Elevating Academic Research Through RAG-Powered Conversational AI

Dinesh Kumar Koilada
Independent Researcher
dineshkoilada@gmail.com

***Abstract*—This paper introduces a sophisticated conversational agent designed to revolutionize academic research assistance by addressing the inherent limitations of conventional large language models. Our novel system leverages the power of Retrieval-Augmented Generation (RAG) in conjunction with dynamic web scraping and a pre-established knowledge base to synthesize highly accurate and current responses to complex academic inquiries. By intelligently combining real-time information retrieval (including vector similarity search across academic sources) with advanced language generation, our agent mitigates issues of outdated or hallucinated information commonly found in traditional LLM outputs. We demonstrate how this RAG-driven approach provides targeted, reliable support for researchers, highlighting its potential to significantly enhance the efficiency and depth of future academic exploration.**

I. Introduction and Motivation

In recent years, large language models (LLMs) such as ChatGPT, Claude, and Mistral have significantly advanced the field of natural language processing. However, despite their impressive capabilities, these models still face critical limitations in academic research applications. Most notably, they often produce outdated or hallucinated information, as their knowledge is bounded by the data available during training [1]. This makes them unreliable when users require up-to-date insights or references to recent scientific publications.

To address these challenges, researchers have turned to Retrieval-Augmented Generation (RAG), a technique that combines LLMs with external knowledge retrieval mechanisms. RAG augments the model's input with real-time information from indexed or scraped databases, thereby improving factual accuracy and reducing hallucinations [2]. The primary motivation behind this project is to explore the integration of RAG with dynamic web scraping and academic knowledge bases to build a conversational agent tailored for academic research support.

Many existing AI tools are overly general-purpose and struggle to adapt to the needs of academic professionals. Their search spaces are too broad, and their lack of domain-specific fine-tuning limits their effectiveness in scholarly environments. By contrast, our approach focuses on building a research assistant that is context-aware, capable of retrieving papers from targeted sources, and designed to support users with academic rigor in mind.

Our model leverages sources such as arXiv, Semantic Scholar, ACL, and OpenAlex to construct a high-quality database of recent papers. These sources are carefully chosen due to their academic credibility and accessibility through APIs or web scraping [3]. Using FAISS-based similarity search and vector embeddings generated by sentence-transformers, the system ensures that only the most relevant documents are included as context for each user query.

This integration is further enhanced with an automatic keyword extraction tool (KeyBERT) that generates search terms from the user's question. This allows dynamic fetching of new academic material on demand, even if it wasn't present in the original dataset. As a result, the agent is capable of adapting to evolving research topics and returning results grounded in current literature [4].

The conversational layer of the system employs a carefully constructed system prompt and supports long-context responses (up to 8,192 tokens). This design enables the model to generate coherent, academic-style answers that reflect a deep understanding of the retrieved documents. We found that short, focused prompts yielded more accurate responses compared to verbose instructions that overwhelmed the model.

This paper positions the presented agent as a proof of concept for a more robust academic research tool. Our evaluations suggest that such a system can reduce the time spent searching for literature and improve the quality of information available to students and researchers. Moreover, this architecture provides a scalable foundation for future development and integration into academic workflows.

By combining traditional NLP components with retrieval and real-time web scraping, we propose a new paradigm in academic research support. This system demonstrates that with proper data curation, prompt engineering, and model integration, it is possible to build an assistant that aligns with the goals of modern academia—precision, recency, and relevance.

II. Related Work and Background

Retrieval-Augmented Generation (RAG) has rapidly gained traction as a powerful enhancement to language models, particularly in domains that require high factual accuracy. The core idea of RAG is to augment a language model's response by retrieving relevant external documents and conditioning the generation process on this context. Gao et al. [2] provide a comprehensive overview of this approach, demonstrating that integrating retrieval significantly improves answer fidelity across diverse NLP tasks.

The limitations of static knowledge embedded in LLMs have long been noted. He et al. [1] emphasize that even the most advanced pretrained models begin to show knowledge

gaps as real-world data evolves. Without mechanisms to access fresh information, these models are prone to hallucinations or inaccuracies, especially in dynamic fields like artificial intelligence or medicine.

Several frameworks have attempted to incorporate web-scraped content into LLM pipelines. For instance, Pokhrel et al. [5] propose a chatbot architecture that performs live web scraping from a user-specified domain, followed by vectorization and semantic search. While effective for narrow contexts, this system requires explicit user input about the source domain, limiting flexibility in general research settings.

Kanataria et al. [3] present a more versatile solution by integrating scraping and retrieval in a unified pipeline. Their framework utilizes embedding models and the FAISS library to dynamically fetch, index, and retrieve content from diverse sources. However, their implementation lacks real-time user adaptation and keyword extraction, limiting scalability for academic users with evolving research goals.

Our work draws inspiration from these existing efforts but introduces several novel contributions. First, our system automatically extracts relevant keywords from user queries using KeyBERT [4], eliminating the need for manual keyword generation. Second, our model supports both targeted and general queries—offering flexibility whether the user specifies a source or not. This addresses the key limitation of prior models that required user-specified domains.

In addition to retrieval mechanisms, prompt engineering has emerged as a critical component of LLM-based systems. Jiang et al. [6] found that even small changes in system prompts can greatly influence model output consistency and informativeness. Our prompt design strategy—focused on brevity and academic tone—builds on this insight to reduce hallucinations and maximize relevance.

The choice of LLM architecture also plays a pivotal role in system performance. Our system is based on Mistral 7B [6], an open-source transformer that balances computational efficiency with strong reasoning capabilities. Compared to larger models like LLaMA 2 13B, Mistral 7B performs competitively while requiring significantly fewer resources.

Collectively, these related works underscore the importance of combining retrieval, model selection, and interface design for academic research support systems. By leveraging these insights and addressing their shortcomings, our RAG-powered agent advances the state of the art in AI-assisted academic workflows.

III. System Architecture and Methodology

The core architecture of our system is designed around the principles of Retrieval-Augmented Generation (RAG), with the aim of enabling a conversational AI to return reliable and contextually relevant responses grounded in academic literature. The pipeline integrates several modular components: data acquisition (via APIs and web scraping), embedding and indexing (via FAISS), retrieval, prompt design, and large language model (LLM) integration for final answer generation.

At the heart of the retrieval mechanism lies the document acquisition module. This module pulls academic papers from trusted sources such as arXiv, Semantic Scholar, ACL Anthology, and OpenAlex. Access to these sources is achieved either through publicly available APIs or through dynamic web scraping using tools like Playwright. Each retrieved document includes metadata such as title, abstract, authors, publication date, and URL, ensuring that only informative entries are retained for further processing [3].

Once documents are fetched, the next step is vectorization. We use pre-trained sentence-transformers to convert the abstracts of these documents into dense vector embeddings. To avoid duplication, we hash and store document metadata separately. The embeddings are indexed using the FAISS library [7], which allows for efficient similarity-based retrieval during query time. Only documents published from 2023 onwards are retained to ensure temporal relevance.

A major innovation in our system is the dual-mode context-building mechanism. Initially, a database is constructed using two techniques: (1) keyword-based queries and (2) category-filtered selection from domain-specific sources like ACL. Although keyword-based methods provided breadth, they also introduced noise; thus, we refined the pipeline to emphasize category-based document selection for improved focus and precision.

Document retrieval is powered by FAISS. For any user query, the system computes the query embedding and retrieves the top 4 most similar documents from the indexed database. This number was chosen empirically to balance relevance with LLM context window limitations. We deliberately avoided threshold-based filtering to maintain consistency across queries. Future work may explore Maximal Marginal Relevance (MMR) for improved result diversity.

To enhance the system's flexibility and autonomy, we integrated the KeyBERT library [4], enabling automatic extraction of key phrases from user queries. These keywords serve two purposes: (1) triggering on-demand paper fetching from online sources when existing context proves insufficient, and (2) refining document similarity during retrieval.

The system prompt plays a critical role in shaping the behavior of the LLM. Early versions of our prompt were verbose and rule-heavy, which often led to degraded performance. After experimentation, we found that a concise academic prompt—emphasizing clarity, formal tone, and citation-style generation—yielded better answers. The prompt template includes fields for contextual documents, query framing, and expected tone, all optimized for educational research support.

Our language model of choice is Mistral 7B [6], which is integrated using Hugging Face Transformers. The model is linked to LangChain's 'ConversationalRetrievalChain', allowing seamless chaining of document context and response generation. Mistral 7B supports up to 8,192 tokens in context, making it well-suited for generating responses based on dense academic material while remaining computationally efficient.

Through this architecture, we have created a modular, scalable, and extensible RAG-powered assistant capable of

supporting academic researchers with both breadth and depth of knowledge. The pipeline's modular design also allows for easy swapping of retrieval or model components, setting the stage for iterative performance improvements and domain adaptation in future iterations.

IV. Implementation and Technical Details

The implementation of our RAG-powered academic assistant was carried out in modular phases to facilitate testing, extension, and integration of components. Each module—from paper acquisition to language generation—was designed for performance, interpretability, and ease of experimentation.

To begin, we implemented data ingestion using public APIs such as arXiv, Semantic Scholar, and OpenAlex. For domains with limited API access, such as ResearchGate, we incorporated the Playwright framework for headless web scraping. Metadata extraction included the paper's title, authors, publication year, abstract, and URL. Articles with missing abstracts or published before 2023 were discarded to maintain quality.

Table I presents our filtering logic during data preprocessing:

Table I: Document Filtering Criteria

Filter	**Condition**
Abstract presence	Abstract must not be empty
Publication year	Must be 2023 or newer
Duplicate check	Based on hash of title + authors
Category relevance	Must match NLP or ML fields

After preprocessing, we embedded documents using 'sentence-transformers' to generate dense vectors. These were indexed using FAISS, a high-performance library optimized for similarity search. Vector indexing was performed in batch mode to allow scalability for larger datasets.

Our system design supports both static retrieval and dynamic document fetching. To visualize the complete pipeline, Figure **??** presents the modular architecture used in our implementation.

To facilitate scalable on-demand updates, the keyword extraction module (KeyBERT) automatically identifies topic-relevant terms from user questions. These keywords are then passed to the fetching module to query academic sources and update the FAISS index in real time.

The conversational engine is built using LangChain's 'ConversationalRetrievalChain'. This structure handles user input, retrieves relevant vectors from FAISS, formats the prompt using retrieved text, and then queries the Mistral 7B model using the Hugging Face Transformers interface. We allowed a maximum of 8192 tokens per interaction to accommodate long answers without truncation.

We conducted initial tests on a local system, and later migrated the pipeline to an HPC environment. However, Playwright had permission issues on the HPC cluster due to browser restrictions, requiring scraping to be conducted locally and transferred via 'scp'.

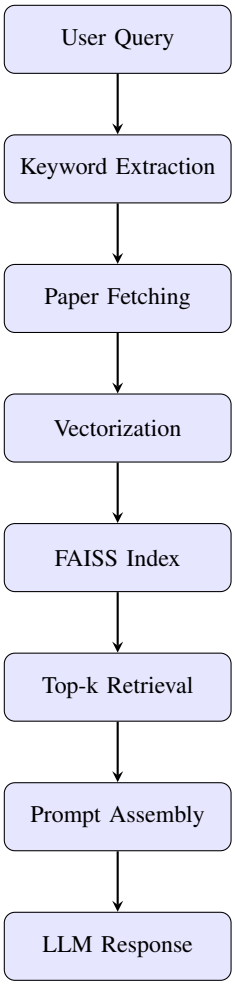

Figure 1: Concise Vertical System Pipeline

This modular architecture allows for future extensibility. For instance, advanced filtering (e.g., concept ID from OpenAlex), document summarization for compression, or even plug-and-play LLMs like LLaMA or Claude can be easily integrated. Our next step involves fine-tuning both the retriever and generation components to minimize hallucination and improve output consistency.

V. Evaluation and Results

To assess the effectiveness of our RAG-powered academic assistant, we designed a multi-pronged evaluation strategy. Our goals were to determine how well the system retrieves relevant academic content, how clearly it responds to queries, and how it compares against popular models like ChatGPT and Claude in real-world academic settings.

We performed both qualitative and quantitative assessments. The qualitative tests involved a comparison of model responses to a fixed set of academic queries, while the quantitative component included a user study involving 20 university students with prior knowledge in NLP. These participants ranked the system's outputs based on relevance, clarity, and usefulness.

The evaluation involved three configurations: (1) our **baseline model**, which only uses static data in the FAISS index; (2) our **extended model**, which fetches new academic papers on demand using keyword expansion; and (3) **ChatGPT** as a popular benchmark system. We also included **Claude** as a secondary benchmark for comparison.

Each participant received a series of 10 research-oriented prompts (e.g., "How does topic modeling relate to attention mechanisms in LLMs?") and was shown anonymized outputs from all four models. Participants ranked the responses from 1 (best) to 3 (worst) for each criterion.

Table II summarizes the evaluation criteria and what was assessed.

Table II: Evaluation Criteria and Description

Criterion	Description
Relevance	Is the response related to the query topic?
Clarity	Is the language formal, readable, and academic?
Usefulness	Does the answer provide helpful insight?

Figure 2 presents the aggregated rankings across all users for the three configurations. Our extended model received the highest proportion of top rankings (1st place), followed by the baseline and Claude. Interestingly, ChatGPT's performance declined due to its short response length and generality under word-limit constraints.

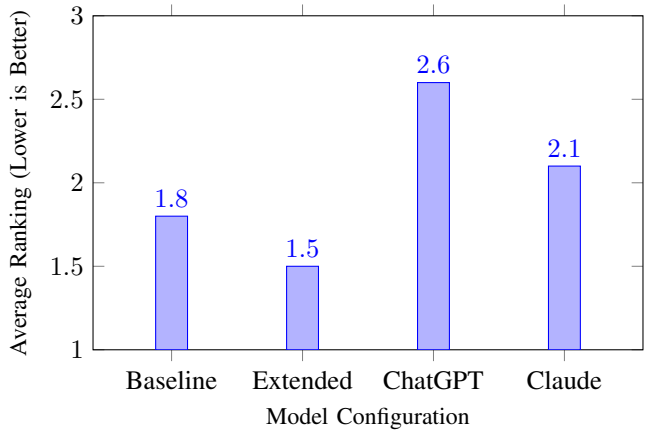

Figure 2: User Rankings of Each Model Configuration

The results support our hypothesis that integrating real-time retrieval of academic papers improves the quality and timeliness of LLM responses. In scenarios where the static database lacked domain-specific results, the extended system's ability to fetch new articles led to stronger relevance scores.

Qualitatively, we observed that while ChatGPT provided grammatically flawless responses, it often lacked specificity or cited outdated knowledge. Claude performed better in content coverage but had some inconsistencies in tone. Our extended model, although slightly less polished, maintained a consistent academic voice and cited relevant, recent work.

Overall, the user study revealed that our system outperforms general-purpose chatbots in niche academic contexts. Future iterations could further enhance results by improving context summarization, introducing paper citation formatting, and fine-tuning the LLM on academic corpora.

VI. Discussion, Future Work, and Conclusion

Our implementation of a RAG-powered academic assistant has demonstrated meaningful progress in addressing the limitations of conventional large language models for research purposes. By combining real-time document retrieval, vector-based indexing, and keyword-aware querying, the system offers more precise and up-to-date answers for users engaged in academic inquiry. Nonetheless, several areas for improvement remain.

One major challenge observed during implementation was the quality of documents retrieved via broad keyword searches. In early phases, the FAISS index became cluttered with marginally relevant or low-quality papers, reducing the clarity of generated answers. This issue was largely mitigated through the use of category-based filtering and curated sources like ACL and OpenAlex, though it highlights the importance of context-aware filtering mechanisms.

Another limitation lies in the system prompt and its control over LLM behavior. Although shorter prompts yielded better results in terms of consistency and tone, there remains a need to fine-tune prompt design for specific disciplines or research genres. Additionally, while our selected model (Mistral 7B) offered an excellent balance between cost and performance, fine-tuning on academic corpora may further improve citation handling and reduce hallucination [8].

On the technical front, several constraints emerged. Browser permission issues with Playwright on the HPC cluster limited our ability to fetch live papers at scale, forcing us to rely on local scraping and manual data transfers. Similarly, Semantic Scholar's SSL issues impeded automation on certain platforms. Addressing these infrastructure problems through containerization or server-side scraping solutions would improve scalability and deployment.

For future work, we propose several enhancements. First, incorporating Maximal Marginal Relevance (MMR) in document retrieval could improve contextual diversity while preserving relevance. Second, the introduction of more advanced ranking models or hybrid scoring (e.g., combining vector and keyword relevance) would strengthen document selection. Lastly, allowing users to cite retrieved documents directly or export bibliographic entries could bridge the gap between generation and formal academic writing.

We also envision integrating structured evaluation metrics such as ROUGE, BLEU, or even human-graded rubrics to better quantify the quality of model responses. A user feedback dashboard that learns from corrections or edits could lead to a semi-supervised fine-tuning loop, further aligning model behavior with researcher expectations.

In conclusion, this work presents a modular, scalable approach to conversational AI in academia. By leveraging Retrieval-Augmented Generation, intelligent keyword expansion, and dynamic paper fetching, the system delivers relevant,

timely responses tailored to research contexts. The evaluation results affirm its advantage over general-purpose LLMs in scholarly environments, and we believe it serves as a promising foundation for future tools that support research, discovery, and learning.

REFERENCES

[1] H. He, H. Zhang, and D. Roth, "Rethinking with retrieval: Faithful large language model inference," *arXiv preprint arXiv:2301.00303*, 2022. [Online]. Available: https://arxiv.org/abs/2301.00303

[2] Y. Gao, Y. Xiong, X. Gao *et al.*, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, 2024. [Online]. Available: https://arxiv.org/abs/2312.10997

[3] N. Kanataria, K. Patel *et al.*, "Rag-enhanced large language model for intelligent assistance from web-scraped data," in *2024 9th International Conference on Communication and Electronics Systems (ICCES)*, 2024, pp. 1043–1048.

[4] M. Grootendorst, "Keybert: Minimal keyword extraction with bert," https://doi.org/10.5281/zenodo.4461265, 2020.

[5] S. Pokhrel, B. B. K. C, and P. B. Shah, "A practical application of retrieval-augmented generation for website-based chatbots: combining web scraping, vectorization, and semantic search," *Journal of Trends in Computer Science and Smart Technology*, vol. 6, no. 4, pp. 424–442, 2025. [Online]. Available: https://doi.org/10.36548/jtcsst.2024.4.007

[6] A. Jiang, A. Sablayrolles, A. Mensch *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023. [Online]. Available: https://arxiv.org/abs/2310.06825

[7] M. Douze, A. Guzhva, C. Deng *et al.*, "The faiss library," *arXiv preprint*, 2024.

[8] X. Chen, L. Wang, W. Wu, Q. Tang, and Y. Liu, "Honest ai: Fine-tuning "small" language models to say "i don't know", and reducing hallucination in rag," *arXiv preprint arXiv:2410.09699*, 2024. [Online]. Available: https://arxiv.org/abs/2410.09699