Evaluation and Architectural Analysis of the "Asta Find Papers" Knowledge Agent

Amey Gupta
Siebel School of Computing and Data Science
University of Illinois Urbana-Champaign
Champaign, United States
amey2@illinois.edu

***Abstract*—The exponential growth of academic literature has made manual information retrieval increasingly inefficient and inaccessible. This report evaluates the performance and architectural design of "Asta Find Papers," an agentic search system developed by the Allen Institute for AI (Ai2). Unlike traditional keyword-based search engines, Asta employs an Agentic Retrieval-Augmented Generation (RAG) workflow to synthesize scientific literature. I compare Asta against Google Scholar using complex, context-heavy queries (e.g., "cross-correlation algorithms for IMU synchronization"). The evaluation reveals that while Google Scholar optimizes for high recall, often retrieving irrelevant papers sharing lexical keywords, Asta achieves superior precision through semantic reasoning and relevance verification. Furthermore, this report analyzes the underlying architecture of Asta, identifies "Relevance Verification" as a crucial step, and details a Python-based reproduction of this mechanism to demonstrate its effectiveness in filtering search noise.**

INTRODUCTION

The rapid growth of scientific literature presents significant challenges for researchers attempting to find and access relevant information. Traditional search engines, such as Google Scholar, rely primarily on lexical keyword matching to retrieve documents. While effective for navigational queries, such as finding a specific known paper, these systems struggle with information synthesis and comprehension. These limitations become apparent when users seek answers to complex questions rather than specific documents. In these scenarios, simple keywords often fail to capture the user's research intent.

"Asta Find Papers" represents a shift toward agentic search. Instead of simply indexing documents, it functions as a research knowledge agent that plans queries, retrieves evidence from multiple sources, and verifies relevance using Large Language Models (LLMs).

This report presents a comprehensive analysis of the "Asta Find Papers" service. Section I evaluates its performance relative to Google Scholar. Section II examines its hybrid, AI-enabled retrieval architecture. Finally, Section III details the implementation of a custom relevance verifier agent, reproducing Asta's core filtering logic to mitigate hallucination and search noise.

I. EVALUATION (ASTA FIND PAPERS VS. GOOGLE SCHOLAR)

A. Methodology

To evaluate the Asta Find Papers service relative to Google Scholar, I adopted a qualitative methodology centered on four core metrics:

- **Precision:** The proportion of retrieved results that are semantically relevant to the user's specific domain.
- **Synthesis Quality:** The system's ability to extract and integrate answers rather than merely listing bibliographic links.
- **Transparency:** The traceability of generated claims back to their original primary sources.
- **Context Awareness:** The extent to which the system captures user intent beyond simple keyword matching.

I tested two distinct classes of queries representing common research workflows:

1) **Survey Query:** "What are the latest mitigation strategies for object hallucination in LVLMs?" (Goal: Discovery and summarization).
2) **Context-Constraint Query:** "How does cross-correlation synchronize IMU and camera data for liveness detection?" (Goal: Focused technical explanation and semantic filtering).

B. Performance of Google Scholar (The Baseline)

I first evaluated the baseline system, Google Scholar, using the survey query regarding object hallucination in Large Vision-Language Models (LVLMs) (See Fig. 1).

Google Scholar demonstrated high recall and low latency, successfully retrieving a raw list of recent papers in milliseconds. However, the lack of synthesis capabilities requires manual verification, where the user must individually click and vet papers to identify specific strategies hidden within the PDF content. While the platform maintains high transparency by providing direct links, it exhibits limited context awareness, often presenting redundant titles without grouping or ranking by semantic relevance.

I next evaluated Google Scholar using the context-constraint query regarding IMU/camera synchronization for liveness detection (See Fig. 2).

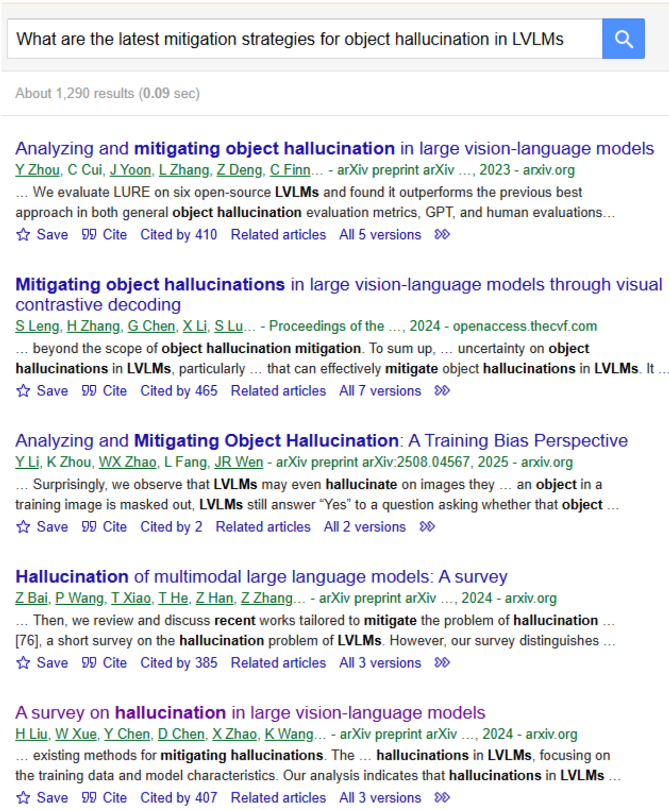

Fig. 1. Google Scholar response to Query 1

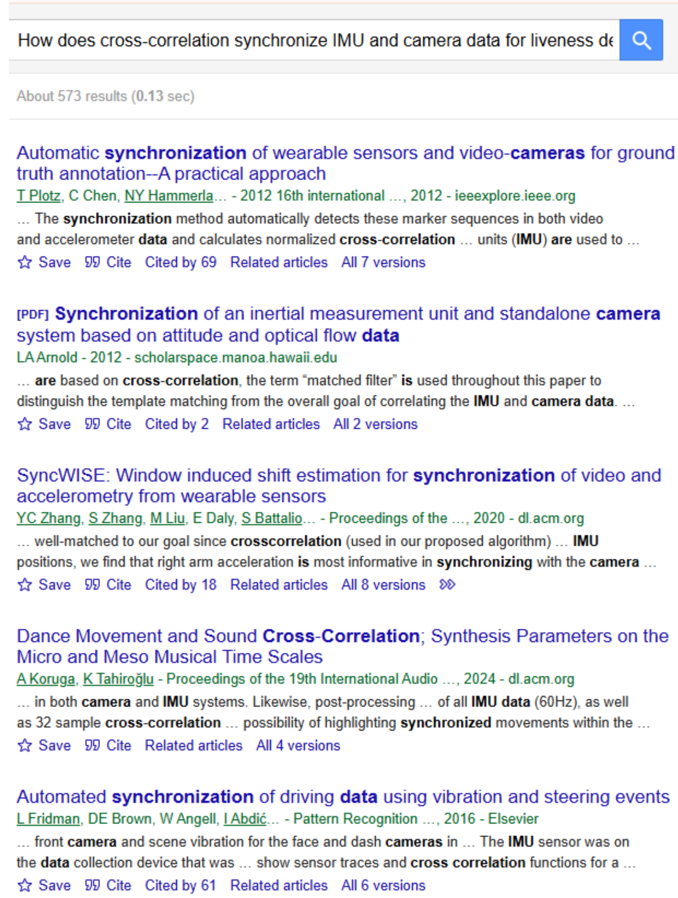

Fig. 2. Google Scholar response to Query 2

The baseline system failed significantly due to lexical noise. Scholar's keyword-based retrieval algorithm prioritized terms like "cross-correlation" and "IMU" without understanding the biometric context of "liveness detection." Consequently, the search returned irrelevant results, such as papers on synchronizing dance movements to music or steering vehicles. This confirms that while Google Scholar excels at lexical matching, it struggles to capture complex research intent when domain-specific constraints are applied.

C. Performance of Asta (The Agent)

Using the same queries, I evaluated the agentic system, "Asta Find Papers."

For the query regarding LVLMs (See Fig. 3), Asta functioned as an intelligent filter rather than a simple indexer. While the process took approximately 33 seconds, the system achieved excellent precision. It did not just list papers but synthesized the content to explicitly name mitigation methods found within the literature, such as LURE, Visual Contrastive Decoding (VCD), and LRV-Instruction. Furthermore, Asta provided a relevance score for each result, justifying its selection with specific evidence snippets. This approach reduces the manual burden of reading while maintaining transparency through the use of direct citations.

For the query regarding how cross-correlation synchronizes IMU and camera data for liveness detection (See Fig. 4), the difference in architecture became most apparent. Asta took over seven minutes to process the query but correctly identified the semantic mismatch in the available literature. Unlike Google Scholar, which forced irrelevant results, Asta correctly reported that it could not find papers specifically linking cross-correlation synchronization to liveness detection. It provided a negative result warning, accurately distinguishing between the presence of keywords and the presence of the specific requested concept. This demonstrates a high degree of active verification, preventing the user from wasting time on semantically irrelevant papers.

D. Comparative Analysis and Summary

The structural differences between the two systems are summarized in Table I.

While Google Scholar remains the better tool for navigational queries (finding a known paper), Asta demonstrates a clear advantage for informational queries where synthesis and noise reduction are critical. Asta effectively bridges the gap between search and knowledge acquisition by extracting specific information that traditional engines leave buried in PDFs. The trade-off for this intelligence is latency, but the computational time is negligible compared to the human hours saved in manual literature review.

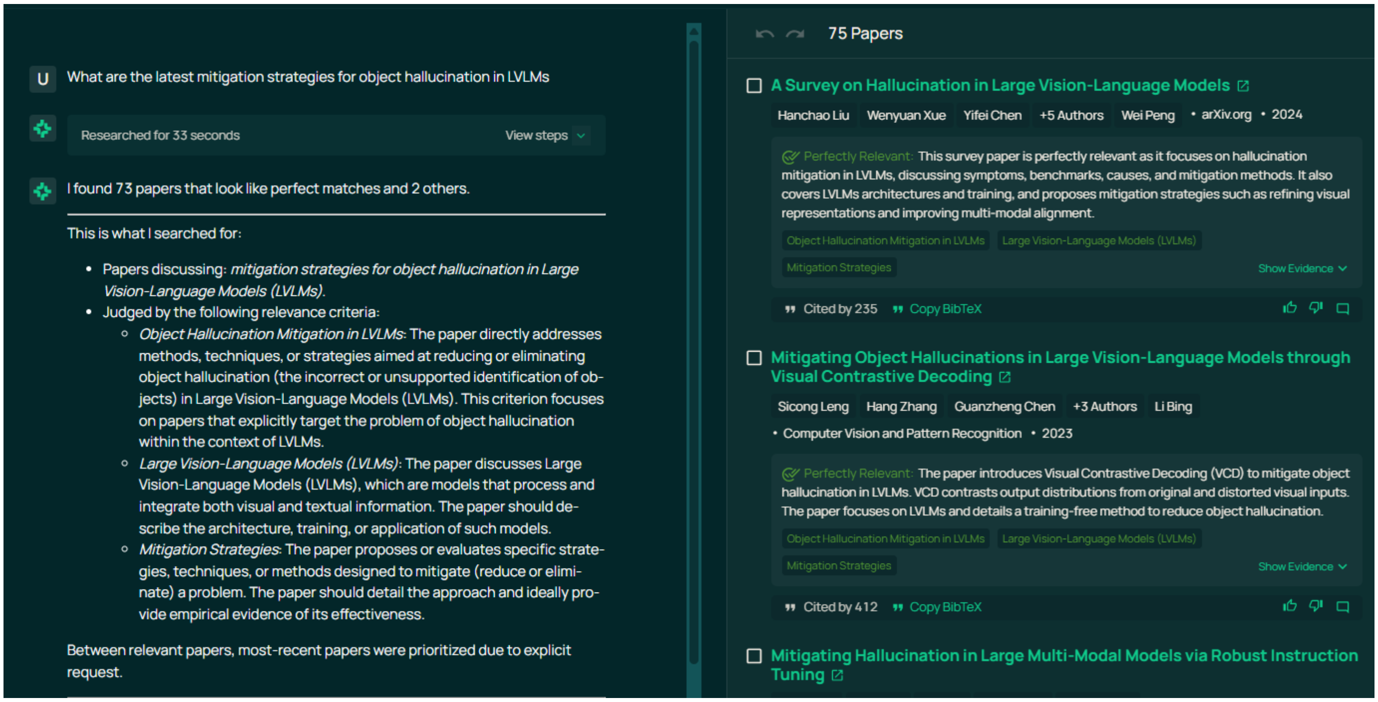

Fig. 3. Asta Find Papers response to Query 1

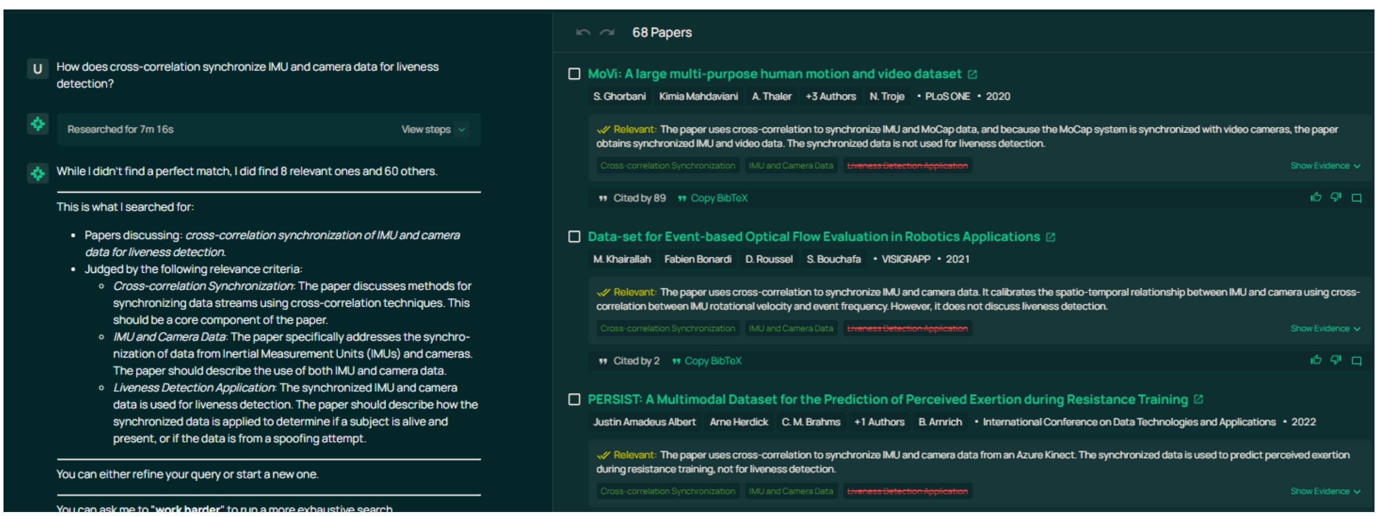

Fig. 4. Asta Find Papers response to Query 2

TABLE I
COMPARISON OF SEARCH ATTRIBUTES

Feature	**Google Scholar**	**Asta (Agentic)**
Search Logic	Lexical (Keyword)	Semantic + Reasoning
Precision	Low (Noisy)	High (Context-Aware)
Output	List of Links	Synthesized Answer
User Effort	High (Manual Reading)	Low (Auto-Extraction)
Failure Mode	Misinterpretation of Intent	Hallucination Risk
Latency	Milliseconds	Seconds to Minutes
User Exp.	Retrieval-Centric	Answer-Centric

II. Architectural Analysis

A. High-Level Architecture: The Agentic Workflow

Asta Find Papers operates not just as a search indexer, but as an autonomous knowledge agent. Unlike Google Scholar, which relies on static retrieval, Asta plans and executes a multi-step research workflow. It mimics the cognitive process of a human researcher: iteratively searching, scanning abstracts, verifying citations, discarding irrelevant results, and refining the search strategy based on intermediate findings.

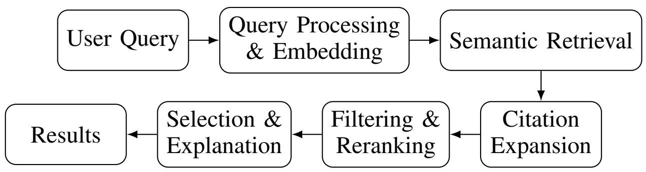

Fig. 5. High-level workflow of Asta Find Papers.

B. The Step-by-Step Workflow From Query to Result

The Asta Paper Finder generally follows the workflow of a standard LLM-driven retrieval system. The steps described below closely align with the methodology described in [1]:

1) **Query Processing & Embedding:** The system translates a user's natural language request into embedding vectors, capturing semantic meaning, key concepts, and metadata constraints.
Challenge: Users often use ambiguous, colloquial terms or vague descriptions, leading to a vocabulary mismatch between the query and the academic terminology.
Solution: Asta utilizes an LLM to analyze the raw query, decomposing it into core scientific concepts and specific metadata constraints before generating high-quality embeddings.
2) **Semantic Retrieval:** The system identifies a candidate set of papers from a database of hundreds of millions using semantic similarity measures (e.g., Semantic Scholar).
Challenge: Vector search at this scale is computationally intensive. Furthermore, purely semantic searches can yield papers that use similar language but discuss entirely different topics.
Solution: Asta employs Approximate Nearest Neighbor (ANN) algorithms for speed, running these in parallel with standard keyword-based API calls to ensure exact matches are not lost by the vector model.
3) **Citation Expansion:** For the top retrieved papers, the system explores citation graphs to discover relevant work that may not match the initial query terms directly.
Challenge: Pure semantic search is limited by the initial vocabulary. Important foundational papers or novel applications may use different terminology. Additionally, citation graphs are noisy, and not all cited papers are relevant to the specific claim being investigated.
Solution: Asta performs a bidirectional traversal of the citation graph starting from high-confidence nodes. It applies relevance-aware filtering to retain only those citations that match the user's research intent.
4) **Filtering & Reranking:** The system narrows the expanded pool to the most pertinent results using LLM-based relevance judgment.
Challenge: A paper might be semantically relevant but outdated, or highly cited but irrelevant to the specific nuances of the user's query.
Solution: An LLM-based reranker evaluates metadata, abstracts, and extracted passages. It assigns a relevance score by combining semantic similarity with other signals, while applying metadata filters.
5) **Selection & Explanation:** The system presents the final results to the user with justifications, evidence snippets, and bibliographic data.
Challenge: Standard search engines provide context-free snippets that force users to read the full text. Additionally, LLM-generated summaries risk hallucination if not strictly grounded.
Solution: Asta utilizes Retrieval-Augmented Generation (RAG) to ensure all summaries are traceable to specific passages. The LLM acts as a relevance judge, outputting both the decision logic and the supporting evidence.

III. Implementation (The Knowledge Agent)

One key task in building Asta is relevance verification via filtering and reranking. This step transforms a system from a passive search engine into an active knowledge agent by having an LLM explicitly evaluate whether each retrieved paper truly addresses the user's intent. Historically, this task was performed by scoring document pairs based on text similarity. However, these models lack logical reasoning capabilities, and the current state-of-the-art has shifted toward LLM-as-a-Judge architectures, which allow the system to not only score a paper but to justify that score with extracted evidence.

To validate this architectural finding, I developed a Python-based relevance verifier that reproduces Asta's core filtering logic. [1]

A. System Design

The implementation integrates the **Semantic Scholar API** for broad retrieval and **Google Gemini 2.5 Flash-Lite** as the reasoning engine. The agent operates in a three-stage workflow:

1) **Broad Retrieval:** The agent fetches a candidate pool of raw papers ($N = 10$) using standard keyword search. This simulates the high-recall/low-precision output of traditional engines.
2) **Agentic Verification Loop:** Instead of displaying these links immediately, the agent enters a verification loop. It iterates through each candidate, passing the abstract and the user's query to the LLM with a strict prompt to determine relevance and provide reasoning and evidence.

[1] Code available at: https://github.com/amey-gupt/Scholar-Verifier-Agent

3) **Evidence Extraction:** The system applies a hard filter (Score > 5) and renders the remaining papers in a structured table, highlighting the specific evidence snippet that justified the selection.

B. Experimental Results

Initially, I tested the agent using the second query from Section I; however, Semantic Scholar returned zero papers because the query was too specific. I therefore evaluated the system using the technical query: *"IMU camera synchronization cross-correlation."*

The broad retrieval stage returned 9 raw candidate papers. As shown in Fig. 6, the agent successfully filtered search noise, reducing the pool to 4 verified papers. The LLM correctly discarded papers that merely mentioned "cross-correlation" in unrelated contexts while retaining papers that explicitly discussed synchronization algorithms.

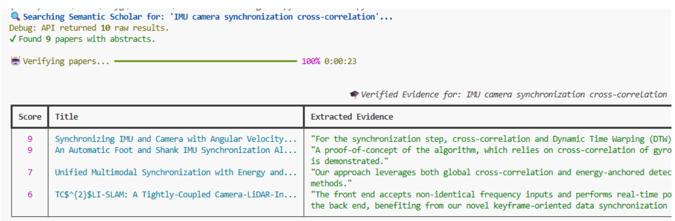

Fig. 6. Relevance Verification Implementation Result. The agent filtered 9 raw papers down to 4, extracting the specific evidence sentence for each match.

This proof-of-concept confirms that integrating LLMs as active judges significantly increases search precision.

C. Limitations and Future Improvements

To further approximate the full Asta system and improve the current implementation, a key next step would be to add a query processing and decomposition module. Rather than relying on the user's raw input, the agent could reformulate the query, decompose it into sub-questions, and better capture user intent before querying Semantic Scholar. This would enable the retrieval of a more diverse and representative candidate pool, further improving relevance verification.

Conclusion

This report evaluated Asta Find Papers and demonstrated its superiority over traditional search engines for complex knowledge discovery tasks. By shifting from semantic retrieval to agentic reasoning, Asta solves the information overload problem, filtering noise and synthesizing answers rather than just listing links. My reproduction of the Relevance Verification step confirms that integrating LLMs as active judges in the search loop is the key architectural shift enabling this performance. As scientific literature continues to grow, such agentic systems will become indispensable tools for researchers.

References

[1] "Introducing Ai2 Paper Finder," Allen Institute for AI Blog. [Online]. Available: https://allenai.org/blog/paper-finder.