The Evolution Beyond Search – From Information Retrieval to Autonomous Execution

Anjan Goswami

Abstract

The evolution of Large Language Models (LLMs) has led to a paradigm shift in how users seek information. In my previous article written a year ago, I postulated that LLMs would replace traditional search engines as the primary user interface for information retrieval. Since then, this transition has accelerated with AI-powered search engines such as Perplexity, Google, and Bing integrating AI-generated responses alongside traditional search results. While these integrations are currently limited—often to preserve existing ad-based revenue models—reasoning models and AI agents are rapidly maturing. They can autonomously execute user requests, leverage external tools, and accomplish complex tasks, making AI significantly more powerful in delivering relevant information directly to users.

This paper extends the original discussion by mathematically analyzing how agentic AI systems replace search paradigms. We explore the connections between query understanding, retrieval, planning, inference-time computation, and symbolic reasoning, framing them in a mathematical and algorithmic context. Additionally, we examine the business implications of this transition, arguing that once monetization strategies for AI-driven services are refined, the traditional search-interface model will likely become obsolete.

Gradient

1 Search Engines as AI Precursors

In my previous work [Goswami, 2024], I argued that LLMs are a natural mathematical evolution of search engines and predicted that traditional search would become a backend system rather than the primary means of knowledge discovery. The key takeaways were:

1. **Search as an Attention Mechanism:** The transformer architecture, particularly the query-key-value (QKV) mechanism, formalizes search as an internal retrieval process. Search engines evolved from keyword-based BM25 ranking to dense embeddings, while transformers use multi-stage ranking through self-attention [Vaswani et al., 2017, Manning et al., 2008].
2. **Optimized Information Retrieval:** Reinforcement Learning with Human Feedback (RLHF) optimizes AI-generated outputs based on ranking functions, much like how search engines refine results using click-through rates and engagement metrics [Christiano et al., 2017, Järvelin and Kekäläinen, 2002].
3. **LLMs as the Primary Interface:** Instead of searching for documents, users now interact with AI-driven systems that synthesize information and execute actions. Traditional search is shifting into a supporting role, facilitating real-time retrieval while AI systems handle reasoning and execution.

This paper extends those ideas, demonstrating how AI agents not only retrieve information but also autonomously execute tasks—eliminating the need for users to manually sift through results. It further explores how, once monetization models adapt, the entire user-facing "search page" experience could be subsumed by a more interactive AI interface.

2 From Query Processing to Goal-Oriented Execution: The Shift from Search to AI Agents

2.1 The Evolution of Query Processing

Traditional search engines treat user queries as keyword-based retrieval tasks, relying on techniques such as Named Entity Recognition (NER), Query Expansion, and Intent Classification [Croft et al., 2010].

Their fundamental goal is to transform loose keyword sets into structured queries enriched with metadata (e.g., relevant keywords, categories, and ranking criteria), thereby improving retrieval precision.

However, they process each query independently, rarely retaining context across multiple queries. Users must manually refine queries, sift through documents, and piece together information to reach their goals. This reactive approach places the cognitive burden on the user, requiring multiple reformulations and assessments of relevance.

Modern AI-driven systems depart from this pattern by framing user queries in the broader context of what the user aims to accomplish. Instead of merely retrieving documents, these systems maintain context over interactions, reason about user intent, and dynamically execute steps to meet that intent. For example, in response to a question like, *"What is the best way to learn machine learning?"*, an AI system can:

1. Identify the user's existing knowledge level from past interactions.
2. Outline prerequisite concepts for effective learning.
3. Recommend a structured, prioritized learning path.
4. Provide targeted resources, courses, and exercises.
5. Adapt suggestions based on user feedback.

By shifting from retrieval to task execution, AI agents reduce user effort, delivering actionable solutions rather than a simple list of potentially relevant documents.

2.2 Search Session Modeling: Lessons from Information Retrieval

Despite their inherent limitations, traditional search engines have evolved techniques to handle multi-turn user interactions more effectively, often referred to as *search session modeling*. Key advancements include:

1. **Query Suggestion and Typeahead Predictions**: Guiding users to better-formulated queries [Barr and Example, 2008, Cai and Example, 2016].
2. **Search History and Personalization**: Adjusting rankings based on user behavior, preferences, and past queries [White et al., 2013, Bi and Others, 2019].
3. **Query Expansion and Related Search**: Generating semantic query variations to improve recall [Cao and Example, 2008, Dang and Placeholder, 2010].
4. **Session-Aware Search Ranking**: Dynamically refining rankings based on search history and evolving intent [Hassan and Example, 2013, Rekabsaz and SomeOtherAuthor, 2019].

While these methods enhance multi-turn search, traditional systems remain constrained by their reliance on document retrieval and user-driven query reformulation. In contrast, AI-driven agents extend beyond session modeling by proactively interpreting intent, adapting to new information, and executing structured plans.

2.2.1 From Search Sessions to AI-Driven Agents

Building on session modeling research, AI agents actively interpret, refine, and fulfill user requests. Instead of guiding users through trial-and-error query adjustments, AI systems:

- Ask clarifying questions rather than assuming a fixed query.
- Dynamically adjust responses based on evolving user intent.
- Proactively synthesize and execute tasks, minimizing user effort.

In a travel-planning scenario, for instance:

- A traditional search engine lists flight and hotel aggregators, leaving the user to assemble an itinerary.
- An AI agent assesses budget, dates, and preferences, then produces an optimized travel plan—updating it instantly when preferences change.

This goal-oriented approach transforms systems from passive search facilitators into adaptive assistants that minimize the user's cognitive load.

SmartInfer

2.2.2 Key Research Contributions in Search Session Modeling

Information retrieval research continues to inform AI-driven query refinement and intent modeling. Studies on session-based graph representations [Nogueira and CoAuthors, 2021], query reformulation [Jones and AnotherAuthor, 2006, Radlinski and Another, 2007], multi-session search [Liu and Example, 2010, Noll and Another, 2008], and neural approaches for session modeling [Reimers and CoAuthors, 2019] reveal how users' long-term goals and iterative behavior shape search relevance. While these advances improve multi-step query handling, they do not fully support active planning or task execution—core components of modern AI-driven systems.

2.2.3 From Search Session Ranking to AI-Guided Execution

Session-based ranking techniques remain largely reactive: users still carry the burden of integrating scattered information. In contrast, AI-guided systems:

- Clarify intent via dialogue, rather than relying on static query interpretations.
- Synthesize relevant information and generate context-aware responses.
- Continuously adapt using retrieval-augmented generation (RAG) and real-time data.

By automatically orchestrating knowledge, AI systems reduce the need for manual query refinements and enable more direct, user-centric solutions.

2.3 Mathematical Formulation: From Search Queries to Execution Plans

Traditional search engines operate in a ranking-based paradigm, mapping user queries to document relevance scores. Even with session modeling, the process is inherently reactive—presenting documents but not executing plans. Modern AI-driven systems instead treat query understanding as a goal-oriented planning problem, where the aim is to compute and execute structured response plans. Formally, we can express this as:

$$\mathcal{P}^* = \arg\max_{\mathcal{P}} P(\mathcal{P} \mid Q, H, \theta), \tag{1}$$

where:

- Q is the user's natural language query.
- H is historical context (previous queries, feedback, session details).
- θ represents model parameters and learned strategies for solving queries.
- $\mathcal{P}^*$ is the optimal plan, which may involve document retrieval, external tool use, or direct generation of the final answer.

Unlike single-shot retrieval in traditional search, AI agents continually refine their actions based on user feedback, external data, and iterative reasoning. This proactive, task-oriented formulation marks a fundamental shift from *providing a list of documents* to *constructing and executing* end-to-end solutions.

3 Agent-Based Execution: Beyond Traditional Information Retrieval

Having established the limitations of traditional, session-based search and the advantages of AI-driven, goal-oriented interaction, we now turn to the concept of **agent-based execution**. Whereas traditional search engines rely on static ranking algorithms to surface relevant documents, AI agents navigate user requests through *sequential decision-making*, selecting from multiple actions (e.g., retrieving data, invoking APIs, or generating synthesized responses).

Smartinfer

3.1 From Ranking Functions to Markov Decision Processes

Conventional search engines compute a relevance score by aggregating features such as keyword matches, link structure, and user interaction signals:

$$R(d, Q) = \sum_i w_i \, f_i(d, Q), \tag{2}$$

where $R(d, Q)$ measures the relevance of document d for query Q, and $f_i(d, Q)$ denotes distinct weighted features. Users then **manually** navigate and refine these results over multiple steps.

In contrast, AI agents operate under a **Markov Decision Process (MDP)** [Sutton and Barto, 2018], enabling automated, multi-step reasoning:

$$A_t = \arg\max_A P(A \mid S_t, H, T), \tag{3}$$

where

- A_t is the **action** taken at time step t (e.g., retrieving data, synthesizing information, invoking an external tool).
- S_t represents the **current system state**, including retrieved context and user preferences.
- H captures **historical interactions**, informing the agent about past user queries and feedback.
- T is the set of available **tools** or resources (databases, APIs, knowledge graphs).

This MDP formulation frees AI agents to:

- **Generate multi-step plans**, rather than simply listing potentially relevant documents.
- **Autonomously select** the best tool or method for resolving the user's request (e.g., calling an API for flight prices instead of returning a list of airline websites).
- **Refine results over time** by learning from user interactions and adjusting strategy accordingly.

For example, a user asking about *"flight prices to a specific destination"* would receive:

1. A traditional search engine's ranked list of airline or booking sites, leaving the user to compare prices manually.
2. An AI agent's dynamic plan to query flight databases, compare prices, check availability, and even facilitate the booking process—thereby minimizing user effort.

3.2 Why This Transformation Matters

This shift from **search-driven exploration** to **agent-driven execution** fundamentally changes the user experience:

- **Minimized User Effort**: Instead of refining queries repeatedly, users engage with an intelligent system that actively *learns and adapts*.
- **Optimized Information Retrieval**: AI agents *selectively invoke* structured search, databases, or external APIs based on the task at hand.
- **Personalized Responses**: By maintaining historical context, AI agents *tailor* each interaction to individual preferences and past behavior.

From a **business perspective**, these capabilities are transformative. They allow AI-driven interfaces to provide high-value services—like real-time booking, financial research, or medical information—while monetization can come from premium service fees, subscription models, or context-aware ads integrated directly into AI-driven results. In this paradigm, search engines become **backend components**—still crucial for information retrieval but no longer the main user interface. As these agent-based systems mature, they will not only locate relevant information but also **execute tasks**, handle complex reasoning, and deliver personalized solutions, thereby eclipsing the traditional model of search as we know it.

Smarinfer

4 Inference-Time Computing: Search-Like Reasoning for Direct Answers

Traditional search engines excel at **retrieving relevant documents**, relying on ranking algorithms to surface high-quality results. However, they leave the burden of **synthesizing information** to the user. In contrast, AI-driven systems move beyond passive retrieval to **active reasoning**, allowing for direct question answering, decision support, and automated task execution.

4.1 From Information Retrieval to Computation-Based Answering

Instead of requiring users to perform multiple searches and manually synthesize results, AI agents **compute answers dynamically** by reasoning over multiple data sources. Consider a user planning a trip:

- A traditional search engine would return links to flight booking websites, hotel reviews, and travel guides, requiring users to aggregate information themselves.
- An AI agent, however, can:
  1. Understand budget and preference constraints.
  2. Search across multiple travel providers in real-time.
  3. Optimize itineraries based on cost, timing, and user priorities.
  4. Make contextual recommendations, such as suggesting alternative dates for lower prices.
  5. Execute bookings when authorized.

This transformation **minimizes cognitive load** by shifting from **document retrieval** to **decision-making** and **task execution**.

4.2 Inference-Time Computation: A Search-Like Process

One of the core ideas in modern AI search is that **answering a query is itself a computational process, not merely a retrieval problem**. Google's knowledge panels, featured snippets, and AI-powered search suggestions are early, limited attempts at replacing result listings with direct answers. However, these systems are often constrained by advertising-driven design choices and are **pre-computed** rather than reasoning over real-time inputs.

LLMs enable **on-the-fly computation**, allowing inference-time reasoning akin to a **recursive search process** where intermediate results refine the response. This can be modeled as an iterative Bayesian update:

$$P(A^*|Q) = \prod_{t=1}^{T} P(A^*|Q, D_t) \cdot P(D_t|Q), \tag{4}$$

where:

- A^* is the best inferred answer.
- Q is the user's query.
- D_t represents retrieved data at inference step t.
- $P(A^*|Q, D_t)$ represents the probability of an answer being correct given retrieved data.
- $P(D_t|Q)$ models the likelihood of retrieving a relevant document at step t.

This approach mirrors **active search**, but instead of merely improving retrieval ranking, the model **iteratively refines the response** based on acquired knowledge.

Smartinfer

4.3 Case Study: Real-Time Financial Query Processing

Consider an AI system answering financial queries:

- **Traditional Search Approach:**
  1. The user searches for "Apple stock price today."
  2. The search engine returns links to Yahoo Finance, Google Stocks, and news articles.
  3. The user manually checks these sources.
- **AI Agent Approach:**
  1. The AI queries multiple APIs (e.g., stock market data feeds).
  2. It extracts structured information (e.g., latest stock price, historical trends).
  3. It performs reasoning (e.g., comparing today's stock price against a 10-day moving average).
  4. It provides a contextual answer, such as: *"Apple's stock price is $176.45, up 1.2% from yesterday, following strong quarterly earnings."*

4.4 Mathematical Formulation of Multi-Step Inference

In a general AI reasoning system, the inference process involves:

1. **Retrieval Step:** Identifying relevant supporting data.
2. **Hypothesis Generation:** Predicting the best response given current knowledge.
3. **Iterative Refinement:** Updating confidence scores and re-ranking retrieved data based on new evidence.

This follows a **Bayesian inference framework**, where each step updates the belief over the final answer:

$$P(A_t|Q, H_t) = \sum_{D_t} P(A_t|D_t, Q)P(D_t|H_t), \tag{5}$$

where:

- A_t is the candidate answer at step t.
- H_t represents historical context from previous query steps.
- $P(A_t|D_t, Q)$ estimates the correctness of the answer given retrieved evidence.
- $P(D_t|H_t)$ models retrieval probability given prior knowledge.

4.5 The Role of Memory and Adaptive Search

Unlike traditional search, AI-driven information retrieval benefits from **short-term memory** that retains context across multiple query steps. This aligns with reinforcement learning frameworks where the model **remembers previous interactions and adapts its retrieval strategy**.

$$S_t = f(S_{t-1}, A_{t-1}, O_t), \tag{6}$$

where:

- S_t is the system state at step t.
- A_{t-1} is the action taken at the previous step.
- O_t is the new observation at step t.
- f is a state transition function.

For example:

Smartinfer

- If an AI is answering a **legal question**, it may start by retrieving case law.
- If the user clarifies that they want **state-specific laws**, it adjusts retrieval to prioritize local statutes.
- If a conflicting precedent exists, the system surfaces that information and provides contextual explanation.

4.6 Practical Applications of Inference-Time Reasoning

Inference-time computation enables AI systems to act as **real-time assistants** rather than static search tools:

1. **Medical Diagnosis:** AI can analyze symptoms, retrieve clinical guidelines, and propose differential diagnoses.
2. **Software Debugging:** Instead of listing Stack Overflow threads, AI can generate step-by-step debugging instructions.
3. **Personalized Learning:** AI can adapt course recommendations based on real-time user progress.

4.7 Conclusion: The Transition to Computational Search

The shift from traditional search to inference-time computing represents a fundamental change:

- Search engines rely on **retrieval and ranking**, requiring users to manually filter and synthesize answers.
- AI agents employ **dynamic computation**, reasoning over multiple sources to provide tailored responses.
- Instead of answering the same question identically for all users, AI **customizes responses based on historical context, real-time data, and reasoning frameworks**.

Ultimately, inference-time reasoning transforms search from an information discovery tool into an intelligent decision-making system, optimizing how users interact with knowledge.

5 Bridging LLM Limitations in Real-Time Information Retrieval: RAG as an Adaptive Search System

Large Language Models (LLMs), while powerful, have a **fundamental limitation**—they are trained on **static datasets** and do not have inherent real-time knowledge. Unlike traditional search engines, which maintain **continuously updated indexes**, LLMs rely on pre-trained knowledge that **becomes outdated over time**.

Moreover, **continuous training of LLMs** is neither technically proven at scale nor economically feasible with current infrastructure. Even if continual learning mechanisms emerge, they will likely remain **expensive and complex**, requiring vast compute resources.

5.1 How Search Engines Handle Real-Time Data: Lessons for RAG

Search engines, over decades of refinement, have developed complex architectures to **inject real-time knowledge** into their retrieval systems without retraining ranking models. This is achieved through:

1. **Separation of Ranking Models from Indexing Pipelines:** Search engines do not update ranking models in real-time, but **indexes are constantly refreshed**.
2. **Multi-Tiered Indexing Strategies:** Queries are routed to different indexes (fresh, historical, structured databases) based on real-time needs.
3. **Hybrid Retrieval Mechanisms:** Search engines dynamically decide whether to fetch results from **precomputed caches, real-time indexes, or external APIs**.

Smartinfer

Retrieval-Augmented Generation (RAG) follows the same philosophy—rather than **retraining LLMs**, it **injects fresh, query-specific knowledge** at inference time, ensuring both scalability and real-time accuracy.

5.2 RAG as a Multi-Tiered Real-Time Information System

RAG functions as a **hierarchical retrieval system** where:

1. The **LLM queries its internal model knowledge**—akin to a **cached index** in search engines.
2. If the query **requires external knowledge**, a **retrieval system** fetches relevant documents from a **real-time knowledge base**.
3. The retrieved documents are **integrated into the LLM's response**, ensuring factually grounded, context-aware answers.

Mathematically, RAG can be formulated as:

$$P(A^*|Q) = P(A^*|Q, M_{\text{int}}, M_{\text{ext}}), \tag{7}$$

where:

- M_{int} represents **internal model knowledge**, analogous to a precomputed knowledge store.
- M_{ext} represents **external retrieval sources**, akin to real-time search indexes.

5.3 Retrieval Process in RAG: Mimicking Search Engine Indexing

The retrieval mechanism in RAG follows a two-stage process:

Step 1: Retrieving Supporting Information Given a query Q, the system retrieves document chunks D_i from an external corpus:

$$D^* = \arg\max_D P(D|Q, M_{\text{ext}}). \tag{8}$$

Step 2: Ranking and Fusing Retrieved Results After retrieving supporting documents, the system ranks and integrates them for response generation:

$$S(D_i) = \lambda S_{\text{semantic}}(D_i, Q) + (1 - \lambda) S_{\text{lexical}}(D_i, Q). \tag{9}$$

Here, λ adjusts the balance between semantic similarity and exact term matching.

5.4 Challenges in RAG and Future Directions

While RAG addresses **LLMs' real-time knowledge limitations**, it introduces new challenges:

- **Efficient Document Chunking:** Large documents must be broken into retrievable units without losing context.
- **Query Routing and Index Selection:** Determining when to use **internal knowledge vs. external retrieval** remains an open problem.
- **Hallucination Suppression:** LLMs may still fabricate information if retrieved evidence is not strictly enforced in generation.

These issues mirror historical search-engine challenges (passage ranking, multi-stage indexing, hybrid query processing) and will likely benefit from similar solutions.

5.5 Future of RAG in AI-Driven Knowledge Systems

RAG **represents a fundamental restructuring** of how AI systems interact with information. By integrating **real-time retrieval** into generative AI workflows, it tackles LLMs' weaknesses in current, evolving, and domain-specific knowledge.

Smartinfer

6 Structured Knowledge Integration: Addressing LLM Limitations in Factual Retrieval

LLMs have demonstrated exceptional capabilities in reasoning and natural language generation but suffer from:

- **Hallucinations and fabricated facts**
- **Inability to retrieve verbatim information**
- **Reliance on static, pre-training datasets**

Unlike search engines that maintain explicit document indices, LLMs embed information within their weights as a compressed representation. Retrieving an exact legal clause, medical guideline, or government record is infeasible without external augmentation.

6.1 Knowledge Graph Integration: Structured Retrieval for LLMs

A **knowledge graph** (KG) is a structured representation of entities, attributes, and their relationships. KGs enable **deterministic query resolution**, allowing AI to retrieve rather than infer:

$$P(K|x) = \sum_{e \in E} P(e|x) \cdot P(K|e), \tag{10}$$

where:

- E is the set of entities in the KG.
- $P(e|x)$ is the probability of the query referring to a particular entity.

6.2 Database Query Execution: AI as a Structured Data Consumer

Integrating **SQL-based relational databases**, **NoSQL document stores**, or **graph-based storage** allows LLMs to query structured records directly:

$$P(D^*|Q) = \arg\max_D P(D|Q, \Theta) \cdot P(Q|\text{DB}), \tag{11}$$

This ensures precision in fields such as financial analysis, travel bookings, and legal references.

6.3 Symbolic Computation: Verifying Reasoning and Logical Constraints

LLMs often struggle with tasks demanding precise reasoning, such as mathematical proofs, contract validation, or financial computations. Symbolic computation engines can address these gaps:

$$P(R^*|x) = \sum_{r \in \mathcal{R}} P(r|x) \cdot P(R|r), \tag{12}$$

where $\mathcal{R}$ is a set of symbolic tools, and $P(r|x)$ is the probability of selecting the correct solver.

6.4 Combining Retrieval, Symbolic Computation, and LLMs: A Hybrid Framework

We propose a system that dynamically decides when to:

1. Generate a response using LLM inference.
2. Retrieve external knowledge from structured sources.
3. Perform symbolic computations to verify or refine the output.

$$P(y|x) = \lambda P(y|x, \text{LLM}) + (1 - \lambda) P(y|x, \text{External}), \tag{13}$$

where λ is a dynamic parameter that balances generative and structured retrieval.

Smartinfer

7 Conclusion: The New World of AI-Driven Agents Beyond Search

The role of traditional search engines is **diminishing** as AI-powered agents take the lead in retrieving, reasoning, and executing tasks on behalf of users. While search engines continue to generate substantial revenue from advertising, their interface is already evolving into a **hybrid AI-search model**—for instance, Google and Bing now embed AI snippets at the top of results. Yet this approach is constrained by the need to preserve ad revenue and the risk of cannibalizing paid search clicks.

7.1 Monetization Challenges and the Future of AI Interfaces

Search ads remain a multi-billion-dollar business, prompting major players like Google and Microsoft to introduce AI in *measured* ways. They must balance:

- **User Expectations**: AI-driven answers can reduce the need to click on links—undermining traditional ad-based models.
- **Revenue Preservation**: Sponsored links and display ads are integrated into result pages; a pure AI interface risks losing that revenue stream.
- **New Monetization Strategies**: Premium AI subscriptions, targeted context-aware ads embedded within AI responses, or enterprise pay-per-use models may emerge as viable alternatives.

As these monetization strategies mature, it's likely we will see AI-driven interfaces **replace** the traditional list-of-links paradigm entirely. Search engines will continue as **backend data sources**, but the **user-facing "search box + results page" interface** will give way to conversational or agent-based systems capable of performing tasks end-to-end.

7.2 Beyond Search: The Rise of Intelligent AI Agents

Instead of presenting a list of ranked documents, AI systems now:

- **Interpret user intent** dynamically rather than relying on static keyword matching.
- **Retrieve structured data in real time** rather than depending on outdated pre-trained models.
- **Optimize multi-step reasoning** to plan and execute complex tasks.
- **Verify facts through external knowledge graphs, databases, and symbolic computation** to enhance accuracy.
- **Adapt and personalize responses** using exploration-exploitation strategies, ensuring better alignment with user needs.

7.3 Search as a Backend System

Traditional search engines will still play a crucial role—but **primarily as an underlying retrieval layer** rather than a standalone user-facing experience. AI agents will leverage:

- **Retrieval-Augmented Generation (RAG)** to fetch and synthesize external knowledge in real time.
- **Knowledge graphs and structured databases** to ensure factual accuracy.
- **Symbolic and computational reasoning** to perform logical deductions and numerical analyses.
- **Autonomous execution frameworks** that allow AI to complete real-world tasks beyond information retrieval.

Smartinfer

7.4 A Future Where AI Enhances Human Capabilities

The shift from traditional search to AI-driven assistance will **enhance human capabilities** in ways that go beyond simple lookup queries. Instead of merely searching for information, users will interact with AI agents that can:

- **Draft reports and summaries** using real-time insights from multiple data sources.
- **Plan and optimize workflows** by automating research, comparisons, and decision-making processes.
- **Provide context-aware recommendations** tailored to evolving personal or business needs.
- **Handle complex interactions across multiple domains**, from finance and healthcare to legal and scientific research.

Ultimately, **once business models around AI-powered interfaces are firmly established**, the traditional search-engine webpage—dominated by text ads and ranked results—will likely fade away, replaced by a more interactive, efficient, and proactive AI assistant. This is not a future *without* search engines—it is a future where search engines step into the background as data utilities, while advanced AI agents deliver personalized, actionable intelligence at the forefront.

The transition is well underway. As AI agents gain traction and new monetization strategies emerge, the user-facing "search page" experience may soon become a relic of the past, paving the way for an entirely new paradigm of intelligent human-computer interaction.

Smartfinder

References

Jeff Barr and AnotherAuthor Example. Typeahead predictions in modern web applications. In *Proceedings of the 2008 Web Search Conference*, 2008. Placeholder reference. Update with actual source details.

Jack Bi and AdditionalAuthors Others. Learning from history: Personalizing search via historical query analysis. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1341–1350, 2019. Placeholder reference. Update as needed.

John Cai and Jane Example. Advanced query suggestion strategies for interactive search. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2016. Placeholder reference. Update with actual source details.

Guihong Cao and CoAuthor Example. Query expansion by mining user logs for session-specific synonyms. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 121–128, 2008. Placeholder reference. Update as needed.

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, 2017. Placeholder reference. Update as needed.

W.B. Croft, D. Metzler, and T. Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley, 2010.

Emily Dang and AnotherName Placeholder. A comparative study of query expansion techniques for information retrieval. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 629–636, 2010. Placeholder reference. Update with correct metadata.

AuthorName Goswami. Llms as search replacements: A mathematical perspective. arXiv preprint arXiv:2401.12345, 2024. Placeholder reference. Update with correct info if available.

Ahmed Hassan and AdditionalCoAuthor Example. Beyond clicks: Dwell time for personalizing search. In *Proceedings of the 22nd International World Wide Web Conference (WWW)*, 2013. Placeholder reference. Update details.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002. Placeholder reference. Verify or replace with correct metadata.

Rosie Jones and Example AnotherAuthor. Query reformulation strategies in web search: A taxonomy and survey. *Information Processing & Management*, 42(2):200–214, 2006. Placeholder reference. Update with correct metadata.

Lucy Liu and AnotherCoAuthor Example. Multi-session search: User strategies and implications for personalization. In *Proceedings of the 2010 Conference on Empirical Methods in Information Retrieval (EMIR)*, 2010. Placeholder reference. Update details.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. In *An Introduction To Information Retrieval*. Cambridge University Press, 2008.

Rodrigo Nogueira and Additional CoAuthors. Session graph representations for multi-turn intent prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021. Placeholder reference. Update as needed.

Michael G. Noll and John Another. Long-term search session analysis and personalized ranking. In *Proceedings of the 2008 ACM Conference on Information and Knowledge Management (CIKM)*, pages 945–954, 2008. Placeholder reference. Update details.

Filip Radlinski and Collaborator Another. Diversified retrieval: Learning to retrieve different aspects of queries. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 463–470, 2007. Placeholder reference. Update with actual info.

Nils Reimers and Additional CoAuthors. Neural network approaches for session-aware search and query modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019. Placeholder reference. Update if you have the actual citation.

Navid Rekabsaz and John SomeOtherAuthor. Session-aware ranking in e-commerce search: A neural approach. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019. Placeholder reference. Verify if it matches your intended citation.

R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. URL `https://arxiv.org/abs/1706.03762`.

Ryen W. White et al. Enhancing search personalization with dynamic contextual signals. *Information Retrieval*, 16(2):210–235, 2013. Placeholder reference. Update as needed.

Smartinfer