Revolutionary Advances in Computing and Electronics (RACE): An International Journal
Volume 02, Issue 02, pp.33-39, 2026
ISSN (Online): 3107-8540

Received: 02/06/2026, Accepted: 02/07/2026

Deep Research Agent: An AI-Powered System for Automated Research Paper Analysis and Citation-Based Answer Generation

Mukul Negi, Rajdeep Ramola, Vipul Bijalwan, Gopal Datt, Aakanksha Pundir

Department of Computer Science and Engineering, Shivalik College of Engineering, Dehradun, Uttarakhand, India

negimukul2004@gmail.com, rajdeepramola958@gmail.com, vipulbijalwan112@gmail.com, gopaldattsharma90@gmail.com, aakanksha.pundir@sce.org.in

Abstract

The rapid growth of scholarly publications makes it difficult for researchers and students to locate relevant evidence, compare findings, and produce source-grounded answers efficiently. This paper presents Deep Research Agent, an AI-powered system for the automated analysis of research papers and the generation of citation-based answers. The system ingests uploaded PDF research papers, extracts and cleans the text, segments the content into retrievable passages, generates semantic embeddings using Sentence-BERT, stores the vectors in ChromaDB, and combines dense retrieval with BM25 keyword matching. Retrieved evidence is passed to a large language model via citation-aware prompts, ensuring that generated answers remain linked to the source passages. A Neo4j knowledge graph layer supports exploration of entity and topic relationships, while a Streamlit interface provides an accessible workflow for uploading papers and asking research questions. Compared with generic RAG assistants, the proposed system focuses specifically on local scholarly-paper analysis, hybrid retrieval, explicit evidence mapping, and citation-grounded responses. The revised manuscript also discusses practical deployment issues, including document quality, retrieval latency, privacy, evaluation, and scalability.

Keywords: *Deep Research Agent, Retrieval-Augmented Generation, Large Language Models, Citation-Aware Generation, Scholarly Retrieval, ChromaDB, Neo4j.*

1. Introduction

Researchers increasingly work with large collections of papers, technical reports, and preprints. Manual reading remains essential, but the early stages of literature exploration often involve repetitive tasks: extracting key claims, locating relevant passages, comparing related work, and verifying the source of a generated answer. Large language models (LLMs) can summarize and answer questions fluently, yet their answers may be unsupported, outdated, or difficult to verify when used without retrieval and citation controls.

Retrieval-Augmented Generation (RAG) addresses part of this problem by grounding generation in retrieved external evidence. However, many general RAG systems are optimized for broad enterprise search or open-domain question answering rather than scholarly-paper workflows. Research papers contain dense terminology, section-specific context, figures, equations, references, and claims that must be attributed carefully. A useful research assistant must therefore

*Corresponding author: Mukul Negi, Department of Computer Science and Engineering, Shivalik College of Engineering, Dehradun, India. (negimukul2004@gmail.com)

combine semantic retrieval, keyword retrieval, metadata tracking, citation mapping, and transparent answer generation [1-5].

This paper proposes Deep Research Agent, a lightweight system that allows users to upload research papers and receive citation-based answers grounded in the uploaded corpus. The system is designed for academic and technical users who need a practical tool for paper analysis without having to build a full-scale digital library platform.

1.1 Novelty and Contributions

The novelty of the proposed work lies not in the use of RAG alone, but in the integration of several components into a citation-aware scholarly workflow. The system differs from general RAG assistants in four ways:

- It targets uploaded research papers rather than general web or enterprise documents, preserving paper metadata and source passage identifiers.
- It combines dense SBERT-based retrieval with BM25 keyword matching to improve recall for both semantic queries and exact technical terms.
- It maintains a citation map between retrieved chunks and generated answer statements, reducing unsupported responses and improving traceability.
- It augments vector retrieval with a Neo4j knowledge graph layer so users can explore relationships among topics, methods, datasets, and papers.

2. Literature Review

Early RAG work established the value of combining parametric language models with non-parametric retrieval for knowledge-intensive tasks. Recent studies from 2024 onward have moved the field toward modular retrieval pipelines, citation-aware generation, and domain-specific research assistance. These studies show that retrieval improves factual grounding, but they also emphasize that citation quality, retrieval robustness, context selection, and evaluation remain unresolved challenges.

2.1 Recent Literature from 2024-2026

Table 1 extends the literature review with recent work on RAG, scholarly retrieval, citation-aware generation, and deployed LLM pipelines. The comparison also clarifies the research gap that the proposed Deep Research Agent addresses.

Study	Focus	Key contribution	Limitation for scholarly QA
Fan et al. (2024) [6]	Retrieval-augmented LLMs	Surveyed RA-LLM architectures, training strategies, and applications.	Broad survey; does not provide an end-user citation workflow for uploaded papers.

Huang and Huang (2024) [7]	RAG text generation	Organized RAG into pre-retrieval, retrieval, post-retrieval, and generation stages.	Primarily methodological; limited treatment of paper-level citation traceability.
Zhao et al. (2024) [8]	RAG for AI-generated content	Reviewed RAG foundations, benchmarks, modalities, and engineering practices.	Not specialized for research-paper analysis or citation-grounded answers.
Li et al. (2024) [9]	Citation-enhanced generation	Used retrieval and NLI-style checks to improve citation support in chatbot responses.	Post-hoc citation support is useful, but it is not designed as a full research-paper ingestion system.
Qian et al. (2024) [10]	LLM citation capacity	Evaluated citation correctness and proposed citation refinement metrics and methods.	Focuses on citation quality evaluation rather than integrated scholarly document analysis.
Wang et al. (2025) [11]	ScholarCopilot	Introduced citation-aware academic writing support with retrieval tokens and scholarly reference lookup.	Targets academic writing generation; less emphasis on local PDF analysis and knowledge graph exploration.
Jiang et al. (2025) [12]	Retrieval and structuring augmented generation	Highlighted structured representations, taxonomies, and knowledge integration for LLMs.	Conceptual survey; implementation details for lightweight student/researcher tools remain open.
Fingleton et al. (2026) [13]	RAG for software testing and inspection	Demonstrated practical RAG benefits in automated software V&V workflows.	Domain-specific to software inspection; not focused on research literature retrieval.
Proposed Deep Research Agent (2026)	Citation-aware research assistant	Combines PDF ingestion, hybrid BM25+dense retrieval, ChromaDB, Neo4j relationships, and grounded answer generation.	Current prototype requires broader evaluation, live database integrations, and scaled deployment testing.

2.2 Research Gap

The reviewed literature indicates three gaps. First, many RAG studies focus on general benchmarks rather than practical scholarly paper workflows. Second, citation-generation studies

often evaluate citation correctness but do not provide a complete local pipeline for the analysis of uploaded PDFs. Third, systems that support academic writing do not always expose the path to the evidence they retrieve to the user. Deep Research Agent addresses these gaps by combining local document ingestion, hybrid retrieval, citation-aware prompting, and graph-based relationship exploration into a single prototype.

3. Proposed Methodology

The system follows a modular pipeline consisting of document ingestion, indexing, retrieval, evidence management, answer generation, and user interaction.

- Document ingestion: Uploaded PDF papers are parsed, cleaned, and divided into manageable chunks while retaining file name, page number, and section metadata where available.
- Embedding and storage: Each chunk is represented using Sentence-BERT embeddings and stored in ChromaDB for efficient semantic search.
- Hybrid retrieval: A BM25 index supports exact keyword matching, while dense retrieval captures semantic similarity. The retrieved candidates are fused and re-ranked.
- Evidence management: The top passages are assigned stable source identifiers and passed to the LLM with instructions to answer only from retrieved evidence.
- Citation-based generation: The answer generator produces concise responses with citations linked to source chunks, enabling users to verify statements.
- Knowledge graph support: Neo4j stores extracted entities and relationships such as paper-topic, method-dataset, and concept-concept links.

4. System Architecture

Figure 1 replaces the earlier low-resolution architecture with a clearer, high-resolution view of module interactions. The diagram separates ingestion, indexing, retrieval, evidence management, generation, graph exploration, and verified output to make the execution flow easier to follow.

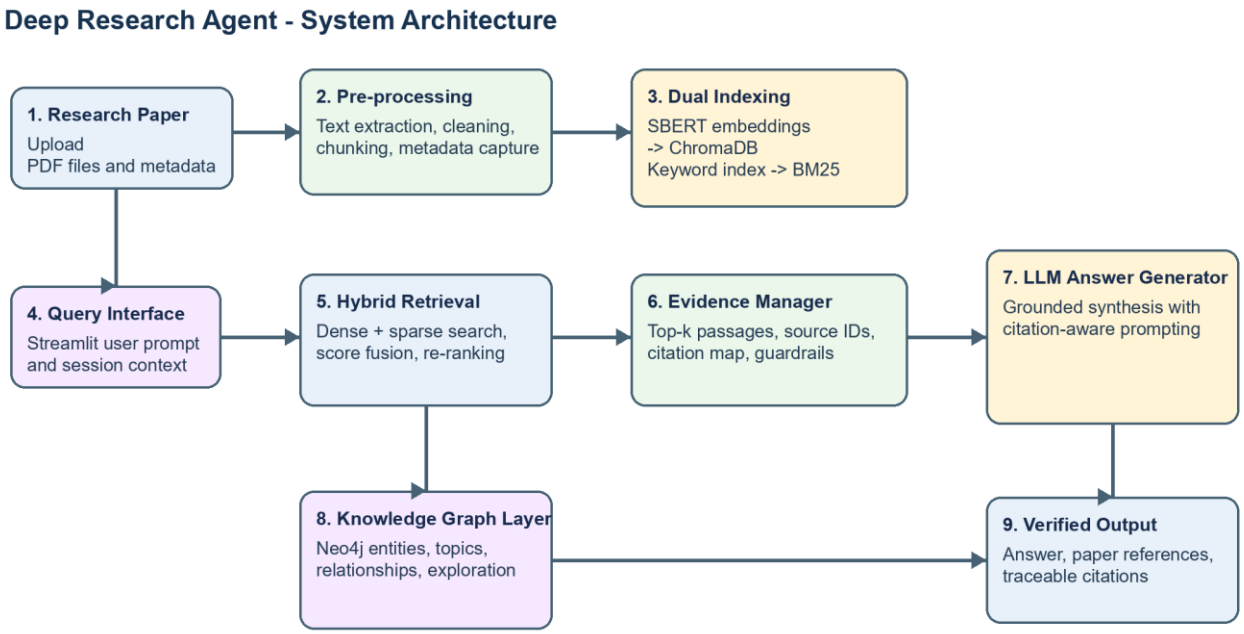

Figure 1. Revised high-resolution architecture showing data ingestion, hybrid retrieval, evidence management, LLM generation, and citation verification.

Figure 1. Revised system architecture of Deep Research Agent.

5. Implementation

The prototype is implemented in Python, using LangChain for pipeline orchestration, ChromaDB for vector storage, Neo4j for relationship storage, BM25 for sparse retrieval, and Streamlit for the user interface. The typical workflow begins when a user uploads one or more PDF papers. The system extracts text, preprocesses the content, generates embeddings, stores chunks with metadata, and builds a keyword index. When the user submits a query, the system retrieves relevant chunks using hybrid search and sends a compact evidence bundle to the LLM. The generated answer includes source identifiers so the user can inspect the supporting passages.

The implementation deliberately separates retrieval and generation. This separation supports future replacement of the LLM, vector database, embedding model, or ranking method without redesigning the whole system. It also makes evaluation easier because retrieval quality and answer quality can be measured independently.

6. Results and Discussion

In prototype testing, the system successfully processed uploaded research papers, generated vector representations, retrieved relevant passages, and produced citation-based answers through the Streamlit interface. The hybrid retrieval design is useful because scholarly queries often contain both broad semantic intent and exact technical terms such as model names, datasets, algorithms, or software libraries. Dense retrieval improves semantic matching, while BM25 reduces the risk of missing exact terminology.

The main benefit of the system is improved answer traceability. Instead of returning unsupported summaries, the answer generator receives a restricted evidence bundle and produces responses tied to the retrieved passages. This design does not eliminate all hallucination risks, but it gives users a practical verification path and helps identify weak answers when the retrieved evidence is insufficient.

7. Practical Deployment Challenges

Real-world deployment requires attention to document quality, privacy, cost, latency, and evaluation. PDF extraction can fail when papers contain scanned pages, complex tables, equations, or multi-column layouts. Retrieval latency may increase as the corpus grows, requiring approximate nearest-neighbor search, caching, and batch indexing. Privacy controls are necessary when users upload unpublished papers or institutional documents. The system should also expose uncertainty when evidence is weak, rather than forcing an answer.

Scalable deployment would benefit from user authentication, persistent workspaces, background indexing jobs, API-based access to scholarly databases, and systematic evaluation using retrieval metrics such as recall and answer metrics such as citation precision, faithfulness, and user-rated usefulness.

8. Conclusion and Future Work

This paper presented Deep Research Agent, a citation-aware AI system for automated analysis and answer generation of research papers. The revised contribution is clearly positioned against recent RAG and citation-aware systems: the proposed agent combines processing of uploaded papers, hybrid dense and sparse retrieval, citation mapping, and graph-based exploration within a lightweight academic workflow. The system can help students, researchers, and technical teams quickly inspect papers and generate evidence-linked answers while preserving a path for human verification. Future work will focus on larger-scale corpus evaluation, stronger citation verification, multilingual paper support, integration with academic databases, improved table and equation extraction, collaborative workspaces, and cloud deployment. Additional experiments should compare retrieval strategies and measure citation faithfulness across diverse research domains.

References

1. Lewis P., Perez E., Piktus A., Petroni F., Karpukhin V., Goyal N., et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems. 2020.
2. Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of EMNLP-IJCNLP. 2019.
3. Robertson S., Zaragoza H. The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends in Information Retrieval. 2009.

4. Luan Y., Eisenstein J., Toutanova K., Collins M. Sparse, Dense, and Attentional Representations for Text Retrieval. Transactions of the Association for Computational Linguistics. 2021.
5. Gao Y., Xiong Y., Gao X., Jia K., Pan J., Bi Y., et al. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997. 2023.
6. Fan W., Ding Y., Ning L., Wang S., Li H., Yin D., Chua T.-S., Li Q. A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models. arXiv:2405.06211. 2024.
7. Huang Y., Huang J. A Survey on Retrieval-Augmented Text Generation for Large Language Models. arXiv:2404.10981. 2024.
8. Zhao P., Zhang H., Yu Q., Wang Z., Geng Y., Fu F., et al. Retrieval-Augmented Generation for AI-Generated Content: A Survey. arXiv:2402.19473. 2024.
9. Li W., Li J., Ma W., Liu Y. Citation-Enhanced Generation for LLM-based Chatbots. arXiv:2402.16063. 2024.
10. Qian H., Fan Y., Zhang R., Guo J. On the Capacity of Citation Generation by Large Language Models. arXiv:2410.11217. 2024.
11. Wang Y., Ma X., Nie P., Zeng H., Lyu Z., Zhang Y., et al. ScholarCopilot: Training Large Language Models for Academic Writing with Accurate Citations. arXiv:2504.00824. 2025.
12. Jiang P., Ouyang S., Jiao Y., Zhong M., Tian R., Han J. A Survey on Retrieval and Structuring Augmented Generation with Large Language Models. arXiv:2509.10697. 2025.
13. Fingleton Z., Siavash N., Moin A. Enhancing Large Language Models with Retrieval Augmented Generation for Software Testing and Inspection Automation. arXiv:2604.15270. 2026.