A Survey on AI Search with Large Language Models

Jian Li[1,3], Xiaoxi Li[2], Yan Zheng[1], Yizhang Jin[1], Shuo Wang[1], Jiafu Wu[1], Yabiao Wang[1], Chengjie Wang[1], Xiaotong Yuan[3]

[1]Tencent YouTu Lab, [2]Renmin University of China, [3]Nanjing University

Abstract

Seeking information is a complex task that requires considerable effort. Although search engines have transformed the way we access information, they often struggle to fully understand intricate human intentions. Recently, Large Language Models (LLMs) have demonstrated remarkable capabilities in natural language understanding and generation. However, LLMs are limited in their ability to acquire external knowledge and access up-to-date information. AI search has evolved by leveraging LLMs into the search process, allowing it to tackle complex real-world challenges through comprehensive information retrieval and multi-step reasoning, enabling us to browse and search the web effectively. Over the past few years, significant efforts have been made to improve AI search. This paper presents a comprehensive review of these methods, focusing on Text-based AI Search, Web Browsing Agent, Multimodal AI Search, Benchmarks, Software and Products. Finally, we discuss the limitations of the current AI search methods and explore promising future directions. For more details, please visit our website.

1 Introduction

Searching for information is a fundamental daily need for humans. To address the need for rapid acquisition of desired information, some core web search technologies, such as PageRank [1, 2, 3] have been developed to support information retrieval systems. These include search engines like Google, Bing, and Baidu, which retrieve relevant web pages in response to user queries, providing convenient and efficient access to information on the internet. Natural language processing (NLP) [4] and information retrieval (IR) [5] technologies have been developed to enhance the ability of machines to accurately fetch content from the vast array of websites available online. However, as user queries become more complex and the expectation for precise, contextually relevant, and up-to-date responses increases, traditional search technologies face challenges in fully understanding intricate human intention, and users must manually open, read, and synthesize multiple webpages to answer complex questions.

Recently, Large Language Models (LLMs) [6] have recently captured significant attention in both academic and industrial domains. LLMs such as ChatGPT [7], LLaMA [8] have showcased remarkable progress in language understanding, reasoning, and information integration. However, LLMs are limited in their ability to acquire external knowledge and access up-to-date information. To address these problems, researchers integrate the remarkable abilities of LLMs with search engines or websites, aiming to improve real-time evidence gathering and reflective reasoning. The complementary advantages of LLMs and search engines highlight an opportunity for their combination, where the reasoning ability of LLMs can be complemented by the extensive web information accessible via search engines. This integration is transforming the way we seek and integrate web-based information and ushering in a new era of search technology known as Artificial Intelligence (AI) Search. In this

[1]Project leader: Yabiao Wang; Corresponding author: Xiaotong Yuan

Preprint.

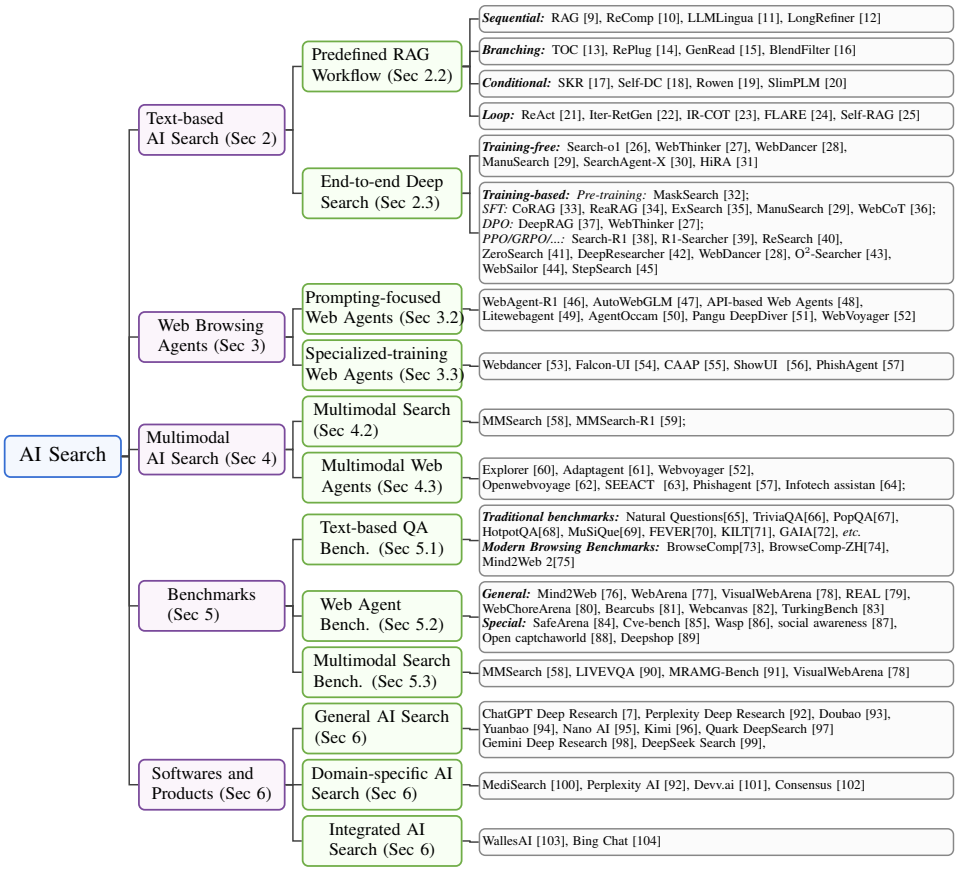


Figure 1: Taxonomy of research on AI search: investigating text-based AI search, web browsing agents, multimodal AI search, benchmarks, softwares and products.

survey, we aim to provide an overview of recent advancements in the rapidly evolving field of AI Search. As illustrated in Fig. 1, we organize the literature in a taxonomy consisting of five primary categories. (1) Text-based AI Search, (2) Web Browsing Agents, (3) Multimodal AI Search. (4) Benchmarks, (5) Software and Products.

The classic **Text-based AI Search** is a Retrieval-Augmented Generation (RAG) [105] workflow. RAG retrieves relevant passages from search engines based on the input query and incorporates them into the LLM's context for response generation. This allows the LLM to leverage external knowledge when answering questions. Another text-based AI Search is a deep search method that acquires external knowledge by calling search engines within an end-to-end coherent reasoning process to solve complex information retrieval problems. This approach does not require predefined workflows; instead, the model autonomously decides when to invoke search-related tools during its reasoning process, making it more flexible and effective. Different from Text-based AI Search, which browses pure text on the web, **Web Browsing Agent** completes a specific task on the target website through a series of actions by leveraging a thought-action-observation paradigm. For example, on the open street map, you want the web agent to calculate the driving time to reach Beijing from my stay at Shanghai. Web agents are evolving along two main paths: one utilizes a small language model trained specifically to filter actions or identify relevant HTML elements. The other path focuses on prompting LLMs, employing different agentic modules to accomplish complex web navigation tasks more effectively. In addition, with the rise of visual web-oriented benchmarks and the development of Multimodal Large Language Models, many agents use screenshots as sensory input to offer more comprehensive web environmental understanding. Different from **Multimodal AI Search**, most current AI search methods are limited to text-only settings, neglecting the multimodal user queries and the text-image interleaved nature of website information. particularly given the complexity and interleaved nature of modern websites. For example, consider a scenario where You captured an

antique photo at the museum but are unaware of its specific knowledge, such as history. A multimodal AI search engine could match photographs of these medals with an interleaved table of images and text retrieved from the Internet, thereby explaining to you the history and story. Hence, a multimodal AI search engine is crucial for advancing information retrieval and analysis.

Furthermore, this paper presents a comprehensive review of these **Benchmarks** for the relative methods. Evaluating the search capabilities of AI models, particularly large language models (LLMs), is essential for assessing their proficiency in effectively retrieving, filtering, and reasoning over web-based information. This evaluation is vital for understanding the true web-browsing competence of LLMs and their potential to address real-world tasks that require dynamic information retrieval. In recent years, substantial efforts have been dedicated to examining AI search from various perspectives. This paper offers a comprehensive review of benchmarks and evaluations related to AI search, focusing on three key areas: text-based question-answering benchmarks, web agent benchmarks, and multimodal benchmarks. The **Software and Products** of AI Search like Perplexity [92], can change our daily lives. We introduce a wide range of state-of-the-art open-source and proprietary models, software, as well as mainstream AI search products, aiming to provide a diverse, comprehensive display of AI Search. Finally, we discuss the limitations of the current AI search methods and explore promising future directions. To illustrate the evolution of AI search methods over time, Fig. 2 presents a timeline of recent AI search technologies, related methods, and products.

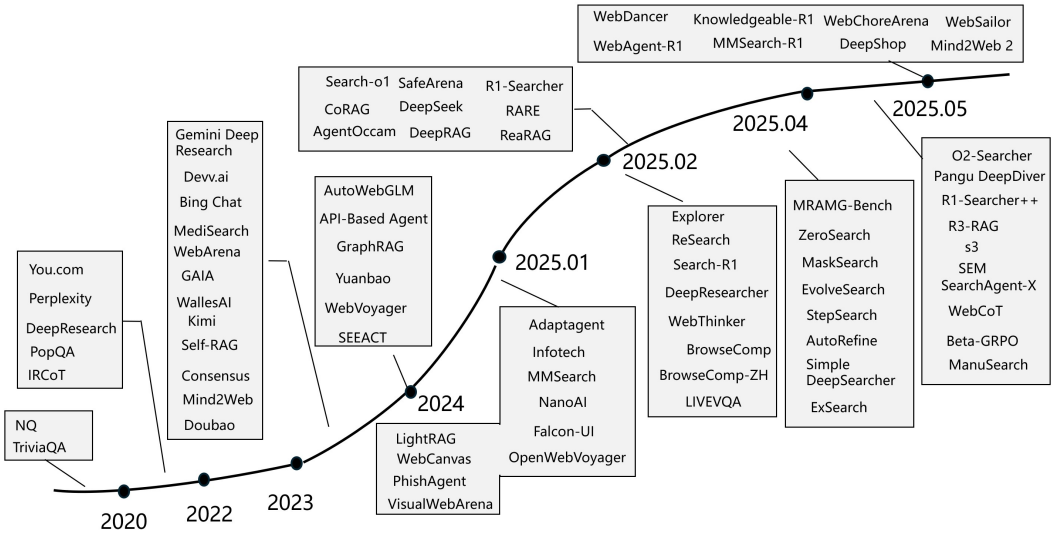


Figure 2: A timeline of existing AI Search and relatives methods and products in recent years. The timeline was established mainly according to the release date (e.g., the submission date to arxiv) of the technical paper for a model.

2 Text-based AI Search

AI search represents a transformative advancement in information retrieval systems, evolving from traditional search engines to sophisticated approaches incorporating RAG workflows and Deep Search capabilities, as shown in Figure 3. This section provides an overview of the key components and cutting-edge developments in modern text-based AI search technologies.

2.1 Traditional Search Engines

Traditional search engines form the backbone of modern search engines. They employ a variety of techniques to efficiently process user queries and return relevant results. Two key components of these systems are document retrieval and post-ranking, which work in tandem to provide users with the most pertinent information [106].

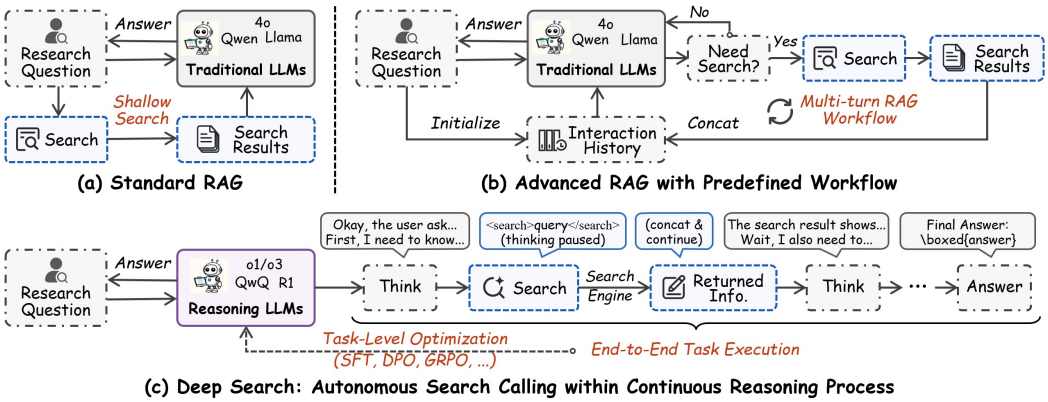


Figure 3: Evolution of text-based AI search paradigms, from (a) standard RAG that retrieves once per query, to (b) advanced RAG workflows capable of multi-turn search and decision-making, and finally to (c) fully autonomous, reasoning-model-powered Deep Search.

Document Retrieval. Document retrieval is the process of identifying relevant documents from a collection based on a user query. It is a crucial step in information retrieval, as it determines which documents are most relevant to the user's query. Traditional document retrieval systems typically employ techniques like inverted indexing, term frequency-inverse document frequency (TF-IDF), and BM25 models [107, 108]. More advanced approaches incorporate semantic matching using dense vector representations and neural ranking models [109, 110, 111, 112]. The retrieval process often involves query preprocessing, document indexing, similarity computation, and efficient search algorithms to handle large-scale document collections. In recent years, some work has explored LLM-based generative retrieval [113, 114, 115, 116], eliminating the need to build document indexes and directly generating document identifiers through LLMs.

Post-Ranking. Post-ranking is the process of refining the results of a search query after the initial retrieval stage. It is used to improve the quality of the search results by applying additional filters and reranking algorithms. Post-ranking systems typically employ learning-to-rank algorithms, neural reranking models, and LLM-based reranking models that combine multiple ranking signals [117, 118, 119]. This stage is crucial for improving search precision and user satisfaction by promoting the most relevant documents to top positions [117].

2.2 Retrieval-Augmented Generation with Pre-defined Workflows

Retrieval-Augmented Generation (RAG) enhances generative models by integrating a retrieval mechanism, allowing the model to ground its responses in external, reliable knowledge [9, 120]. Typically, a RAG system consists of a retriever and a generator, and the interaction between these components gives rise to four main RAG paradigms [105]:

Sequential RAG. Sequential RAG follows a linear "retrieve-then-generate" workflow, where the retriever first fetches relevant documents and the generator produces the final response based on these documents [9, 121, 122, 120, 123, 124]. Early works explored joint or separate training of retriever and generator, while recent approaches often use a frozen generator and focus on optimizing the retriever [125, 126, 127]. Pre-retrieval modules (e.g., rewriters [123]) and post-retrieval compressors [128, 129, 12, 10, 11, 130] further improve efficiency and response quality.

Branching RAG. Branching RAG processes the input query through multiple parallel pipelines, each potentially involving its own retrieval and generation steps, and then merges the outputs for a comprehensive answer [13, 16, 14, 15]. This approach enables finer-grained handling of complex queries, such as decomposing questions into sub-questions [13], augmenting queries with additional knowledge [16], or merging generated and retrieved content [14, 15].

Conditional RAG. Conditional RAG introduces a decision-making module to adaptively determine whether retrieval is necessary for a given query, improving flexibility and robustness [17, 20, 18, 19].

Methods include training classifiers to predict the need for retrieval [17, 20], using model confidence to guide retrieval [18], or employing consistency checks across perturbed queries [19].

Loop RAG. Loop RAG features iterative and interactive retrieval-generation cycles, enabling deep reasoning and handling of complex queries [21, 22, 23, 24, 25, 131, 132]. These methods alternate between retrieval and generation [22, 23], dynamically decide when to retrieve [24, 25], or decompose and answer sub-questions with verification steps to reduce misinformation [131, 132].

2.3 End-to-end Deep Search within Reasoning Process

Unlike traditional RAG workflows, Deep Search methods acquire external knowledge by calling search engines within an end-to-end coherent reasoning process to solve complex information retrieval problems. This approach does not require predefined workflows; instead, the model autonomously decides when to invoke search-related tools during its reasoning process, making it more flexible and effective [133, 134, 135].

Training-free Methods These methods aim to enhance the reasoning model's search capabilities by designing instructions that make the model aware of its task and how to use search tools. Initially, Search-o1 [26] proposed an agentic RAG mechanism that allows the reasoning model to autonomously retrieve external knowledge when encountering uncertain information during the main reasoning process, addressing the knowledge gaps in long Chain-of-Thought (CoT) reasoning. They also introduced a Reason-in-Documents process, which deeply analyzes the content of retrieved documents after each search call in the main reasoning process, returning consice and helpful information to the main reasoning chain. Experiments demonstrated significant performance improvements across mathematical, scientific, coding, and multi-hop question answering tasks.

Following this paradigm, a series of works such as WebThinker [27], WebDancer [28], ManuSearch [29] and HiRA [31] have proposed advanced frameworks. Typically, these methods introduce browsing of collected webpage URLs to achieve in-depth web exploration. Additionally, to improve search efficiency, SearchAgent-X [30] proposed an efficient reasoning framework that aims to increase system throughput and reduce latency through high-recall approximate retrieval, priority-aware scheduling, and non-stagnant retrieval mechanisms. Beyond directly answering users' information seeking questions, some works like WebThinker [27] have explored autonomously writing research reports while gathering information, offering users more comprehensive and cutting-edge knowledge.

Training-based Methods These methods design various training strategies to incentivize or enhance the LLM's search capabilities within the reasoning process. These strategies span pre-training, supervised fine-tuning (SFT), and reinforcement learning (RL).

During pre-training, the MaskSearch [32] framework introduces a Retrieval-Augmented Mask Prediction (RAMP) task, which trains the model to use search tools to fill in masked text, thereby enhancing its retrieval and reasoning abilities.

For Supervised Fine-Tuning (SFT), several methods focus on synthesizing long chain-of-thought data that incorporates search actions [33, 34, 35, 136, 28, 29, 36, 44]. Specifically, CoRAG [33] addresses the lack of intermediate retrieval steps in existing RAG datasets by automatically generating retrieval chains through rejection sampling. ReaRAG [34] avoids complex reinforcement learning by building a specialized dataset for fine-tuning via policy distillation. ExSearch [35] introduces an iterative self-incentivization framework based on the Generalized Expectation-Maximization (GEM) algorithm, enabling the model to learn from its own generated search trajectories. SimpleDeepSearcher [136] simulates user search behavior in a real-world web environment to synthesize multi-turn reasoning trajectories, which are then curated using a multi-criteria strategy. ManuSearch [29] leverages its multi-agent framework to decompose the deep search process and generate structured reasoning data. Lastly, WebCoT [36] synthesizes training data by reconstructing successful and failed trajectories, explicitly embedding reasoning skills like reflection, branching, and rollback into the chain of thought.

Furthermore, RL-based training has recently garnered significant attention. Some works leverage Direct Preference Optimization (DPO) [137]. For instance, DeepRAG [37] introduces a chain of calibration method to refine the model's atomic decisions, thereby synthesizing preference data for training. WebThinker [27] constructs positive and negative pairs based on the model's ability to correctly complete research tasks while efficiently using tools. By iteratively constructing data

and training the model with DPO, it implements on-policy RL training, improving performance on complex reasoning and report generation tasks.

Another line of work has explored training strategies based on PPO [138], GRPO [139], REINFORCE++ [140], and others. Initially, Search-R1 [38], R1-Searcher [39], ReSearch [40], and WebSailor [141] used the accuracy of the generated answer as a rule-based reward to encourage the LLM to use Wikipedia-based search tools during reasoning, achieving significant performance improvements in multi-hop QA tasks. Subsequently, a series of studies have investigated various enhancement strategies. These include leveraging web search capabilities [42, 28, 142, 46, 44], refining retrieved information [42, 143, 43], enabling multi-tool usage [144], developing improved sampling techniques [145], designing advanced reward functions [146], combining outcome and process rewards [147, 45], enhancing training efficiency [41], and implementing iterative SFT and RL training cycles [148]. To optimize search efficiency, methods such as SEM [149], β-GRPO [150], and s3 [151] have been proposed, which design training algorithms and reward functions for more efficient and accurate use of search tools.

3 Web Browsing Agent

A Web Browsing Agent is an artificial intelligence (AI)-based autonomous program capable of simulating human-like interactions within web browsers to perform tasks such as information retrieval, task execution, and dynamic environment adaptation. This section elaborates on the definition, categorization, evaluation criteria, and state-of-the-art technologies pertaining to Web Browsing Agents and Web Agents in general.

3.1 Agent

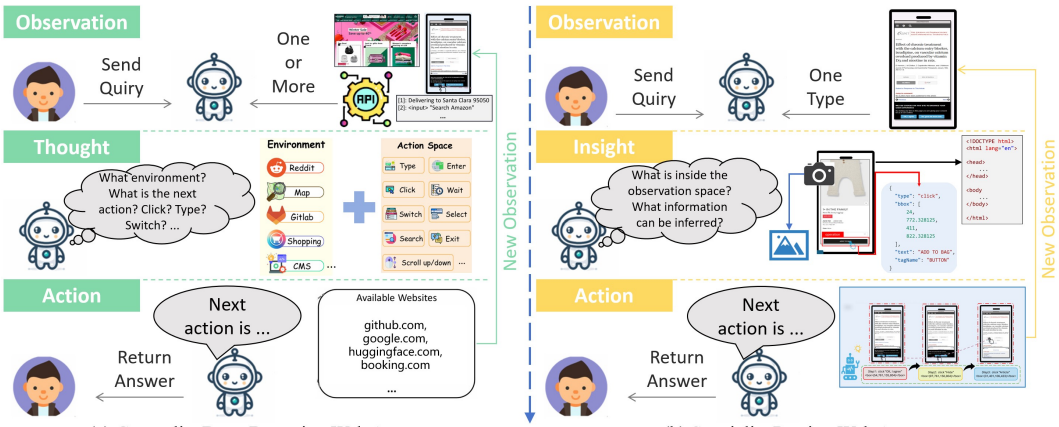
Aagent is a system that perceives environments, makes autonomous decisions, and executes tasks, aiming to simulate human cognition. With LLMs' emergence, LLM-based agents have become a key research direction. The core of LLM-based autonomous agents lies in two key aspects: architecture design and capability acquisition. In terms of architecture design, researchers aim to fully leverage the powerful language understanding and generation capabilities of LLMs through various network structures and modular combinations(e.g., AgentVerse's unified framework[152]). Regarding capability acquisition, two primary methods are employed[153]. First, Fine-tuning optimizes performance for specialized tasks by training the model with domain-specific data (e.g., SFT). Second, prompt engineering elicits the model's latent capabilities through carefully designed prompts.

Building upon general agent research, Web Agents' key distinction lies in handling the diversity and dynamism of web pages, which imposes stricter demands on perception modules and safety-aware design. Most current Web Agent frameworks adopt a Markov Decision Process (MDP) formulation[49], where each decision step is governed by a 4-tuple (S, A, T, R): S (State Space): Represents the environment state, typically the current webpage's HTML content. A (Action Space): Encompasses possible web interactions (e.g., button clicks, scrolling, text input). T (Transition Function): Defines how executing action A in state S alters the webpage state. R (Reward Function): Evaluates the quality of interactions to guide learning. While this MDP-based approach serves as the foundation, variants exist where steps are adapted (e.g., simplified or extended) based on practical requirements.

Based on the adopted training strategies, current Web Agents can be categorized into two types shown in Fig. 4: Generalist Deep Browsing Web Agents that enhance the model's ability to perform more complex web browsing tasks, especially across multiple types of web pages; Specialist Parsing Web Agents that employ dedicated training procedures to make the model focus specifically on action sequences or interface elements[154].

3.2 Generalist Deep Browsing Web Agents

Due to the open-ended nature of Web Agent applications, conventional static dataset-based training methods exhibit significant limitations, particularly when handling complex web navigation tasks. To enhance Web Agent capabilities in such scenarios, it is crucial to dynamically prompt the model during training to facilitate optimal action selection in corresponding situations. Reinforcement learning



(a) Generalist Deep Browsing Web Agents (b) Specialist Parsing Web Agents

Figure 4: Illustration of Web Agents. (a) Generalist Deep Browsing Web Agents comprises three iterative stages: (i) Observation;(ii) Thought;(iii) Action.(b) Specialist Analytical Agent follows three distinct phases:(i) Observation;(ii) Insight;(iii) Action. The loop terminates when required information is obtained, returning results to the user.

(RL) has become a key technology that enables web agents to adapt to dynamic environments in real-time through exploration and interactive feedback.

For instance, WebAgent-R1 is the first purely end-to-end RL-trained Web Agent[46]. It employs a multi-turn end-to-end RL framework, where the agent is trained through online interactions guided by rule-based outcome rewards. During training, it extends the standard Group Proximal Policy Optimization (GRPO) method into Multi-Group GRPO[155], utilizing multiple parallel interaction trajectories to enhance training efficacy. Additionally, WebAgent-R1 implements dynamic context compression for the state space (S) in the Markov Decision Process (MDP). When a new state arrives, earlier states are simplified to reduce context length while preserving complete history, thereby minimizing memory consumption. AutoWebGLM adopts a multi-stage training approach, integrating Supervised Fine-Tuning (SFT), RL, and Rejection Sampling Fine-Tuning (RFT)[47]. And it retains erroneous samples during training to facilitate learning from mistakes. Microsoft proposed an "API-first" Web Agent based on the CodeAct architecture[48], which replaces traditional browser interactions with API calls and selectively accesses browser APIs to retrieve feedback. For websites with limited APIs, the agent directly incorporates complete API documentation into its prompts. For websites with extensive APIs, it first generates a dictionary mapping each API to its documentation and then filters relevant APIs based on task descriptions. This dynamic approach enhances adaptability, with API-based prompts proving more concise and effective than direct browsing. AgentOccam simplifies the action space (A) by replacing multiple operations with functionally equivalent single actions and abstracting knowledge-dependent operations[50]. And it yields a streamlined yet effective Web Agent workflow by reducing the state space (S) by merging repetitive elements or structures and selectively replaying historical information.

Fine tuning before RL is also critical. This process establishes basic web interaction skills in the action space A. The result of the process directly affects the effect of reinforcement learning in the later stage. For example, Huawei's Pangu DeepDiver integrates cold-start SFT with reward allocation and scheduling mechanism[51], transitioning from lenient to strict scoring to stabilize RL training. The action enhances the model's ability to couple multiple reasoning and action steps.

Multimodality is also one of the methods to improve the effect of RL, and many web agents are developing towards multimodality. For example: WebVoyager leverages both visual (screenshots) and textual (HTML elements) modalities for interaction[52]. It utilizes the GPT-4V-ACT tool to annotate visual inputs (e.g., screenshots with numbered bounding boxes), which are then mapped to auxiliary text descriptions. This approach bypasses the need to parse complex HTML DOM or accessibility trees, simplifying structural representation. A detailed discussion of multimodal Web Agents is provided in Section 4.3.

3.3 Specialist Parsing Web Agents

Due to the complexity of web environments and the diversity of user objectives, Web Agents can acquire information from multiple sources. While, in principle, accessing a broader range of information types is preferable, focusing on a specific category for in-depth filtering and analysis can also yield effective results. This approach imposes lower demands on the model, making it suitable for lightweight Web Agents. Consequently, the goal-specific Web Agents—those specialized in target elements or actions—emerges. Training such specialized Web Agents typically requires well-defined objectives and dedicated datasets[55, 56].

For example, WebDancer specializes in QA pair parsing, aiming to extract high-quality trajectories from QA pairs to guide fine-tuning and reinforcement learning[28]. To extend the reasoning depth and hop count of existing QA datasets, the authors developed two datasets: CRAWLQA and E2HQA. CRAWLQA collects data from root URLs of official websites such as arXiv, GitHub, and Wikipedia, while E2HQA constructs its corpus by reformulating initially simple questions into more complex, multi-step queries. A ReAct-based Web Agent employs rejection sampling to extract trajectories from these QA datasets, forming both short and long chain-of-thought (CoT) trajectories. During training, the agent proceeds to formal RL using QA data not utilized in the SFT phase, internalizing CoT generation as an active behavioral component of the model. This process leverages the Dynamic Adaptive Policy Optimization (DAPO) algorithm. Falcon-UI is another example[54], focusing on graphical user interface (GUI) interactions. For its training, raw data was sourced from Common Crawl, followed by standard deduplication and denoising procedures. The researchers then use APIs with varying resolutions and platform types to simulate diverse device environments (e.g., Android, iOS, Windows, and Linux).The data integration platform interacts with the GUI interface and records the generated new interaction data. Unlike traditional full-page textual datasets, Falcon-UI exclusively logs visible elements, mimicking human-like interactions. The resulting hybrid dataset is then used to train Falcon-UI, significantly improving its GUI processing performance.

In some cases, depending on the task requirements, a Web Agent may employ multiple models. For instance, PhishAgent specializes in phishing website detection by identifying target website brands and their domains[57]. To recognize the brand of a target website, PhishAgent utilizes both textual and visual models. In many scenarios, textual information alone suffices for brand identification, in which case only LLM is used. However, if textual cues are insufficient or obscured by adversarial attacks, PhishAgent activates its brand extractor (IBE) based on multimodal large language model (MLLM), which identifies brand names from webpage screenshots. Upon successful brand recognition, PhishAgent proceeds to cross-reference the target domain with authentic domain information, which is obtained through both offline and online interactions, to determine whether the site is phishing.

4 Multimodal AI Search

Most current AI search methods are limited to text-only settings, neglecting the multimodal user queries and the text-image interleaved nature of website information. particularly given the complexity and interleaved nature of modern websites. For example, consider a scenario where You captured an antique photo at the museum but are unaware of its specific knowledge, such as history. A multimodal AI search engine could match photographs of these medals with an interleaved table of images and text retrieved from the Internet, thereby explaining to you the history and story. Hence, a multimodal AI search engine is crucial for advancing information retrieval and analysis.

4.1 Multimodal Large Language Models

Recently, Multimodal Large Language Models (MLLMs) or Large Multimodal Models(LMMs) [156] have demonstrated remarkable performance in various applications such as visual question answering, visual perception, understanding, and reasoning. Closed-source models is represented by GPT-4V [157], GPT-4o [158] and Claude 3.5 Sonnet [159]. Open-source MLLMs such as BLIP [160, 161], LLaVA [162, 163], Qwen-VL [164], Gemini [165], InternVL [166], EMU [167]. The standard MLLM framework can be divided into three main modules: a visual encoder g tasked with receiving and processing visual inputs, a pre-trained language model that manages the received multimodal signals and performs reasoning, and a visual-language projector P which functions as a bridge to align

the two modalities. Over the past few years, significant efforts in MLLM benchmarks survey [168] have been made to examine MLLMs from multiple perspectives.

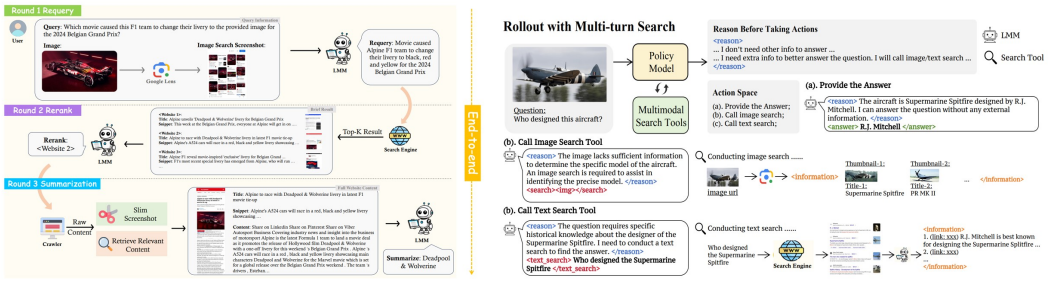


Figure 5: Illustration of Multimodal AI Search. (a) MMSearch [58] pipeline comprises three sequential stages executed by an MLLM:(i)requery,(ii)rerank,(iii)summarization. (b) A detailed view of the MMSearch-R1 [59] with rollout process and search tool execution.

4.2 Multimodal Search

Inspired by Text-based AI Search, it is necessary to explore a framework for MLLMs to function as multimodal AI search engines. MMSearch [58] in Fig. 5 proposes a multimodal AI search engine pipeline named MMSEARCH-ENGINE, empowering any MLLMs with advanced search capabilities. MMSEARCH-ENGINE maximizes the utilization of MLLMs' multimodal information comprehension abilities, incorporating both visual and textual website content as information sources within the searching process: requery, rerank, and summarization. MMSearch-R1 [59] in Fig. 5 is an initial effort to equip MLLMs with active image search capabilities through an end-to-end RL framework with the assistance of image search tools. This method is to train models not only to determine when to invoke the image search tool but also to effectively extract, synthesize, and utilize relevant information to support downstream reasoning. This work represents a foundational step toward enabling MLLMs to dynamically interact with external tools in a goal-directed manner, thereby enhancing their performance on long-tailed and knowledge-intensive VQA tasks.

4.3 Multimodal web agents

Websites are the primary Graphical User Interfaces(GUIs) medium through which humans interact with digital devices. Web agents can significantly enhance the user experience. By leveraging the ability of MLLMs to process and interpret web, Multimodal web agents [169] can autonomously execute user instructions, simulating human-like interactions such as clicking and typing on websites. Recent years have witnessed significant advancements in multi-modal research, with many Web Agents evolving in this direction[60][61].

WebVoyager[170] first instantiates a web browser and then performs operations using visual signals (i.e., screenshots) and textual signals (i.e., HTML elements) from the web. Its successor, OpenWebVoyager[62], further refines this approach through an iterative "exploration-feedback-optimization" loop.First, OpenWebVoyager adopts the more multimodal-capable Idefics2 model as its backbone LLM. Second, it abandons WebVoyager's method of tagging screenshots to establish mappings with text, instead leveraging accessibility trees for association. This enhancement enables autonomous optimization in real-world web environments while eliminating WebVoyager's dependency on closed-source models. As a result, OpenWebVoyager demonstrates improved capability in handling more complex web navigation tasks.

SEEACT [63] is a generalist web agent that harnesses the power of MLLMs for integrated visual understanding and acting on the web to solve web-based tasks (e.g., "Rent a truck with the lowest rate" in the car rental website). This work leverages an MLLM like GPT-4V to visually perceive websites and generate plans in textual form. The textual plans are then grounded onto the HTML elements and operations to act on the website. Beyond general domains, Multi-modal Web Agents exhibit high versatility in specialized fields[57]. For example, InfoTech Assistant employs a "retrieve-generate" collaborative mechanism[64], integrating 41 types of bridge technology data scraped from the FHWA

InfoTechnology website with a raw WebAgent to create a specialized agent for bridge evaluation and infrastructure technology.

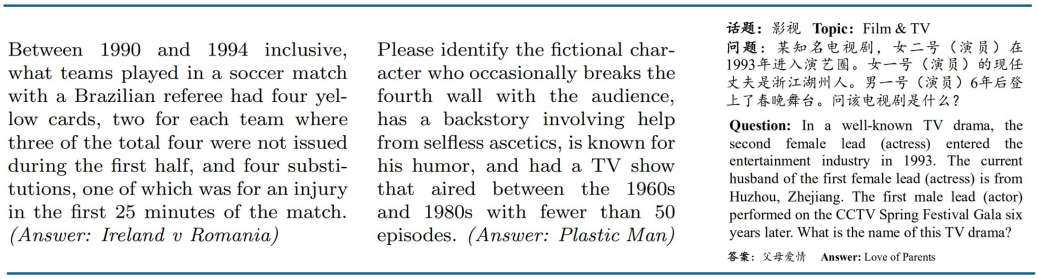
Open-source MLLM agents have made remarkable progress in offline evaluation benchmarks. However, their performance in more realistic online settings still lags significantly behind human-level capabilities. A major challenge lies in the scarcity of diverse, large-scale trajectory-level datasets across various domains, as collecting such data is both costly and resource-intensive. To address this, Explorer [171] has synthesized the most extensive and diverse trajectory-level dataset to date. Notably, the work employs comprehensive web exploration and iterative refinement techniques to capture a wide range of task intents, ensuring the dataset's diversity and utility.

5 Benchmarks

5.1 Text-based QA benchmark

As large language models (LLMs) evolve into tool-using agents, the ability to browse the web in real-time has become a critical yardstick for measuring their reasoning and retrieval competence. A variety of widely used English benchmarks have been proposed to assess retrieval capabilities, including TriviaQA, HotpotQA, FEVER, KILT, GAIA, *etc.* These datasets cover multi-hop reasoning, knowledge-intensive QA, and fact checking, typically relying on structured sources like Wikipedia and StackExchange.

Traditional Benchmarks Natural Questions (NQ)[65] is a large-scale QA dataset using real Google search queries and corresponding Wikipedia pages, requiring models to provide both long-form and short-form answers. TriviaQA[66] is a reading comprehension dataset characterized by complex, compositional questions with significant lexical variation from their evidence, often demanding multi-sentence reasoning. PopQA[67] is an entity-centric QA dataset designed to test factual knowledge recall across a long-tail distribution of entity popularity. HotpotQA[68] is a multi-hop QA dataset that requires reasoning across multiple documents and providing sentence-level supporting facts, making it a benchmark for explainable QA. 2WikiMultiHopQA[172] is a more challenging multi-hop QA dataset that integrates Wikipedia with Wikidata, using structured triples to explain complex reasoning paths. MuSiQue[69] is a multi-hop QA dataset emphasizing connected reasoning and includes unanswerable examples to challenge models that rely on shortcuts. FEVER[70] is a benchmark for fact verification, requiring systems to classify claims as `SUPPORTED`, `REFUTED`, or `NOTENOUGHINFO` against Wikipedia and provide sentence-level evidence. KILT[71] unifies 11 knowledge-intensive NLP tasks under a single Wikipedia snapshot, providing a standardized framework for evaluating both task performance and evidence retrieval. GAIA[72] evaluates general-purpose AI assistants with real-world questions that require a combination of reasoning, tool use, and multi-modality, revealing a large gap between AI and human performance. TREC Health Misinformation Track[173] provides datasets with binary "yes/no" health questions based on medical consensus to evaluate a system's ability to combat health misinformation.



(a) BrowseComp (b) BrowseComp-ZH

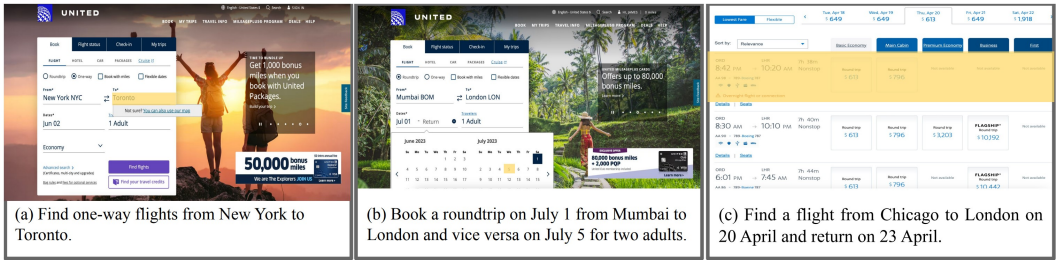
Figure 6: Illustration of Modern Browsing Benchmarks with complex and challenging queries. (a) BrowseComp [73]. (b) BrowseComp-ZH [74].

Modern Browsing Benchmarks. While traditional benchmarks mentioned above have effectively measured an AI's ability to retrieve straightforward information through basic queries (e.g., single-hop fact lookup), their simplicity has led to saturation—modern models now achieve near-perfect

scores on these tasks. This progress reveals a critical gap: real-world information needs often require persistent navigation through complex data landscapes. These challenges mirror the evolutionary jump from arithmetic tests to mathematical proofs—where success depends less on recall and more on strategic problem-solving.

BrowseComp[73] is a benchmark dataset introduced to evaluate web-browsing AI agents. It contains 1,266 challenging questions requiring persistent navigation of the internet to find entangled information. Key features include: (1) *High difficulty* - questions are designed to be unsolvable by humans within 10 minutes; (2) *Verifiability* - short reference answers enable easy validation; (3) *Diverse topics* spanning sports, fiction, and academic publications; and (4) *Core capability measurement* focusing on persistence, factual reasoning, and creative search strategies. BrowseComp-ZH[74] benchmark is a high-difficulty Chinese web browsing evaluation dataset consisting of 289 multi-hop questions across 11 domains (e.g., Art, Film&TV, Medicine). Each question is reverse-engineered from verifiable factual answers and undergoes rigorous two-stage quality control to ensure retrieval difficulty and answer uniqueness. Fig. 6 illustrates these two benchmarks and shows some complex and challenging queries. Mind2Web 2 [75] is also a modern benchmark with 130 realistic, high-quality, and long-horizon tasks that require real-time web browsing and extensive information synthesis.

5.2 Web agent benchmark



(a) Find one-way flights from New York to Toronto.

(b) Book a roundtrip on July 1 from Mumbai to London and vice versa on July 5 for two adults.

(c) Find a flight from Chicago to London on 20 April and return on 23 April.

Figure 7: Sample tasks of Mind2Web [76]. The web agent benchmark can test an agent's generalizability across tasks on the same website (a vs. b), similar tasks on different websites (a vs. c).

Web Agent Benchmark refer to a standardized set of test tasks and evaluation frameworks designed to assess the performance of web agents. These benchmarks simulate interactive tasks in real-world web environments to quantify an agent's capabilities in navigation, operation, and reasoning. A Web Agent Benchmark consists of two key components: tasks (data) and metrics. Tasks refer to a series of operational requirements posed to web agents, mimicking typical human web activities, such as clicking buttons, filling out forms, or navigating between pages. More complex tasks may involve multi-step processes. Metrics are the standards used to evaluate a web agent's performance, which vary depending on the agent's functionality and objectives[78, 83, 79, 81].

Mind2Web[76] shown in Fig. 7 is the first dataset designed for developing and evaluating general-purpose Web Agents. Its tasks include five top-level domains (travel, shopping, services, entertainment, and information). Each task consists of three core components: Task description: This outlines the high-level goal of the task. Action sequence: This is the sequence of actions required to complete the task on the website. Each action includes the target element and the corresponding operation. Webpage snapshot: It captures the webpage environment during task execution. In short, these three components contain all the task-related information. WebArena[77] analyzed real-world web browser histories and abstracted four prominent categories: e-commerce, social forums, collaborative software development, and content management. For task design, the authors curated three task types: Information seeking – requiring multi-page navigation. Website navigation – using interactive elements (e.g., search functions and links) to locate specific information in webpages. Content/configuration manipulation – creating, modifying, or configuring content (settings). Task evaluation involves: (1) comparing outputs for information seeking tasks, and (2) reward-based assessment of intermediate states for navigation and manipulation tasks. WebChoreArena[80] adheres to WebArena's design principles. However, its benchmark has new tasks. Their key characteristics include: (1)Emphasis on memory-intensive analytical tasks. (2)Reduction of ambiguity in task instructions and evaluation, a notable departure from WebArena. (3)Template-based task construction and expansion. Experiments on GPT-4o indicate that WebChoreArena presents greater challenges than WebArena. WebCanvas[82]

introduces a dynamic evaluation framework using "critical nodes". Critical nodes refer to essential steps that must be completed in any viable path to accomplish a given web task. To enhance realism, the authors derive Mind2Web-Live from tasks in the Mind2Web dataset. Mind2Web-Live includes critical nodes and meticulously annotated steps. Subsequent experiments demonstrate that in partially web environments, evaluating solely the final state or outcome is insufficient.

For task-specific objectives, general-purpose benchmarks often prove inadequate for evaluating Web Agent performance in their domains, necessitating dedicated benchmarks[87, 85, 86, 88]. For instance, DeepShop focuses on e-commerce, generating query tasks across five popular online shopping categories[89]. During assessment, it adopts a fine-grained approach by separately evaluating product attributes, matching accuracy, and ranking performance, ultimately synthesizing a comprehensive evaluation. SafeArena[84], the first benchmark dedicated to assessing malicious use cases of Web Agents, comprises 250 security tasks and 250 harmful tasks. Its evaluation metrics include the standard Task Completion Rate (TCR), and specialized measures rarely adopted by other benchmarks: Normalized Security Score (NSS) and Rejection Rate.

5.3 MM Search benchmark

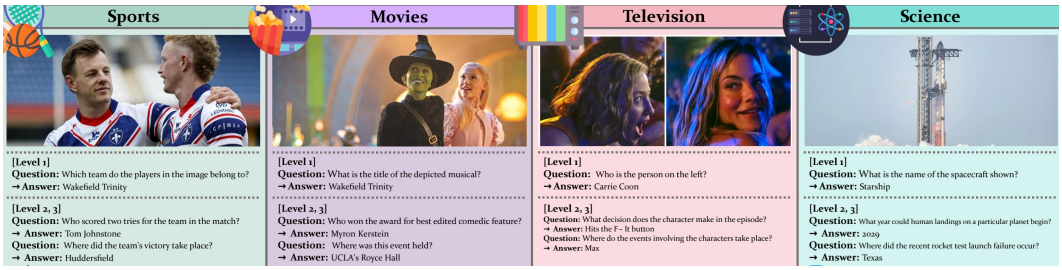


Figure 8: Illustration of four categories of LiveVQA [90]. QA pair for basic image for understanding, and two multimodal multi-hop QA pairs for deeper reasoning.

Large language models (LLMs) have made remarkable progress in understanding and reasoning about live textual content when integrated with a search engine. However, while live textual knowledge understanding has advanced significantly, a critical question remains unanswered: has other modality knowledge in live contexts, such as visual knowledge—been similarly solved? Are there multimodal search benchmarks for these methods?

MMSearch [58] introduced a multimodal AI search engine benchmark to comprehensively evaluate MLLMs' searching performance, and it serves as the first evaluation dataset to measure LMMs' multimodal searching capabilities. LIVEVQA [90] in Fig. 8 is an automatically collected benchmark dataset specifically designed to evaluate current AI systems on their ability to answer questions requiring live visual knowledge. However, current benchmarks for evaluating this critical task suffer from a notable lack of suitable datasets and scientifically rigorous evaluation metrics. MRAMG-Bench [91] is a novel benchmark designed to evaluate the MRAMG task comprehensively. MRAMG-Bench consists of six carefully curated English datasets, comprising 4,346 documents, 14,190 images, and 4,800 QA pairs, sourced from three domains: Web, Academia, and Lifestyle, across seven distinct data sources.

VisualWebArena [78] is designed primarily for visual web agent tasks, with both textual and visual content sourced from real-world environments. It comprises 910 real-world tasks across three distinct web environments: Classifieds – Introduces visually grounded tasks centered around typical user interactions (e.g., posting, searching, commenting) in classified advertisement websites. Shopping – Adopts the e-commerce environment from WebArena, incorporating product information and content from Amazon. Forum – Represents a social forum platform, featuring 31,464 posts with embedded images across various subforums and discussion threads. A key distinguishing feature of VisualWebArena is that all tasks require agents to process and interpret visual information, rather than relying solely on textual or HTML-based cues. For evaluation, the metrics follow WebArena's framework, but extend it by incorporating image verification in addition to the original two assessment methods.

6 Softwares and Products

AI search ecosystem has rapidly diversified into general-purpose platforms, domain-specific tools, and integrated assistants, each leveraging large language models (LLMs), retrieval-augmented generation (RAG), and agentic workflows to redefine information retrieval. Below, we will introduce the key products driving this transformation.

Global General-Purpose AI Search Engines. A pioneer in generative AI, ChatGPT Deep Research [7] integrates Bing's real-time web search to provide concise, conversational responses, sparking a surge of interest among researchers in large language models. Perplexity Deep Research [92] combines GPT-4 and Claude 3 with real-time web crawling, providing source-attributed answers. Its Discover feature tracks trending topics, making it ideal for academic literature reviews and technical writing. You.com [174] prioritizes privacy and personalization, allowing model switching (e.g., GPT-4, Claude) mid-session. Its Smart mode offers free access, while Research mode supports deep investigations with citation exports. Gemini Deep Research [98] embeds multi-modal capabilities into Pixel phones and Wear OS, enabling real-time translation via camera and health data-driven recommendations, reinforcing its "hardware-software" synergy in high-end markets. Optimized for speed and cost-efficiency, Doubao [93] integrates seamlessly with Douyin for video-content searches. Yuanbao [94] redefines "search-as-service" by embedding within WeChat's ecosystem. Its three-layer architecture—base model (trillion-parameter MoE), industry-specific tuning (e.g., medical diagnostics), and mini-program integration—enables seamless service execution (e.g., generating travel itineraries with bookings). This ecosystem approach has driven rapid adoption. Nano AI [95] is China's first "super search agent" that autonomously plans tasks (e.g., travel itineraries, market reports) by integrating data from walled gardens. Its DeepSearch technology parses tables, formulas, and video comments, enabling cross-platform verification for reliable decision-making. Kimi [96] can process 200 K-context windows, ideal for academic paper analysis. Users highlight its semantic search for Chinese literature. DeepSeek Search [99] represents a paradigm shift in cost-efficient, open-source AI search. Quark DeepSearch [97] relies on Qwen-QWQ inference model. Unlike traditional search engines that rely on keyword matching, the model understands natural language and performs semantic analysis to more accurately grasp user intent.

Domain-Specific AI Search Tools. MediSearch [100] provides evidence-based medical answers (e.g., drug interactions, treatment protocols), trusted by 74% of healthcare professionals for clinical decision support. Devv.ai [101] is a code-specific search engine offering real-time debugging snippets and GitHub integration. It supports Chinese queries but is limited to programming contexts. Consensus [102] accesses 200 M+ scientific papers, using NLP to extract hypotheses and methodologies. Researchers report 50% time savings in literature reviews.

Integrated AI Search Assistants WallesAI [103] is a browser-sidebar assistant that reads PDFs, videos, and webpages, enabling cross-document Q&A and content export. Bing Chat [104], deeply integrated into Edge's ecosystem, delivers citation-backed answers through real-time web indexing and source attribution, establishing a unified search-browser experience.

7 Challenges and Future Research

Despite the notable progress, this field still faces many unresolved challenges, and there is considerable room for improvement. We finally highlight several promising directions based on the reviewed progress:

- **Methods** More complex problems lead to a prolonged search process and additional actions, resulting in an extended search context. This extended context can limit the effectiveness of AIS methods and the ability of LLMs, causing search performance to degrade as the inference length increases.
- **Evaluations** There is a strong need for systematic and standardized evaluation frameworks in AI search. The datasets used for evaluation should be meticulously curated to closely resemble real-world scenarios, featuring complex, dynamic, and citation-supported answers.
- **Applications** The potential real-world applications of AI Search are significant. Beyond user scenarios, there are numerous applications across various industries. We hope to see the development of more AIS software and products to enhance the interaction between humans and machines.

8 Conclusion

Seeking and accessing information is a fundamental daily need for humans. In this survey, we provide a thorough overview of the latest research on AI Search based on LLMs. Our goal is to identify and highlight areas that require further research and suggest potential avenues for future studies. We start by introducing the traditional information retrieval systems, large language models (LLMs), and AI Search based on LLMs. Subsequently, we classify existing studies into four categories: Text-based AI Search, Web Browsing Agent, Multimodal AI Search, and Benchmarks. Then, we spotlight a range of current and significant Software and products within the realm of AI search. Finally, we discuss the limitations of the current AI search methods and explore promising future directions.

References

[1] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[2] Pavel Berkhin. A survey on pagerank computing. *Internet mathematics*, 2(1):73–120, 2005.

[3] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96, 2005.

[4] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, 2011.

[5] Mei Kobayashi and Koichi Takeda. Information retrieval on the web. *ACM computing surveys (CSUR)*, 32(2):144–173, 2000.

[6] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.

[7] ChatGPT Deep Research. https://openai.com/index/introducing-deep-research, 2022.

[8] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[9] Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[10] Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: improving retrieval-augmented lms with compression and selective augmentation. *CoRR*, abs/2310.04408, 2023.

[11] Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376. Association for Computational Linguistics, December 2023.

[12] Jiajie Jin, Xiaoxi Li, Guanting Dong, Yuyao Zhang, Yutao Zhu, Yongkang Wu, Zhonghua Li, Qi Ye, and Zhicheng Dou. Hierarchical document refinement for long-context retrieval-augmented generation, 2025.

[13] Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 996–1009, Singapore, December 2023. Association for Computational Linguistics.

[14] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. REPLUG: retrieval-augmented black-box language models. *CoRR*, abs/2301.12652, 2023.

[15] Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. Generate rather than retrieve: Large language models are strong context generators. *arXiv preprint arXiv:2209.10063*, 2022.

[16] Haoyu Wang, Tuo Zhao, and Jing Gao. Blendfilter: Advancing retrieval-augmented large language models via query generation blending and knowledge filtering, 2024.

[17] Yile Wang, Peng Li, Maosong Sun, and Yang Liu. Self-knowledge guided retrieval augmentation for large language models. *arXiv preprint arXiv:2310.05002*, 2023.

[18] Hongru Wang, Boyang Xue, Baohang Zhou, Tianhua Zhang, Cunxiang Wang, Guanhua Chen, Huimin Wang, and Kam-fai Wong. Self-dc: When to retrieve and when to generate? self divide-and-conquer for compositional unknown questions. *arXiv preprint arXiv:2402.13514*, 2024.

[19] Hanxing Ding, Liang Pang, Zihao Wei, Huawei Shen, and Xueqi Cheng. Retrieve only when it needs: Adaptive retrieval augmentation for hallucination mitigation in large language models, 2024.

[20] Jiejun Tan, Zhicheng Dou, Yutao Zhu, Peidong Guo, Kun Fang, and Ji-Rong Wen. Small models, big insights: Leveraging slim proxy models to decide when and what to retrieve for llms. *CoRR*, abs/2402.12052, 2024.

[21] Shunyu Yao, Jeffrey Zhao, Dian Yu, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

[22] Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy, 2023.

[23] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*, 2022.

[24] Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7969–7992. Association for Computational Linguistics, 2023.

[25] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *CoRR*, abs/2310.11511, 2023.

[26] Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. *CoRR*, abs/2501.05366, 2025.

[27] Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. *CoRR*, abs/2504.21776, 2025.

[28] Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webdancer: Towards autonomous information seeking agency, 2025.

[29] Lisheng Huang, Yichen Liu, Jinhao Jiang, Rongxiang Zhang, Jiahao Yan, Junyi Li, and Wayne Xin Zhao. Manusearch: Democratizing deep search in large language models with a transparent and open multi-agent framework, 2025.

[30] Tiannuo Yang, Zebin Yao, Bowen Jin, Lixiao Cui, Yusen Li, Gang Wang, and Xiaoguang Liu. Demystifying and enhancing the efficiency of large language model based search agents, 2025.

[31] Jiajie Jin, Xiaoxi Li, Guanting Dong, Yuyao Zhang, Yutao Zhu, Yang Zhao, Hongjin Qian, and Zhicheng Dou. Decoupled planning and execution: A hierarchical reasoning framework for deep search, 2025.

[32] Weiqi Wu, Xin Guan, Shen Huang, Yong Jiang, Pengjun Xie, Fei Huang, Jiuxin Cao, Hai Zhao, and Jingren Zhou. Masksearch: A universal pre-training framework to enhance agentic search capability, 2025.

[33] Liang Wang, Haonan Chen, Nan Yang, Xiaolong Huang, Zhicheng Dou, and Furu Wei. Chain-of-retrieval augmented generation, 2025.

[34] Zhicheng Lee, Shulin Cao, Jinxin Liu, Jiajie Zhang, Weichuan Liu, Xiaoyin Che, Lei Hou, and Juanzi Li. Rearag: Knowledge-guided reasoning enhances factuality of large reasoning models with iterative retrieval augmented generation, 2025.

[35] Zhengliang Shi, Lingyong Yan, Dawei Yin, Suzan Verberne, Maarten de Rijke, and Zhaochun Ren. Iterative self-incentivization empowers large language models as agentic searchers, 2025.

[36] Minda Hu, Tianqing Fang, Jianshu Zhang, Junyu Ma, Zhisong Zhang, Jingyan Zhou, Hongming Zhang, Haitao Mi, Dong Yu, and Irwin King. Webcot: Enhancing web agent reasoning by reconstructing chain-of-thought in reflection, branching, and rollback, 2025.

[37] Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, and Jie Zhou. Deeprag: Thinking to retrieve step by step for large language models, 2025.

[38] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning, 2025.

[39] Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning, 2025.

[40] Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. Research: Learning to reason with search for llms via reinforcement learning, 2025.

[41] Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. Zerosearch: Incentivize the search capability of llms without searching, 2025.

[42] Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments, 2025.

[43] Jianbiao Mei, Tao Hu, Daocheng Fu, Licheng Wen, Xuemeng Yang, Rong Wu, Pinlong Cai, Xinyu Cai, Xing Gao, Yu Yang, Chengjun Xie, Botian Shi, Yong Liu, and Yu Qiao. O^2-searcher: A searching-based agent model for open-domain open-ended question answering, 2025.

[44] Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. Websailor: Navigating super-human reasoning for web agent, 2025.

[45] Ziliang Wang, Xuhui Zheng, Kang An, Cijun Ouyang, Jialu Cai, Yuhang Wang, and Yichao Wu. Stepsearch: Igniting llms search ability via step-wise proximal policy optimization, 2025.

[46] Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, Hyokun Yun, and Lihong Li. Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning, 2025.

[47] Hanyu Lai, Xiao Liu, Iat Long Iong, Shuntian Yao, Yuxuan Chen, Pengbo Shen, Hao Yu, Hanchen Zhang, Xiaohan Zhang, Yuxiao Dong, and Jie Tang. Autowebglm: A large language model-based web navigating agent. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 5295–5306, New York, NY, USA, 2024. Association for Computing Machinery.

[48] Yueqi Song, Frank Xu, Shuyan Zhou, and Graham Neubig. Beyond browsing: Api-based web agents, 2025.

[49] Danqing Zhang, Balaji Rama, Jingyi Ni, Shiying He, Fu Zhao, Kunyu Chen, Arnold Chen, and Junyu Cao. Litewebagent: The open-source suite for vlm-based web-agent applications, 2025.

[50] Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik Chaudhari, George Karypis, and Huzefa Rangwala. Agentoccam: A simple yet strong baseline for llm-based web agents, 2025.

[51] Wenxuan Shi, Haochen Tan, Chuqiao Kuang, Xiaoguang Li, Xiaozhe Ren, Chen Zhang, Hanting Chen, Yasheng Wang, Lifeng Shang, Fisher Yu, and Yunhe Wang. Pangu deepdiver: Adaptive search intensity scaling via open-web reinforcement learning, 2025.

[52] Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. WebVoyager: Building an end-to-end web agent with large multimodal models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6864–6890, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[53] Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webdancer: Towards autonomous information seeking agency, 2025.

[54] Huawen Shen, Chang Liu, Gengluo Li, Xinlong Wang, Yu Zhou, Can Ma, and Xiangyang Ji. Falcon-ui: Understanding gui before following user instructions, 2024.

[55] Junhee Cho, Jihoon Kim, Daseul Bae, Jinho Choo, Youngjune Gwon, and Yeong-Dae Kwon. Caap: Context-aware action planning prompting to solve computer tasks with front-end ui only, 2024.

[56] Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Weixian Lei, Lijuan Wang, and Mike Zheng Shou. Showui: One vision-language-action model for gui visual agent, 2024.

[57] Tri Cao, Chengyu Huang, Yuexin Li, Wang Huilin, Amy He, Nay Oo, and Bryan Hooi. Phishagent: A robust multimodal agent for phishing webpage detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(27):27869–27877, Apr. 2025.

[58] Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, Pengshuo Qiu, Pan Lu, Zehui Chen, Guanglu Song, Peng Gao, Yu Liu, et al. Mmsearch: Unveiling the potential of large models as multi-modal search engines. In *The Thirteenth International Conference on Learning Representations*.

[59] Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. Mmsearch-r1: Incentivizing lmms to search, 2025.

[60] Vardaan Pahuja, Yadong Lu, Corby Rosset, Boyu Gou, Arindam Mitra, Spencer Whitehead, Yu Su, and Ahmed Awadallah. Explorer: Scaling exploration-driven web trajectory synthesis for multimodal web agents. 2025.

[61] Gaurav Verma, Rachneet Kaur, Nishan Srishankar, Zhen Zeng, Tucker Balch, and Manuela Veloso. Adaptagent: Adapting multimodal web agents with few-shot learning from human demonstrations, 2024.

[62] Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Hongming Zhang, Tianqing Fang, Zhenzhong Lan, and Dong Yu. Openwebvoyager: Building multimodal web agents via iterative real-world exploration, feedback and optimization, 2024.

[63] Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*, 2024.

[64] Sai Surya Gadiraju, Duoduo Liao, Akhila Kudupudi, Santosh Kasula, and Charitha Chalasani. Infotech assistant: A multimodal conversational agent for infotechnology web portal queries. In *2024 IEEE International Conference on Big Data (BigData)*, pages 3264–3272, 2024.

[65] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

[66] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[67] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*, 2022.

[68] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

[69] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

[70] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

[71] Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. Kilt: a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv:2009.02252*, 2020.

[72] Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*, 2023.

[73] Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

[74] Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, et al. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese. *arXiv preprint arXiv:2504.19314*, 2025.

[75] Boyu Gou, Zanming Huang, Yuting Ning, Yu Gu, Michael Lin, Weijian Qi, Andrei Kopanev, Botao Yu, Bernal Jiménez Gutiérrez, Yiheng Shu, Chan Hee Song, Jiaman Wu, Shijie Chen, Hanane Nour Moussa, Tianshu Zhang, Jian Xie, Yifei Li, Tianci Xue, Zeyi Liao, Kai Zhang, Boyuan Zheng, Zhaowei Cai, Viktor Rozgic, Morteza Ziyadi, Huan Sun, and Yu Su. Mind2web 2: Evaluating agentic search with agent-as-a-judge, 2025.

[76] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 28091–28114. Curran Associates, Inc., 2023.

[77] Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents, 2024.

[78] Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 881–905, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[79] Divyansh Garg, Shaun VanWeelden, Diego Caples, Andis Draguns, Nikil Ravi, Pranav Putta, Naman Garg, Tomas Abraham, Michael Lara, Federico Lopez, James Liu, Atharva Gundawar, Prannay Hebbar, Youngchul Joo, Jindong Gu, Charles London, Christian Schroeder de Witt, and Sumeet Motwani. Real: Benchmarking autonomous agents on deterministic simulations of real websites, 2025.

[80] Atsuyuki Miyai, Zaiying Zhao, Kazuki Egashira, Atsuki Sato, Tatsumi Sunada, Shota Onohara, Hiromasa Yamanishi, Mashiro Toyooka, Kunato Nishina, Ryoma Maeda, Kiyoharu Aizawa, and Toshihiko Yamasaki. Webchorearena: Evaluating web browsing agents on realistic tedious web tasks, 2025.

[81] Yixiao Song, Katherine Thai, Chau Minh Pham, Yapei Chang, Mazin Nadaf, and Mohit Iyyer. Bearcubs: A benchmark for computer-using web agents, 2025.

[82] Yichen Pan, Dehan Kong, Sida Zhou, Cheng Cui, Yifei Leng, Bing Jiang, Hangyu Liu, Yanyi Shang, Shuyan Zhou, Tongshuang Wu, and Zhengyang Wu. Webcanvas: Benchmarking web agents in online environments, 2024.

[83] Kevin Xu, Yeganeh Kordi, Tanay Nayak, Adi Asija, Yizhong Wang, Kate Sanders, Adam Byerly, Jingyu Zhang, Benjamin Van Durme, and Daniel Khashabi. TurkingBench: A challenge benchmark for web agents. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3694–3710, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[84] Ada Defne Tur, Nicholas Meade, Xing Han Lù, Alejandra Zambrano, Arkil Patel, Esin Durmus, Spandana Gella, Karolina Stańczak, and Siva Reddy. Safearena: Evaluating the safety of autonomous web agents, 2025.

[85] Yuxuan Zhu, Antony Kellermann, Dylan Bowman, Philip Li, Akul Gupta, Adarsh Danda, Richard Fang, Conner Jensen, Eric Ihli, Jason Benn, Jet Geronimo, Avi Dhir, Sudhit Rao, Kaicheng Yu, Twm Stone, and Daniel Kang. Cve-bench: A benchmark for ai agents' ability to exploit real-world web application vulnerabilities, 2025.

[86] Ivan Evtimov, Arman Zharmagambetov, Aaron Grattafiori, Chuan Guo, and Kamalika Chaudhuri. Wasp: Benchmarking web agent security against prompt injection attacks, 2025.

[87] Haoyi Qiu, Alexander Fabbri, Divyansh Agarwal, Kung-Hsiang Huang, Sarah Tan, Nanyun Peng, and Chien-Sheng Wu. Evaluating cultural and social awareness of LLM web agents. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3978–4005, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[88] Yaxin Luo, Zhaoyi Li, Jiacheng Liu, Jiacheng Cui, Xiaohan Zhao, and Zhiqiang Shen. Open captchaworld: A comprehensive web-based platform for testing and benchmarking multimodal llm agents, 2025.

[89] Yougang Lyu, Xiaoyu Zhang, Lingyong Yan, Maarten de Rijke, Zhaochun Ren, and Xiuying Chen. Deepshop: A benchmark for deep research shopping agents, 2025.

[90] Mingyang Fu, Yuyang Peng, Benlin Liu, Yao Wan, and Dongping Chen. Livevqa: Live visual knowledge seeking. *arXiv preprint arXiv:2504.05288*, 2025.

[91] Qinhan Yu, Zhiyou Xiao, Binghui Li, Zhengren Wang, Chong Chen, and Wentao Zhang. Mramg-bench: A beyondtext benchmark for multimodal retrieval-augmented multimodal generation. *arXiv preprint arXiv:2502.04176*, 2025.

[92] Perplexity Deep Research. https://www.perplexity.ai, 2022.

[93] Doubao. https://www.doubao.com, 2023.

[94] Yuanbao. https://yuanbao.tencent.com, 2024.

[95] Nano AI. https://www.n.cn, 2025.

[96] Kimi. https://www.kimi.com, 2023.

[97] Quark DeepSearch. https://quark.sm.cn, 2025.

[98] Gemini Deep Research. https://gemini.google/overview/deep-research, 2023.

[99] DeepSeek. https://www.deepseek.com, 2025.

[100] MediSearch. https://medisearch.io, 2023.

[101] Devv.ai. https://devv.ai/zh, 2023.

[102] Consensus. https://consensus.app, 2022.

[103] walles.ai. https://walles.ai/, 2023.

[104] Bing Chat. http://bing.com, 2023.

[105] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.

[106] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*, 2023.

[107] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer, 2003.

[108] Stephen E. Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, 2009.

[109] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP*, pages 6769–6781, 2020.

[110] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *ICLR*, 2020.

[111] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *CoRR*, abs/2212.03533, 2022.

[112] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings, 2024.

[113] Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. From matching to generation: A survey on generative information retrieval. *ACM Transactions on Information Systems*, 43(3):1–62, 2025.

[114] Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Prakash Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. Transformer memory as a differentiable search index. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[115] Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, et al. A neural corpus indexer for document retrieval. *Advances in Neural Information Processing Systems*, 35:25600–25614, 2022.

[116] Xiaoxi Li, Zhicheng Dou, Yujia Zhou, and Fangchao Liu. Corpuslm: Towards a unified language model on corpus for knowledge-intensive tasks. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 26–37, 2024.

[117] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

[118] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM, 2020.

[119] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is chatgpt good at search? investigating large language models as re-ranking agents. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 14918–14937. Association for Computational Linguistics, 2023.

[120] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*, 2020.

[121] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR, 2020.

[122] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR, 2022.

[123] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore, December 2023. Association for Computational Linguistics.

[124] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*, 2023.

[125] Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. Augmentation-adapted retriever improves generalization of language models as generic plug-in. *arXiv preprint arXiv:2305.17331*, 2023.

[126] Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. Retrieve anything to augment large language models. *CoRR*, abs/2310.07554, 2023.

[127] Lingxi Zhang, Yue Yu, Kuan Wang, and Chao Zhang. Arl2: Aligning retrievers for black-box large language models via self-guided adaptive relevance labeling, 2024.

[128] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023. arXiv:2307.03172.

[129] Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for rag systems, 2024.

[130] Haoyan Yang, Zhitao Li, Yong Zhang, Jianzong Wang, Ning Cheng, Ming Li, and Jing Xiao. PRCA: fitting black-box large language models for retrieval question answering via pluggable reward-driven contextual adapter. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5364–5375. Association for Computational Linguistics, 2023.

[131] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore, December 2023. Association for Computational Linguistics.

[132] Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context, 2023.

[133] Yuxuan Huang, Yihang Chen, Haozheng Zhang, Kang Li, Meng Fang, Linyi Yang, Xiaoguang Li, Lifeng Shang, Songcen Xu, Jianye Hao, Kun Shao, and Jun Wang. Deep research agents: A systematic examination and roadmap, 2025.

[134] Weizhi Zhang, Yangning Li, Yuanchen Bei, Junyu Luo, Guancheng Wan, Liangwei Yang, Chenxuan Xie, Yuyao Yang, Wei-Chieh Huang, Chunyu Miao, Henry Peng Zou, Xiao Luo, Yusheng Zhao, Yankai Chen, Chunkit Chan, Peilin Zhou, Xinyang Zhang, Chenwei Zhang, Jingbo Shang, Ming Zhang, Yangqiu Song, Irwin King, and Philip S. Yu. From web search towards agentic deep research: Incentivizing search with reasoning agents, 2025.

[135] Renjun Xu and Jingwen Peng. A comprehensive survey of deep research: Systems, methodologies, and applications, 2025.

[136] Shuang Sun, Huatong Song, Yuhao Wang, Ruiyang Ren, Jinhao Jiang, Junjie Zhang, Fei Bai, Jia Deng, Wayne Xin Zhao, Zheng Liu, Lei Fang, Zhongyuan Wang, and Ji-Rong Wen. Simpledeepsearcher: Deep information seeking via web-powered reasoning trajectory synthesis, 2025.

[137] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024.

[138] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[139] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

[140] Jian Hu, Jason Klein Liu, and Wei Shen. Reinforce++: An efficient rlhf algorithm with robustness to both prompt and reward models, 2025.

[141] Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. Websailor: Navigating super-human reasoning for web agent, 2025.

[142] Wenxuan Shi, Haochen Tan, Chuqiao Kuang, Xiaoguang Li, Xiaozhe Ren, Chen Zhang, Hanting Chen, Yasheng Wang, Lifeng Shang, Fisher Yu, and Yunhe Wang. Pangu deepdiver: Adaptive search intensity scaling via open-web reinforcement learning, 2025.

[143] Yaorui Shi, Sihang Li, Chang Wu, Zhiyuan Liu, Junfeng Fang, Hengxing Cai, An Zhang, and Xiang Wang. Search and refine during think: Autonomous retrieval-augmented reasoning of llms, 2025.

[144] Guanting Dong, Yifei Chen, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui Zhou, Zhicheng Dou, and Ji-Rong Wen. Tool-star: Empowering llm-brained multi-tool reasoner via reinforcement learning, 2025.

[145] Chenyu Lin, Yilin Wen, Du Su, Fei Sun, Muhan Chen, Chenfu Bao, and Zhonghou Lv. Knowledgeable-r1: Policy optimization for knowledge exploration in retrieval-augmented generation, 2025.

[146] Hongjin Qian and Zheng Liu. Scent of knowledge: Optimizing search-enhanced reasoning with information foraging, 2025.

[147] Yuan Li, Qi Luo, Xiaonan Li, Bufan Li, Qinyuan Cheng, Bo Wang, Yining Zheng, Yuxin Wang, Zhangyue Yin, and Xipeng Qiu. R3-rag: Learning step-by-step reasoning and retrieval for llms via reinforcement learning, 2025.

[148] Dingchu Zhang, Yida Zhao, Jialong Wu, Baixuan Li, Wenbiao Yin, Liwen Zhang, Yong Jiang, Yufeng Li, Kewei Tu, Pengjun Xie, and Fei Huang. Evolvesearch: An iterative self-evolving search agent, 2025.

[149] Zeyang Sha, Shiwen Cui, and Weiqiang Wang. Sem: Reinforcement learning for search-efficient large language models, 2025.

[150] Peilin Wu, Mian Zhang, Xinlu Zhang, Xinya Du, and Zhiyu Zoey Chen. Search wisely: Mitigating sub-optimal agentic searches by reducing uncertainty, 2025.

[151] Pengcheng Jiang, Xueqiang Xu, Jiacheng Lin, Jinfeng Xiao, Zifeng Wang, Jimeng Sun, and Jiawei Han. s3: You don't need that much data to train a search agent via rl, 2025.

[152] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors, 2023.

[153] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6), March 2024.

[154] Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. Webwalker: Benchmarking llms in web traversal, 2025.

[155] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

[156] Yizhang Jin, Jian Li, Yexin Liu, Tianjun Gu, Kai Wu, Zhengkai Jiang, Muyang He, Bo Zhao, Xin Tan, Zhenye Gan, et al. Efficient multimodal large language models: A survey. *arXiv preprint arXiv:2405.10739*, 2024.

[157] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[158] OpenAI. Hello gpt-4o, 2024.

[159] Anthropic. Claude 3.5 sonnet, 2024.

[160] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

[161] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[162] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[163] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.

[164] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[165] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[166] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Zhong Muyan, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023.

[167] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Zhengxiong Luo, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, et al. Generative multimodal models are in-context learners. *arXiv preprint arXiv:2312.13286*, 2023.

[168] Jian Li, Weiheng Lu, Hao Fei, Meng Luo, Ming Dai, Min Xia, Yizhang Jin, Zhenye Gan, Ding Qi, Chaoyou Fu, et al. A survey on benchmarks of multimodal large language models. *arXiv preprint arXiv:2408.08632*, 2024.

[169] Shuai Wang, Weiwen Liu, Jingxuan Chen, Yuqi Zhou, Weinan Gan, Xingshan Zeng, Yuhan Che, Shuai Yu, Xinlong Hao, Kun Shao, et al. Gui agents with foundation models: A comprehensive survey. *arXiv preprint arXiv:2411.04890*, 2024.

[170] Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*, 2024.

[171] Vardaan Pahuja, Yadong Lu, Corby Rosset, Boyu Gou, Arindam Mitra, Spencer Whitehead, Yu Su, and Ahmed Awadallah. Explorer: Scaling exploration-driven web trajectory synthesis for multimodal web agents. *arXiv preprint arXiv:2502.11357*, 2025.

[172] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*, 2020.

[173] Marcos Fernández-Pichel, Juan C Pichel, and David E Losada. Evaluating search engines and large language models for answering health questions. *arXiv preprint arXiv:2407.12468*, 2024.

[174] You.com. https://you.com, 2021.