Enhancing Document Retrieval Using AI and Graph-Based RAG Techniques

2024 5th International Conference on Communication, Computing & Industry 6.0 (C2I6) | 979-8-3503-7988-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/C2I663243.2024.10895931

Vikas Kamra
CSE Department
Amity University Uttar Pradesh
Noida, India
kamra1984@gmail.com

Lakshya Gupta
CSE Department
Amity University Uttar Pradesh
Noida, India
lakshyagupta2602@gmail.com

Dhruv Arora
CSE Department
Amity University Uttar Pradesh
Noida, India
arora.dhruv26@gmail.com

Ashwin Kumar Yadav
CSE Department
Amity University Uttar Pradesh
Noida, India
ashwin.yka@gmail.com

***Abstract*—Retrieval-Augmented Generation (RAG) has emerged as a potent method for enhancing the capabilities of large language models (LLMs) by integrating them with external knowledge sources. While traditional RAG models rely heavily on textual similarity for retrieval, often leading to issues like context drift and hallucinations, graph-based RAG offers a more sophisticated approach. By representing documents and their relationships within a graph structure, graph-based RAG enables more context-aware retrieval, reducing hallucinations, and facilitating multi-hop reasoning. This abstract provides an overview of the RAG landscape, contrasting traditional and graph-based approaches, and highlights the advantages of graph-based RAG in addressing the limitations of traditional methods. The application of graph-based RAG to various domains, such as question answering, dialogue systems, and recommendation systems, is also explored. The abstract concludes by emphasizing the potential of graph-based RAG to revolutionize information access and retrieval in diverse AI applications.**

Keywords*—*Graph-Based RAG, Document Retrieval, Evaluation, Knowledge Graphs, Graph Neural Networks

I. Introduction

Document retrieval is a cornerstone of modern AI applications, particularly in question answering, semantic search, and recommendation systems. Traditional Retrieval Augmented Generation (RAG) techniques have enhanced Large Language Models (LLMs) by integrating external information, improving response accuracy and relevance. [1] This approach retrieves relevant documents and uses them as context for LLMs to generate more informative responses. However, traditional RAG faces limitations such as context drift and hallucinations, where retrieved documents may lack true contextual relevance or LLMs generate inaccurate information.

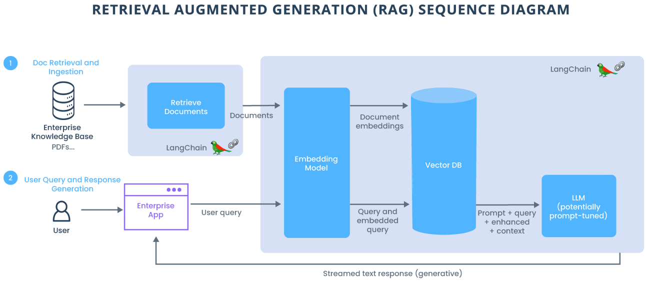
RETRIEVAL AUGMENTED GENERATION (RAG) SEQUENCE DIAGRAM
1
Doc Retrieval and Ingestion
Enterprise Knowledge Base
PDFs...
Retrieve Documents
LangChain
Documents
Document embeddings
Embedding Model
Vector DB
LangChain
2
User Query and Response Generation
User
Enterprise App
User query
Query and embedded query
Prompt + query + enhanced + context
LLM (potentially prompt-tuned)
Streamed text response (generative)

Fig. 1. Retrieval Augmented Generation (Rag) Sequence Diagram [2]

Graph-Based RAG offers a sophisticated solution to these challenges by employing graph structures to represent the rich interconnections between documents, entities, and concepts. This approach captures a more comprehensive understanding of relationships between documents, moving beyond simple textual similarities. By representing information as a network of interconnected entities and relationships, Graph-Based RAG provides a structured context that enhances retrieval accuracy and contextual relevance while mitigating issues like hallucinations. This paper surveys recent advancements in Graph-Based RAG techniques for document retrieval, analyzing their methodologies, effectiveness, and future research directions. [3]

A. *Significance of Document Retrieval*

Document retrieval plays a crucial role in modern AI applications, serving as a bridge between vast amounts of unstructured information and the AI systems that need to process it. [4] In question answering systems, it identifies relevant text passages to answer user queries. For semantic search, it goes beyond keyword matching to understand user intent and retrieve semantically related documents. In recommendation systems,

979-8-3503-7988-4/24/$31.00 ©2024 IEEE

it helps identify items likely to interest users based on their behavior and preferences (refer to Fig. 1).

The importance of effective document retrieval cannot be overstated. It enables AI systems to access and process the most relevant information, which is crucial for performing tasks accurately, efficiently, and insightfully. As AI applications become more sophisticated, the demand for advanced document retrieval techniques continues to grow, driving innovation in this field.

B. Traditional RAG and its Limitations

Traditional Retrieval Augmented Generation (RAG) represents a significant advancement in document retrieval and language model capabilities. By incorporating external information sources, it addresses many limitations of standalone LLMs, leading to more accurate and relevant responses. However, traditional RAG faces several challenges that hinder its overall effectiveness, primarily stemming from its reliance on text-based retrieval and the difficulties in effectively integrating retrieved information with LLM generation.

Key limitations of traditional RAG include context drift and superficial matching, where retrieved documents may share textual similarities with the query but lack true contextual relevance. Hallucinations pose another significant concern, with LLMs sometimes generating factually incorrect or irrelevant information. Additionally, traditional RAG often struggles with complex queries requiring multi-hop reasoning or understanding of intricate relationships between concepts. These limitations highlight the need for more sophisticated approaches to enhance the effectiveness and reliability of RAG systems.

C. Graph-Based RAG as a Solution

Graph-Based RAG emerges as a sophisticated solution to the challenges faced by traditional RAG approaches. By leveraging graph structures to capture the rich interconnections between documents, entities, and concepts, it introduces a fundamental shift in how information is represented and processed. This approach moves beyond relying solely on textual similarities, offering a more nuanced and contextually aware method of information retrieval and generation.

The graph-based approach enhances contextual awareness and reduces hallucinations by constructing a knowledge network that serves as grounded evidence for LLM responses. It improves reasoning capabilities, particularly for complex queries, by enabling multi-hop traversal through interconnected documents. Furthermore, Graph-Based RAG has the potential to significantly enhance retrieval accuracy and relevance by considering not just textual similarities but also semantic relationships and the interconnectedness of documents within the knowledge graph. This paradigm shift promises a more comprehensive and reliable retrieval experience across various AI applications.

D. Paper's Objective: Survey of Advancements in Graph-Based RAG for Document Retrieval

This paper aims to provide a comprehensive survey of recent advancements in Graph-Based RAG techniques for document retrieval. Our objective is to present a clear picture of the current landscape, analyzing key methodologies and their effectiveness, while identifying potential future research directions. We will explore how Graph-Based RAG addresses the limitations of traditional RAG by leveraging graph structures to capture rich, interconnected relationships between documents, entities, and concepts.

The survey will examine the methodologies used in Graph-Based RAG, including techniques for graph construction, node embedding, and graph-based ranking. We will analyze the effectiveness of these approaches using benchmark datasets and relevant metrics. Additionally, we will highlight areas for further investigation, such as scalability to large graphs and dynamic knowledge integration. By providing this thorough examination, we aim to contribute to the advancement of this promising field and its application in various AI-driven domains.

II. Literature Review

Retrieval-Augmented Generation (RAG) is a two-step process designed to enhance the accuracy and relevance of Large Language Models (LLMs) by integrating external information sources. The retrieval phase identifies and retrieves documents relevant to a user query, often using dense vector embedding techniques and similarity search tools. The generation phase then utilizes LLMs to process the retrieved documents as context and generate a response grounded in the provided information.

Traditional RAG systems, while an improvement over standalone LLMs, face challenges such as context drift and hallucinations. These limitations arise from the reliance on textual similarities and the difficulty in effectively integrating retrieved information with LLM generation. Graph-Based RAG systems address these issues by incorporating a graph representation of documents and their relationships, providing a more structured context and enabling multi-hop reasoning.

A. Retrieval-Augmented Generation (RAG)

The retrieval phase in RAG systems aims to identify and retrieve documents relevant to a user's query from a large corpus. This process typically employs dense vector embedding techniques, using models like BERT or SBERT to encode both queries and documents into numerical vector representations. These embeddings capture semantic meaning, allowing for comparisons based on content rather than just keyword matching. Similarity search tools, such as FAISS, are then used to efficiently find the documents closest to the query in the vector space.

In the generation phase, LLMs process the retrieved documents as context to generate a response grounded in the provided information. This approach improves accuracy by providing the LLM with factual information directly relevant to the query and increases relevance by ensuring the response stays focused on the specific information need. However, traditional RAG models can still encounter challenges like context drift and hallucinations, highlighting the need for more sophisticated approaches that incorporate a deeper understanding of relationships and context. [5]

B. Graph-Based Retrieval

Graph-Based Retrieval leverages graph structures to represent and reason about information in a more comprehensive and contextually aware manner. At its core are Knowledge Graphs (KGs), which represent information as a network of interconnected entities and their relationships. In document retrieval, entities might represent key concepts, topics, or named entities extracted from documents, while relationships capture various connections between these entities, such as semantic similarity, citation links, or ontological relationships.

The power of KGs is further enhanced when combined with Graph Neural Networks (GNNs), specialized neural networks designed to operate on graph-structured data. GNNs can learn to capture complex dependencies and patterns within the graph, enabling advanced capabilities such as node embedding, graph-based ranking, and multi-hop reasoning. This combination of KGs and GNNs allows Graph-Based Retrieval systems to move beyond simple keyword matching, understanding semantic relationships between documents and leading to more accurate and contextually relevant retrieval results.

C. Motivating Graph-Based RAG

Graph-Based RAG addresses several limitations of traditional RAG systems. Context drift and superficial matching are mitigated by representing documents and their relationships in a graph structure, allowing for a more nuanced understanding of context. The graph representation also helps reduce hallucinations by grounding LLM responses in interconnected evidence. Additionally, Graph-Based RAG can overcome the shortcut behavior observed in traditional systems by enabling deeper analysis of the relationships between documents and concepts.

A key motivation for Graph-Based RAG is its ability to enhance reasoning and relevance through multi-hop reasoning. Graph traversal algorithms and GNNs support the exploration of indirect relationships between documents, leading to more comprehensive and accurate responses. This capability is particularly valuable for handling complex queries that require understanding nuanced relationships and contextual interpretations, areas where traditional RAG systems often struggle.

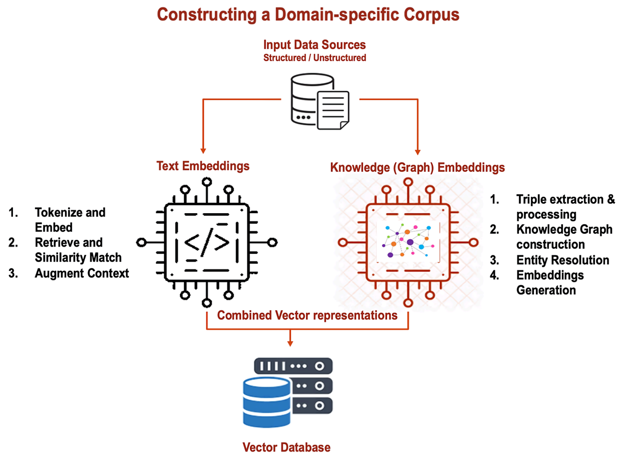
Constructing a Domain-specific Corpus
Input Data Sources
Structured / Unstructured
Text Embeddings
Knowledge (Graph) Embeddings
1. Tokenize and Embed
2. Retrieve and Similarity Match
3. Augment Context
1. Triple extraction & processing
2. Knowledge Graph construction
3. Entity Resolution
4. Embeddings Generation
Combined Vector representations
Vector Database

Fig. 2. Graph-Based RAG Architectures [7]

D. Graph-Based RAG Architectures and Techniques

Graph-Augmented Retrieval focuses on utilizing graph structures to enhance the document retrieval process. Node embedding methods play a crucial role in this approach, learning low-dimensional vector representations of nodes (documents or entities) in the graph. Techniques like Node2Vec, TransE, and Graph Convolutional Networks (GCNs) capture structural and semantic information, enabling efficient similarity comparisons. Graph-Based Ranking methods, such as Personalized PageRank, leverage connectivity information to rank documents based on their relevance to the user query and importance within the graph structure. [6]

Graph-Enhanced Generation integrates graph information into the generation phase of RAG. Graph attention mechanisms allow the LLM to selectively focus on the most relevant parts of the retrieved graph structure during response generation, improving coherence, relevance, and factual accuracy. Graph-Structured Contextualization techniques organize and contextualize information from multiple documents according to the relationships encoded in the graph, helping the LLM better understand connections between different pieces of information and generate more coherent and contextually appropriate responses.

III. Methodology

Graph-Based RAG architectures leverage graph structures to represent and process relationships between documents, offering a more contextually aware and accurate retrieval process compared to traditional RAG. These architectures typically consist of two main components: Graph-Augmented Retrieval and Graph-Enhanced Generation.

A. Graph-Augmented Retrieval

Graph-Augmented Retrieval utilizes graph structures to enhance the process of retrieving relevant documents from a corpus. This approach represents documents and their relationships as nodes and edges in a graph, allowing for a richer understanding of the information landscape.

Node embedding methods are crucial in this process, transforming the graph's structural and semantic information into low-dimensional vector representations. These embeddings enable efficient similarity comparisons between documents and queries. (refer to Fig. 2) Various techniques exist for creating these embeddings, ranging from random walk-based approaches to more sophisticated neural network methods that can capture complex relationships within the graph.

Graph-Based Ranking methods build upon these embeddings by leveraging the connectivity information within the graph. These algorithms consider not just the content of individual documents, but also their relationships and importance within the overall structure. This approach allows for more nuanced and contextually relevant document ranking, often incorporating user preferences or query-specific information to personalize results.

B. Graph-Enhanced Generation

Graph-Enhanced Generation integrates graph information into the language generation phase of RAG. This integration enables Large Language Models (LLMs) to produce responses that are not only grounded in retrieved documents but also consistent with the relationships captured by the graph structure.

Graph attention mechanisms play a key role in this process. They allow the LLM to focus selectively on the most relevant parts of the retrieved graph structure during response generation. By assigning varying levels of importance to different nodes or edges in the graph, these mechanisms guide the LLM to prioritize information most relevant to the query and current context. This selective attention contributes to improved coherence, relevance, and factual accuracy in generated responses.

Graph-Structured Contextualization techniques further enhance the generation process by organizing and contextualizing information from multiple documents according to the relationships encoded in the graph. Rather than presenting the LLM with a simple list of documents or passages, these techniques provide a structured representation that highlights connections between different pieces of information. This approach aids the LLM in understanding complex relationships, synthesizing knowledge from multiple sources, and generating more coherent and contextually appropriate responses.

By combining these graph-based retrieval and generation techniques, Graph-Based RAG systems can offer significant improvements over traditional RAG approaches. They provide a more holistic view of the information landscape, enable more accurate and relevant document retrieval, and support the generation of responses that demonstrate a deeper understanding of context and relationships within the knowledge domain.

IV. Benchmarks and Evaluation

Evaluating the effectiveness of Retrieval-Augmented Generation (RAG) systems, particularly in the context of document retrieval, requires comprehensive benchmarks and robust evaluation metrics. This section summarizes key benchmarks and evaluation methodologies used to assess the performance of Graph-Based RAG systems compared to traditional RAG approaches.

A. CRAG (Comprehensive RAG Benchmark)

The Comprehensive RAG Benchmark (CRAG) is designed to evaluate RAG systems across various real-world scenarios. [8] It includes 4,409 question-answer pairs spanning five domains and eight question categories. CRAG consists of three tasks: Retrieval Summarization, KG and Web Retrieval Augmentation, and End-to-end RAG. [9] These tasks test different aspects of RAG systems, from answer generation to handling structured data and ranking larger sets of retrieval results.

CRAG uses several evaluation metrics, including Accuracy, Hallucination Rate, Missing Rate, and a composite Score (refer to TABLE I). These metrics provide a comprehensive view of a system's performance, considering not just correct answers but also the quality and relevance of the generated responses. [10]

The benchmark evaluates RAG systems through three tasks:

1) Retrieval Summarization: Provides up to five potentially relevant web pages per question to test answer generation.
2) KG and Web Retrieval Augmentation: Offers web pages and mock KG APIs to assess querying and synthesis of structured and unstructured data.
3) End-to-end RAG: Presents 50 web page candidates to evaluate ranking and handling of larger, noisier result sets.

These tasks assess different aspects of RAG systems, from basic answer generation to complex information synthesis and ranking.

B. RAGFoundry

RAGFoundry is an open-source framework for augmenting large language models for RAG use cases [11]. It integrates data creation, training, inference, and evaluation into a single workflow. The framework uses several evaluation metrics, including Exact Match (EM), Faithfulness, Relevancy, and F1 Score, to assess the performance (refer to TABLE II) of RAG models across different tasks and datasets. [9]

C. Cohere Rerank

Cohere's Rerank model is designed to enhance enterprise search and RAG systems by improving the ranking of retrieved documents. It handles multi-aspect and semi-structured data and provides multilingual coverage. [12] The evaluation metrics for Cohere Rerank include Score, which measures the overall performance of the reranking model, and Differential,

TABLE I. CRAG Benchmark Results [8]

	Model	Accuracy (%)	Hallucination (%)	Missing (%)	Score (%)
LLM only	Llama 3 70B Instruct	32.3	28.9	38.8	3.4
	GPT-4 Turbo	33.5	13.5	53.0	20.0
Task 1	Llama 3 70B Instruct	35.6	31.1	33.3	4.5
	GPT-4 Turbo	35.9	28.2	35.9	7.7
Task 2	Llama 3 70B Instruct	37.5	29.2	33.3	8.3
	GPT-4 Turbo	41.3	25.1	33.6	16.2
Task 3	Llama 3 70B Instruct	40.6	31.6	27.8	9.1
	GPT-4 Turbo	43.6	30.1	26.3	13.4

which quantifies the improvement in performance (refer to TABLE III) compared to the baseline. [13]

These benchmarks and evaluation metrics provide valuable insights into the effectiveness of Graph-Based RAG systems, helping researchers and practitioners identify areas

TABLE II. RAGFoundry Benchmark Results [11]

Model	Method	TriviaQA			ASQA			PubmedQA			
		EM	Faith.	Rel.	STR EM	Faith.	Rel.	Acc	F1	Faith.	Rel.
Phi-3 3.8B	Baseline	0.630	-	-	0.109	-	-	0.476	0.290	-	-
	RAG	0.876	0.821	0.836	0.294	0.685	0.895	0.530	0.281	-	-
	RAG-sft	0.878	0.777	0.750	0.252	0.717	0.833	0.720	0.491	-	-
	CoT	0.923	0.555	0.741	0.367	0.263	0.826	0.574	0.439	0.477	0.705
	CoT-sft	0.795	0.793	0.749	0.386	0.749	0.839	0.620	0.458	0.631	0.853
Llama-3 8B	Baseline	0.722	-	-	0.200	-	-	0.560	0.366	-	-
	RAG	0.828	0.783	0.746	0.285	0.610	0.861	0.556	0.398	-	-
	RAG-sft	0.916	0.704	0.714	0.291	0.653	0.854	0.770	0.537	-	-
	CoT	0.896	0.518	0.764	0.395	0.536	0.730	0.684	0.480	0.378	0.732
	CoT-sft	0.851	0.808	0.697	0.422	0.768	0.790	0.694	0.485	0.777	0.883

TABLE III. Cohere Rerank Benchmark Results [13]

	Model (and reranking method)	Score (%)	Differential (%)
Semi-Structured Data (JSON) with Recall@5	BM25	47.5	
	BM25 + Rerank English 3	60.3	12.70%
	Embed English v3	47.8	
	Embed English v3 + Rerank English 3	62.7	13.10%
Code Evaluations (6 Datasets) with NDCG@10	BM25	34	
	BM25 + Rerank 2	37.6	3.60%
	BM25 + Rerank 3	51.7	14.10%
Multilingual Evaluations (MIRACL) with NDCG@10	BM25	36.5	
	BM25 + Rerank Multilingual 3	62.8	26.30%
	Embed Multilingual v3	63.7	
	Embed Multilingual v3 + Rerank Multilingual 3	70.3	6.60%
Long Context Evaluations (7 Datasets) with NDCG@10	BM25	54.6	
	BM25 + Rerank 2	63.1	8.60%
	BM25 + Rerank 3	69	5.90%

for improvement and advance the state-of-the-art in document retrieval technology.

V. Applications and Use Cases

This section explores practical applications and use cases of the concepts and techniques discussed earlier.

A. Enterprise Search and Knowledge Management

Enterprise search and knowledge management systems can leverage Graph-Based RAG techniques to enhance information retrieval within organizations. [14] By improving the accuracy of search results, these systems empower employees to access information more efficiently, leading to improved productivity and decision-making. For instance, a team searching for the most up-to-date version of a design document can benefit from a search engine that prioritizes relevant documents based on semantic understanding, ensuring quick access to required information. [15]

B. Question Answering and Dialogue Systems

Graph-Based RAG can significantly enhance question answering and dialogue systems, making them more accurate and efficient at retrieving relevant information. [16] These systems can be integrated into customer service chatbots, virtual assistants, or educational platforms to provide precise answers to user queries. For example, a customer support chatbot can quickly identify the most relevant knowledge base articles or FAQs related to a customer's question, improving customer satisfaction and reducing the workload on human agents. [17]

C. Recommendation Systems

Recommendation systems can leverage the improved retrieval accuracy offered by Graph-Based RAG to provide users with more relevant and personalized suggestions. By understanding the semantic relationships between items and user preferences, these systems can enhance user experience across various domains, including e-commerce, entertainment, and content discovery. [18] For instance, an online retailer can improve product recommendations based on a user's browsing history, past purchases, and product reviews, leading to increased sales and customer satisfaction. [19]

VI. Conclusion

Graph-based Retrieval-Augmented Generation (RAG) has emerged as a powerful approach to enhance document retrieval and information processing in AI systems. This survey has explored the architectures, techniques, and applications of graph-based RAG, highlighting its advantages over traditional methods. As we conclude, it's important to reflect on the key contributions of this field and consider its future potential and challenges.

A. Summary of Key Contributions and Future Challenges

Graph-based RAG addresses key limitations of traditional RAG systems by leveraging graph structures to represent complex relationships between documents and concepts. This approach enhances contextual awareness, reduces hallucinations, and enables multi-hop reasoning. The architectures and techniques discussed, such as graph-augmented retrieval and graph-enhanced generation, have shown promising results in various applications, including enterprise search, question answering, and recommendation systems. However, the field faces important challenges moving forward, including scalability to large graphs, dynamic knowledge integration, and ensuring explainability and trustworthiness. [20] Addressing these challenges will be crucial for realizing the full potential of graph-based RAG in real-world applications.

B. Transformative Potential and Future Directions

Graph-based RAG has the potential to revolutionize how information is accessed, retrieved, and utilized across diverse AI applications. Its ability to handle complex queries effectively and provide accurate, contextually relevant information contributes to more insightful and reliable AI systems. [21] Future directions for the field include exploring advanced graph algorithms to improve reasoning capabilities, integrating multimodal data to enhance the richness of knowledge representation, and developing new evaluation metrics that can better capture the nuanced performance of these systems. As graph-based RAG continues to evolve, it promises to play a pivotal role in shaping the next generation of intelligent information retrieval and processing systems, opening up new possibilities for AI-driven insights and decision-making across various domains. [22]

References

[1] S. Zhao, Y. Yang, Z. Wang, Z. He, L. K. Qiu, and L. Qiu, "Retrieval Augmented Generation (RAG) and Beyond: A Comprehensive Survey on How to Make your LLMs use External Data More Wisely." 2024. doi: 10.48550/arXiv.2409.14924.

[2] S. Siriwardhana, R. Weerasekera, E. Wen, T. Kaluarachchi, R. Rana, and S. Nanayakkara, "Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering." arXiv, 2022. doi: 10.48550/ARXIV.2210.02627.

[3] M. Blazevic, L. B. Sina, C. A. Secco, M. Siegel, and K. Nazemi, "Real-Time Ideation Analyzer and Information Recommender," *Electronics*, vol. 13, no. 9, p. 1761, Jan. 2024, doi: 10.3390/electronics13091761.

[4] J. Bayarri-Planas, A. K. Gururajan, and D. Garcia-Gasulla, "Boosting Healthcare LLMs Through Retrieved Context." arXiv, Sep. 2024. doi: 10.48550/arXiv.2409.15127.

[5] G. Zare, N. Jafari Navimipour, M. Hosseinzadeh, and A. Sahafi, "Network link prediction via deep learning method: A comparative analysis with traditional methods," *Engineering Science and Technology, an International Journal*, vol. 56, p. 101782, Aug. 2024, doi: 10.1016/j.jestch.2024.101782.

[6] Y. Zhang, J. Wang, L.-C. Yu, D. Xu, and X. Zhang, “Personalized LoRA for Human-Centered Text Understanding.” 2024. doi: 10.48550/arXiv.2403.06208.

[7] N. Nashid, M. Sintaha, and A. Mesbah, “Retrieval-Based Prompt Selection for Code-Related Few-Shot Learning,” in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, Melbourne, Australia: IEEE, May 2023, pp. 2450–2462. doi: 10.1109/ICSE48619.2023.00205.

[8] X. Yang *et al.*, “CRAG – Comprehensive RAG Benchmark.” arXiv, Jun. 2024. doi: 10.48550/arXiv.2406.04744.

[9] H. Touvron *et al.*, “LLaMA: Open and Efficient Foundation Language Models.” arXiv, Feb. 2023. doi: 10.48550/arXiv.2302.13971.

[10] S. Ivanova, F. A. Andreassen, M. Jentoft, S. Wold, and L. Øvrelid, “NorQuAD: Norwegian Question Answering Dataset.” arXiv, May 2023. doi: 10.48550/arXiv.2305.01957.

[11] D. Fleischer, M. Berchansky, M. Wasserblat, and P. Izsak, “RAG Foundry: A Framework for Enhancing LLMs for Retrieval Augmented Generation.” arXiv, 2024. doi: 10.48550/ARXIV.2408.02545.

[12] H. Que *et al.*, “HelloBench: Evaluating Long Text Generation Capabilities of Large Language Models.” arXiv, Sep. 2024. doi: 10.48550/arXiv.2409.16191.

[13] W. Sun *et al.*, “Instruction Distillation Makes Large Language Models Efficient Zero-shot Rankers.” 2023. doi: 10.48550/arXiv.2311.01555.

[14] S. Agrawal, S. Bansal, and V. Kamra, “Navigating the Skies: A Comprehensive Analysis of Airport Automation and Its Impact on Passenger Experience,” in *2024 International Conference on Intelligent Systems for Cybersecurity (ISCS)*, Gurugram, India: IEEE, May 2024, pp. 1–6. doi: 10.1109/ISCS61804.2024.10581214.

[15] P. Kumar, M. Rakhimzhanova, S. Rawat, A. Orynbek, and V. Kamra, “Deep learning based COVID and Pneumonia detection using chest X-ray,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 34, no. 3, p. 1944, Jun. 2024, doi: 10.11591/ijeecs.v34.i3.pp1944-1952.

[16] D. Edge *et al.*, “From Local to Global: A Graph RAG Approach to Query-Focused Summarization.” arXiv, Apr. 2024. doi: 10.48550/arXiv.2404.16130.

[17] V. Kamra, P. Kumar, and M. Mohammadian, “Diagnosis Support System for General Diseases by Implementing a Novel Machine Learning Based Classifier,” *International Journal of Computing and Digital Systems*, vol. 10, no. 1, pp. 737–746, Nov. 2021, doi: 10.12785/ijcds/100168.

[18] J. Wu, J. Zhu, and Y. Qi, “Medical Graph RAG: Towards Safe Medical Large Language Model via Graph Retrieval-Augmented Generation.” arXiv, Aug. 2024. doi: 10.48550/arXiv.2408.04187.

[19] J. S, “Artificial Intelligence (AI) in Retailing- A Systematic Review and Research Agenda.” Mar. 2022. doi: 10.2139/ssrn.4134959.

[20] Y. Gao *et al.*, “Retrieval-Augmented Generation for Large Language Models: A Survey.” 2024. doi: 10.48550/arXiv.2312.10997.

[21] C. J. van Rijsbergen and I. Ruthven, University of Glasgow M. Lalmas, Queen Mary and Westfield College, “Retrieval through explanation: an abductive inference approach to relevance feedback.” Accessed: Oct. 01, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:6482830

[22] M. H. Shuvo, M. T. Rahman, and S. M. M. Ahsan, “WBC Subtype Detection using Deep Learning with Optimizing Hyperparameters by Genetic Algorithm,” in *2023 International Conference on Next-Generation Computing, IoT and Machine Learning (NCIM)*, Jun. 2023, pp. 1–6. doi: 10.1109/NCIM59001.2023.10212778.