Surveying and Building an Automatic Knowledge Agent: An Evaluation of LLM-Based Literature Retrieval Systems

1. Evaluation of Asta and Comparison with Keyword-Based Databases

As generative artificial intelligence (GenAI) changes how we learn and research, understanding how users interact with AI agents during literature search is important for improving research efficiency. This section evaluates Asta's "Find Papers" module—a search agent driven by meaning—and contrasts its performance with traditional keyword-based databases like Google Scholar.

1.1. Performance Evaluation of the Asta Agent

Asta shows strong abilities in understanding the real meaning behind users' natural language prompts. Unlike traditional databases that require highly precise search terms, Asta allows researchers to describe their ideas using everyday language. This reduces the mental effort and lowers the chance of missing important literature.

Furthermore, Asta replaces fixed traditional filters with interactive conversational refinement. If an initial query does not yield great results, researchers can provide more explanations to get better articles. Beyond understanding natural language, Asta uses strict rules to stay on topic. During a test asking "what is 1+1", Asta redirected: "If you're interested in academic papers related to mathematics... I can help". This shows the system actively steers off-topic dialogue back to finding literature.

Crucially, Asta reduces GenAI hallucination risks through a clear and transparent process. The system manages an evidence-based selection process by extracting specific text quotes from papers to support its choices. When asked about a completely fake concept (e.g., "Quantum-Neuro Flux Capacitor Model"), Asta successfully avoided making up fake citations—showing a strict reliance on real data.

However, Asta has a few practical limits. The system restricts its results, often keeping them under 200 papers. Also, it lacks automated quality checks like journal impact factors, requiring users to use their own metacognitive monitoring to check source reliability before fully trusting the results.

1.2. Case Studies in Refining Searches Step-by-Step

To test Asta's capabilities, two search scenarios were analyzed to show how it handles follow-up questions and context.

In the first case, searching for "disciplinary practice" using simple keywords returned fewer than 100 papers. However, performance improved when the search was refined through conversation. By breaking down the task with clear context—such as focusing on students' higher-order abilities and providing similar examples like "scientific practice"—the agent successfully connected the idea to a highly relevant set of literature.

In the second case, searching a popular domain like "AI-empowered instructional design" initially returned too many documents. Asta's advantage showed during multi-turn interactive refinement. When the user added follow-up rules—specifically

focusing on helping teachers and explicitly excluding student interaction with AI—the agent actively created new categories. This allowed the literature to be filtered through specific tags rather than fixed database filters.

1.3. Comparative Analysis: Asta vs. Google Scholar

The main difference between Asta and Google Scholar is their underlying search methods. Google Scholar relies mostly on exact keyword matching, which is still better for fast, citation-based ranking. However, for vague or descriptive queries, this keyword focus often leads to poor matching.

In contrast, Asta uses semantic search to understand the context behind descriptive questions. As a result, the user experience is very different: while traditional databases provide a flat list of links, Asta categorizes results into a three-relevance level ("Perfectly Relevant," "Relevant," and "Slightly Relevant") and presents extracted evidence for each. Furthermore, Asta retains the conversational history, allowing researchers to easily backtrack and adjust their search parameters. Although Asta trades processing speed (~40 seconds per query) for deeper meaning compared to Scholar's instant retrieval, it takes on the heavy lifting of initial reading, saving researchers hours of manual work.

2. System Architecture

To understand how Asta works, practical observations of its execution logs were analyzed alongside recent Information Retrieval (IR) literature. Asta shifts from traditional retrieve-and-rank databases to a Criteria-Driven Retrieval-Augmented Generation (RAG) pipeline (Lewis et al., 2020). The workflow has four main stages.

Step 1: Query Understanding. Traditional databases fail when users use descriptive or noisy language. Asta addresses this challenge by using a Large Language Model (LLM) to translate conversational prompts into structured sub-queries and filtering rules before searching (Dong & Zhang, 2025; Peng et al., 2024). Similar to zero-shot dense retrieval techniques like Hypothetical Document Embeddings (HyDE), this step successfully bridges the gap between a researcher's description and formal academic terms (Gao et al., 2023).

Step 2: Hybrid Retrieval over Dynamic Corpora. Scanning the full text of millions of papers word-by-word is impossible. Asta uses a combined approach that pairs meaning-based dense vector embeddings (e.g., SPECTER) (Cohan et al., 2020; Karpukhin et al., 2020) with traditional keyword matching (e.g., BM25) (Ahluwalia et al., 2024; Monir et al., 2024). This ensures Asta maintains high recall across a constantly updating database (Chen et al., 2023).

Step 3: Citation Search and Text Extraction. Relying only on abstracts can lead to shallow scoring. Asta runs a detailed evidence-gathering phase by breaking full texts into readable sections and exploring citation networks(Ammar et al., 2018). By extracting "citances"—the specific sentences containing the citation—the system uses the external academic community's context to approximate a human researcher's literature review process (Li & Ouyang, 2025; Syed et al., 2023).

Step 4: Criteria-Based Ranking and Synthesis. Traditional RAG models often score documents with math and then force the LLM to make up an explanation later, which increases hallucination risks (Lewis et al., 2020). Asta reverses the order. The LLM acts as the final judge, comparing the extracted text from Step 3 strictly against the specific criteria from Step 1. This evidence-first approach ensures that recommendations are logically justified conclusions checked against the actual text (Edelman & Skolnick, 2025; Huang et al., 2024).

3. Key Task Implementation

3.1. Task Definition

Based on the workflow analyzed, Criteria-Grounded Reranking and Synthesis (Step 4) was identified as the most critical task to build. The goal is to automate the evaluation of a candidate paper's relevance based on dynamic criteria, while extracting clear text evidence to justify the decision.

To achieve this, a Python-based prototype was developed. The complete source code for this prototype is publicly available on GitHub (*https://github.com/Beichen-H/Research-Agent*). The key idea is simple: instead of trusting the model, I force it to justify every decision with evidence and then verify it.

3.2. State of the Art (SOTA) Analysis

This core idea is driven directly by recent advancements in State of the Art (SOTA) research. Traditionally, systems would score documents first and then ask the LLM to write a summary, which often led to made-up, after-the-fact explanations.

To fix this, recent literature shows a shift towards "Attributed LLMs". Rather than relying on simple text search, current SOTA frameworks require models to actively resist hallucinations through strict workflows:

1. Forced Evidence Extraction: Techniques like the "Evidence to Generate" (E2G) strategy (Parvez, 2025) ensure that the AI isolates verified text quotes *before* making any judgments.
2. Logical Reasoning Chains: Advanced models now use frameworks like TRACE (Fang et al., 2024; Sun et al., 2025) to explicitly map out how different sentences connect to form a logical argument.
3. Self-Verification: Tools like the AGREE framework (Ye et al., 2024) require the AI to check its own citations and confirm that its claims are grounded in the extracted text.

By combining these innovations, modern systems can function as verifiable academic judges, similar to large-scale claim verification tools like Valsci (Edelman & Skolnick, 2025).

3.3. Solution Implementation

This Python-based prototype moves beyond simple prompts by using a strict, multi-step pipeline designed to act as a discerning evaluator:

1. Query Expansion: The system translates conversational queries into structured academic screening criteria.

2. Combined Scoring: A Multi-Agent system calculates both a semantic vector score and a logical LLM-judge score to filter out completely mismatched documents.
3. Evidence-Based Reasoning Chain: To reduce hallucination risks during the final step, the LLM is forced to first extract exact quotes without summarizing (Parvez, 2025). Next, it connects these quotes logically to build a reasoning chain (Fang et al., 2024). Finally, it writes a relevance report based only on the extracted evidence.
4. Faithfulness Check (Self-Verification): An independent LLM auditor does a final check to confirm that the generated answer does not exaggerate and is truly supported by the extracted quotes (Ye et al., 2024).

This pipeline is directly demonstrated in the accompanying video, where the system enforces evidence extraction and rule-based filtering in real time.

3.4. Limitations and Future Improvements

While the proposed Python-based prototype system effectively reduces text-based hallucination risks through its grounded synthesis module, rigorous stress-testing revealed a few natural limits of zero-shot RAG pipelines that need future improvement:

1. Semantic Role Confusion, the LLM sometimes failed to distinguish between a paper's "actual research method" and its "testing dataset". For example, when searching for "using LLMs to generate quiz questions," the model incorrectly retrieved and highly scored a medical paper simply because it used "medical quiz cases" to test its own model. This indicates that zero-shot attention mechanisms are vulnerable to keyword-based interference, causing the model to confuse the actual role of a concept within the text.
2. Over-Generalizing Negative Constraints Using strict exclusion rules (such as "exclude model fine-tuning") exposed a critical flaw in LLM reasoning. The system often applied these negative rules too broadly, incorrectly assuming that any paper proposing a "new framework" automatically involved model training. This shows the logical weakness of negative prompting when dealing with complex academic topics.
3. Citation Bias in Hybrid Scoring Early tests of the multi-agent system showed that a very high citation score (e.g., over 100 citations) could artificially hide a low relevance score. This scoring bias allowed highly-cited but off-topic papers to incorrectly dominate the top rankings, creating a "fame over quality" issue where popularity overrides rule-following.
4. The Precision-Recall Trade-off (Risk of False Negatives) While enforcing strict rejection rules effectively lowers hallucination risks, it naturally makes the system too inflexible. This high-precision approach may lead to "false negatives," dropping valuable, cross-disciplinary papers simply because they briefly mention an excluded term (e.g., mentioning "fine-tuning" only for comparison). Finding the right balance between strict quality control and acceptable flexibility remains a practical challenge.

Future Improvements: To address these limitations, future versions should move beyond zero-shot prompting by using "Dynamic Few-Shot In-Context Learning (ICL)".

Providing the LLM with clear guiding examples—such as showing the exact difference between "RAG" and "Fine-tuning"—will greatly improve the logical strength of the reasoning chain. Furthermore, future research should test and adjust the scoring system, perhaps replacing the absolute zero score with flexible penalty weights to balance precision and recall.

References

[1] Aman Ahluwalia, Bishwajit Sutradhar, Karishma Ghosh, Indrapal Yadav, Arpan Sheetal, and Prashant Patil. Hybrid semantic search: Unveiling user intent beyond keywords. arXiv preprint arXiv:2408.09236, 2024.

[2] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. SPECTER: Document-level representation learning using citation-informed transformers. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2270–2282, 2020. https://doi.org/10.18653/v1/2020.acl-main.207.

[3] Benjamin Edelman and Jeremy Skolnick. Valsci: An open-source, self-hostable literature review utility for automated large-batch scientific claim verification using large language models. BMC Bioinformatics, 26:140, 2025. https://doi.org/10.1186/s12859-025-06159-4.

[4] Guanting Dong and Yue Zhang. Leveraging LLM-assisted query understanding for live retrieval-augmented generation. arXiv preprint arXiv:2506.21384, 2025.

[5] Hao Sun, Hengyi Cai, Yuchen Li, Xuanbo Fan, Xiaochi Wei, Shuaiqiang Wang, Yan Zhang, and Dawei Yin. Enhancing retrieval-augmented generation via evidence tree search. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 24116–24127, 2025. https://doi.org/10.18653/v1/2025.acl-long.1175.

[6] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, and Yixing Fan. Continual learning for generative retrieval over dynamic corpora. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pages 306–315, 2023. https://doi.org/10.1145/3583780.3614821.

[7] Jinyuan Fang, Zaiqiao Meng, and Craig MacDonald. TRACE the evidence: Constructing knowledge-grounded reasoning chains for retrieval-augmented generation. In Findings of the Association for Computational Linguistics: EMNLP 2024, pages 8472–8494, 2024. https://doi.org/10.18653/v1/2024.findings-emnlp.496.

[8] Lei Huang, Xiaocheng Feng, Weitao Ma, Yuxuan Gu, Weihong Zhong, Xiachong Feng, Weijiang Yu, et al. Learning fine-grained grounded citations for attributed large language models. In Findings of the Association for Computational Linguistics: ACL 2024, pages 14095–14113, 2024. https://doi.org/10.18653/v1/2024.findings-acl.838.

[9] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, pages 1762–1777, 2023. https://doi.org/10.18653/v1/2023.acl-long.99.

[10] Md Rizwan Parvez. Chain of evidences and evidence to generate: Prompting for context grounded and retrieval augmented reasoning. In Proceedings of the 4th International Workshop on Knowledge-Augmented Methods for Natural Language Processing, pages 230–245, 2025. https://doi.org/10.18653/v1/2025.knowledgenlp-1.21.

[11] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33:9459–9474, 2020.

[12] Shahbaz Syed, Ahmad Dawar Hakimi, Khalid Al-Khatib, and Martin Potthast. Citance-contextualized summarization of scientific papers. In Findings of the

Association for Computational Linguistics: EMNLP 2023, pages 8551–8568, 2023. https://doi.org/10.18653/v1/2023.findings-emnlp.573.

[13] Solmaz Seyed Monir, Irene Lau, Shubing Yang, and Dongfang Zhao. VectorSearch: Enhancing document retrieval with semantic embeddings and optimized search. arXiv preprint arXiv:2409.17383, 2024.

[14] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 6769–6781, 2020. https://doi.org/10.18653/v1/2020.emnlp-main.550.

[15] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, et al. Construction of the literature graph in Semantic Scholar. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 84–91, 2018. https://doi.org/10.18653/v1/N18-3011.

[16] Wenjun Peng, Zilong Wang, Guiyang Li, Dan Ou, Derong Xu, and Tong Xu. Large language model based long-tail query rewriting in Taobao search. In Companion Proceedings of the ACM Web Conference 2024, pages 20–28, 2024. https://doi.org/10.1145/3589335.3648298.

[17] Xi Ye, Ruoxi Sun, Sercan Arik, and Tomas Pfister. Effective large language model adaptation for improved grounding and citation generation. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6237–6251, 2024. https://doi.org/10.18653/v1/2024.naacl-long.346.

[18] Xiangci Li and Jessica Ouyang. Explaining relationships among research papers. In Proceedings of the 31st International Conference on Computational Linguistics, pages 1080–1105, 2025.