensively model diverse system-user interaction scenarios. Extensive experimental evaluations conducted on two publicly available Conversational Recommender System (CRS) datasets demonstrate that iEvaLM achieves substantial performance enhancements over conventional evaluation methodologies. Notably, this framework introduces a novel dimension by incorporating the assessment of recommendation interpretability, thereby addressing a critical aspect often overlooked in traditional evaluation paradigms. Agent4Rec [167] generates 1,000 LLM-enabled agents with different social traits and preferences by initializing from the MovieLens-1M dataset. The agents interact with the RS on a page-by-page basis, performing operations such as watching, rating, evaluating, exiting, and interviewing. Experimental results show that the generative agent can effectively identify and respond to items that match user preferences, and the feedback provided by the agent can be used as enhanced data for iterative training and improvement of the recommendation strategy, thus forming a dual-track approach for comprehensive evaluation of recommendation algorithms. Yoon et al. [160] introduced a novel evaluation framework designed to quantitatively assess the behavioral alignment between LLMs and human users in CRS. This comprehensive protocol encompasses five distinct evaluation tasks: (1) item selection for discussion, (2) binary preference expression, (3) open-ended preference articulation, (4) recommendation solicitation, and (5) feedback provision. Through systematic experimentation, their study not only identifies significant behavioral discrepancies between LLM-generated responses and human interactions but also provides empirical insights into mitigating these deviations through optimized model selection criteria and advanced prompting techniques. Zhu et al. [188] conducted an analysis of the inherent limitations in current LLM-based user simulation approaches, particularly focusing on the issue of inadvertent data leakage. To address these challenges, they proposed SimpleUserSim, an innovative framework that implements a controlled information disclosure mechanism. This novel simulator employs a direct conversational strategy to steer dialogue topics toward target items while maintaining strict information constraints - specifically, it restricts access to target item titles and operates solely on item attribute information until successful recommendation completion. The RecAgent [135] framework has a detailed definition of LLM agent simulation users, including the Profile Module (ensuring that the agent has a unique personality), Memory Module (remember past behaviors and evolve dynamically in the environment), and Action Module (including RS-related searches, browsing, clicks, and page turns, as well as some social behaviors), so that it can effectively simulate real human behavior. The CSHI framework [189] introduces control mechanisms, enhances the scalability of simulation, and incorporates elements of human-computer collaboration, enabling agents to flexibly adjust their behavior based on real-time context and historical data, thereby providing a more accurate personalized experience in conversational RSs. Beyond acting as user simulators in conversational recommendation scenarios, LLM agents have further broadened their application areas, playing a vital role in reinforcement learning and attacks on RS. In the latest research, the SUBER framework [21] ingeniously utilizes LLM agents to simulate human behavior, thereby constructing a synthetic environment specifically for the training and evaluation of reinforcement learning-based RSs. This innovative approach effectively mitigates the high training costs associated with the scarcity of online data and addresses the challenges in evaluating RS models, particularly in accurately measuring model performance without directly impacting the real user experience. Zhang et al. [177] addresses the shortcomings of current user simulators such as SUBER and Agent4Rec (including high computational cost, susceptibility to "hallucinations", and failure to fully capture the complex user-system interaction dynamics) by explicitly modeling user preferences, combining the advantages of logical reasoning and statistical learning, and designing an effective evaluation framework to improve the reliability and efficiency of the simulators. CheatAgent [91] first analyzes the parts of the text that have the most significant impact on recommendation results. Then, it utilizes an LLM agent to generate adversarial perturbations that can mislead

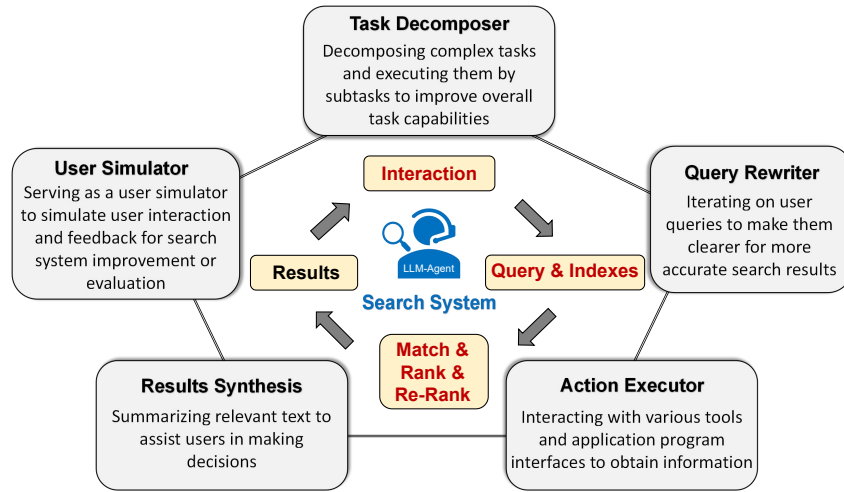


Fig. 4. Five domains of LLM agent's role in search tasks

the RS and optimize the attack strategy through prompt tuning. Finally, it fine-tunes the LLM's attack strategy based on feedback from the target system to enhance the attack's effectiveness.

In summary, these studies range from simple dialogue interactions to complex reinforcement learning environments, collectively representing the advancing application landscape of LLM agents in simulating user behavior.

4.2 LLM Agents for Searching

4.2.1 Taxonomy Introduction. Typically, the interaction between users and search engines can be divided into four key steps: goal, query writing, search execution, and result understanding [145]. Figure 4 shows the five domains used by LLM agents for search tasks, which can correspond to the typical IR interaction process mentioned 2. Detailed descriptions of each domain are as follows:

- **Task Decomposers:** LLM agents can break down complex tasks into smaller, manageable components to align with user intent and enhance overall task execution. In this case, LLM agents serve as the leading interfaces of search engines.
- **Query Rewriters:** LLM agents are good at refining user queries iteratively, making them clearer and more specific, leading to more accurate search results. In this case, LLM agents serve as the query modeling modules.
- **Action Executors:** LLM agents can interact with various tools and APIs to gather necessary information on behalf of the users. In this case, LLM agents serve as the matching and ranking modules.
- **Results Synthesizers:** LLM agents are suitable for summarizing large amounts of text and helping users quickly grasp the essential details to make decisions. In this case, LLM agents bring new abilities into the traditional searching process.
- **User Simulators:** LLMs can serve as user simulators to mimic user action and interaction for search systems improvement or evaluation. In this case, LLM agents give feedback to search engines.

We categorize research work related to search tasks according to the descriptions in the abovementioned domains and list typical works in the five domains in Table 2.

Table 2. A list of representative works of LLM agents in search.

Role of agent	Paper	What agents can do (ability)
Decomposer	Laser [85]	Uses state-space exploration for web navigation tasks
Decomposer	Knowagent [190]	Integrates knowledge base for task decomposition and logical action execution
Decomposer	Deng et al. [29]	Utilizes self-reflection memory enhancement planning for web navigation tasks
Decomposer	Gur et al. [50]	Learns from experience to complete tasks and divide complex instructions
Decomposer	SteP [124]	Introduces dynamic strategy combination through task decomposition
Decomposer	Koh et al. [69]	Enhances web navigation using tree search algorithms
Decomposer	Agent Q [100]	Integrates MCTS-guided search with self-critique for multi-step reasoning
Rewriter	CoSearchAgent [46]	Enables collaborative search through plug-ins that understand and refine queries
Rewriter	Joko et al. [66]	Assists in constructing personalized dialogue datasets to enhance query quality
Rewriter	Aliannejadi et al. [5]	Utilizes internal knowledge of LLMs for better retrieval and response generation
Rewriter	Jain et al. [61]	Proposes RAG-powered agents with multi-stream ensemble for semantic code search
Executor	AVATAR [150]	Utilizes a comparator LLM to teach the agent how to use tools
Executor	EASYTOOL [162]	extracts key information from tool documentation and designs a unified interface
Executor	CodeAct [136]	Integrates LLM agents with a Python interpreter in order to execute code actions
Executor	CodeNav [49]	Proposes code-as-tool paradigm through semantic code search engines
Synthesizer	PersonaRAG [165]	Uses real-time personalized data to enhance the relevance of the returned results.
Synthesizer	ChatCite [77]	Mimics human methods to extract key points and write summaries for literature reviews.
Synthesizer	PaSa [53]	Utilizes a selector to determine whether search results should be included or not.
Simulator	Sekuli et al. [115]	Explores user emulators in conversation search systems for multi-round clarification
Simulator	Usimagent [169]	Simulates users' query, click, and stop behavior in search tasks
Simulator	BASES [108]	Establishes a parameterized user profiling system with validation framework
Simulator	Chatshop [14]	Introduces LLM-simulated shoppers to evaluate agents' multi-turn interaction

4.2.2 Task Decomposers. The existing search technologies typically focus on learning the relevance between user queries and documents without any thoughtful processing. This limitation is critical for complex search tasks. For example, when a user searches for “where to go on vacation?”, his/her expectation might be a comparison of various travel destinations, details on travel itinerary, and hotel/flight prices. As discussed in Section 3, LLM agents possess strong reasoning and task decomposition abilities, enabling them to collect and aggregate information step by step, providing comprehensive results to the user.

Laser [85] introduces a method that views web navigation tasks as state-space exploration. LLM agents decompose tasks into the pre-defined set of states to handle unfamiliar scenarios and flexible backtracking. Deng et al. [29] propose a new task called Conversational Web Navigation, which requires complex multi-turn interactions with both users and the environment. In cases where tasks involve intricate actions, the agents will initially decompose the task into smaller sub-tasks. The decomposition results will later be verified and corrected by human annotators. Gur et al. [50] introduce a LLM agent named WebAgent. WebAgent has the capability to decompose instructions into standardized sub-instructions and plan ahead, which enables it to better understand and execute complex tasks. For instance, in a complex web operation task, it can first identify the main steps that need to be executed and then carry out these steps sequentially. SteP [124] proposes a policy stack to better coordinate different actions and accomplish complex tasks. By decomposing tasks into various policy layers, each policy can focus on specific aspects of the task, thereby improving efficiency and accuracy in task execution. For example, in handling web tasks, one policy might be responsible for page navigation while another policy might handle form filling. Together, these policies form a policy stack that collaboratively completes the entire task. Koh et al. [69] propose an inference-time search algorithm where agents do a tree search progress

to decide which action to take according to the current state. This method helps LLM agents better perform multi-step planning in interactive web environments. Knowagent [190] utilizes an action knowledge base and a knowledge-driven self-learning approach to guide the action paths during the planning phase. The action knowledge base furnishes the agent with extensive action knowledge, allowing for more rational action selection during planning. Meanwhile, the knowledge-driven self-learning approach enables the agent to continuously learn and refine its planning strategies, enhancing overall planning performance. Xie et al. [153] introduce a human-like reasoning framework to improve the planning ability that can help LLM agents when solving multi-phase problems like travel planning. Agent Q [100] proposes a framework that integrates guided Monte Carlo Tree Search (MCTS) with iterative fine-tuning for agents. This approach enables LLM agents to learn effectively from all trajectories, including both successful and unsuccessful ones, thereby enhancing their ability to generalize in complex, multi-step reasoning tasks.

Overall, the task decomposition capability of LLM agents significantly enhances the efficiency of users in obtaining information for complex search tasks. Some of the existing approaches predefine the action space and do search progress to decide the next move. These approaches offer strong control but are limited to solving specific problems. In contrast, the others allow the model to autonomously plan, offering greater flexibility, while the decomposition results may be in turn.

4.2.3 Query Rewriters. In traditional search scenarios, users must decide on the search terms themselves and master complex syntax to emphasize which keywords should necessarily be included or excluded from the search, and the length and context of these queries are usually limited. These limitations can prevent users from fully expressing their true intentions. However, these obstacles can be solved through LLM agents. LLM agents allow users to input longer context and fully express their intentions and then formulate more effective queries by themselves, thereby enhancing search efficiency.

CoSearchAgent [46] is a lightweight collaborative search agent powered by LLMs and serves as a search plugin on the instant message platform Slack. This agent can autonomously extract the user's search intent based on the conversation between users, formulate it into a query, and then invoke a search engine to return the results. The entire process does not require the user to explicitly input a query, thereby avoiding disruption to the conversational experience. Joko et al. [66] propose a scheme called LAPS (LLM-Augmented Personalized Self-Dialogue) to gather personalized dialogues, which can be used to train preference extraction and personalized response generation. As more conversational data is collected, LLM agents can gradually learn about users' various preferences, for instance, a user can be vegetarianism. Subsequently, when the same user inquires about recipes, the agents can account for vegetarian needs and directly search for vegetarian-related rather than normal recipes. Aliannejadi et al. [5] organize a competition on how to utilize users' prior interactions and present context and observed two main pipelines, namely, multi-turn retrieve-then-generate ($R \rightarrow G$) and generate-retrieve-then-generate ($G \rightarrow R \rightarrow G$). In this context, "G" includes conversational query expansion and conversational rewriting. Jain et al. [61] propose an approach utilizing Retrieval Augmented Generation (RAG) to enable LLM agents to enrich user queries with additional information. Specifically, by leveraging RAG, the agents can augment user queries with pertinent details sourced from GitHub repositories.

In summary, LLM agents can significantly enhance users' ability to utilize search systems. They can eliminate the need for users to learn search syntax, obviate the necessity of explicitly writing search queries, and even supplement information based on historical interactions or external data to improve search accuracy.

4.2.4 Action Executors. The searching process often involves the use of numerous tools. For example, in travel planning, users need to gather various types of information, such as geographical locations, hotels, flights, and weather. Specially designed LLM agents can learn how to judiciously select the appropriate tools and execute these actions on behalf of the user.

AVATAR [150] is a novel automatic framework to optimize the performance of LLM agents when using external tools and knowledge to complete complex tasks. It consists of an actor LLM and a comparator LLM. During optimization, the actor generates actions to answer queries by utilizing the provided tools. The comparator then evaluates a set of well-performing (positive) and poorly-performing (negative) queries, teaching the actor more effective retrieval strategies and tool usage. EASYTOOL [162] extracts key information from extensive tool documentation from various sources and meticulously designs a unified interface called tool instructions. This interface provides LLM-based agents with standardized tool descriptions and functionalities. By doing so, it effectively reduces the cognitive load on LLMs when understanding tool functionalities, thereby enhancing the efficiency and accuracy of tool usage. CodeAct [136] proposes consolidating LLM agents' actions into a unified action space using executable Python code with a Python interpreter integrated. In multi-turn interaction scenarios, CodeAct can execute code actions, dynamically revise previous actions, or generate new actions based on new observations. CodeNav [49] points out that tool-using LLM agents typically require manual registration of all relevant tools in the LLM context, which presents limitations when dealing with complex real-world codebases. CodeNav addresses this issue by automatically indexing and searching code fragments within the target codebase, eliminating the need for manual tool registration. It can locate relevant code snippets and incorporate them to iteratively develop solutions. Compared to traditional tool-using LLM agents, CodeNav is more flexible and efficient, capable of adapting to different codebase structures and requirements.

In summary, recent research has focused on two key areas. Firstly, how to equip LLMs agents with a broader range of tools, such as code execution and web retrieval. Secondly, how to make LLM agents better select the most appropriate tools from various options.

4.2.5 Results Synthesizers. In traditional search processes, after the search engine returns results, users must review each document individually, determine their relevance, and synthesize conclusions on their own. However, this laborious process can be entirely replaced by LLM agents, which can significantly enhance search efficiency and user experience.

PersonaRAG [165] is an innovative framework that incorporates user-centric agents to select retrieval and generation based on real user preference. This framework works through three steps: retrieval, user interaction analysis, and cognitive dynamic adaptation. PersonaRAG outperforms baseline models, delivering personalized answers that better address user needs on various question-answering datasets. ChatCite [77] is an LLM agent guided by human workflows for comparative literature summarization. The agent mimics the human workflow to extract key elements from relevant literature and generate summaries through a reflective incremental mechanism. This approach enables ChatCite to achieve a deeper understanding of the literature content and produce more targeted and comparative summaries. PaSa [53] introduces a hierarchical architecture comprising Crawler and Selector agents to address complex academic search challenges. The Crawler agent dynamically generates search queries and expands citation networks through tool invocation, while the Selector agent evaluates paper relevance via fine-grained scoring. To optimize this process, a novel conversational PPO algorithm is developed to handle sparse rewards in long-horizon search tasks. The system is trained on AutoScholarQuery, a synthetic dataset with 35k AI-domain queries, and validated on RealScholarQuery containing real-world research needs. Experiments show that PaSa-7B achieves 37.78% higher recall@20 than GPT-4o-enhanced Google Search, demonstrating superior capability in resolving specialized academic queries.

In conclusion, large model agents can assist in summarizing search results from multiple perspectives. On one hand, they can consider personalized user information to filter and highlight content relevant to the user. On the other hand, they can be equipped with better mechanisms for filtering information, such as mimicking human expertise or employing specialized selectors.

4.2.6 User Simulators. Evaluating search effectiveness has always been a challenge. Existing solutions typically measure algorithm performance by conducting A/B tests and collecting the metrics difference between A/B

groups, such as users' click rate and stay time. However, these statistical measures cannot directly reflect user satisfaction, and conducting A/B tests may disrupt real user experiences when the test strategy is suboptimal. LLM agents offer a solution to these issues by simulating user behavior. As previously mentioned, with their independent reasoning and action capabilities, LLM agents can mimic user decision-making processes by learning from real user data. This allows them not only to provide simple feedback like clicks but also to articulate specific reasons for their preferences or dislikes.

Usimagent [169] is a search user behavior emulator based on LLMs. USimAgent can simulate users' query, click, and stop behaviors in search tasks, generating complete search sessions. Through empirical research on actual user behavior datasets, researchers found that USimAgent outperforms existing methods in generating queries and performs similarly to traditional methods in predicting user click and stop behaviors. Sekuli [115] explores the use cases of user simulators in conversation search systems. User simulators that can automatically answer clarification questions in multiple rounds of conversations and evaluate the rationality of system search results. Additionally, user simulators can generate large amounts of training data, thereby enhancing the performance of conversational systems. BASES [108] is an innovative user simulation framework based on LLM agents, designed to simulate web search user behavior comprehensively. It is capable of generating unique user profiles on a large scale, leading to diverse search behaviors. Specifically, through learning from vast amounts of web knowledge, the large language model can achieve human-like intelligence and generalization capabilities. Chen et al. [14] proposed an online shopping task named ChatShop, where an agent needs to interact with the shopper to understand their needs and preferences gradually. They used LLM agents simulated shoppers to test the capabilities of the agent and found that the simulated shopper makes the task more realistic, increasing its complexity and challenge like real human shoppers.

On the whole, the use of LLM agents as user simulators in conversational search systems shows great promise. They not only enhance evaluation efficiency but also offer valuable insights for improving these systems. Future research could explore further refinements and applications of such simulators to address the challenges in evaluating conversational search results.

4.3 Benchmark and Datasets

Existing benchmarks on LLM primarily focus on language understanding and generation. However, these datasets and benchmarks are inadequate for comprehensively assessing the agents on their decision-making and interaction capabilities in complex, dynamic environments. In this subsection, we will elaborate on the research approaches and progress in this area.

Furuta et al. [42] introduced a benchmark, CompWoB, to highlight the limitations of language model agents in sequential task compositions and to emphasize the need for robust and generalized LLM agents for task combinations. CompWoB is a benchmark consisting of 50 new composite web automation tasks that reflect more realistic scenarios. Experimental results show that existing language model agents (such as gpt-3.5-turbo or gpt-4) achieve an average success rate of 94.0% on basic tasks, but their success rate drops to 24.9% on composite tasks. Agentbench [83] includes diverse agent-task scenarios and assesses LLMs' agent capabilities through task performance data. This benchmark not only pinpoints the strengths and weaknesses of LLMs in agent tasks but also provides a unified standard for comparison. WebArena [186] and MIND2WEB [28] provide environments that replicate realistic web browsing experiences, incorporating fully functional websites from diverse domains, such as e-commerce and social forums, with URLs, open tabs, and keyboard and mouse interactions. Cocktail [24] is a benchmark providing a comprehensive tool for evaluating information retrieval models in the LLM era. Cocktail comprises 16 different datasets covering various text retrieval tasks and domains, including a mix of human-written and LLM-generated corpora. The diversity of these datasets enables researchers to assess the performance of information retrieval models across different scenarios. To avoid potential biases from datasets

previously included in LLM training, the authors introduce a new dataset named NQ-UTD. This dataset features queries originating from recent events, ensuring the fairness and validity of the evaluation.

Overall, current research has introduced various datasets and benchmarks to evaluate the performance of LLM agents in IR scenarios. Unlike conventional LLM evaluation datasets that primarily focus on textual information, these datasets often include real online browsing tasks and user interaction behaviors.

5 EMBODIED LLM AGENTS: TOWARDS NEXT GENERATION RECOMMENDATION AND SEARCH

Although LLM agents enhance traditional recommendation and search models with comprehensive capabilities, they are often limited to analyzing static user-item interactions, such as pre-training from fixed-size and limited-domain data. This limitation hinders their ability to respond to emerging interactions in changing environments such as smart devices, and provide proactive services from both user and environmental perspectives. Recent embodied agents take a forward step over the LLM agent above by actively perceiving and interacting with environments. Their applications in the cyber world, such as autonomous exploration and online content retrieval in web services, naturally encompass a myriad of information retrieval processes that closely align with the task of recommendation and search. In this section, we discuss the latest development of embodied agents in the cyber environment, their potential applications in recommendation and search, limitations, and promising directions.

5.1 Recent Advance of Embodied Agents in Cyber Environment

Embodied agents. LLMs have been recognized with notable reasoning abilities and capacities to utilize external tools and knowledge. Building on this foundation, researchers have integrated these capabilities into unified LLM agents with diverse domain profiles and knowledge of specific targets. However, these LLM agents remain constrained by their dependence on passive learning from static data of images, videos, and text, which are pre-sampled from real-world information systems, like the Internet.

In recent years, researchers have been exploring the next piece of the puzzle of realizing AGI. Reinforcement Learning (RL) and Embodied AI have emerged in response to the thriving research of LLM agents. Consequently, there has been a shift from "Internet agents" that focus on learning from constant datasets collected from the Internet, towards "Embodied agents" which enable LLM agents to learn through interactions with their surroundings [36]. Just as its name implies, the term *embodied* grants the agent a physical substrate, manifesting in the real world as using robotic arms to measure and interact with the environment or in cyber environments as the "driving software" of information systems. Web agents serve as one of the key representations of embodied agents within the cyber environment. These agents must not only learn to navigate websites by interacting with GUI (Graphical User Interface) functions but also respond effectively to feedback from both web systems and human users. Such advancement inherently involves information retrieval tasks, requiring agents to perform complex multi-step searching and multi-modal retrievals. As a result, embodied agents are evolving into a new testbed and paradigm for research in recommendation and search areas.

How embodied agents interact with cyber environment. The current enthusiasm for building web agents is mainly divided into two directions; they consider static and dynamic environments respectively [51, 105, 186]. Most existing approaches follow the static environments by comparing an agent's action trajectory to a pre-collected human demonstration. For example, AiTW [106], the most commonly used dataset, comprises over 700K sequential samples with screenshot-action pairs. This lightweight environment offers greater flexibility for designing model architectures and promotes the adoption of more complex technology stacks, such as Chain-of-Thought (CoT) [27]. However, agents trained within this constrained framework often follow a predetermined path to complete tasks, which can result in a limited understanding of real-world scenarios and difficulty in responding to new targets. To achieve more realistic evaluations, recent research has developed dynamic environments where agents learn interactively from their mistakes, enabling them to test the boundaries of these systems.

AndroidWorld [105] is the representative benchmark where a simulated Android system is well-exhibited without restricted action trajectories. This introduces complexity and instability of the real environment, posing a huge challenge to the agent's robustness and ability to cope with complex tasks. In summary, both of these directions entail extensive multi-modal retrieval tasks, which could represent either novel intersection fields or evaluation environments for recommendation and search research.

Applications. Along with these approaches, embodied agents have been developed in diverse fields related to recommendation and search areas. In website search, web agents have been used to instantiate search engine models into autonomous agents, equipped with external web-browsing tools [52, 67, 68, 182]. In general GUI control, GUI agents have been supported by fine GUI element grounding capability to retrieve potential interactive objects [17, 54, 87, 166]. Other domains, such as games [181] and 3D environments [57], have also witnessed the flourishing application of embodied agents.

5.2 Why Embodied Agents Can Be Used for Recommendation and Search

The widespread application of embodied agents is based on two key factors: the inherent human-like language reasoning ability of LLM and the interactive learning of embodied agents. The following discussion delves into the intrinsic qualities behind these factors that make embodied agents viable tools for recommendation and search tasks, by reviewing their capabilities as lifelong learners, one-for-all (One4all) models, and personal assistants, respectively.

- **Lifelong learner:** One of the core strengths of embodied agents is their ability for continuous and even lifelong learning [132]. Unlike traditional machine learning models that are typically trained on a fixed dataset, embodied agents can persistently expand their knowledge and skills through ongoing interaction with environments. In the recommendation field, previous lifelong works [76, 81, 161] continually focus on identifying long-term behavior dependencies and incorporating emerging intentions, while these traditional task-specific models constantly suffer from catastrophic forgetting. The nature of embodied agents as lifelong learners could be a novel solution.
- **One4all model:** Another crucial advantage of embodied agents is their ability to achieve the One4all model, i.e., generalist problem solver. Differing from learning lifelong, the capability of one4all modeling concentrates on cross-domain tasks, where one4all agents are expected to efficiently adapt to unseen domains at minimal cost (in terms of adaptation time, fine-tuning data, loss of performance, etc.). Existing recommendation work [18, 39, 40, 72] has made great efforts on this topic thanks to techniques such as large-scale pre-training and efficient adaptation. However, a universal representation that seamlessly spans various domains is still out of reach [75, 122, 171]. Embodied agents could contribute massive supportive information to knowledge transfer through rapid adaptation to new scenarios.
- **Personal assistant:** Embodied agents can be omnipotent personal assistants. The characteristics of learning lifelong and being flexible as the one4all model could make the embodied agent a competent assistant for perceiving the rapidly changing personalized needs in the spatial-temporal dimension. In recommendation and search tasks, the possibility of condensing well-trained models into cloud-edge personal services gradually becomes a hot spot in both academics and industries. Prior research [104, 140, 159, 163] have delved into the path of achieving lightweight on-device recommender systems and search tools in privacy-preserving and cybersecure approaches, where embodied agents could be the next generation of solutions.

Embodied Agent Designing for Recommendation and Search. Existing embodied agents within cyber environment primarily focused on interaction with GUI. These agents concentrate on GUI operations and task-solving on a wide range of information systems like websites, desktops, and smartphones, which are typical scenarios for recommendation and search tasks. This close relationship could inspire innovative approaches and solutions for the next generation of recommendations and search. WorkArena [35] introduces the first

environment that supports chat-based agent-user interactions for the development and evaluation of web agents. Through a chat interface, the real human user can exchange messages with web agents. This allows for information retrieval tasks where a specific answer is expected from the agent, but also more practical tasks where user instructions change over time. This advancement may inspire further research and discussion around recommendation and search tasks, as agents would need to complete information retrieval tasks and output specific actions or plans. SmartAgent [172] defines a new task known as embodied personalized learning, where embodied cognition and item recommendation are interlaced as in many real-world personal assistance scenarios. To address this, a novel reasoning paradigm called Chain-of-User-Thought (COUT) is proposed to align user feedback with embodied agent actions under the progressive thoughts from basic GUI navigation to explicit and implicit user requirements. Leveraging the COUT paradigm, SmartAgent demonstrates the first full-stage embodied personalized capabilities in collaboration with a new dataset called SmartSpot, which for the first time supports a range of diverse embodied personalized environments.

In summary, the combination of embodied agents with recommendation and search systems could lead to the creation of intelligent systems offering more personalized services, but may also realize truly bionic personal assistants. This could be a potentially important direction for the next generation of recommendation and search research.

5.3 Limitation and Promising Directions

Limitation. Embodied agents have shown promise in the fields of recommendation and search, but they are also challenged by several key limitations. One major challenge is the limited generalization capability of current agents. Most existing agents are primarily designed for static environments, resulting in suboptimal performance when faced with dynamic real-world scenarios. Additionally, these static environments typically represent one-sided functions of specific systems, which further complicates the training of general embodied agents across diverse situations. This further leads to the second issue, where currently embodied agents are still not ready to effectively handle information retrieval tasks involved in complex systems. This is mainly due to their limited understanding of environments, as they are still stuck on learning systems' functionality. However, real-world application scenarios frequently involve personalized needs, which are often expressed through non-standard paths that extend beyond the training samples. Goal-completion-oriented embodied agents may struggle to address these personalized requirements underlying, resulting in lower user engagement and stickiness. Finally, there are still other critical technical bottlenecks that have not been fully addressed. For example, executing advanced cross-app operations on smartphone devices and multi-web page tasks on desktop environments remains challenging.

Promising directions. These limitations also spur new research directions. One of them is leveraging the one-for-all capability of embodied agents for transferable recommendation or search. Currently, large-scale pre-trained recommendation system models require extremely high costs to transfer to new scenarios. In some new scenarios where the gap is relatively large, the current pre-training and fine-tuning (PEFT) paradigm finds it difficult to achieve better domain adaptation. The ability of embodied agents to understand dynamic changes in the environment may be able to help with this. Another promising direction is the use of lightweight embodied agents for on-device recommendation. LLMs are often too resource-intensive to deploy on mobile devices, hindering the implementation of personalized services. Embodied agents, with their smaller footprint, may provide a solution to this problem. Additionally, the demand for zero-shot services on end-user devices also poses new requirements for embodied agent capabilities, which could drive more research. Finally, as embodied agents extend the capabilities of LLMs to recommendation and search areas within personal settings, the issue of privacy preservation becomes more crucial. Balancing user behavior modeling and individual privacy protection while providing intelligent services will be a critical and urgent problem.

6 OPEN PROBLEMS AND FUTURE DIRECTIONS

Even though LLM agents have shown great potential to bring promising advancement to IR, there are still many challenges to address and numerous directions to explore. In this section, we will discuss these topics.

Hallucinations. Hallucinations in LLMs refer to situations where the content generated by LLMs is inconsistent with facts or cannot be verified. For LLM agents in the IR field, hallucinations can result in the retrieval of incorrect information, thereby degrading the user experience.. Hallucinations can be divided into factuality hallucinations and faithfulness hallucinations [58]. The former refers to the inconsistency between the content generated by the LLM and the verifiable facts of the real world, while the latter refers to the inconsistency between the content generated by the model and the user's instructions. Researchers have proposed many methods to alleviate these two types of hallucinations. For factual hallucinations, solutions such as RAG [45, 103, 156], CoT [144] and consistency enhancing [93, 120, 138] can constrain LLM to obtain data from given sources. For command-type hallucinations, prompt engineering, RLHF, and other alignment technologies [99, 117, 143] can strengthen the consistency of execution.

Bias. Bias is a classic issue in the IR field, typically arising from improper data collection or training methods, leading to a significant discrepancy between the model results and real data, thereby degrading the accuracy of information retrieval. The capabilities of LLMs are largely constrained by their training data, making them more likely to encounter bias problems. Studies [25, 26, 86, 184] indicate that LLMs can introduce various biases, such as gender discrimination [31], political extremism [33], and echo chamber [118]. Dai et al. [26] find that LLM-based retrievers tend to prioritize content generated by LLMs over human-written text and introduce a bias-oriented loss to alleviate this trend.

Deployment Cost. LLM agents are typically composed of multiple LLM modules, resulting in a higher number of invocations and model parameters compared to general LLM applications. So, deploying LLM agents requires substantial resources, resulting in longer inference latency, which imposes higher hardware requirements on the end devices and may impact users' experience. Researchers have put a lot of effort into this issue; Wilkins et al. [146] optimize LLM efficiency by proposing a cost-based scheduling framework, while Avatar [149] and QueryAgent [59] utilize a tool or environmental feedback to reduce inference time.

Multi-Modal Agent. Multi-modal LLM agents can integrate data from different modalities, such as images and text, to more comprehensively understand user queries. For instance, when querying about a product, a user might provide both a textual description and an image of the product. A multi-modal LLM agent can analyze both the text and image information simultaneously, better understanding the product's features and thereby providing more accurate retrieval results [154, 175]. Recently, numerous studies have begun exploring how to leverage multi-modal capabilities to enhance LLM agents' ability to operate on web pages [41, 183] and mobile devices [92, 133]. These studies typically utilize visual perception capabilities to identify visual and textual elements of mobile applications accurately. This forms a foundational basis for the planning and executing of subsequent operation tasks.

Domain Specific Agent. In specialized domains with substantial domain knowledge, general LLM agents often perform poorly due to their reliance on common knowledge and data sources. However, data in these specialized domains is relatively sparse, making it insufficient to fully train LLMs. Therefore, efficiently and economically transforming general LLM agents into domain-specific LLM agents is a crucial direction for future research. Currently, related research has already been applied to relatively niche but popular categories such as clinical [65, 74, 111, 130, 158] and law legal [73, 126]. In the future, as LLM agents improve their ability to utilize sparse data, there may also be opportunities for improvement in fields that have struggled with effective information dissemination due to their niche nature, such as psychological counseling, job searching, etc.

Multi Agent Interaction. In multi-agent systems, individual agents can solve complex problems through information sharing and collaboration. Different agents may possess distinct knowledge and skills, and by

cooperating, they can leverage their strengths to enhance overall performance information retrieval [20, 147]. Recently, studies [95, 96] explore the interaction and evolution processes among multiple LLM agents, and several multi-agent frameworks [15, 101, 128] have been proposed. Infogent [107] utilized multi-agent systems to simulate human cognition for gathering information from the web, while Chateval [12] and MAD [80] improve agents' generating quality through multi-agent debate.

Personalization. Different users often have varying needs and preferences when performing information retrieval. For instance, the search term "Apple" might refer to Apple Inc. for a tech investor, whereas it might mean the fruit for an average person. By leveraging the memory module and long context processing capabilities of LLM agents, personalized information retrieval tailored to individual users can be achieved [78]. One approach to do so is to treat the user's interaction history as context and input it into the model [4, 89, 110]. Mo et al. [89] incorporate conversational and personalized elements with LLM agents to fulfill user needs through multi-turn interactions. They indicate that Personal Textual Knowledge Bases (PTKBs) can effectively enhance conversational information retrieval, allowing retrieval results to be more closely aligned with the user's background.

7 CONCLUSIONS

In this paper, we take the pioneering step in reviewing the advanced application of LLM agents in recommender and search systems. We begin the survey with a brief introduction to the concepts of recommendation and search tasks, as well as LLM agents, offering newcomers a foundational overview and critical background knowledge. Then, we elucidate the intrinsic nature of LLM agents, detailing three specific aspects to underscore why LLM agents can be used for recommendation and search. Moreover, we provide a detailed introduction to how to apply LLM agents in each task. For each phase, we provide a detailed taxonomy to categorize the major techniques and roles, drawing connections among the existing publications. Additionally, we extensively discuss the potential of applying Embodied agents in both tasks, delving into the rationale behind the advanced development. Finally, we summarize several challenges and promising directions in this field, which are expected to guide potential future directions.

REFERENCES

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).
- [2] Charu C Aggarwal et al. 2016. *Recommender systems*. Vol. 1. Springer.
- [3] Qingyao Ai, Ting Bai, Zhao Cao, Yi Chang, Jiawei Chen, Zhumin Chen, Zhiyong Cheng, Shoubin Dong, Zhicheng Dou, Fuli Feng, et al. 2023. Information retrieval meets large language models: a strategic report from chinese ir community. *AI Open* 4 (2023), 80–90.
- [4] Rachid Aknouche, Ounas Asfari, Fadila Bentayeb, and Omar Boussaid. 2012. Integrating query context and user context in an information retrieval model based on expanded language modeling. In *International Conference on Availability, Reliability, and Security*. Springer, 244–258.
- [5] Mohammad Aliannejadi, Zahra Abbasiantaeb, Shubham Chatterjee, Jeffery Dalton, and Leif Azzopardi. 2024. Trec ikat 2023: The interactive knowledge assistance track overview. *arXiv preprint arXiv:2401.01330* (2024).
- [6] Anthropic. 2024. Claude 3 haiku: our fastest model yet. <https://www.anthropic.com/news/claude-3-haiku>.
- [7] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 1999. *Modern information retrieval*. Vol. 463. ACM press New York.
- [8] Keqin Bao, Jizhi Zhang, Xinyu Lin, Yang Zhang, Wenjie Wang, and Fuli Feng. 2024. Large Language Models for Recommendation: Past, Present, and Future. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2993–2996.
- [9] Alan H Bond and Les Gasser. 2014. *Readings in distributed artificial intelligence*. Morgan Kaufmann.
- [10] Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems* 30, 1-7 (1998), 107–117.
- [11] Shihao Cai, Jizhi Zhang, Keqin Bao, Chongming Gao, and Fuli Feng. 2024. FLOW: A Feedback LOop FrameWork for Simultaneously Enhancing Recommendation and User Agents. *arXiv preprint arXiv:2410.20027* (2024).
- [12] Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201* (2023).

- [13] Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, et al. 2024. When large language models meet personalization: Perspectives of challenges and opportunities. *World Wide Web* 27, 4 (2024), 42.
- [14] Sanxing Chen, Sam Wiseman, and Bhuwan Dhingra. 2024. ChatShop: Interactive Information Seeking with Language Agents. *arXiv preprint arXiv:2404.09911* (2024).
- [15] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. 2023. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848* (2023).
- [16] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.
- [17] Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. 2024. SeeClick: Harnessing gui grounding for advanced visual gui agents. *arXiv preprint arXiv:2401.10935* (2024).
- [18] Yu Cheng, Yunzhu Pan, Jiaqi Zhang, Yongxin Ni, Aixun Sun, and Fajie Yuan. 2024. An Image Dataset for Benchmarking Recommender Systems with Raw Pixels. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*. SIAM, 418–426.
- [19] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).
- [20] Yong S Choi and Suk I Yoo. 1998. Multi-agent learning approach to www information retrieval using neural network. In *Proceedings of the 4th international conference on Intelligent user interfaces*. 23–30.
- [21] Nathan Corecco, Giorgio Piatti, Luca A Lanzendörfer, Flint Xiaofeng Fan, and Roger Wattenhofer. [n. d.]. SUBER: An RL Environment with Simulated Human Behavior for Recommender Systems. ([n. d.]).
- [22] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.
- [23] W Bruce Croft, Donald Metzler, and Trevor Strohman. 2010. *Search engines: Information retrieval in practice*. Vol. 520. Addison-Wesley Reading.
- [24] Sunhao Dai, Weihao Liu, Yuqi Zhou, Liang Pang, Rongju Ruan, Gang Wang, Zhenhua Dong, Jun Xu, and Ji-Rong Wen. 2024. Cocktail: A Comprehensive Information Retrieval Benchmark with LLM-Generated Documents Integration. *arXiv preprint arXiv:2405.16546* (2024).
- [25] Sunhao Dai, Chen Xu, Shicheng Xu, Liang Pang, Zhenhua Dong, and Jun Xu. 2024. Unifying Bias and Unfairness in Information Retrieval: A Survey of Challenges and Opportunities with Large Language Models. *arXiv preprint arXiv:2404.11457* (2024).
- [26] Sunhao Dai, Yuqi Zhou, Liang Pang, Weihao Liu, Xiaolin Hu, Yong Liu, Xiao Zhang, and Jun Xu. 2023. Llm may dominate information access: Neural retrievers are biased towards llm-generated texts. *arXiv preprint arXiv:2310.20501* (2023).
- [27] Shihan Deng, Weikai Xu, Hongda Sun, Wei Liu, Tao Tan, Jianfeng Liu, Ang Li, Jian Luan, Bin Wang, Rui Yan, et al. 2024. Mobile-Bench: An Evaluation Benchmark for LLM-based Mobile Agents. *arXiv preprint arXiv:2407.00993* (2024).
- [28] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2024. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems* 36 (2024).
- [29] Yang Deng, Xuan Zhang, Wenxuan Zhang, Yifei Yuan, See-Kiong Ng, and Tat-Seng Chua. 2024. On the Multi-turn Instruction Following for Conversational Web Agents. *arXiv preprint arXiv:2402.15057* (2024).
- [30] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [31] Harnoor Dhingra, Preetiha Jayashanker, Sayali Moghe, and Emma Strubell. 2023. Queer people are people first: Deconstructing sexual identity stereotypes in large language models. *arXiv preprint arXiv:2307.00101* (2023).
- [32] Yiran Ding, Li Lina Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. Longrope: Extending llm context window beyond 2 million tokens. *arXiv preprint arXiv:2402.13753* (2024).
- [33] Yi Dong, Ronghui Mu, Gaojie Jin, Yi Qi, Jinwei Hu, Xingyu Zhao, Jie Meng, Wenjie Ruan, and Xiaowei Huang. 2024. Building guardrails for large language models. *arXiv preprint arXiv:2402.01822* (2024).
- [34] Ali Dorri, Salil S Kanhere, and Raja Jurdak. 2018. Multi-agent systems: A survey. *Ieee Access* 6 (2018), 28573–28593.
- [35] Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H Laradji, Manuel Del Verme, Tom Marty, Léo Boisvert, Megh Thakkar, Quentin Cappart, David Vazquez, et al. 2024. WorkArena: How Capable are Web Agents at Solving Common Knowledge Work Tasks? *arXiv preprint arXiv:2403.07718* (2024).
- [36] Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. 2022. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence* 6, 2 (2022), 230–244.
- [37] Jiabao Fang, Shen Gao, Pengjie Ren, Xiuying Chen, Suzan Verberne, and Zhaochun Ren. 2024. A multi-agent conversational recommender system. *arXiv preprint arXiv:2402.01135* (2024).

- [38] Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. 2024. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems* 36 (2024).
- [39] Junchen Fu, Xuri Ge, Xin Xin, Alexandros Karatzoglou, Ioannis Arapakis, Jie Wang, and Joemon M Jose. 2024. IISAN: Efficiently adapting multimodal representation for sequential recommendation with decoupled PEFT. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 687–697.
- [40] Junchen Fu, Fajie Yuan, Yu Song, Zheng Yuan, Mingyue Cheng, Shenghui Cheng, Jiaqi Zhang, Jie Wang, and Yunzhu Pan. 2024. Exploring Adapter-based Transfer Learning for Recommender Systems: Empirical Studies and Practical Insights. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM '24)*. ACM. <https://doi.org/10.1145/3616855.3635805>
- [41] Hiroki Furuta, Kuang-Huei Lee, Ofir Nachum, Yutaka Matsuo, Aleksandra Faust, Shixiang Shane Gu, and Izzeddin Gur. 2023. Multimodal web navigation with instruction-finetuned foundation models. *arXiv preprint arXiv:2305.11854* (2023).
- [42] Hiroki Furuta, Yutaka Matsuo, Aleksandra Faust, and Izzeddin Gur. 2024. Exposing Limitations of Language Model Agents in Sequential-Task Compositions on the Web. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.
- [43] Chen Gao, Xiaochong Lan, Nian Li, Yuan Yuan, Jingtao Ding, Zhilun Zhou, Fengli Xu, and Yong Li. 2024. Large language models empowered agent-based modeling and simulation: A survey and perspectives. *Humanities and Social Sciences Communications* 11, 1 (2024), 1–24.
- [44] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhuan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. 2023. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems* 1, 1 (2023), 1–51.
- [45] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* (2023).
- [46] Peiyuan Gong, Jiamian Li, and Jiaxin Mao. 2024. CoSearchAgent: A Lightweight Collaborative Search Agent with Large Language Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2729–2733.
- [47] Alex Graves and Alex Graves. 2012. Long short-term memory. *Supervised sequence labelling with recurrent neural networks* (2012), 37–45.
- [48] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).
- [49] Tanmay Gupta, Luca Weihs, and Aniruddha Kembhavi. 2024. CodeNav: Beyond tool-use to using real-world codebases with LLM agents. *arXiv preprint arXiv:2406.12276* (2024).
- [50] Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. 2023. A real-world webagent with planning, long context understanding, and program synthesis. *arXiv preprint arXiv:2307.12856* (2023).
- [51] Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. 2023. A real-world webagent with planning, long context understanding, and program synthesis. *arXiv preprint arXiv:2307.12856* (2023).
- [52] Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. 2024. WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models. *arXiv preprint arXiv:2401.13919* (2024).
- [53] Yichen He, Guanhua Huang, Peiyuan Feng, Yuan Lin, Yuchen Zhang, Hang Li, et al. 2025. PaSa: An LLM Agent for Comprehensive Academic Paper Search. *arXiv preprint arXiv:2501.10120* (2025).
- [54] Wenyi Hong, Weihang Wang, Qingsong Lv, Jiazhen Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. 2024. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14281–14290.
- [55] Jiri Hron, Karl Krauth, Michael Jordan, and Niki Kilbertus. 2021. On component interactions in two-stage recommender systems. *Advances in neural information processing systems* 34 (2021), 2744–2757.
- [56] Sihao Hu, Tiansheng Huang, Fatih Ilhan, Selim Tekin, Gaowen Liu, Ramana Kompella, and Ling Liu. 2024. A survey on large language model-based game agents. *arXiv preprint arXiv:2404.02039* (2024).
- [57] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. 2023. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871* (2023).
- [58] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232* (2023).
- [59] Xiang Huang, Sitao Cheng, Shanshan Huang, Jiayu Shen, Yong Xu, Chaoyun Zhang, and Yuzhong Qu. 2024. QueryAgent: A Reliable and Efficient Reasoning Framework with Environmental Feedback based Self-Correction. *arXiv preprint arXiv:2403.11886* (2024).
- [60] Xu Huang, Jianxun Lian, Yuxuan Lei, Jing Yao, Defu Lian, and Xing Xie. 2023. Recommender ai agent: Integrating large language models for interactive recommendations. *arXiv preprint arXiv:2308.16505* (2023).
- [61] Sarthak Jain, Aditya Dora, Ka Seng Sam, and Prabhat Singh. 2024. Llm agents improve semantic code search. *arXiv preprint arXiv:2408.11058* (2024).

- [62] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2021. A survey on conversational recommender systems. *ACM Computing Surveys (CSUR)* 54, 5 (2021), 1–36.
- [63] Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024. Llm maybe longlm: Self-extend llm context window without tuning. *arXiv preprint arXiv:2401.01325* (2024).
- [64] Jiarui Jin, Xianyu Chen, Fanghua Ye, Mengyue Yang, Yue Feng, Weinan Zhang, Yong Yu, and Jun Wang. 2023. Lending interaction wings to recommender systems with conversational agents. *Advances in Neural Information Processing Systems* 36 (2023), 27951–27979.
- [65] Qiao Jin, Zhizheng Wang, Yifan Yang, Qingqing Zhu, Donald Wright, Thomas Huang, W John Wilbur, Zhe He, Andrew Taylor, Qingyu Chen, et al. 2024. AgentMD: Empowering Language Agents for Risk Prediction with Large-Scale Clinical Tool Learning. *arXiv preprint arXiv:2402.13225* (2024).
- [66] Hideaki Joko, Shubham Chatterjee, Andrew Ramsay, Arjen P de Vries, Jeff Dalton, and Faegheh Hasibi. 2024. Doing Personal LAPS: LLM-Augmented Dialogue Construction for Personalized Multi-Session Conversational Search. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 796–806.
- [67] Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2024. Language models can solve computer tasks. *Advances in Neural Information Processing Systems* 36 (2024).
- [68] Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. 2024. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649* (2024).
- [69] Jing Yu Koh, Stephen McAleer, Daniel Fried, and Ruslan Salakhutdinov. 2024. Tree Search for Language Model Agents. *arXiv preprint arXiv:2407.01476* (2024).
- [70] Paraskevas Lagakis and Stavros Demetriadis. 2024. EvaAI: A Multi-agent Framework Leveraging Large Language Models for Enhanced Automated Grading. In *International Conference on Intelligent Tutoring Systems*. Springer, 378–385.
- [71] Mark Levene. 2011. *An introduction to search engines and web navigation*. John Wiley & Sons.
- [72] Chenglin Li, Yuanzhen Xie, Chenyun Yu, Bo Hu, Zang Li, Guoqiang Shu, Xiaohu Qie, and Di Niu. 2023. One for All, All for One: Learning and Transferring User Embeddings for Cross-Domain Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23)*. ACM. <https://doi.org/10.1145/3539597.3570379>
- [73] Haitao Li, Junjie Chen, Jingli Yang, Qingyao Ai, Wei Jia, Youfeng Liu, Kai Lin, Yueyue Wu, Guozhi Yuan, Yiran Hu, et al. 2024. LegalAgentBench: Evaluating LLM Agents in Legal Domain. *arXiv preprint arXiv:2412.17259* (2024).
- [74] Qiang Li, Xiaoyan Yang, Hao wen Wang, Qin Wang, Lei Liu, Junjie Wang, Yang Zhang, Mingyuan Chu, Sen Hu, Yicheng Chen, et al. 2023. From beginner to expert: Modeling medical knowledge into general llms. *arXiv preprint arXiv:2312.01040* (2023).
- [75] Ruyu Li, Wenhao Deng, Yu Cheng, Zheng Yuan, Jiaqi Zhang, and Fajie Yuan. 2023. Exploring the upper limits of text-based collaborative filtering using large language models: Discoveries and insights. *arXiv preprint arXiv:2305.11700* (2023).
- [76] Wanda Li, Wenhao Zheng, Xuanji Xiao, and Suhang Wang. 2023. Stan: stage-adaptive network for multi-task recommendation by learning user lifecycle-based representation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 602–612.
- [77] Yutong Li, Lu Chen, Aiwei Liu, Kai Yu, and Lijie Wen. 2024. ChatCite: LLM agent with human workflow guidance for comparative literature summary. *arXiv preprint arXiv:2403.02574* (2024).
- [78] Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, et al. 2024. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459* (2024).
- [79] Jianxun Lian, Yuxuan Lei, Xu Huang, Jing Yao, Wei Xu, and Xing Xie. 2024. RecAI: Leveraging Large Language Models for Next-Generation Recommender Systems. In *Companion Proceedings of the ACM on Web Conference 2024*. 1031–1034.
- [80] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate. *arXiv preprint arXiv:2305.19118* (2023).
- [81] Jianghao Lin, Rongjie Shan, Chenxu Zhu, Kounianhua Du, Bo Chen, Shigang Quan, Ruiming Tang, Yong Yu, and Weinan Zhang. 2023. ReLLa: Retrieval-enhanced Large Language Models for Lifelong Sequential Behavior Comprehension in Recommendation. *Proceedings of the ACM on Web Conference 2024* (2023). <https://api.semanticscholar.org/CorpusID:261065228>
- [82] Qidong Liu, Jiaxi Hu, Yutian Xiao, Xiangyu Zhao, Jingtong Gao, Wanyu Wang, Qing Li, and Jiliang Tang. 2024. Multimodal recommender systems: A survey. *Comput. Surveys* 57, 2 (2024), 1–17.
- [83] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. 2023. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688* (2023).
- [84] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. *Recommender systems handbook* (2011), 73–105.
- [85] Kaixin Ma, Hongming Zhang, Hongwei Wang, Xiaoman Pan, and Dong Yu. 2023. Laser: Llm agent with state-space exploration for web navigation. *arXiv preprint arXiv:2309.08172* (2023).
- [86] Tianhui Ma, Yuan Cheng, Hengshu Zhu, and Hui Xiong. 2023. Large language models are not stable recommender systems. *arXiv preprint arXiv:2312.15746* (2023).

- [87] Xinbei Ma, Zhuosheng Zhang, and Hai Zhao. 2024. CoCo-Agent: A Comprehensive Cognitive MLLM Agent for Smartphone GUI Automation. *arXiv preprint arXiv:2402.11941v3* (2024).
- [88] Marvin Minsky. 1988. *Society of mind*. Simon and Schuster.
- [89] Fengran Mo, Longxiang Zhao, Kaiyu Huang, Yue Dong, Degen Huang, and Jian-Yun Nie. 2024. How to leverage personal textual knowledge for personalized conversational information retrieval. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 3954–3958.
- [90] Guangtao Nie, Rong Zhi, Xiaofan Yan, Yufan Du, Xiangyang Zhang, Jianwei Chen, Mi Zhou, Hongshen Chen, Tianhao Li, Ziguang Cheng, et al. 2024. A hybrid multi-agent conversational recommender system with llm and search engine in e-commerce. In *Proceedings of the 18th ACM Conference on Recommender Systems*. 745–747.
- [91] Liang-bo Ning, Shijie Wang, Wenqi Fan, Qing Li, Xin Xu, Hao Chen, and Feiran Huang. 2024. Cheatagent: Attacking llm-empowered recommender systems via llm agent. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2284–2295.
- [92] Songqin Nong, Jiali Zhu, Rui Wu, Jiongchao Jin, Shuo Shan, Xiutian Huang, and Wenhao Xu. 2024. Mobileflow: A multimodal llm for mobile gui agent. *arXiv preprint arXiv:2407.04346* (2024).
- [93] Sean O'Brien and Mike Lewis. 2023. Contrastive decoding improves reasoning in large language models. *arXiv preprint arXiv:2309.09117* (2023).
- [94] OpenAI. 2022. Introducing ChatGPT. <https://openai.com/blog/chatgpt>. (Accessed on 01/12/2023).
- [95] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–22.
- [96] Joon Sung Park, Carolyn Q Zou, Aaron Shaw, Benjamin Mako Hill, Carrie Cai, Meredith Ringel Morris, Robb Willer, Percy Liang, and Michael S Bernstein. 2024. Generative agent simulations of 1,000 people. *arXiv preprint arXiv:2411.10109* (2024).
- [97] Changhua Pei, Yi Zhang, Yongfeng Zhang, Fei Sun, Xiao Lin, Hanxiao Sun, Jian Wu, Peng Jiang, Junfeng Ge, Wenwu Ou, et al. 2019. Personalized re-ranking for recommendation. In *Proceedings of the 13th ACM conference on recommender systems*. 3–11.
- [98] Qiyao Peng, Hongtao Liu, Hua Huang, Qing Yang, and Minglai Shao. 2025. A Survey on LLM-powered Agents for Recommender Systems. *arXiv:2502.10050 [cs.LG]* <https://arxiv.org/abs/2502.10050>
- [99] Ethan Perez, Sam Ringer, Kamilė Lukošūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. 2022. Discovering language model behaviors with model-written evaluations. *arXiv preprint arXiv:2212.09251* (2022).
- [100] Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. 2024. Agent q: Advanced reasoning and learning for autonomous ai agents. *arXiv preprint arXiv:2408.07199* (2024).
- [101] Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. 2023. Communicative agents for software development. *arXiv preprint arXiv:2307.07924* (2023).
- [102] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789* (2023).
- [103] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics* 11 (2023), 1316–1331.
- [104] Reza Rawassizadeh and Yi Rong. 2023. ODSearch: Fast and Resource Efficient On-device Natural Language Search for Fitness Trackers' Data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 4 (2023), 1–25.
- [105] Christopher Rawles, Sarah Clincemallie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, et al. 2024. AndroidWorld: A dynamic benchmarking environment for autonomous agents. *arXiv preprint arXiv:2405.14573* (2024).
- [106] Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. 2024. Androidinthewild: A large-scale dataset for android device control. *Advances in Neural Information Processing Systems* 36 (2024).
- [107] Revanth Gangi Reddy, Sagnik Mukherjee, Jeonghwan Kim, Zhenhailong Wang, Dilek Hakkani-Tur, and Heng Ji. 2024. Infogent: An agent-based framework for web information aggregation. *arXiv preprint arXiv:2410.19054* (2024).
- [108] Ruiyang Ren, Peng Qiu, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2024. BASES: Large-scale Web Search User Simulation with Large Language Model based Agents. *arXiv preprint arXiv:2402.17505* (2024).
- [109] Paul Resnick and Hal R Varian. 1997. Recommender systems. *Commun. ACM* 40, 3 (1997), 56–58.
- [110] Chris Richardson, Yao Zhang, Kellen Gillespie, Sudipta Kar, Arshdeep Singh, Zeynab Raeesy, Omar Zia Khan, and Abhinav Sethy. 2023. Integrating summarization and retrieval for enhanced personalization via large language models. *arXiv preprint arXiv:2310.20081* (2023).
- [111] Maximilian Frederik Russe, Marco Reiser, Fabian Bamberg, and Alexander Rau. 2024. Improving the use of LLMs in radiology through prompt engineering: from precision prompts to zero-shot learning. In *RöFo-Fortschritte auf dem Gebiet der Röntgenstrahlen und der bildgebenden Verfahren*. Georg Thieme Verlag KG.

- [112] Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems* 36 (2024).
- [113] Raphael Schumann, Wanrong Zhu, Weixi Feng, Tsu-Jui Fu, Stefan Riezler, and William Yang Wang. 2024. Velma: Verbalization embodiment of llm agents for vision and language navigation in street view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 18924–18933.
- [114] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Vol. 39. Cambridge University Press Cambridge.
- [115] Ivan Sekulić, Mohammad Alinannejadi, and Fabio Crestani. 2024. Analysing utterances in llm-based user simulation for conversational search. *ACM Transactions on Intelligent Systems and Technology* 15, 3 (2024), 1–22.
- [116] Ivan Sekulić, Silvia Terragni, Victor Guimarães, Nghia Khau, Bruna Guedes, Modestas Filipavicius, André Ferreira Manso, and Roland Mathis. 2024. Reliable LLM-based user simulator for task-oriented dialogue systems. *arXiv preprint arXiv:2402.13374* (2024).
- [117] Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R Johnston, et al. 2023. Towards understanding sycophancy in language models. *arXiv preprint arXiv:2310.13548* (2023).
- [118] Nikhil Sharma, Q Vera Liao, and Ziang Xiao. 2024. Generative Echo Chamber? Effect of LLM-Powered Search Systems on Diverse Information Seeking. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–17.
- [119] Alex Sherstinsky. 2020. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena* 404 (2020), 132306.
- [120] Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Scott Wen-tau Yih. 2023. Trusting your evidence: Hallucinate less with context-aware decoding. *arXiv preprint arXiv:2305.14739* (2023).
- [121] Wentao Shi, Xiangnan He, Yang Zhang, Chongming Gao, Xinyue Li, Jizhi Zhang, Qifan Wang, and Fuli Feng. 2024. Enhancing Long-Term Recommendation with Bi-level Learnable Large Language Model Planning. *arXiv preprint arXiv:2403.00843* (2024).
- [122] Kyuyong Shin, Hanock Kwak, Kyung-Min Kim, Minkyu Kim, Young-Jin Park, Jisu Jeong, and Seungjae Jung. 2021. One4all user representation for recommender systems in e-commerce. *arXiv preprint arXiv:2106.00573* (2021).
- [123] Yubo Shu, Haonan Zhang, Hansu Gu, Peng Zhang, Tun Lu, Dongsheng Li, and Ning Gu. 2024. RAH! RecSys–Assistant–Human: A Human-Centered Recommendation Framework With LLM Agents. *IEEE Transactions on Computational Social Systems* (2024).
- [124] Paloma Sodhi, SRK Branavan, Yoav Artzi, and Ryan McDonald. 2023. Step: Stacked llm policies for web actions. *arXiv preprint arXiv:2310.03720* (2023).
- [125] Mirco Speretta and Susan Gauch. 2005. Personalized search based on user search histories. In *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*. IEEE, 622–628.
- [126] Jingyun Sun, Chengxiao Dai, Zhongze Luo, Yangbo Chang, and Yang Li. 2024. Lawluo: A chinese law firm co-run by llm agents. *arXiv preprint arXiv:2407.16252* (2024).
- [127] Ryosuke Takata, Atsushi Masumori, and Takashi Ikegami. 2024. Spontaneous Emergence of Agent Individuality through Social Interactions in LLM-Based Communities. *arXiv preprint arXiv:2411.03252* (2024).
- [128] X Team. 2023. Xagent: An autonomous agent for complex task solving. *XAgent blog* (2023).
- [129] Param Thakkar and Anushka Yadav. 2024. Personalized Recommendation Systems using Multimodal, Autonomous, Multi Agent Systems. *arXiv preprint arXiv:2410.19855* (2024).
- [130] Dave Van Veen, Cara Van Uden, Louis Blankemeier, Jean-Benoit Delbrouck, Asad Aali, Christian Bluethgen, Anuj Pareek, Malgorzata Polacin, Eduardo Pontes Reis, Anna Seehofnerová, et al. 2024. Adapted large language models can outperform medical experts in clinical text summarization. *Nature medicine* 30, 4 (2024), 1134–1142.
- [131] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
- [132] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291* (2023).
- [133] Junyang Wang, Haiyang Xu, Jiabo Ye, Ming Yan, Weizhou Shen, Ji Zhang, Fei Huang, and Jitao Sang. 2024. Mobile-agent: Autonomous multi-modal mobile device agent with visual perception. *arXiv preprint arXiv:2401.16158* (2024).
- [134] Kaiye Wang, Qiyue Yin, Wei Wang, Shu Wu, and Liang Wang. 2016. A comprehensive survey on cross-modal retrieval. *arXiv preprint arXiv:1607.06215* (2016).
- [135] Lei Wang, Jingsen Zhang, Xu Chen, Yankai Lin, Ruihua Song, Wayne Xin Zhao, and Ji-Rong Wen. 2023. RecAgent: A Novel Simulation Paradigm for Recommender Systems. *arXiv preprint arXiv:2306.02552* (2023).
- [136] Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. 2024. Executable code actions elicit better llm agents. In *Forty-first International Conference on Machine Learning*.
- [137] Xiaolei Wang, Xinyu Tang, Wayne Xin Zhao, Jingyuan Wang, and Ji-Rong Wen. 2023. Rethinking the evaluation for conversational recommendation in the era of large language models. *arXiv preprint arXiv:2305.13112* (2023).

- [138] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171* (2022).
- [139] Yancheng Wang, Ziyang Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Xiaojiang Huang, Yanbin Lu, and Yingzhen Yang. 2023. Recmind: Large language model powered agent for recommendation. *arXiv preprint arXiv:2308.14296* (2023).
- [140] Zongwei Wang, Min Gao, Junliang Yu, Hao Ma, Hongzhi Yin, and Shazia Sadiq. 2024. Poisoning attacks against recommender systems: A survey. *arXiv preprint arXiv:2401.01527* (2024).
- [141] Zhefan Wang, Yuanqing Yu, Wendi Zheng, Weizhi Ma, and Min Zhang. 2024. Multi-Agent Collaboration Framework for Recommender Systems. *arXiv preprint arXiv:2402.15235* (2024).
- [142] Donald A Waterman. 1985. *A guide to expert systems*. Addison-Wesley Longman Publishing Co., Inc.
- [143] Jerry Wei, Da Huang, Yifeng Lu, Denny Zhou, and Quoc V Le. 2023. Simple synthetic data reduces sycophancy in large language models. *arXiv preprint arXiv:2308.03958* (2023).
- [144] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.
- [145] Ryan W White. 2024. Advancing the search frontier with AI agents. *Commun. ACM* (2024).
- [146] Grant Wilkins, Srinivasan Keshav, and Richard Mortier. 2024. Hybrid Heterogeneous Clusters Can Lower the Energy Consumption of LLM Inference Workloads. In *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems*. 506–513.
- [147] BCM Wondergem, P van Bommel, Theo WC Huibers, and Th P Weide. 1998. Agents in Cyberspace—Towards a Framework for Multi-Agent Systems in Information Discovery. In *20th Annual BCS-IRSG Colloquium on IR*. BCS Learning & Development.
- [148] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2024. A survey on large language models for recommendation. *World Wide Web* 27, 5 (2024), 60.
- [149] Shirley Wu, Shiyu Zhao, Qian Huang, Kexin Huang, Michihiro Yasunaga, Kaidi Cao, Vassilis Ioannidis, Karthik Subbian, Jure Leskovec, and James Y Zou. 2025. AvaTaR: Optimizing LLM Agents for Tool Usage via Contrastive Reasoning. *Advances in Neural Information Processing Systems* 37 (2025), 25981–26010.
- [150] Shirley Wu, Shiyu Zhao, Qian Huang, Kexin Huang, Michihiro Yasunaga, Kaidi Cao, Vassilis N Ioannidis, Karthik Subbian, Jure Leskovec, and James Zou. 2024. AvaTaR: Optimizing LLM Agents for Tool-Assisted Knowledge Retrieval. *arXiv preprint arXiv:2406.11200* (2024).
- [151] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The Rise and Potential of Large Language Model Based Agents: A Survey. *arXiv preprint arXiv:2309.07864* (2023).
- [152] Yihang Xiao, Jinyi Liu, Yan Zheng, Xiaohan Xie, Jianye Hao, Mingzhi Li, Ruitao Wang, Fei Ni, Yuxiao Li, Jintian Luo, et al. 2024. CellAgent: An LLM-driven Multi-Agent Framework for Automated Single-cell Data Analysis. *bioRxiv* (2024), 2024–05.
- [153] Chengxing Xie and Difan Zou. 2024. A Human-Like Reasoning Framework for Multi-Phases Planning Task with Large Language Models. *arXiv preprint arXiv:2405.18208* (2024).
- [154] Junlin Xie, Zhihong Chen, Ruifei Zhang, Xiang Wan, and Guanbin Li. 2024. Large multimodal agents: A survey. *arXiv preprint arXiv:2402.15116* (2024).
- [155] Jun Xu, Xiangnan He, and Hang Li. 2018. Deep learning for matching in search and recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1365–1368.
- [156] Xinhao Xu, Hui Chen, Zijia Lin, Jungong Han, Lixing Gong, Guoxin Wang, Yongjun Bao, and Guiguang Ding. 2024. Tad: A plug-and-play task-aware decoding method to better adapt llms on downstream tasks. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI*.
- [157] Zhenyu Xu, Hailin Xu, Zhouyang Lu, Yingying Zhao, Rui Zhu, Yujiang Wang, Mingzhi Dong, Yuhu Chang, Qin Lv, Robert P Dick, et al. 2024. Can Large Language Models Be Good Companions? An LLM-Based Eyewear System with Conversational Common Ground. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 2 (2024), 1–41.
- [158] Weixiang Yan, Haitian Liu, Tengxiao Wu, Qian Chen, Wen Wang, Haoyuan Chai, Jiayi Wang, Weishan Zhao, Yixin Zhang, Renjun Zhang, et al. 2024. ClinicalLab: Aligning Agents for Multi-Departmental Clinical Diagnostics in the Real World. *arXiv preprint arXiv:2406.13890* (2024).
- [159] Hongzhi Yin, Liang Qu, Tong Chen, Wei Yuan, Ruiqi Zheng, Jing Long, Xin Xia, Yuhui Shi, and Chengqi Zhang. 2024. On-device recommender systems: A comprehensive survey. *arXiv preprint arXiv:2401.11441* (2024).
- [160] Se-eun Yoon, Zhankui He, Jessica Maria Echterhoff, and Julian McAuley. 2024. Evaluating Large Language Models as Generative User Simulators for Conversational Recommendation. *arXiv preprint arXiv:2403.09738* (2024).
- [161] Fajie Yuan, Guoxiao Zhang, Alexandros Karatzoglou, Joemon Jose, Beibei Kong, and Yudong Li. 2021. One person, one model, one world: Learning continual user representation without forgetting. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 696–705.
- [162] Siyu Yuan, Kaitao Song, Jiangjie Chen, Xu Tan, Yongliang Shen, Ren Kan, Dongsheng Li, and Deqing Yang. 2024. Easytool: Enhancing llm-based agents with concise tool instruction. *arXiv preprint arXiv:2401.06201* (2024).
- [163] Wei Yuan, Hongzhi Yin, Fangzhao Wu, Shijie Zhang, Tieke He, and Hao Wang. 2023. Federated unlearning for on-device recommendation. In *Proceedings of the sixteenth ACM international conference on web search and data mining*. 393–401.

- [164] Yankai Zeng, Abhiramon Rajasekharan, Parth Padalkar, Kinjal Basu, Joaquín Arias, and Gopal Gupta. 2024. Automated interactive domain-specific conversational agents that understand human dialogs. In *International Symposium on Practical Aspects of Declarative Languages*. Springer, 204–222.
- [165] Saber Zerhouni and Michael Granitzer. 2024. PersonaRAG: Enhancing Retrieval-Augmented Generation Systems with User-Centric Agents. *arXiv preprint arXiv:2407.09394* (2024).
- [166] Zhuosheng Zhan and Aston Zhang. 2023. You only look at screens: Multimodal chain-of-action agents. *arXiv preprint arXiv:2309.11436* (2023).
- [167] An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. 2024. On generative agents in recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1807–1817.
- [168] An Zhang, Leheng Sheng, Yuxin Chen, Hao Li, Yang Deng, Xiang Wang, and Tat-Seng Chua. 2023. On Generative Agents in Recommendation. *arXiv preprint arXiv:2310.10108* (2023).
- [169] Erhan Zhang, Xingzhu Wang, Peiyuan Gong, Yankai Lin, and Jiaxin Mao. 2024. Usimagent: Large language models for simulating search users. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2687–2692.
- [170] Jizhi Zhang, Keqin Bao, Wenjie Wang, Yang Zhang, Wentao Shi, Wanhong Xu, Fuli Feng, and Tat-Seng Chua. 2024. Prospect Personalized Recommendation on Large Language Model-based Agent Platform. *arXiv preprint arXiv:2402.18240* (2024).
- [171] Jiaqi Zhang, Yu Cheng, Yongxin Ni, Yunzhu Pan, Zheng Yuan, Junchen Fu, Youhua Li, Jie Wang, and Fajie Yuan. 2024. Ninerec: A benchmark dataset suite for evaluating transferable recommendation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).
- [172] Jiaqi Zhang, Chen Gao, Liyuan Zhang, Yong Li, and Hongzhi Yin. 2024. SmartAgent: Chain-of-User-Thought for Embodied Personalized Agent in Cyber World. *arXiv preprint arXiv:2412.07472* (2024).
- [173] Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian McAuley, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2024. Agentcf: Collaborative learning with autonomous language agents for recommender systems. In *Proceedings of the ACM on Web Conference 2024*. 3679–3689.
- [174] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)* 52, 1 (2019), 1–38.
- [175] Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, et al. 2024. A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4314–4325.
- [176] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th acm international conference on information and knowledge management*. 177–186.
- [177] Zijian Zhang, Shuchang Liu, Ziru Liu, Rui Zhong, Qingpeng Cai, Xiangyu Zhao, Chunxu Zhang, Qidong Liu, and Peng Jiang. 2024. LLM-Powered User Simulator for Recommender System. *arXiv preprint arXiv:2412.16984* (2024).
- [178] Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 19632–19642.
- [179] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).
- [180] Yuyue Zhao, Jiancan Wu, Xiang Wang, Wei Tang, Dingxian Wang, and Maarten de Rijke. 2024. Let Me Do It For You: Towards LLM Empowered Recommendation via Tool Learning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1796–1806.
- [181] Zhonghan Zhao, Wenhao Chai, Xuan Wang, Li Boyi, Shengyu Hao, Shidong Cao, Tian Ye, Jenq-Neng Hwang, and Gaoang Wang. 2023. See and think: Embodied agent in virtual environment. *arXiv preprint arXiv:2311.15209* (2023).
- [182] Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614* (2024).
- [183] Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. 2024. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614* (2024).
- [184] Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2023. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*.
- [185] Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2023. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406* (2023).
- [186] Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, et al. 2023. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854* (2023).
- [187] Wangchunshu Zhou, Yuchen Eleanor Jiang, Long Li, Jialong Wu, Tiannan Wang, Shi Qiu, Jintian Zhang, Jing Chen, Ruipu Wu, Shuai Wang, et al. 2023. Agents: An open-source framework for autonomous language agents. *arXiv preprint arXiv:2309.07870* (2023).

- [188] Lixi Zhu, Xiaowen Huang, and Jitao Sang. 2024. How Reliable is Your Simulator? Analysis on the Limitations of Current LLM-based User Simulators for Conversational Recommendation. In *Companion Proceedings of the ACM on Web Conference 2024*. 1726–1732.
- [189] Lixi Zhu, Xiaowen Huang, and Jitao Sang. 2024. A LLM-based Controllable, Scalable, Human-Involved User Simulator Framework for Conversational Recommender Systems. *arXiv preprint arXiv:2405.08035* (2024).
- [190] Yuqi Zhu, Shuofei Qiao, Yixin Ou, Shumin Deng, Ningyu Zhang, Shiwei Lyu, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. 2024. Knowagent: Knowledge-augmented planning for llm-based agents. *arXiv preprint arXiv:2403.03101* (2024).
- [191] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107* (2023).