NExT-Search: Rebuilding User Feedback Ecosystem for Generative AI Search

Sunhao Dai
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
sunhaodai@ruc.edu.cn

Wenjie Wang*
University of Science and Technology of China
Hefei, China
wenjiewang96@gmail.com

Liang Pang*
CAS Key Laboratory of AI Safety
Institute of Computing Technology
Chinese Academy of Sciences
Beijing, China
pangliang@ict.ac.cn

Jun Xu
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
junxu@ruc.edu.cn

See-Kiong Ng
National University of Singapore
Singapore
seekiong@nus.edu.sg

Ji-Rong Wen
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
jrwen@ruc.edu.cn

Tat-Seng Chua
National University of Singapore
Singapore
dcscts@nus.edu.sg

Abstract

Generative AI search driven by large language models (LLMs) is reshaping information retrieval by offering end-to-end answers to complex queries, reducing users' reliance on manually browsing and summarizing multiple web pages. However, while this paradigm enhances convenience, it disrupts the feedback-driven improvement loop that has historically powered the evolution of traditional Web search. Web search can continuously improve their ranking models by collecting large-scale, fine-grained user feedback (e.g., clicks, dwell time) at the document level. In contrast, generative AI search operates through a much longer search pipeline—spanning query decomposition, document retrieval, and answer generation—yet typically receives only coarse-grained feedback on the final answer. This introduces a *feedback loop disconnect*, where user feedback for the final output cannot be effectively mapped back to specific system components, making it difficult to improve each intermediate stage and sustain the feedback loop.

To address this limitation, we envision **NExT-Search**, a next-generation paradigm designed to reintroduce fine-grained, process-level feedback into generative AI search. NExT-Search integrates two complementary modes: **User Debug Mode**, which allows engaged users to intervene at key stages—such as refining query decomposition, rating retrieved documents, and editing initial generated responses—and **Shadow User Mode**, where a personalized user agent simulates user preferences and provides AI-assisted feedback for less interactive users. As these feedback signals serve as valuable resources for refining the whole search pipeline, we also introduce a feedback store mechanism that encourages users to share and monetize their debugging efforts, further incentivizing participation. Furthermore, we envision how these feedback signals can be leveraged through **online adaptation**, which refines current search outputs in real-time, and **offline update**, which aggregates interaction logs to periodically fine-tune query decomposition, retrieval, and generation models. By restoring human control over key stages of the generative AI search pipeline, we believe NExT-Search offers a promising direction for building feedback-rich AI search systems that can evolve continuously alongside human feedback.

CCS Concepts

• **Information systems → Users and interactive retrieval**.

Keywords

Generative AI Search, User Feedback, Large Language Model

ACM Reference Format:
Sunhao Dai, Wenjie Wang, Liang Pang, Jun Xu, See-Kiong Ng, Ji-Rong Wen and Tat-Seng Chua. 2025. NExT-Search: Rebuilding User Feedback Ecosystem for Generative AI Search. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3726302.3730353

1 Introduction

Search engines have served as a primary gateway to information access for decades, helping users address an enormous spectrum of information needs [7, 9, 31]. Despite continuous advances in ranking algorithms [6, 43], user interface design [16], and large-scale log data analysis [8, 20], recent estimates suggest that nearly

*Corresponding authors

SIGIR '25, Padua, Italy

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3730353

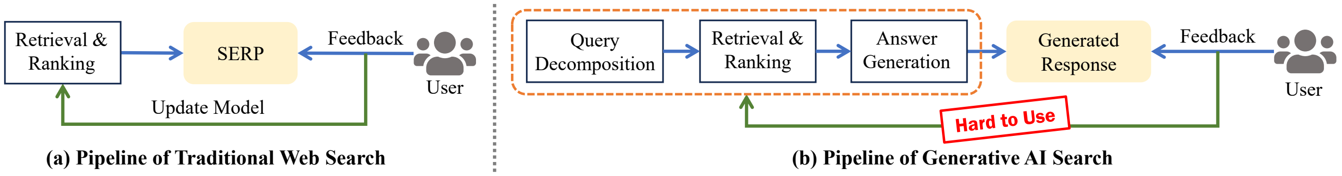

Figure 1: Comparison of the paradigm of traditional web search engines and generative AI search engines. (a) Web search retrieves and ranks results, presenting them as a Search Engine Results Page (SERP), where user feedback on the document level can be directly leveraged to update the ranking model. (b) Generative AI search automates multiple steps to generate direct answers, but its extended pipeline complicates the effective use of user feedback for refining individual components.

half of all Web search queries fail to yield relevant results [1]. Many of these queries lie on the *complex* end of the search spectrum: they require users to break down their search goal and iteratively check different returned search results, aggregating disparate pieces of information in a manual, time-consuming process [15, 41].

Recently, large language model (LLM)–driven generative AI search systems have promised to address these more complex queries in an end-to-end fashion [26, 42, 46, 49]. Users can now pose open-ended or creative requests (e.g., *"Plan a trip to attend SIGIR 2025"*), and the generative AI search automatically parses them, retrieves relevant documents in smaller chunks, and then aggregates and synthesizes the extracted information into a cohesive answer [11, 12]. Thus, by automating many of the steps traditionally performed by users in Web search, generative AI search *expands the task boundary* of search, enabling users to solve complex, multi-faceted queries with reduced effort and cognitive load.

However, by first revisiting and comparing the foundational paradigms of both traditional Web search and advanced generative AI search in Section 2 and Figure 1, we find that while the paradigm of generative AI search introduces notable advantages, it also loses a critical component that has historically driven the success of traditional search: **the user feedback ecosystem**.

- **Traditional Web search** thrives on a *feedback-driven improvement loop*, where ranking models continuously evolve by leveraging large-scale user feedback on search results. Users can provide feedback at the document level, such as clicks, dwell time, and bounce rates. These fine-grained signals serve as direct supervision for refining retrieval and ranking models, enabling search engines to iteratively enhance result relevance and improve search quality over time [1, 20, 21, 23].
- **Generative AI search** operates through a much longer pipeline that directly synthesizes answers, significantly reducing human control over the search process. Users can only provide feedback in coarse-grained forms to the final generated response, such as simple likes/dislikes or written comments [41]. This feedback loop disconnect prevents effective attribution of user dissatisfaction to specific pipeline components—whether query decomposition, retrieval, or answer generation—making it difficult to improve intermediate stages.

Thus, while generative AI search systems such as Microsoft Bing Copilot [2] and Perplexity AI [3] gained remarkable traction in their early launches, they still account for only a small share of the global search engine market [4]. Addressing these limitations is crucial for enabling generative AI search engines to achieve scalable, iterative optimization while retaining the benefits of automation.

To this end, we envision a new paradigm called **NExT-Search**, aimed at restore—and potentially enhance—the feedback-driven ecosystem for generative AI search. The central idea is to incorporate two complementary modes of interaction: an active **User Debug Mode**, which enables users to engage with and refine each stage of the search pipeline, and a passive **Shadow User Mode**, which leverages a personalized user agent to simulate feedback when users prefer minimal involvement. In the User Debug Mode, users can examine and modify different stages of the generative search workflow—such as query decomposition, document retrieval, and answer generation—providing fine-grained feedback as needed. In contrast, the Shadow User Mode engages a personalized user agent that learns from past interactions and user profiles to provide AI-assisted feedback throughout the pipeline, thereby reducing user effort while maintaining the flow of valuable supervision signals. Together, these two modes offer a potential path toward gathering higher-quality feedback at various junctures in the pipeline.

Building on the fine-grained feedback envisioned in the NExT-Search paradigm, we further discuss how such feedback could be leveraged through two complementary update strategies. First, **online adaptation** enables generative AI search to refine current sessions in real time—for instance, re-ranking documents or partially regenerating answers in response to new feedback from either users or a personalized agent. Second, **offline update** aggregate logs from multiple interactions to periodically retrain or fine-tune crucial pipeline modules, preserving the iterative feedback-driven improvement loop that once propelled the success of traditional Web search. To further incentivize user participation, we envision a **Feedback Store** mechanism that enables users to share and potentially monetize their debugging contributions, making feedback not only a technical asset but also a user-valued commodity. In this way, our NExT-Search paradigm has the potential to not only enhance search quality and user experience but also foster a mutually beneficial ecosystem where users are rewarded for their engagement while driving continuous model improvements.

[1] https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web/

[2] https://copilot.microsoft.com

[3] https://www.perplexity.ai

[4] https://gs.statcounter.com/search-engine-market-share

Finally, as a perspective paper, our aim is to spark new thinking about why and how user feedback should be reimagined in the evolving paradigm of generative AI search. To this end, we also outline three promising research directions: building personalized user simulators to generate realistic AI-assisted feedback at scale, designing human-centric interfaces to make pipeline-level debugging accessible and efficient, and developing learning algorithms that can effectively leverage both human and AI-assisted feedback. Together, these directions chart a roadmap for building next-generation AI search systems that can continuously evolve in tandem with human feedback.

In summary, the main contributions of this paper are as follows:

- We provide a systematic analysis of the transition from traditional Web search to generative AI search, highlighting the core reasons why generative AI search has struggled to achieve large-scale success—chiefly, the loss of rich user feedback loops.
- We envision a new paradigm for generative AI search, called NExT-Search, which aims to rebuild the user feedback ecosystem with two modes: a *User Debug Mode* that enables step-by-step user debugging across the pipeline and a *Shadow User Mode* that simulates user feedback using personalized user agents to support minimal-interaction users.
- We outline how fine-grained feedback can be utilized through both online adaptation and offline model updates, and propose a feedback store to incentivize user participation, laying the foundation for a sustainable and self-improving search ecosystem.

2 Web Search vs. Generative AI Search

In this section, we first revisit the pipelines of both traditional Web search and advanced generative AI search. By comparing the two pipelines, we underscore how the shift from a retrieval system with web page ranking lists to an end-to-end natural language answer-generating system leads to both gains and losses.

2.1 The Pipeline of Traditional Web Search

Traditional Web search engines, such as Google and Bing, have evolved over decades of research and industrial practice in information retrieval (IR) [7, 9, 31]. As shown in Figure1(a), the core pipeline typically consists of the following stages:

(1) Retrieval & Ranking. Upon receiving a user query (typically expressed as a set of keywords), the search engine initiates the retrieval process by identifying candidate documents from its index. This process leverages a hybrid approach that combines traditional keyword matching techniques (e.g., BM25 [35]) with advanced semantic or embedding-based matching methods [13, 25, 43] (e.g., DSSM [18]). Following retrieval, the system usually applies a learning-to-rank (LTR) model [29] that integrates multiple features, including term-matching signals, link-based authority metrics (e.g., PageRank), user behavior patterns (e.g., click-through rates), temporal relevance indicators, and other contextual signals.

(2) Result Delivery & User Feedback. The retrieval and ranking stage then produces a Search Engine Results Page (SERP) [17], which generally lists tens or hundreds of hyperlinks accompanied by titles and snippets [22]. Users then interact with and provide implicit feedback on these results, primarily through measurable engagement metrics such as click-through patterns, dwell time duration, bounce rates, and other interaction signals, all of which are systematically captured in real-time logs and subjected to continuous computational analysis [1, 20, 21, 23].

(3) Model Update. The large-scale feedback signals collected from user interactions then serve as direct supervision for refining ranking and retrieval models. User engagement at the document level—such as clicks on relevant results and skips on irrelevant ones—acts as a learning signal that continuously informs the ranking function, optimizing document relevance estimation.

The above stages drive an iterative refinement process for traditional web search known as the **data flywheel** [14, 24]: the more interactions occur, the more feedback is accumulated; the more training data is available, the better the ranking model becomes; and as search quality improves, more users engage with the system, further reinforcing the cycle. This feedback-driven improvement loop has been fundamental to the scalability and success of traditional Web search engines (e.g., Google and Bing).

2.2 The Pipeline of Generative AI Search

While traditional Web search has been highly effective in retrieving relevant documents, it struggles with complex or multi-step queries. Users often need to iteratively refine their searches, aggregate information from multiple sources, and manually synthesize answers to fulfill their information needs. Recent advancements in LLMs have given rise to the emergence of generative AI search engines, enabling them to go beyond hyperlink retrieval and provide end-to-end, synthesized answers tailored to user queries [26, 42, 49]. Instead of relying on users to extract and combine information, generative AI search automates the entire process—from decomposition and retrieval to synthesis—providing more direct responses. A representative pipeline of this process is illustrated in Figure 1(b).

(1) Query Decomposition. Generative AI search systems often adopt a retrieval-augmented generation (RAG) framework [12, 28, 41], which begins with decomposing complex user queries into one or more coherent sub-queries. This decomposition enables the system to iteratively refine and address partial information needs, ensuring a more precise and contextually relevant final response. By breaking down multifaceted queries, the system can better align with user intent and improve the overall search experience.

(2) Retrieval & Ranking. Following task decomposition, the system retrieves and ranks relevant passages for each sub-query, akin to traditional search engines. However, generative AI search typically operates at a finer granularity, retrieving and ranking text chunks or paragraphs rather than entire documents [12]. This approach ensures that the retrieved evidence is both semantically aligned with the user's intent and sufficiently detailed to support high-quality answer generation.

(3) Answer Generation. The core of generative AI search lies in its ability to synthesize retrieved evidence into a coherent, natural-language response using LLMs [26]. The LLM integrates information from multiple sources, producing fluent and contextually rich answers that directly address the user's query. To enhance transparency and trustworthiness, the system may optionally include citations or references to the original sources, allowing users to verify the information and trace its provenance [28].

(4) Result Delivery & User Feedback. Generative AI search engines present their outputs through conversational or chat-style interfaces, offering users a summarized and digestible answer in a single interaction. However, the user feedback in generative AI search is often more sparse than in traditional systems, typically limited to simple likes/dislikes or brief written comments to the final answer, posing challenges for fine-grained system improvement.

Unlike traditional Web search, which benefits from fine-grained feedback at the document level, the coarse-grained feedback from generative AI search primarily presents a fundamental challenge: dissatisfaction with the final answer does not directly indicate whether errors originated from query decomposition, retrieval, or answer generation, making it difficult to pinpoint and correct specific weaknesses in the pipeline.

2.3 Comparison and Analysis

After illustrating the pipelines of both traditional Web search and generative AI search, we now summarize the key shifts in these two paradigms, highlighting the gains in usability and efficiency, as well as the challenges posed by reduced human control and weakened feedback loops:

- **Potential for End-to-End Solutions.** Generative AI search demonstrates significant potential for addressing complex, multi-faceted queries that require synthesis and creativity, surpassing the capabilities of traditional top-10 hyperlink-based results. However, this capability comes with the risk of generating inaccurate or hallucinated content, underscoring the need for robust mechanisms to ensure factual grounding and reliability.
- **Search Result Delivery.** Traditional search engines provide users with a list of links and snippets, enabling direct comparison and exploration of multiple sources. In contrast, generative AI search delivers a unified, synthesized answer, streamlining the user experience but potentially sacrificing transparency and human control for the entire search process.
- **User Feedback Mechanism.** Traditional search systems benefit from a rich and granular feedback loop, driven by user interactions such as clicks and dwell time. These signals enable continuous refinement of retrieval and ranking models. Generative AI search, however, typically relies on coarser feedback mechanisms, such as binary likes/dislikes or free-text corrections, which provide limited insights into specific failures within the pipeline (e.g., query decomposition, relevant document retrieval, or answer generation).

In summary, the evolution from traditional link-based retrieval systems to generative AI search, which provides direct answers, has undeniably enhanced user convenience and streamlined the search experience. However, this shift introduces a **fundamental challenge**: the once-critical feedback loop that fueled constant improvements in Web search engines is now at risk of stalling. Generative search engines consolidate multiple user-driven interactions—such as decomposing information needs into sub-queries, selecting and examining multiple SERPs, and aggregating knowledge from retrieved documents—into an end-to-end pipeline. While this integration significantly improves usability, it simultaneously reduces the granularity of user feedback, making it difficult to diagnose and address specific weaknesses in the system. For instance, user dissatisfaction with the final answer could arise from various underlying issues, such as flawed task decomposition, inadequate retrieval, or errors in the LLM's summarization process. Yet, feedback limited to the final output fails to provide the necessary insights to pinpoint the root cause of these failures across the complex pipeline. This limitation poses a significant barrier to iterative refinement and large-scale self-improvement, which have long been successful hallmarks of traditional search systems.

3 NExT-Search Paradigm

In this section, we present **NExT-Search**, a new paradigm aimed at rebuilding the user feedback ecosystem for generative AI search by reintegrating fine-grained user feedback throughout the whole search pipeline, akin to the success of traditional Web search.

3.1 Motivation and Overview

As discussed in Section 2.3, while generative AI search improves usability by offering end-to-end answers, it sacrifices the fine-grained feedback mechanisms that once enabled traditional Web search engines to improve iteratively. Most current systems only receive coarse signals, such as likes or dislikes for final answers, which provide little insight into which specific component—query decomposition, retrieval, or generation—led to user dissatisfaction.

To this end, **NExT-Search** is motivated by the need to reintroduce such fine-grained feedback without sacrificing the advanced user experience of modern generative AI search. As illustrated in Figure 2, NExT-Search integrates two complementary feedback mechanisms: **User Debug Mode** allows engaged users to intervene at various stages of the search pipeline—editing queries, re-ranking results, and refining generated answers—while **Shadow User Mode** employs a *personalized user agent* to infer and simulate user preferences when explicit feedback is unavailable. By combining these modes, NExT-Search enables that every search interaction—whether fully guided by users or passively inferred—contributes structured signals for continuous optimization.

In the following sections, we describe these two feedback mechanisms in detail (Section 3.2 and Section 3.3) with a concrete query example of "Plan a trip to attend SIGIR 2025" and explore their synergy (Section 3.4) and introduce an incentive mechanism to encourage user participation (Section 3.5).

3.2 User Debug Mode

In *User Debug Mode*, NExT-Search empowers users with deep, hands-on control over the generative AI search pipeline. By providing an interactive window and a transparent breakdown of each search stage, users can contribute fine-grained feedback that immediately affects system output (*online adaption* in Section 4.1) and is recorded for later model refinement (*offline update* in Section 4.2). This mode benefits scenarios where users have the motivation and expertise to debug and refine the search process, and it paves the way for more precise, interpretable, and adaptive generative AI search. In what follows, we detail how a user under this debug mode may intervene at each stage to debug the system.

3.2.1 Debug in Query Decomposition. Generative AI search engines initially attempt to break a user's broad or ambiguous query

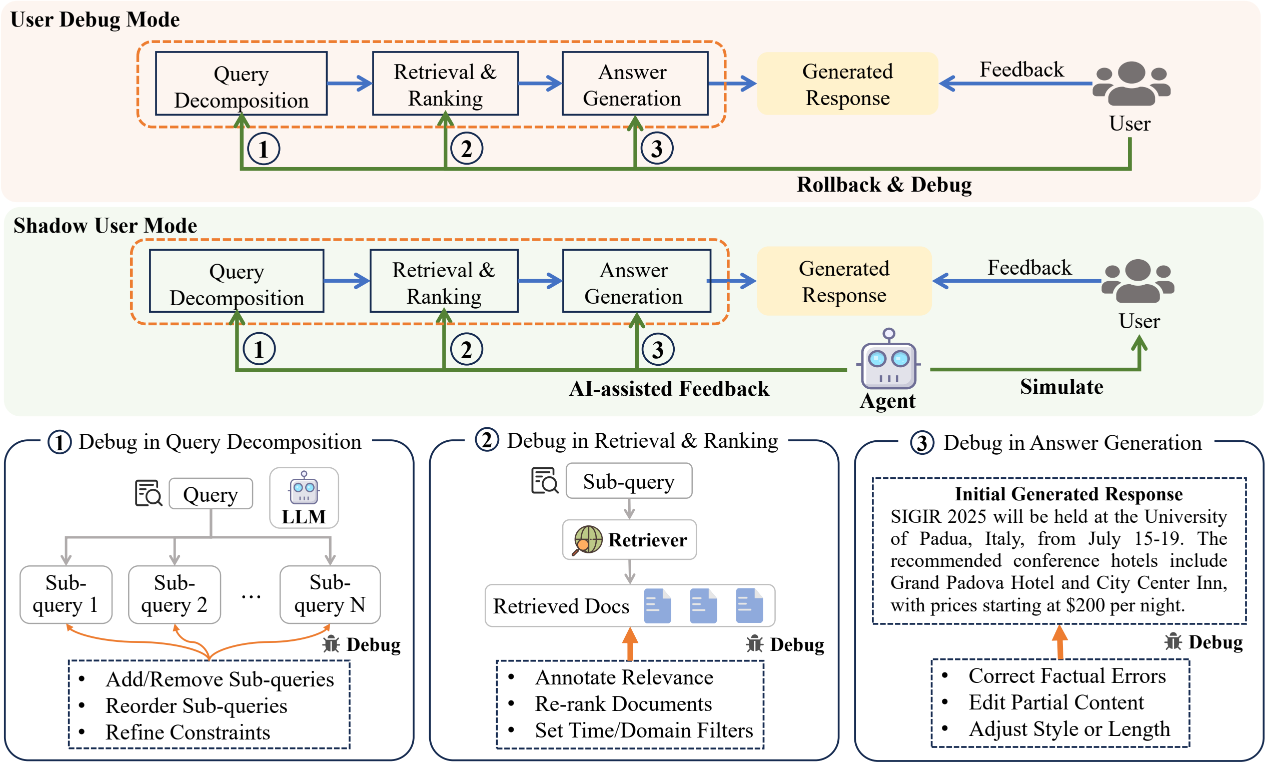

Figure 2: Illustration of our proposed NExT-Search paradigm. NExT-Search introduces a dual feedback mechanism to enhance generative AI search: In *User Debug Mode*, users can intervene at different stages—query decomposition, retrieval, and answer generation—to refine search results with granular feedback. In *Shadow User Mode*, a personalized agent simulates user behavior to assist in providing feedback with minimal user interaction, reducing user engagement costs.

into multiple sub-queries (or subtasks). However, several types of errors commonly arise at this stage:

- *Misses key subtasks*: The system may overlook essential queries required to fulfill the user's intent (e.g., omitting "registration fees and process" when planning to attend a conference).
- *Includes irrelevant queries*: Extraneous queries may be introduced, leading to inefficiencies (e.g., including "local sightseeing recommendations" when the primary goal is attending a conference).
- *Improperly orders subtasks*: Subtasks may be sequenced incorrectly, causing inefficiencies in information retrieval (e.g., searching for hotel bookings before confirming the schedule).

To tackle these issues and reintroduce fine-grained user signals, *User Debug Mode* enables the following feedback mechanisms for user debugging in this stage:

- *Add/Remove Sub-queries:* Users can introduce missing queries or remove redundant ones to align with their actual information needs (e.g., adding a query about "registration fees" or removing "local sightseeing recommendations").
- *Reorder Sub-queries:* When task dependencies exist (e.g., the user must confirm the conference dates before booking flights), users can adjust the order of execution accordingly.
- *Refine Constraints:* Users can fine-tune specific constraints within sub-queries, such as adjusting budget limits.

An example of debugging at this stage is shown below:

User Debug in Query Decomposition

Initial Query Decomposition:

- **[Q1]** What are the best flight options from [User's City] to the [conference location]?
- **[Q2]** Where and when will SIGIR 2025 be held?
- **[Q3]** What are the recommended hotels near the conference venue?
- **[Q4]** What are some sightseeing attractions near the conference venue?

User Debugging:

- **Remove [Q4]**: User's primary goal is to attend SIGIR 2025, and sightseeing is not a priority.
- **Add [Q5]**: "What is the registration process and cost for SIGIR 2025?"— a prerequisite for travel planning.
- **Reorder [Q2] and [Q1]**: Adjust order to book travel only after confirming the event schedule.

By logging each user edit (e.g., frequently added subtasks, repeatedly deleted constraints), the system amasses valuable data to refine its decomposition strategy over time.

3.2.2 Debug in Retrieval & Ranking. After decomposition, the system retrieves relevant documents or passages from indexed sources. However, common errors may occur:

- *Retrieving irrelevant documents*: Results may include off-topic or outdated content.
- *Missing high-quality or authoritative sources*: Important references such as official government websites or conference organizers' pages might be underrepresented in the retrieved results.
- *Suboptimal ranking*: The system may rank less useful documents higher than more relevant ones.

To address these retrieval and ranking issues while incorporating fine-grained user feedback, *User Debug Mode* allows users to refine the retrieval and ranking process through several mechanisms:

- *Annotate Relevance:* Users can explicitly mark documents as *relevant*, *partially relevant*, or *irrelevant*.
- *Re-rank Documents:* Users can manually adjust document priority, ensuring that the most useful sources are emphasized.
- *Set Time/Domain Filters:* Users can refine retrieval criteria by restricting results to specific timeframes or limiting sources to specific domains (e.g., only include results from "sigir.org").

An example of debugging at this stage is shown below:

User Debug in Retrieval & Ranking

Initial Retrieved Results for sub-query: "Where and when will SIGIR 2025 be held?"

- **[D1]** News article on 2025 AI conferences briefly mentioning SIGIR (Partially relevant)
- **[D2]** SIGIR 2025 announcement from ACM SIGIR website (Highly relevant)
- **[D3]** 2023 SIGIR proceedings mentioning past conference locations (Irrelevant)

User Debugging:

- **Remove [D3]**: Outdated sources.
- **Re-rank [D2] above [D1]**: Prioritize official SIGIR sources over news articles.
- **Apply domain filter**: Restrict results to "sigir.org" to focus on authoritative sources.

By logging each user edit (e.g., frequently excluded irrelevant documents, consistent source preferences, and re-ranking adjustments), the system accumulates valuable data to refine its retrieval and ranking models offline.

3.2.3 Debug in Answer Generation. After retrieving and ranking relevant documents, the system synthesizes a response with LLMs. While this process provides a seamless, end-to-end search experience, it also introduces potential issues, including:

- *Factual inaccuracies*: The model may generate hallucinated claims or misinterpret retrieved documents.
- *Incomplete or excessive information*: The response might omit important details or contain unnecessary elaboration.
- *Inappropriate style or tone*: The output may be too formal, too casual, overly technical, or lack proper structuring.

To empower users in refining the final response, *User Debug Mode* introduces the following debugging mechanisms:

- *Correct Factual Errors:* Users can highlight incorrect statements or ask for additional supporting evidence.
- *Edit Partial Content:* Users can directly modify specific sections of the response by adding, removing, or restructuring content for improved clarity and accuracy.
- *Adjust Style or Length:* Users can refine verbosity and tone based on their needs.

An example for debugging in this stage is shown below:

User Debug in Answer Generation

Initial Generated Answer:
"SIGIR 2025 will be held at the University of Padua, Italy, from July 15-19. The recommended conference hotels include NH Hotel Padova and Best Western Hotel Biri, with prices starting at 120€ per night."

User Debugging:

- **Correct Factual Error:** The conference dates and venue are incorrect; verify with the official SIGIR website.
- **Edit Partial Content:** Instead of listing only two hotels, request more budget-friendly options.
- **Adjust Style or Length:** Summarize the hotel details in bullet points for easy comparison.

Revised Answer:
"According to the official website, SIGIR 2025 will be hosted at the Padova Congress Center in Padua, Italy, from July 13–17. Recommended accommodations include:

- *NH Hotel Padova* (120€/night, 10 min walk)
- *Best Western Hotel Biri* (165€/night, 20 min walk)
- *B&B Hotel Padova* (90€/night, 10 min walk)

For more lodging options, refer to the official SIGIR 2025 accommodation page: https://sigir2025.dei.unipd.it/recommended-hotels.html."

By logging frequent user interventions, such as recurring factual corrections, content modifications, or style adjustments, *User Debug Mode* gathers valuable data to update for answer generation models, ensuring better alignment with user preferences over time.

3.3 Shadow User Mode

In scenarios where users prefer minimal interaction, **Shadow User Mode** employs a *personalized user agent* to simulate user behavior and generate fine-grained feedback. This agent serves as an intelligent intermediary, assisting users who wish to refine their search process but find manual debugging cumbersome. By proactively suggesting AI-assisted feedback at each key stage, the agent reduces the interaction cost while still incorporating user preferences. As the agent continuously learns and improves its ability to mimic user decisions, it can increasingly replace direct user intervention, ensuring that generative AI search accumulates meaningful feedback without requiring extensive manual input.

The implementation of the *personalized user agent* consists of two key components: (1) **User Preference Learning**, which constructs and maintains a dynamic user profile based on past interactions to infer likely search preferences; (2) **AI-assisted Feedback Generation**, which predicts pseudo-feedback when users do not explicitly engage with intermediate steps.

3.3.1 User Preference Learning. The goal of user preference learning is to construct and continuously update a dynamic user profile that captures individual preferences and behavioral patterns. By analyzing various signals—including demographic attributes, search behaviors, click interactions, and browsing history—the system models user-specific tendencies to better align search outcomes with their expectations [44]. Conceptually, this process identifies overarching user preferences that influence search behavior. For example, if a user consistently prioritizes convenience over cost when booking accommodations, the system recognizes and encodes this preference. Ultimately, this phase will generate a structured user profile that guides the personalized user agent in providing more relevant and personalized AI-assisted feedback at each stage of the search pipeline, reducing unnecessary user effort while maintaining high-quality search refinements.

3.3.2 AI-assisted Feedback Generation. With the constructed user profile, the *personalized user agent* can assist users who prefer minimal interaction but still seek to refine their search results. When a user finds the generated answer unsatisfactory and wants to debug the process, the agent provides targeted correction suggestions based on their preferences, requiring only confirmation rather than manual intervention. This reduces user effort while ensuring valuable process-level feedback is continuously integrated. Next, we illustrate how the agent can generate intermediate feedback to assist users in debugging the search pipeline.

Simulating User Feedback in Query Decomposition. The personalized user agent leverages the learned user preferences to autonomously debug query decomposition by deciding whether to remove, reorder, or refine sub-queries. Instead of requiring the user to manually adjust them, the agent presents modification suggestions for confirmation. Once approved, it executes these refinements and provides a brief explanation of the changes. Below is a potential implementation prompt example:

Simulating User Feedback in Query Decomposition

System Prompt: You are simulating a user who wants to refine a query decomposition process. Based on the provided user profile, you will review the initial sub-queries and identify necessary adjustments. You can perform the following actions: {*Description of the actions in this stage*}
User Profile: {*User-specific preferences*}
User Query: {*User query*}
Initial Query Decomposition: {*Original sub-queries*}
Task Prompt: Analyze the given sub-queries in light of the user profile, highlighting any necessary modifications with clear explanations. Then, generate a refined list of sub-queries that better align with the user's needs.

Simulating User Feedback in Retrieval & Ranking. In this stage, the personalized user agent refines the retrieval and ranking process based on learned user preferences and context. Instead of requiring users to manually sift through documents, the agent proactively suggests adjustments and presents a revised list for user confirmation. Below is a potential implementation prompt example:

Simulating User Feedback in Retrieval & Ranking

System Prompt: You are simulating a user who wants to refine a retrieval & ranking process. Based on the provided user profile, you will review the initial retrieved results and apply necessary adjustments. You can perform the following actions: {*Description of the actions in this stage*}
User Profile: {*User-specific preferences*}
User Query: {*User query*}
Initial Retrieved Results: {*Original ranked document list*}
Task Prompt: Analyze the retrieved documents in light of the user profile and context, identifying any necessary refinements with clear justifications. Then, generate a revised ranked list that best aligns with the user's intent.

Simulating User Feedback in Answer Generation. Finally, the agent refines the generated answer based on the user's learned preferences. Instead of requiring users to manually adjust factual correctness, content structure, or stylistic elements, the agent proactively suggests modifications aligned with their past preferences. Below is a potential implementation prompt example:

Simulating User Feedback in Answer Generation

System Prompt: You are simulating a user who wants to refine an AI-generated answer. Based on the provided user profile, you will review the initial answer and apply necessary adjustments. You can perform the following actions: {*Description of the actions in this stage*}
User Profile: {*User-specific preferences*}
User Query: {*User query*}
Initial Generated Answer: {*Original generated answer*}
Task Prompt: Analyze the generated answer in light of the user profile and query context, highlighting necessary modifications with clear justifications. Then, generate a revised answer that best aligns with the user's needs.

3.4 Synergy of Dual Feedback Modes

Both *User Debug Mode* and *Shadow User Mode* share the common objective of maintaining a continuous flow of user-driven signals to support both *online* (current session) and *offline* (long-term) model refinement. In *User Debug Mode*, engaged users can exert full control over the search pipeline by modifying sub-queries, re-ranking retrieved documents, and refining generated answers. However, this increased level of human control comes at the cost of greater interaction complexity. *Shadow User Mode* mitigates this by deploying a personalized user agent that learns user behaviors and provides fine-grained AI-assisted feedback when explicit debugging is absent. As the agent progressively refines its ability to simulate user preferences, it can deliver increasingly high-quality feedback, reducing the need for manual intervention. Over time, users can gradually delegate more of the debugging process to the agent, trusting it to make refinements on their behalf.

This synergy ensures that every search session—whether actively debugged or passively simulated—contributes valuable feedback for improving the generative AI search system. *User Debug Mode* provides high-fidelity "gold" signals when users directly interact with the pipeline, while *Shadow User Mode* supplies continuous AI-assisted feedback in cases of minimal engagement. Together, these modes reconstruct a robust, multi-stage feedback ecosystem: immediate user corrections address urgent errors, while aggregated logs—comprising both explicit user input and agent-generated AI-assisted feedback—fuel iterative improvements to query decomposition, ranking, and generation modules.

3.5 Motivating User Engagement

Encouraging users to actively participate in NExT-Search's feedback mechanisms is crucial for maintaining a robust and continuously improving generative AI search ecosystem. However, engaging users in deep debugging processes requires additional effort. Without clear incentives, users may be reluctant to invest the necessary time and cognitive resources. To address this, we introduce a **feedback store** as a promising mechanism to motivate user participation. In this marketplace, users can package their optimized debugging processes into reusable templates and offer them to others facing similar search challenges. These templates can be listed for purchase, where contributors receive direct financial compensation when others adopt their solutions, or they can generate passive income based on usage metrics such as views, downloads, or successful query resolutions.

This feedback store creates a closed-loop knowledge monetization loop, enabling experienced users to capitalize on their expertise while allowing less experienced users to benefit from high-quality, pre-optimized search workflows without the need for manual refinement. By bridging the gap between expert contributors and general users, the feedback store fosters a self-sustaining ecosystem for search refinement and continuous improvement. Users not only achieve more efficient and accurate search outcomes but also receive tangible incentives, ultimately creating a mutually beneficial dynamic between the platform and its contributors.

4 Leveraging Feedback: From Online to Offline

With the step-by-step feedback collected under our NExT-Search paradigm, we explore two complementary strategies to leverage the feedback in Figure 3: **Online Adaptation**, which refines responses *in real time* based on user feedback within the current active session, and **Offline Update**, where accumulated interaction logs drive model retraining, reinforcing *a long-term self-improvement loop.*

4.1 Online Adaptation

Online adaptation focuses on dynamically improving response quality during an ongoing user session. As feedback is provided—whether through explicit corrections in *User Debug Mode* or inferred AI-assisted feedback in *Shadow User Mode*—the system applies immediate refinements to better align with the user's intent. Once a user modifies any stage in the search pipeline, all subsequent stages are re-executed accordingly, akin to debugging a program where each adjustment propagates downstream to ensure consistency.

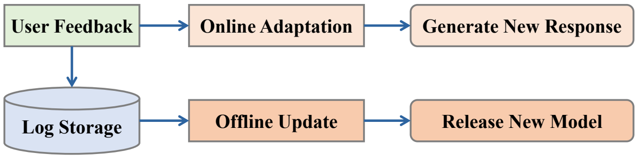

Figure 3: Two mechanisms for leveraging feedback.

For example, when users modify sub-queries—such as adding a missing query—the system immediately reprocesses the updated formulation, ensuring that all downstream stages reflect the changes. When users annotate retrieved documents for relevance or apply filtering criteria, the system dynamically re-ranks results, improving the quality of the knowledge pool before answer synthesis. Finally, if users correct factual errors or request additional details in the generated response, the system selectively regenerates affected sections while preserving validated content, reducing unnecessary recomputation and enhancing efficiency. Through these real-time adaptations, NExT-Search allows generative AI search to continuously align with evolving user intent.

4.2 Offline Update

Beyond immediate corrections, offline update aggregates multi-session interaction logs to drive long-term system improvements. User feedback—both explicit and simulated—serves as structured supervision signals for continuously refining key components of the generative AI search pipeline. In industrial search systems, a widely adopted strategy is *daily incremental updates* [24, 29, 31], where user interaction and feedback logs are periodically processed to generate positive and negative training samples. These samples are then used to perform incremental training on top of the previous day's model parameters. Once training is completed, the updated model is deployed to production.

For each of the three core stages of the pipeline, we discuss how offline update can be effectively constructed using user feedback:

Update for Query Decomposition. User-corrected sub-queries provide direct supervision signals for improving query decomposition. The system collects pairs of *original sub-queries* (before correction) and *revised sub-queries* (after user modification), treating the latter as positive examples and the former as negative examples. These structured samples can then be used to refine the LLM's decomposition abilities through techniques such as instruction fine-tuning [33] or direct preference optimization [34].

Update for Retrieval & Ranking. Feedback from user annotations—including relevance labels, source preferences, and re-ranking actions—serves as essential signals for improving retrieval and ranking models. Positive samples consist of documents that users frequently engage with (e.g., those marked as relevant, clicked, or cited in responses), while negative samples include documents that users downvoted or explicitly filtered out. These signals are then used to fine-tune retrieval models [48] and ranking models [43] to better reflect user preferences.

Update for Answer Generation. Corrections made to generated responses, such as factual fixes or content expansions, are logged as supervised learning signals for improving the LLM's answer synthesis capabilities. Positive samples include sections that users accepted or minimally modified, while negative samples are those that were corrected or flagged as hallucinations. These signals can be used to fine-tune the LLM—e.g., through reinforcement learning from human feedback (RLHF) [2, 33]—to improve factual accuracy and better align responses with user expectations.

By continuously leveraging these structured feedback signals, offline update enables generative AI search to iteratively refine the search pipeline over the long term.

5 Potential Research Opportunities

Under **NExT-Search** paradigm, several open challenges and research directions emerge, offering rich avenues for future investigation. We outline the following research opportunities:

LLM for Personalized User Simulation. One critical challenge in realizing the full potential of *Shadow User Mode* under the NExT-Search paradigm lies in building personalized user agents capable of reliably simulating user behavior and producing high-quality, fine-grained feedback from limited user interaction data. Future work may explore advanced user modeling techniques that combine large reasoning models with behavioral data to infer preferences more accurately [44]. Promising directions include combining RAG with personalized in-context learning [36], constructing dynamic memory modules [47] to retain user-specific history across sessions, and leveraging preference-conditioned generation to align feedback suggestions with individual goals. Furthermore, building such simulators often requires access to sensitive signals such as user profiles, engagement logs, or contextual attributes, which may raise privacy concerns [3, 10, 38]. As a result, balancing personalization with privacy is another fundamental challenge. Techniques such as federated learning [40] and on-device adaptation [32] could offer promising pathways for privacy-preserving user simulation.

Learning from Human and AI-Assistant Feedback. While the NExT-Search paradigm envisions a renewed user feedback ecosystem for generative AI search, effectively leveraging the collected signals to drive system improvement remains an open challenge. In Section 4, we outline two complementary strategies: *online adaptation* for immediate refinements and *offline updates* for longer-term model retraining. However, how to efficiently implement these strategies—particularly in leveraging fine-grained user feedback—deserves deeper exploration in future work. One promising direction is to design training procedures that utilize users' step-by-step feedback trajectories to supervise various components of the search pipeline. Integrating recent advances in LLM reasoning-aware training [27] may help construct richer learning strategies and improve pipeline robustness. Moreover, another core challenge lies in integrating heterogeneous feedback sources. While *User Debug Mode* offers high-quality but sparse feedback and *Shadow User Mode* supplies abundant but potentially noisy signals, effectively combining these complementary signals remains a key research challenge. Thus, exploring adaptive learning techniques such as multi-task learning [30] or curriculum learning [5] may offer promising avenues for training on different feedback types. More broadly, the question of how to maximize system robustness and learning efficiency from diverse feedback streams is central to realizing the full potential of the NExT-Search paradigm.

Human-Centric Interaction Design. Although *Shadow User Mode* is an effective complement to *User Debug Mode*—using LLMs to proactively suggest feedback and reduce user effort—the paradigm still fundamentally depends on user engagement. Thus, the system must strike a careful balance between transparency, control, and interaction burden. This raises several important design questions: How should intermediate steps (e.g., sub-query decomposition or retrieved documents) be presented to encourage actionable feedback? What level of user intervention is appropriate across different users and tasks? Collaboration with researchers from human-computer interaction and user behavior studies [4, 19, 39] could yield innovative UI designs or interactive workflows that solicit targeted feedback, minimize user frustration, and gradually train novices to handle more complex tasks. Another open challenge is how to dynamically manage transitions between *Shadow User Mode* and *User Debug Mode*. Developing adaptive mode-switching mechanisms—based on task complexity, user expertise, or predicted benefit-to-cost ratios—presents a promising research direction. Techniques such as user modeling [37] or reinforcement learning [45] could be leveraged to personalize interaction strategies.

While our NExT-Search paradigm offers an initial step toward rethinking feedback in generative AI search, future work should investigate how feedback mechanisms can be continuously refined and adapted in response to the rapidly evolving pipelines and interaction patterns of generative search systems.

6 Conclusion

In this work, we envision **NExT-Search**, a paradigm aimed at reintroducing fine-grained user feedback into generative AI search. By integrating *User Debug Mode* for direct user interventions and *Shadow User Mode* for implicit feedback simulation, NExT-Search enables the collection of structured, stage-level signals across the entire search pipeline. These feedback signals can be further leveraged through online adaptation for real-time answer refinements and offline update for long-term model improvements. Additionally, we introduced a feedback store mechanism to motivate user engagement. While NExT-Search remains a forward-looking framework, we believe it offers valuable insights for the design of next-generation generative AI search, ensuring a balance between automation, user control, and continuous self-improvement. Due to the lack of publicly available datasets, we leave empirical validation and system implementation to future work.

Acknowledgments

This research/project is supported by the National Research Foundation, Singapore under its National Large Language Models Funding Initiative, (AISG Award No: AISG-NMLP-2024-002), the National Natural Science Foundation of China (No.62276248), and the Youth Innovation Promotion Association CAS under Grants (No.2023111). This research is supported by A*STAR, CISCO Systems (USA) Pte. Ltd and National University of Singapore under its Cisco-NUS Accelerated Digital Economy Corporate Laboratory (Award I21001E0002). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

[1] Eugene Agichtein, Eric Brill, and Susan Dumais. 2006. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 19–26.

[2] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862* (2022).

[3] Krisztian Balog and ChengXiang Zhai. 2025. User Simulation in the Era of Generative AI: User Modeling, Synthetic Data Generation, and System Evaluation.

[4] Marcia J. Bates. 1989. The design of browsing and berrypicking techniques for the online search interface. *Online Review* 13(5) (1989), 407–424.

[5] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. 41–48.

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

[7] Gobinda G Chowdhury. 2010. *Introduction to modern information retrieval*. Facet publishing.

[8] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining*. 87–94.

[9] W Bruce Croft, Donald Metzler, and Trevor Strohman. 2010. *Search engines: Information retrieval in practice*. Vol. 520. Addison-Wesley Reading.

[10] Sunhao Dai, Chen Xu, Shicheng Xu, Liang Pang, Zhenhua Dong, and Jun Xu. 2024. Bias and Unfairness in Information Retrieval Systems: New Challenges in the LLM Era. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 6437–6447.

[11] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 6491–6501.

[12] 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).

[13] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM international on conference on information and knowledge management*. 55–64.

[14] Alexander Halavais. 2017. *Search engine society*. John Wiley & Sons.

[15] Ahmed Hassan Awadallah, Ryen W White, Patrick Pantel, Susan T Dumais, and Yi-Min Wang. 2014. Supporting complex search tasks. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*. 829–838.

[16] Marti Hearst. 2009. *Search user interfaces*. Cambridge university press.

[17] Nadine Höchstötter and Dirk Lewandowski. 2009. What users see–Structures in search engine results pages. *Information sciences* 179, 12 (2009), 1796–1812.

[18] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 2333–2338.

[19] Jiepu Jiang, Daqing He, and James Allan. 2014. Searching, browsing, and clicking in a search session: changes in user behavior by task and over time. *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval* (2014).

[20] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 133–142.

[21] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2017. Accurately interpreting clickthrough data as implicit feedback. In *Acm Sigir Forum*, Vol. 51. Acm New York, NY, USA, 4–11.

[22] Diane Kelly and Leif Azzopardi. 2015. How many results per page? A study of SERP size, search behavior and user experience. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 183–192.

[23] Diane Kelly and Jaime Teevan. 2003. Implicit feedback for inferring user preference: a bibliography. In *Acm Sigir Forum*, Vol. 37. ACM New York, NY, USA, 18–28.

[24] Mark Levene. 2011. *An introduction to search engines and web navigation*. John Wiley & Sons.

[25] Hang Li, Jun Xu, et al. 2014. Semantic matching in search. *Foundations and Trends® in Information Retrieval* 7, 5 (2014), 343–469.

[26] Yongqi Li, Xinyu Lin, Wenjie Wang, Fuli Feng, Liang Pang, Wenjie Li, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2024. A survey of generative search and recommendation in the era of large language models. *arXiv preprint arXiv:2404.16924* (2024).

[27] Zhongzhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhijiang Guo, Le Song, and Cheng-Lin Liu. 2025. From System 1 to System 2: A Survey of Reasoning Large Language Models. *ArXiv* (2025).

[28] Nelson F Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. *arXiv preprint arXiv:2304.09848* (2023).

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

[30] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1930–1939.

[31] Christopher D Manning. 2009. *An introduction to information retrieval*.

[32] Rafael Mendoza, Isabella Cruz, Richard Liu, Aarav Deshmukh, David Williams, Jesscia Peng, and Rohan Iyer. 2024. Adaptive self-supervised learning strategies for dynamic on-device llm personalization. *arXiv preprint arXiv:2409.16973* (2024).

[33] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

[34] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* 36 (2024).

[35] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[36] Alireza Salemi and Hamed Zamani. 2024. Comparing retrieval-augmentation and parameter-efficient fine-tuning for privacy-preserving personalization of large language models. *arXiv preprint arXiv:2409.09510* (2024).

[37] Xuehua Shen, Bin Tan, and ChengXiang Zhai. 2005. Implicit user modeling for personalized search. In *Proceedings of the 14th ACM international conference on Information and knowledge management*. 824–831.

[38] Xuehua Shen, Bin Tan, and ChengXiang Zhai. 2007. Privacy protection in personalized search. In *ACM SIGIR Forum*, Vol. 41. ACM New York, NY, USA, 4–17.

[39] Sofia Eleni Spatharioti, David M. Rothschild, Daniel G. Goldstein, and Jake M. Hofman. 2023. Comparing Traditional and LLM-based Search for Consumer Choice: A Randomized Experiment. *ArXiv* abs/2307.03744 (2023).

[40] Alysa Ziying Tan, Han Yu, Li zhen Cui, and Qiang Yang. 2021. Towards Personalized Federated Learning. *IEEE Transactions on Neural Networks and Learning Systems* 34 (2021), 9587–9603.

[41] Ryen W White. 2024. Advancing the search frontier with AI agents. *Commun. ACM* 67, 9 (2024), 54–65.

[42] Haoyi Xiong, Jiang Bian, Yuchen Li, Xuhong Li, Mengnan Du, Shuaiqiang Wang, Dawei Yin, and Sumi Helal. 2024. When search engine services meet large language models: visions and challenges. *IEEE Transactions on Services Computing* (2024).

[43] 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.

[44] Yiyan Xu, Jinghao Zhang, Alireza Salemi, Xinting Hu, Wenjie Wang, Fuli Feng, Hamed Zamani, Xiangnan He, and Tat-Seng Chua. 2025. Personalized Generation In Large Model Era: A Survey. *arXiv preprint arXiv:2503.02614* (2025).

[45] Jing Yao, Zhicheng Dou, Jun Xu, and Ji-Rong Wen. 2020. RLPer: A reinforcement learning model for personalized search. In *Proceedings of The Web Conference 2020*. 2298–2308.

[46] ChengXiang Zhai. 2024. Large Language Models and Future of Information Retrieval: Opportunities and Challenges. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 481–490.

[47] Kai Zhang, Yejin Kim, and Xiaozhong Liu. 2024. Personalized llm response generation with parameterized memory injection. *arXiv preprint arXiv:2404.03565* (2024).

[48] Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2023. Dense Text Retrieval based on Pretrained Language Models: A Survey. *ACM Trans. Inf. Syst.* (dec 2023).

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