AI Research Assistant

Mihai Gheorghe[1,*,†], Cătălina Chinie[1,†] and Dumitru Roman[1,2,†]

[1] *Bucharest University of Economic Studies, Piața Romană 6, Bucharest, Romania*
[2] *SINTEF AS, Oslo, Norway*

Abstract

The increasing volume of scientific literature and the abundance of publicly accessible data present a substantial hurdle for researchers aiming to stay informed and effectively derive valuable insights. In this paper we discuss the use of LLMs in the context of extracting information from scientific literature and introduce an AI-driven Research Assistant that uses custom Retrieval Augmented Generation (RAG) as a Service and other techniques to streamline processes such as literature review, information extraction, and knowledge discovery.

Keywords

Artificial Intelligence, Retrieval Augmented Generation, Automated literature review, Information extraction

1. Introduction

Large Language Models (LLMs) have gained significant attention across diverse industries, with their remarkable reasoning abilities enabling time savings and idea generation across numerous domains. Scientific research stands to benefit greatly, however, the probabilistic nature of LLM inference can lead to inaccurate responses for specialized queries. To address this, the novel paradigm of Retrieval Augmented Generation (RAG) has emerged over the last years. RAG enhances LLM query context by retrieving factual information from external sources such as documents, databases, and APIs, thus mitigating reliance solely on LLM training data [1].

Furthermore, the RAG paradigm not only improves the accuracy of LLM outputs but also expands their capabilities by enabling access to real-time information and specialized knowledge bases. This dynamic integration of external information allows LLMs to evolve beyond their static training data and stay abreast of the latest developments in rapidly changing fields such as scientific research.

While Retrieval Augmented Generation (RAG) offers significant improvements in LLM performance, it is not without limitations. The common RAG approach of retrieving top-k semantically similar passages based on vector embeddings and metrics like cosine similarity can encounter challenges. For instance, queries demanding multi-hop reasoning or complex relationships can fall short. Consider the query: "Which is the GDP of the country where the highest mountain peak in the world is?" While a RAG system trained on geography data may accurately answer "Which is the highest mountain peak in the world?", the second query necessitates identifying the country associated with the mountain and then retrieving its GDP, a task potentially beyond simple semantic similarity matching.

RuleML+RR'24: Companion Proceedings of the 8th International Joint Conference on Rules and Reasoning, September 16--22, 2024, Bucharest, Romania
* Corresponding author.
† These authors contributed equally.
mihai.gheorghe@csie.ase.ro (M. Gheorghe); catalina.chinie@fabiz.ase.ro (C. Chinie); dumitru.roman@sintef.no (D. Roman)
0009-0001-9976-7660 (M. Gheorghe); 0000-0002-4281-3312 (C. Chinie)

A variety of techniques are being explored to overcome these limitations [2]. These include:

- **Reranking:** Employing a two-step retrieval process, first retrieving a larger set of candidates using efficient methods like cosine similarity, then reranking them using more sophisticated techniques like cross-encoders or even LLMs themselves, to better capture relevance and address multi-hop reasoning.
- **Hybrid RAG:** Combining semantic similarity-based retrieval with knowledge graphs to incorporate explicit relationships and facilitate more complex reasoning [3].
- **Large Context LLMs** such as Claude with its 200k token context or Gemini with its 1M+ token context, can significantly enhance RAG systems by incorporating more extensive context directly into the model. This reduces the need for explicit retrieval of external information in many cases, allowing the LLM to draw upon a broader knowledge base to understand and reason over larger chunks of information [4].
- **Hierarchical Embeddings:** Leveraging embeddings of the original text alongside various levels of summaries, such as in the RAPTOR model, to enhance retrieval accuracy and efficiency [5].
- **Multi-Hop Query Answering:** Decomposing complex queries into simpler sub-queries and employing techniques like Chain-of-Thought prompting to guide LLMs through multi-step reasoning [6, 7, 8].
- **Autonomous Agents:** Utilizing AI agents to navigate diverse information sources and construct more intricate prompts for the LLM, incorporating logic and multi-step reasoning [9].
- **Self-RAG:** Where the LLM itself participates in the retrieval process, potentially leading to more adaptive and context-aware retrieval [10].

These advancements highlight the active research and development in the RAG paradigm, aiming to address its limitations and enable LLMs to tackle increasingly complex and nuanced information needs.

Despite these advancements, constructing a RAG system specifically for scientific research remains a difficult task. Challenges arise in handling diverse content types commonly found in research papers, such as tables, images, and formulas, which often necessitate conversion into natural language text for effective retrieval. Moreover, the research process often involves snowballing, where the corpus of relevant references expands iteratively from an initial set of studies.

A review of existing AI assistants and RAG systems tailored towards researchers reveals a lack of a universally applicable solution, although several platforms demonstrated advanced techniques and some even appeared customized for scientific research. Existing solutions in this area can be categorized into three distinct classes:

- **AI Document Assistants:** Constituting the most prevalent category, these solutions range from freely available to premium licensed offerings. While they leverage cutting-edge LLMs and enable users to upload documents in various formats, answer questions based on those documents, and perform summarization, they often exhibit limitations specifically concerning academic research support. These constraints include operating on a limited number of documents (typically restricted by the LLM's context window), lacking specialization in scientific papers, and generally not performing multi-hop reasoning across multiple documents.
- **General purpose RAG as a Service:** Augmenting document assistants with retrieval capabilities enables access to a substantially larger corpus of documents. Solutions within this category encompass both commercial and freely available options. The following are noteworthy examples:

    - **RAGflow:** An open-source RAG engine with a comprehensive feature set and a user-friendly, modern interface [11]. It offers partial customization in terms of embedding models and LLMs, alongside fine-grained control over chunking strategies. It also provides tools for constructing custom AI agents. However, as a general-purpose product, it relies on paragraph-length chunking rather than semantic or logical separation. It lacks automated corpus construction and multi-modal capabilities. Despite these limitations, the product shows promise due to its ongoing development.
    - **Vectara:** A company specializing in RAG solutions [12], with a strong reputation in the field. However, they do not offer an out-of-the box readily available Academic RAG as a Service product, offering on demand, customized software.
    - **Other solutions:** Several other solutions exist, such as Humata AI [13], Digilist [14], Weaviate Verba [15], Anything LLM [16], and RAGify [17]. However, these options lack certain crucial features, including multi-hop capabilities, custom structured information extraction, re-ranking, traceability, and agentic behavior.
- **Academic Research Specialized Assistants:** This category comprises solutions specifically designed for academic research. Notable examples include:
    - **Sakana AI Scientist:** Primarily focused on autonomous generation of complete academic papers [18], its Q&A capabilities are not the central feature. Although the demonstrated results are impressive, concerns persist regarding benchmarks for factuality and ethical considerations. Similarly, Insilico's Dora [19] also generates full papers without chat or Q&A functionalities, and appears to have fewer features compared to Sakana. Unriddle.ai [20], another solution in this category, generates full papers and even offers LaTeX rendering, but lacks traceability, multi-hop capabilities, LLM or embedding model agnosticism, and structured information extraction. Notably, these solutions do not operate as traditional RAG as a Service platforms.
    - **Clarivate AI Academia:** Recently announced by Clarivate [21], detailed information about its features and performance remains limited. However, given Clarivate's established track record, the product has strong potential to become a noteworthy contender in this space.

To overcome the above mentioned limitations we initiated the development of a RAG as a Service Research Assistant whose features we briefly introduce in the following section.

2. AI Research Assistant

2.1. Dynamic corpus construction

Researchers can upload their own PDF papers or initiate automated downloads for specific queries through the implementation of ArXiv and Semantic Scholar APIs, thus having access to millions of papers. GROBID processes papers, resulting in structured XML representations with clearly defined sections, figures, tables, and references [22]. The automated download function can also recursively expand the corpus by extracting references from the initial document set.

2.2. Retrieval and question-answering

Full plain text sections from papers are indexed in ChromaDB [23] using cutting-edge BAAI/bge-m3 [24] dense embeddings. Deviating from most Q&A RAG systems, we employ large paragraph chunks (often entire chapters/sections) to maintain context. Oversized paragraphs are divided into subsections while preserving sentence integrity. Question answering employs cosine

similarity retrieval, with results re-ranked using BAAI/bge-reranker-v2-m3 [25]. In cases where no relevant documents are retrieved, the system transparently informs the researcher that the answer is not grounded in the corpus. Each answer is accompanied by source paper sections, promoting transparency and facilitating further exploration.

2.3. Custom information extraction

Our AI Research Assistant facilitates the extraction of custom-structured information from scientific papers by generating valid JSON schemas from natural language queries. The standard schema extracts a comprehensive set of data including definitions, indicators, hypotheses, key findings, topics, and summaries from each paper. These summaries can be indexed within the vector database as well, enabling the system to also respond to high-level conceptual queries, in contrast to specific questions grounded in isolated paper sections which is addressed by the previously mentioned Q&A RAG features. Cost analysis indicates that leveraging a Model as a Service (for instance, Claude Haiku [26]) for information extraction incurs an estimated cost of 1 USD per 250 papers.

2.4. Architecture and scalability

The system adopts a decoupled multi-server architecture for scalability, with LangChain [27] partially managing orchestration. A GPU-intensive machine is required for vector embedding, re-ranking, and local inference when utilizing local LLMs. The workflow exposes API endpoints that can be consumed by a web application to manage user access.

2.5. AI Agents

The project incorporates agentic behavior, utilizing a ZERO-SHOT Classifier to direct user queries to the most suitable tool. These tools include classic RAG/Q&A based on articles, structured information extraction, and deterministic queries to SQL-like datasets (e.g., locally hosted EUROSTAT data). LangChain's agentic implementation allows for chaining multiple tools in a single user query. Therefore, an answer to a user query can be grounded in both the scientific corpus and a relevant dataset.

3. Relevance to Rule-Based AI, Decisions, and Reasoning

Our approach achieves deterministic and explainable results through:

- Deterministic LLM use (temperature set to 0)
- Grounding question-answering in the scientific literature corpus and indicating the paper sources along the answer
- Enforcing JSON schemas for structured information extraction
- Verbose mode for agentic workflow, enhancing explainability
- Integration of conventional programming tools for querying structured data sources

This research assistant offers a significant contribution to AI-powered literature analysis, providing researchers with a valuable tool for navigating the expansive landscape of scientific knowledge.

4. Further directions

We aim to integrate image and figure extraction from papers, leveraging multi-modal LLMs to enrich the dataset. Open models such as PaliGemma [28] or Idefics [29] can do image-to-text inference locally on reasonably accessible hardware.

Related to multi-hop reasoning, we plan to employ techniques like query decomposition and graph neural networks to address complex, multi-step queries.

Additionally, we plan to develop more agentic tools capable of handling various datasets, expanding the system's capabilities and adaptability.

In the mid to long term, we intend to enhance our system by integrating knowledge graph retrieval with the existing semantic similarity-based approach. This knowledge graph will be constructed during the paper processing and parsing stage, extracting relevant entities like topics, affiliations, and named entities as nodes. This hybrid approach aims to facilitate more nuanced and complex query handling, enabling the AI Research Assistant to better understand and leverage the intricate relationships within scientific literature.

Acknowledgement

This research is financed under the Romanian National Recovery and Resilience Plan, by the Romanian Government, under the contract number 268/29.11.2022, Entitled “CAUSEFINDER - CAUSALITY IN THE ERA OF BIG DATA”.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, et al., Retrieval-augmented generation for knowledge-intensive NLP tasks, in: NIPS '20: Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020, pp. 9459-9474. doi: 10.48550/arXiv.2005.11401.

[2] X. Wang, Z. Wang, X. Gao, F. Zhang, Y. Wu, Z. Xu, et al. Searching for Best Practices in Retrieval-Augmented Generation (2024). doi: 10.48550/arXiv.2407.01219.

[3] B. Sarmah, B. Hall, R. Rao, S. Patel, S. Pasquali, and D. Mehta, HybridRAG: Integrating Knowledge Graphs and Vector Retrieval Augmented Generation for Efficient Information Extraction (2024). doi: 10.48550/arXiv.2408.04948.

[4] Z. Li, C. Li, M. Zhang, Q. Mei and M. Bendersky, Retrieval Augmented Generation or Long-Context LLMs? A Comprehensive Study and Hybrid Approach (2024). doi: 10.48550/arXiv.2407.16833.

[5] P. Sarthi, S. Abdullah, A. Tuli, S. Khanna, A. Goldie, and C. D. Manning, Raptor: Recursive abstractive processing for tree-organized retrieval (2024). doi: 10.48550/arXiv.2401.18059.

[6] Z. Jiang, M. Sun, L. Liang and Z. Zhang, Retrieve, Summarize, Plan: Advancing Multi-hop Question Answering with an Iterative Approach (2024). doi: 10.48550/arXiv.2407.13101.

[7] Y. Tang, Y. Yang, Multihop-RAG: Benchmarking Retrieval-Augmented Generation for Multi-Hop Queries (2024). doi: 10.48550/arXiv.2401.15391.

[8] W. Xiong, X. L. Li, S. Iyer, J. Du, P. Lewis, W. Y. Wang, et al., Answering Complex Open-Domain Questions with Multi-Hop Dense Retrieval (2020). doi:10.48550/arXiv.2009.12756.

[9] G. Gamage, N. Mills, D. De Silva, M. Manic, H. Moraliyage, A. Jennings, & D. Alahakoon, Multi-Agent RAG Chatbot Architecture for Decision Support in Net-Zero Emission Energy Systems, in 2024 IEEE International Conference on Industrial Technology (ICIT), Bristol, United Kingdom, 2024, pp. 1-6. doi: 10.1109/ICIT58233.2024.10540920.

[10] A. Asai, Z. Wu, Y. Wang, A. Sil and H. Hajishirzi, Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection (2023). doi: 10.48550/arXiv.2310.11511.

[11] https://ragflow.io

[12] https://vectara.com

[13] https://blog.invgate.com/humata-ai

[14] https://www.diligist.io

[15] https://github.com/weaviate/verba

[16] https://github.com/Mintplex-Labs/anything-llm
[17] https://github.com/kanad13/RAGify
[18] C. Lu, C. Lu, R. T. Lange, J. Foerster, J. Clune, and D. Ha, The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery (2024). doi: 10.48550/arXiv.2408.06292
[19] https://insilico.com/science42/dora
[20] https://www.unriddle.ai
[21] G. Ben-Porat, Introducing the Clarivate Academic AI Platform, 2024. URL: https://clarivate.com/blog/introducing-the-clarivate-academic-ai-platform
[22] GROBID [Computer software] (2008–2024). URL: https://github.com/kermitt2/grobid
[23] https://www.trychroma.com
[24] https://github.com/FlagOpen/FlagEmbedding/tree/master/FlagEmbedding/BGE_M3
[25] https://github.com/FlagOpen/FlagEmbedding/tree/master/FlagEmbedding/llm_reranker
[26] https://www.anthropic.com/news/claude-3-haiku
[27] https://www.langchain.com
[28] https://huggingface.co/google/paligemma-3b-pt-896
[29] https://huggingface.co/HuggingFaceM4/Idefics3-8B-Llama3