DESIDOC Journal of Library & Information Technology, Vol. 45, No. 2, March 2025, pp.109-115, DOI : 10.14429/djlit.20206

Designing Conversational Search for Libraries: Retrieval Augmented Generation through Open Source Large Language Models

Parthasarathi Mukhopadhyay
Department of Library and Information Science, Kalyani University, Kalyani - 741 235 West Bengal, India
Email: psm@klyuniv.ac.in

ABSTRACT

Large language models (LLMs) from the commercial domain like BERT and GPT have made machine learning technologies accessible to everyone. On the other hand, the open-source LLMs like Llama, Mistral, and Orca are equally effective and are now widely available. Librarians and information professionals around the world are exploring how to use these models to improve library systems, particularly in the area of searching and finding information, and in building question-answer based search systems. This research study aims to use open-source large language models to develop a conversational search system that can answer questions in natural language on the basis of a given set of documents. The system is based on a Retrieval Augmented Generation (RAG) pipeline, which helps to overcome two major issues with large language models: providing false or imaginary information (hallucination) and giving outdated or unrelated answers. Through two case studies, this research demonstrates that using a RAG-based approach can effectively address these issues and provide more accurate and relevant results. The study proves that an open-source RAG framework can be used to incorporate large language models into library search systems. This integration allows users to receive direct answers to their questions, rather than just a list of potentially relevant documents. In the coming future, the conversational search system can be designed to work in Indian languages, allowing users to ask questions and receive answers in their preferred language.

Keywords: RAG (Retrieval Augmented Generation); LLM (Large Language Model); Generative AI; Conversational search; Library retrieval

1. INTRODUCTION

Large Language Models (LLMs) from commercial providers (Anthropic [Claude], Google [Gemini], and OpenAI[GPT]and so on..) – all are very costly for use programmatically for large-scale projects) as well as from open-source domains (Llama series, Orca, Mistral, etc.) have made significant advancements in generating human-like text but still face challenges such as outdated information and hallucinations (feature of LLMs to produce coherent and grammatically correct text but factually incorrect or ludicrous). To address these issues, two main approaches are employed: “fine-tuning” and “Retrieval-Augmented Generation (RAG).” Fine-tuning involves re-training LLMs to enhance their understanding of specific topics, but at a high cost in terms of resources and expertise[1]. On the other hand, RAG uses relevant content (retrieved through semantic matching against a query) to improve response accuracy without extensive training[2]. While RAG excels at quickly generating reliable answers based on provided dataset[3], fine-tuning is more suitable for specialised tasks and creative writing[4], despite potential transparency and accuracy measurement concerns[5]. In the given context, this research aims to investigate the potential of an open-source RAG pipeline in libraries for developing a conversational search system that delivers accurate and relevant answers to user queries, thereby addressing the limitations inherent in large language models. Historically, library professionals have been early adopters of technological innovations right from the 1970s. However, the widespread adoption of LLM technologies in libraries has been hindered by the tendency of these models to generate responses that are often hallucinated, outdated, or out of context. By examining the feasibility of an open-source RAG pipeline, this research seeks to bridge the gap between the eagerness to embrace new technologies and the practical challenges associated with deploying LLMs in a library setting.

Received : 11 May 2024, Revised : 28 September 2024
Accepted : 12 December 2024, Online published : 27 February 2025

2. RAG IN LIBRARIES

The introduction of LLMs like ChatGPT in November 2022 has sparked interest in generative AI within the library community. Phil Bradley[6] and Kent Fitch[7] foresee the possible integration of conversational AI models like ChatGPT in library search systems, marking a new era in information retrieval with both opportunities and challenges for professionals and users. Kent Fitch developed a prototype search interface in 2023 as a proof-of-concept for transforming library search systems using LLMs. The prototype focuses on improving keyword indexing and text retrieval and implementing summarisation and a “chat” interface to enhance the user experience, proving that RAG has the potential to revolutionise library services with the assistance of LLMs. It could enhance information retrieval by enabling conversational search and providing personalised recommendations based on user preferences[7]. RAG can improve library services by enabling conversational search, personalised recommendations, improved research assistance, interactive tutorials, accessibility tools, and curated content creation[8-10].

3. GENESIS OF RAG

In 2021, researchers from Facebook AI Research (FAIR) introduced Retrieval Augmented Generation (RAG) to enhance LLMs by merging information retrieval with text generation, improving factual accuracy and reliability[11]. RAG reduces LLMs' tendency to fabricate data, ensures coherence by integrating external knowledge, and adapts to text generation seamlessly without requiring retraining. A potential solution for real-world applications like question-answering and informative writing, the RAG pipeline consists of LLMs and external datasets within a system involving ingestion, retrieval, and generation stages[6,7,11]. Fig. 1 depicts a simple architecture of a RAG system. There are two primary RAG pipelines available in the open source domain: LlamaIndex and LangChain. They differ in their focus and data handling approaches, with LlamaIndex offering a more straightforward setup for RAG applications[11]. This study selected LlamaIndex as a RAG pipeline to develop a prototype conversational search system.

4. OBJECTIVES

The primary objective of this study is to develop and evaluate a conversational search system for a library setup using an open-source Large Language Model (LLM) in a Retrieval-Augmented Generation (RAG) pipeline. The specific objectives of this study are: 1) To select a suitable open-source LLM and a RAG framework for the development of the conversational search system; 2) To design and deploy the conversational search system using the selected LLM (Mistral 7.3B parameter model) and RAG framework (LlamaIndex-based PrivateGPT framework); 3) To collect, curate, and ingest two sets of documents into the RAG framework for the purpose of conducting two case studies; and 4) To test, debug, and evaluate the efficacy of the conversational search system in providing accurate and contextually relevant responses to user queries.

By achieving these objectives, this study aims to demonstrate the potential of using an open-source LLM in an open-source RAG pipeline for developing a conversational search system that can effectively respond to user queries in a library setup.

5. METHODS AND TOOLS

Kent Fitch's work[7] showcased the efficacy of conversational search systems in delivering precise, timely, and contextually relevant answer to users' queries. By employing GPT-4 as a LLM and ada-002 as an embedding tool, both sourced from OpenAI, in conjunction with text data from Trove and Wikipedia, the prototype illustrated the utility of knowledge graphs in enhancing response accuracy. This study underscores the importance of conversational search systems in libraries to support natural language based informative interactions. Furthermore, within the domain of AI and machine learning for libraries, researchers are not only exploring semantic annotation for document organisation[12,13] but also showing keen interest in developing conversational search systems as a promising retrieval mechanism[14,15].

5.1 Methodology

This research study aims to build a RAG-based conversational search system for a library setup using a five-part methodology:

5.1.1 System Configuration

- Domain: Identify the specific domain the system will focus on (here a conversational search system based on external document sets).
- Content: Gather and curate documents that will provide relevant information for retrieval (here newspaper items and journal articles for the prototype – all open access).
- LLM: Select an LLM that is well-suited for the desired conversational responses and compatible with a CPU-based local system.
- RAG Framework: Choose a framework based on

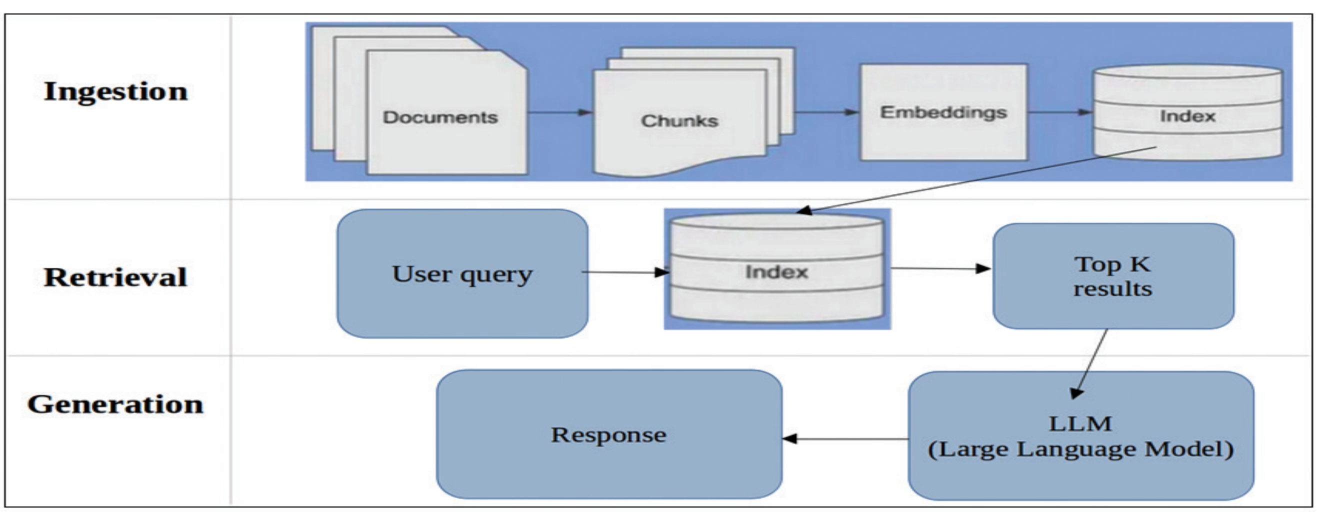

Figure 1. Basic RAG pipeline.

(Top K results based on semantic search is augmented to generate response from the deployed LLM)

LlamaIndex that facilitates communication between the retrieval system and the LLM.

5.1.2 Ingestion and Retrieval

- Indexing: After installing and configuring the RAG framework, the very fist job is to create a searchable index of the curated documents sets using techniques like sentence embedding or semantic indexing. Data indexing process involves two major activities:
- Generate Embeddings: This study uses a Hugging Face model (BAAI/bge-small-en-v1.5) to create vector embeddings for each document, transforming them into numerical representations (vectors) to capture semantic meaning.
- Load to Vector Database: After embedding, the generated vectors are uploaded to a vector database (Qdrant) for indexing. Qdrant stores these high-dimensional vectors to facilitate similarity-based searching.
- Similarity Search: User queries are converted into vectors, and the Qdrant index is searched for documents whose vectors are most similar to the query vector (top K results based on semantic similarities). This identifies the most relevant documents for the response generation step.

5.1.3 Generation

- Content Generation: Once the top K most relevant documents are retrieved, the system generates a natural language response by utilising the backend LLM. The LLM processes the retrieved document set and composes a coherent and contextually appropriate answer based on the information found in the documents.
- Content Delivery: The generated response is then delivered to the user through the RAG interface. Depending on the system's integration (e.g., chatbot, virtual assistant, or web-based interface), the response is displayed in a user-friendly manner, ensuring ease of access and readability. The interface is designed to allow users to interact with the system conversationally, receiving clear and informative answers to their queries.

5.1.4 Tuning and Evaluation

- Tuning: Adjustments are made to the retrieval system using sample queries to ensure accurate and contextually relevant responses. The pipeline includes a re-ranking algorithm (cross-encoder/ms-marco-MiniLM-L-2-v2) to refine semantic matching.
- Evaluation: The system is tested using real user queries, and its performance is assessed based on its ability to provide accurate and informative responses. The RAGAS framework (https://docs.ragas.io/en/stable/index.html) is used as a guideline for evaluation.

5.1.5 Deployment and Maintenance

- Deployment: Once the RAG pipeline performs satisfactorily, it can be integrated into a library platform (e.g., chatbot, virtual assistant, web interface, OPAC) to offer user services.
- Maintenance: Continuous monitoring, gathering of user feedback, and periodic updates to data sources are required. Retraining the system may also be necessary.

5.2 Tools

This study achieved the above stated methodology by using an array of open-source tools and open content (see Table 1).

Table 1.Tools for RAG pipeline

S.No.	Role in the system	Name of tool	Reason for selection
1.	Operating system	Ubuntu LTS 22.04	A stable Linux distro with huge user base
2.	Programming environment	Python 3.11.9 (along with Poetry V. 1.8.2)	Most of the RAG systems are Python based applications
3.	RAG pipeline	LlamaIndex	As it is more RAG specific indexing and retrieval tool in comparison to LangChain
4.	RAG framework	PrivateGPT	It is based on LlamaIndex and supports Ollama integration and API-based access alongside Web UI
5.	Vector store	Qdrant	The default vector database and vector similarity search engine in the deployed RAG pipeline
6.	Embedding tool	BAAI/bge-small-en-v1.5	Developed by Hugging Face and it is the default embedding model in the deployed system
7.	Reranker	ms-marco-MiniLM-L-2-v2	Developed by Hugging Face and it is the default reranker tool in the deployed system
8.	LLM local deployment	Ollama	A tool to run open source LLMs locally with support for an array of such LLMs
9.	LLM in use	mistral-7b-instruct-v0.2.Q4_K_M.gguf (open source LLM)	A large language model (LLM) with 7 billion parameters. It uses a space-efficient 4-bit quantisation technique, for deployment on CPU
10.	Web UI	Gradio	A tool to build intuitive Web UI for machine learning models with a few lines of code

6. CASE STUDIES

Once the RAG system is operational and the workflow is tested with a set of documents, this study is ready for extensive real-world experiments. Two experiments are conducted to assess the effectiveness of the RAG pipeline outlined in previous sections. The testing procedure includes: i) selecting a domain and gathering approximately 150+ objects; ii) ingestion of the objects into the RAG pipeline for vectorisation; iii) creating a suitable system prompt for guidance (prompt engineering); iv) formulating questions based on the provided content; v) evaluating the accuracy, correctness, and completeness of responses from the RAG system; and vi) comparing these responses with two large-scale LLM services: ChatGPT and Gemini.

6.1 Case Study I: Newspaper Articles on Chandrayaan-3

The first case study is based on a total of 157 newspaper items on the Chandrayaan-3 project of ISRO in view of the international impact of this scientific endeavor in India. The news items on the project were gathered from national and international newspapers of repute, along with press releases and other reports from external space agencies like NASA and ESA.

Figure 2 depicts the response from the RAG pipeline (named LibGPT), and Table 2 shows a brief comparison of the performances of LibGPT, ChatGPT, and Gemini against a set of queries.

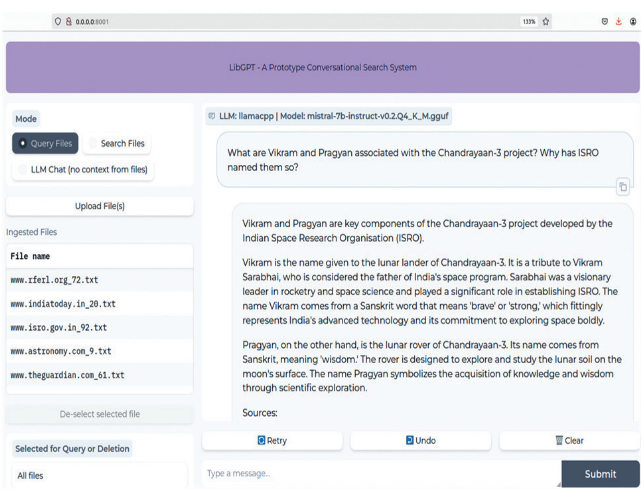

Figure 2. News items based RAG.

6.2 Case study II: Journal Articles on Cataloguing

The second case study is based on a set of journal articles on cataloguing (around 150 full-text papers) published in three open access journals, namely College & Research Libraries (ranging from 1940 to 2023), College & Research Libraries News (from 1988 to 2022), and Information Technology and Libraries (2018 to 2023). These articles deal with topics ranging from older research areas like cataloguing codes to emerging research areas like MARC framework, authority data integration, linked open data in cataloguing, and so on.

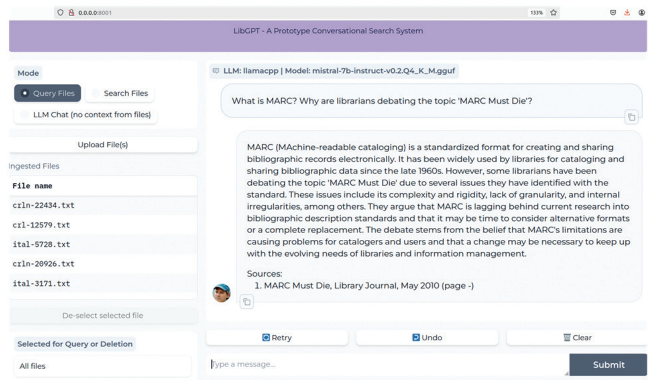

Figure 3. Journal articles based RAG.

Figure 3 depicts the response from LibGPT (this RAG pipeline) against a very specific but critical question, and Table 3 shows a comparison of performances of three systems against a set of sample questions (five are mentioned to save space).

7. FINDINGS

The RAG pipeline, tested through two case studies, demonstrates a strong capability to generate precise, human-like responses with embedded references. This feature provides a cost-effective solution for libraries looking to implement conversational search engines[16]. Unlike traditional systems that deliver entire documents, the RAG pipeline delivers direct answers based on the content, improving the user experience by addressing common issues such as hallucinations and non-contextual responses that are often seen in standalone LLMs. The prototype system, named LibGPT, outperformed commercial AI tools like ChatGPT and Gemini in both response quality and accuracy, as it focused on providing precise answers based on documents that are matched with a query through a semantic retrieval process. However, a notable limitation was the slower response time, primarily due to the use of a CPU-based system. This latency issue can be mitigated by transitioning to a GPU-based cloud server, which would facilitate parallel processing-crucial for speeding up machine learning tasks like vector similarity search and response generation[17]. As a whole, the findings suggest that with improved hardware, a RAG-based search system can be a highly effective tool for enhancing user services in libraries, offering both accuracy and reliability in information retrieval by accepting queries in natural language and in providing complete answers rather than merely retrieving a set of relevant documents.

8. CONCLUSION

The recent emergence of the RAG concept shows potential to transform library retrieval methods. The challenges encountered by professionals in library and information science (LIS) regarding retrieval are not unprecedented; similar issues arose with the advent of large-scale search engines such as Google, Yahoo, and Alvista in the late 1990s. In response, libraries developed Google-like-simple and OPAC-like-elegant search interfaces, called library discovery systems, in the early 2000s. It is plausible that the current metadata-based indexing systems, which provide links and concise document metadata in response to user queries, will be replaced by conversational search systems that furnish direct answers rather than document links[7,16]. This study serves as a proof-of-concept in that direction, with the anticipation of numerous enhancements to RAG-based conversational search systems underway. For instance, the introduction of re-rankers aims to refine semantic search results to ensure answer accuracy[18]. Research is expanding in two key areas of Retrieval-Augmented Generation (RAG) systems. First, there's a growing focus on developing multilingual RAG-based search systems. Second, researchers are working to establish a mathematical framework to assess various components of RAG systems, including embedding techniques, retrieval methods, and the performance of large language models.

Table 2. Performance comparison for the case study I

S. No.	Question	Responses				
		System	Faithful ?	Comprehensive?	References?	Time?
Q1	Identify the most important scientific findings of Chandrayaan-3.	LibGPT	Yes	Yes	Yes	82 seconds
		ChatGPT	No	No	No	Instant
		Gemini	Yes	Yes	No	5 seconds
Remark: LibGPT answered that the most significant finding related to the elemental composition of lunar soil and rocks in the south polar region of the Moon, whereas Gemini said two findings (missed 'the most' phrase): higher lunar surface temperature and the presence of sulfur on the lunar surface (missed the south polar region). ChatGPT answers all wrong (understandable as it was trained up to January 2022).						
Q2	What are Vikram and Pragyan associated with the Chandrayaan-3 project? Why has ISRO named them so?	System	Faithful ?	Comprehensive?	References?	Time?
		LibGPT	Yes, fully	Yes	Yes	79 seconds
		ChatGPT	No	No, wrong	No	Instant
		Gemini	Yes	Yes	No	7 seconds
Remark: The quality of the answer from LibGPT along with two relevant sources is obvious from Figure 2. ChatGPT said that it knows nothing of Chandrayaan-3 (as trained with information up to January 2022) but synthesised wrong information from the Chandrayaan-2 project. Gemini answered correctly and comprehensively, as it was a recent one, and got training on the event with lots of public documents.						
Q3	What is the name of the place where Chandrayaan-3 landed on the lunar surface?	System	Faithful ?	Comprehensive?	References?	Time?
		LibGPT	Yes, fully	Yes	Yes	63 seconds
		ChatGPT	No	No	No	Immediately
		Gemini	No	No	No	7 seconds
Remark: Gemini rightly provided the coordinates of the landing place but not the name. LibGPT, on the basis of the supplied news items, can tell that PM Narendra Modi named the place Shiva Shakti on August 26, 2023, and later it was approved by the IAU on March 19, 2024.						
Q4	Can you compare the Chandrayaan-3 project of India and the Luna-25 project of Russia?	Sysetm	Faithful ?	Comprehensive?	References?	Time?
		LibGPT	Yes	Yes	Yes	72 seconds
		ChatGPT	No	No	No	4 seconds
		Gemini	Yes, partially	No	No	3 seconds
Remark: LibGPT answered the differences completely and said in clear terms that Luna-25 of Russia was a failed mission, whereas Gemini compared both projects comprehensively but said nothing about the failure of the Luna-25 project. ChatGPT answered it all wrong, obviously because there was no training on it.						
Q5	Who is Veeramuthuvel? How was he related to Chandryaan-3?	Sysetm	Faithful ?	Comprehensive?	References?	Time?
		LibGPT	Yes	Yes	Yes	99 seconds
		ChatGPT	No	No	No	Readily
		Gemini	Yes	Yes, reasonably	No	3 seconds
Remark: Except for ChatGPT, both Gemini and LibGPT produced correct and comprehensive responses.						

Table 3. Performance comparison for the case study II

S. No.	Question	Responses				
		System	Faithful ?	Comprehensive?	References?	Time?
Q1	Who was Henriette Avram? Why was she called the mother of MARC?	LibGPT	Yes, fully	Yes	Yes	82 seconds
		ChatGPT	Partially	Yes	No	6 seconds
		Gemini	Partially	Yes	No	5 seconds
Remark: Both ChatGPT and Gemini referred to Ms. Avram as a computer programmer and system analyst; LibGPT referred to her as an American librarian and computer scientist. The comprehensiveness of answers from ChatGPT and Gemini is possibly based on a full-length Wikipedia article on Henriette Avram.						
Q2	What is MARC? Why are librarians debating the topic 'MARC Must Die'?	System	Faithful ?	Comprehensive?	References?	Time?
		LibGPT	Yes, fully	Reasonably	Yes	76 seconds
		ChatGPT	For first part	No, vague	No	5 seconds
		Gemini	For first part	No, unclear	No	Instant
Remark: The quality of the answer from LibGPT along with a relevant source is obvious from Figure 3. ChatGPT and Gemini answered correctly to the first part of the question, but both responded vaguely to the second part of the question, which is more critical in nature.						
Q3	What is LEAF (Linking and Exploring Authority Files)? How is it related to VIAF?	System	Faithful ?	Comprehensive?	References?	Time?
		LibGPT	Yes, fully	To some extent	Yes	93 seconds
		ChatGPT	For second part	No	No	Immediately
		Gemini	For first part	Yes	No	4 seconds
Remark: ChatGPT wrongly described LEAF as a current project, while Gemini incorrectly said that the LEAF project has provided a technological foundation for the VIAF project. LibGPT said rightly (on the basis of the content provided) that these two are similar but unrelated projects, and it also mentioned rightly that the scope of the of the LEAF project included both name authority and subject authority.						
Q4	Can you make a summary of the journal paper entitled "Searching for Meaning Rather Than Keywords and Returning Answers Rather Than Links" in simple sentences?	Sysetm	Faithful ?	Comprehensive?	References?	Time?
		LibGPT	Yes	Yes, summarised	Yes	79 seconds
		ChatGPT	No	No	No	4 seconds
		Gemini	No	No	No	3 seconds
Remark: ChatGPT confidently answered but vaguely explained the meaning of the title, whereas Gemini said that it does not have access to this paper but attempted summarisation, which is wrong. LibGPT briefly explained the scope of this paper as it is included in the collection.						
Q5	Can you comment on the applications of graph theory in library cataloguing?	Sysetm	Faithful ?	Comprehensive?	References?	Time?
		LibGPT	Yes	Reasonably	Yes	121 seconds
		ChatGPT	No	No	No	Readily
		Gemini	No	No	No	3 seconds
Remark: ChatGPT and Gemini both explained in length about graph theory but were silent about specific applications of graph theory in cataloguing. LibGPT attempted to answer it (with the right reference) on the basis of a paper in the collection by Murray and Tillett entitled "Cataloging Theory in Search of Graph Theory and Other Ivory Towers Object: Cultural Heritage Resource Description Networks".						

REFERENCES

1. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K. & Zettlemoyer, L. Deep contextualised word representations. *In* Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 2227–2237 (Association for Computational Linguistics, New Orleans, Louisiana, 2018). doi: 10.18653/v1/N18-1202
2. Li, M.; Kilicoglu, H.; Xu, H. & Zhang, R. BiomedRAG: A retrieval augmented large language model for biomedicine. *Cornell University – arXiv (*preprint*),* 2024. doi: 10.48550/arxiv.2405.00465
3. Das, R.; Zaheer, M.; Thai, D.; Godbole, A.; Perez, E.; Lee, J.; Tan, L.; Polymenakos, L. & McCallum, A. Case-based reasoning for natural language queries over knowledge bases. *arXiv,* 2021. doi: 10.48550/arxiv.2104.08762
4. Li, J.; Yuan, Y. & Zhang, Z. Enhancing LLM factual accuracy with RAG to counter hallucinations: A case study on domain-specific queries in private knowledge-bases. *arXiv,* 2024. doi: 10.48550/arxiv.2403.10446
5. Ovadia, O.; Brief, M.; Mishaeli, M. & Elisha, O. Fine-tuning or retrieval? comparing knowledge injection in LLMs. *arXiv,* 2023. doi: 10.48550/arxiv.2312.05934
6. Bradley, P. The future of search is intelligent. *Comput. in Lib.,* 2023, **43**(3). Available at: https://www.infotoday.com/cilmag/apr23/Bradley--The-Future-of-Search-Is-Intelligent.shtml (accessed on 5 May, 2024).

7. Fitch, K. Searching for meaning rather than keywords and returning answers rather than links. *The Code4Lib Journal,* 2023, **57**. Available at: https://journal.code4lib.org/articles/17443 (accessed on 6 May 2024).
8. Berant, J.; Chou, A.; Frostig, R.; & Liang, P. Semantic parsing on freebase from question-answer pairs. 2013. In Empirical Methods in Natural Language Processing, Vol. 2. Association for Computational Linguistics, **6**, pp. 1533-1544. Available at: https://aclanthology.org/D13-1160.pdf (accessed on 4 May 2024).
9. Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, M. & Wang, H. Retrieval augmented generation for large language models: A survey. *arXiv* (on going work), 2023. doi: 10.48550/arxiv.2312.10997
10. Yasunaga, M.; Aghajanyan, A.; Shi, W.; James, R.; Leskovec, J.; Liang, P.; Lewis, M.; Zettlemoyer, L. & Yih, Wen-tau. Retrieval augmented multimodal language modeling. *Cornell University - arXiv,* 2022. doi: 10.48550/arxiv.2211.12561
11. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.; Rocktäschel, T.; Riedel, S. & Kiela, D. Retrieval-augmented generation for knowledge-intensive NLP tasks. *arXiv*, 2021. doi: 10.48550/ARXIV.2005.11401
12. Ahmed, M.; Mukhopadhyay, M. & Mukhopadhyay, P. Automated knowledge organisation: AI/ML based subject indexing system for libraries. *DESIDOC J. of Lib. Inf. Technol.,* 2023, **43**(1), 45-54. doi: 10.14429/djlit.43.01.18619
13. Mitra, R. & Mukhopadhyay, P. Machine learning applications in digital humanities: Designing a semi automated subject indexing system for a low resource domain. *DESIDOC J. of Lib. Inf. Technol.,* 2023, **43**(4), 219-225. doi: 10.14429/djlit.43.04.19227
14. Setty, S.; Jijo, K.; Chung, E. & Vidra, N. Improving retrieval for RAG based question answering models on financial documents. a*rXiv* (Preprint), 2024. doi: arXiv:2404.07221
15. Lund, B.D. & Wang, T. Chatting about ChatGPT: How may AI and GPT impact academia and libraries? *Lib. Hi Tech. News,* 2023, **40**(3), 26-29. doi: 10.1108/LHTN-01-2023-0009
16. Brzustowicz, R. From ChatGPT to CatGPT: The implications of artificial intelligence on library cataloging. *Inf. Technol. and Lib.*, 2023, **42**(3). doi: 10.5860/ital.v42i3.16295
17. Gozalo-Brizuela, R. & Garrido-Merchan, E. C. ChatGPT is not all you need: A state of the art review of large generative AI models. *arXiv*, 2023. doi: 10.48550/ARXIV.2301.04655
18. Wu, K.; Wu, E. & Zou, J. How faithful are RAG models? quantifying the tug-of-war between RAG and LLMs' internal prior. a*rXiv*, 2024. doi: 10.48550/ARXIV.2404.10198

CONTRIBUTOR

Prof. Parthasarathi Mukhopadhyay, from the Department of Library and Information Science at the University of Kalyani, is an enthusiastic researcher in the application of open source and open standards in Library and Information Science (LIS), data carpentry, library discovery systems, and AI/ML-based applications.