National College of Ireland

A Retrieval-Augmented Generation Framework for Medical Question Answer

MSc Research Project
MSc in Artificial Intelligence

Madni Ali Hussain
Student ID: 23158859

School of Computing
National College of Ireland

Supervisor: Paul Stynes

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Madni Ali Hussain
Student ID:	23158859
Programme:	MSc in Artificial Intelligence **Year:** 2024
Module:	MSc Practicum/Internship part 2
Supervisor:	Paul Stynes
Submission Due Date:	12 December 2024
Project Title:	A Retrieval-Augmented Generation Framework for Medical Question Answer
Word Count:	3251 **Page Count:** 9

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Madni Ali Hussain

Date: 12 December 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	✓
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	✓
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	✓

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Retrieval-Augmented Generation Framework for Medical Question Answer

Madni Ali Hussain
23158859
Practicum
National College of Ireland

Abstract

Large Language Models (LLMs) can generate textual information that can be factually incorrect. This occurs when the model fails to accurately represent or reason about the real world. Retrieval-Augmented Generation (RAG) allows an LLM to refer to external data during the generation process to enhance and improve the factual correctness and contextual understanding of textual information. As of right now existing generation models are not up to par with academic standards and excessively hallucinate information that they don't have in their training. The challenge is to minimize the hallucination problem of LLM and to generate accurate answers as it is very important in the medical field. This research proposes a RAG framework that uses Large Language Models for accurately generate answers for medical datasets. The framework combines Retrieval-Augmented Generation techniques, LLMs, and knowledge base stored in a vector database. The LLM models are implemented using Llama 3, Gemma2, Mistral, GPT2. The results of these models, integrated with RAG, are evaluated based on the following metrics: 'BLEU', 'ROUGE-1', 'BERT P', 'BERT R', 'BERT F1', 'Perplexity' score. This research will benefit medical researchers and doctors in getting semantically correct medical information without hallucinations.

Keywords— **LLM, RAG, Data Embeddings, Vector Database**

1 Introduction

The rapid advancement in natural language processing and Large Language Models has changed the world and multiple applications have been built using this technology. Recent advancements have shown that LLMs are great at understanding context Lewis et al. (2021) Chen et al. (2024). The challenge is that LLMs struggle to generate contextually relevant content in subjects that requires domain specific knowledge because they are heavily prone to hallucination with some LLMs having hallucination rates of 50 percent [1][2]. RAG techniques overcome this challenge by using vector database knowledge with retrieval techniques [3][7]. Which results in contextually accurate answers. This contextual awareness is necessary in academic circles to assist in the creation of relevant student assessments hence reducing the time and effort for professors. Benchmarking RAG based frameworks against existing text generation systems remains a limitation.

The aim of this research is to investigate to what extent a RAG-based framework using LLMs can be able to generate a contextually aware system that can generate correct answers without hallucination

To address the research question, the following specific sets of research objectives were derived:

1. Investigate the state-of-the-art LLMS for exam generation.
2. Compare and contrast the existing RAG techniques for relevant text generation.
3. Design a new RAG based framework to assist professors in exam creation.
   Evaluate this RAG framework based on 'BLEU', 'ROUGE-1', 'BERT P', 'BERT R', 'BERT F1', 'Perplexity' score.

The major contribution of this research is the creation of a novel Retrieval-Augmented Generation Framework that combines a Multi Query RAG technique, LLM and a medical question answers knowledge base. To identify the optimal LLM this research compares Llama 3, Gemma2, Mistral, GPT2. Based on accuracy, size of the model, 'BLEU', 'ROUGE-1', 'BERT P', 'BERT R', 'BERT F1', 'Perplexity' score.

This paper discusses LLMs, RAG techniques and vector databases for medical question answers.[1] The research methodology is discussed in Section 3 and Section 4 discusses the design components for the RAG based framework. The implementation of this research is discussed in Section 5 Section 6 presents and discusses the evaluation results. Section 7 concludes the research and discusses future work.

2 Related Work

Large Language models have seen an uplifting leap in these years and to do complex tasks with the use of RAG. Research has been done to improve the efficiency of LLM through RAG and to stop the hallucination. This section will explore different methods, dataseta and models used in LLM for integrating RAG and critically analyze previous work in this topic.

Guangzhi Xiong and Qiao Jin (2024) used RAG techniques in medical field [3][4]. The major contribution of this study is that they benchmarked RAG systems in medical question answering. They introduced Mirage framework to benchmark and evaluate RAG systems performance using medical QA datasets. The main results in this study were that they achieved 18 percent improvement of accuracy over standard models. This paper addresses a critical need by benchmarking RAG in medicine. The limitation in this study is the lack of diversity in the source of data and lack of explanation of cleaning of the data. The study is directed to further developed RAG system for underrepresented medical field.

Victor Zhong and Caiming Xiong (2017) introduced Seq2Sql model that used reinforcement learning to transalate the natural language questions into query able SQL queries. This study used a rewards system by executing queries on the database to learn a policy for generating SQL queries. This research improved the accuracy of the results from 35.9 percent to 59.4 percent. This paper introduced a novel approach of learning SQL queries generation and also improved the results from the previous models

proposed. But the focused on the synthetic and template data is a limitation as it will not be generalised well in the natural and variable inputs. In the future they proposed to further investigate complex and diverse datasets.

Xi Fang and Weijie Xu (2024) surveyed the use of LLMS in various tasks specified in tabular data. It provides a comprehensive overview of techniques. This survey is extensive and provides a detail in the taxonomy of the field. But the lack of exploration in the real-world adoption of these methods for used.

Wang et al. (2024) explored RAG to enhance the ability of code generation in the study CODERAG-BENCH. This research is innovative as it provides a novel approach to combine RAG technique with code generation to make a new standard for benchmarking for future studies. The evaluation shows that retrieval of data from multiple sources and documents chunks of 200-800 shows optimal performance. This paper also highlights the challenges as using multiple sources can restrict the practical.

In conclusion, the reviewed studies have made significant progress in this field but also shows the importance of research in this area. The current research [2][4][6] indicates that the current models are good at generating text but can hallucinate specially in the fields where there is technical knowledge required. The used of RAG techniques can help in solving the hallucinations problem. So, there is a need of research for techniques using RAG for scientific and technical fields and as this research don't dive into real world problems and generalize the RAG techniques for knowledge so there is also a need for research to solve a particular problem in the LLM world.

3 Methodology

The research methodology consists of five steps namely data gathering, data preprocessing, chunking documents, embedding chunks, indexing chunks in vector database and evaluation, as shown in fig 1.

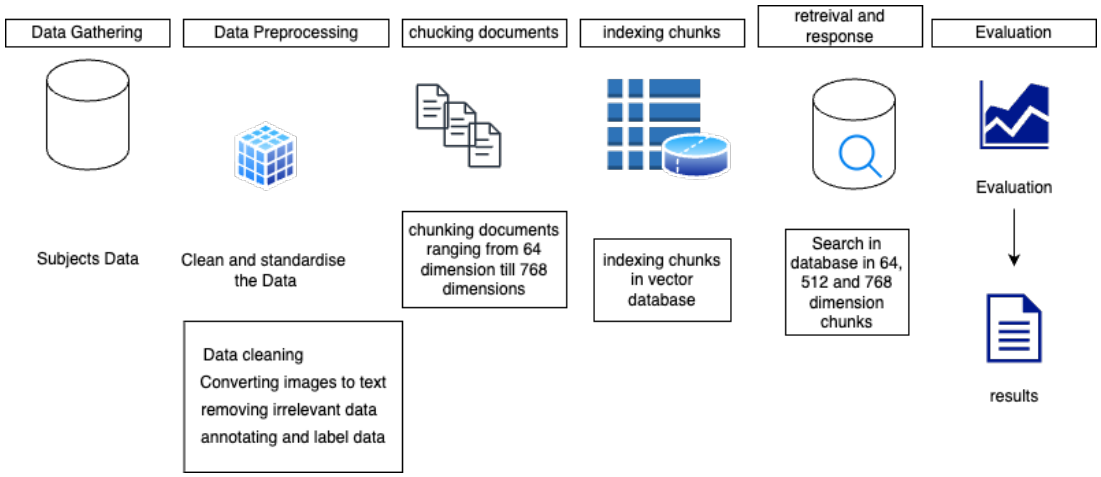

Fig 1. Research and methodology

The first step, Data gathering involves collecting data and this research we used MedQuad [3] that has 13,915 rows containing question, answer and type column.

The second step, Data preprocessing involves converting standardizing format across all documents, convert text to lower case to avoid case sensitive preprocessing issues, reducing irrelevant and redundant information, data annotation using labels and tags for guiding the LLM.

The third step chucking documents involves segmentation of documents into manageable chunks to fit inside the context window of embedding models and LLMs. Large Language Models and Embedding Models have context window of how much token they can handle. So, the data needs to segment into suitable manageable chunks that the model can handle, and the size of each chunk depends on LLM, Embedding Model and the Data (In our case Question and Answers). The quality and size of the chunks contribute to Generation accuracy as these chunks is embedded and then stored in vector database and in the retrieval process the top results are retrieved. Three embeddings' models were used. mxbai-embed-large has a parameter size of 334M, nomic-embed-text and allmini llm have parameter size of 137M and 23M respectively. The choice of embedding model depends on the type of information we are parsing. Each text chunk is parsed to the embedding model, and it outputs a vector embedding of each chunk. The dataset was divided into 64 dimensions to 768 dimensions depending on the type and context. With 512-dimension 20-page text chunks at 5 MB and 768 dimensions the chunks are converted at 8MB data.

The fourth step involves indexing the chunks in the vector database for fast retrieval. We have used PostgreSQL vector extension PGvector for indexing and fast retrieval mechanisms. A user query will be embedded into vector database and that makes the query fast. We accomplish a 20 percent improvement in overall generation response time and for the same query an improvement of 70 percent. We used three four kind of RAG techniques our proposed Multi Query, RAG Fusion, Hyde and Chain of Thoughts.

The fifth step is the retrieval and response generation. On receiving the query, the query is converted to three types of embedding on the basis of dimension, 64, 512 and 768 dimensions. On receiving the query, the vector database is searched for similar embedding. We used k-nearest neighbour KNN searches to find the closer embeddings. The closeness is measured through k-nearest neighbour and cosine similarity. we need to select the most contextually relevant vector knowledge (in our case embedded relevant questions and answer) to feed to the LLM. We selected cosine similarity for the similarity as for our data we needed to neglect the frequency and deal with the relative difference as semantic meaning matters for our data regardless of size of information, this improved our retrieval. The retrieved chunks are then ranked on their similarity score. And then feed up to the LLM. For different LLM we have used different size of top chunks depending on the context window of the LLM.

The sixth step is the evaluation of the retrieval and response generation. Evaluation and results involve evaluating on the following metrics Context Recall, BLEU Score, ROUGE Score and BERT score. The three embedding models and four open sources were evaluated. The metrics used for evaluation is important. BERT score combines precision, recall, and F1 score and indicates the overlap of n-grams, word sequences, and word pairs between the generated answer and the reference answer. And it indicates how much the generated answer is semantically similar to the stored knowledge. Experiment with different model, RAG technique helped us optimize the final pipeline of the system and improvement in response generation.

4 Design Specification

The Medrag framework architecture combines vector embedding and a self-retrieval mechanism as shown in fig 2. The component of the embedding includes fetching data, converting text in to

embedding and saving in a vector database as discussed in section 4.1. components of self-retrieval mechanism are discussed in section 4.2.

4.1 Machine Learning Embedding Model

Embedding model starts when a medical research student upload relevant data on a device that is equipped with a GPU. Each question in the data is first cleaned and the question is saved in the dataset question type that is a meta data used for filtering the query. Then we cleaned the data using a python library to clean the data from all the noise that can disturb the result. We converted the data into chunks of information. The right size for each chunk helps in the retrieval process as we don't want the chunk of information to be so big that it exceeds the context window of models, and not too small that it misses the important information. In the embedding process, we embed this knowledge and save it to the vector database. The embedding model for the user query and knowledge stored should have the same vector size, so it affects the effectiveness of information retrieval. In the chunking and embedding process, we also do the indexing of the chunks of information in the vector database so a similar user query will be retrieved faster, as it is already indexed in our database, which improves the efficiency of information retrieval. For MultiQuery we used only the question and convert the data chunks so that the LLM can embedding model can parse it easily. We created chunks of 10 and then using an embedding model. The machine learning embedding model is based on nomic-embed-text and mxbai-embedded-large. To ensure that the embedding is accurate a model of parameter size of 137M is choose. A meta data that includes the original question, answer and qtype is also stored in the vector database.

4.2 Self-Retrieval and Generation

The self-retrieval element employed includes LLM Model, Embedding Model, Metadata filtering and RAG technique.

The selected RAG techniques (Multi Query, RAG Fusion, and Chain of Thoughts) change the pipeline of the retrieval and generation process. Selecting the right technique plays a role in the overall system performance `and query response.

RAG Fusion: RAG Fusion is similar to Multi Query, but before the generation process, we rank the retrieved information through reciprocal rank. This helps in filtering out the information but may omit some knowledge. It helps in efficiency during generation, as the amount of information passed to the LLM is less than in Multi Query, but some knowledge can be missing.

Chain of Thoughts: In Chain of Thoughts, we structure the user query into reasoning steps that improve the logical reasoning of the LLM during the generation step. For our data (Question and Answer), the system needs more coherence and knowledge rather than logical reasoning.

Multi Query: In Multi Query, we diversify the user query by using an LLM to create multiple queries with the same semantic meaning. It helps to retrieve a broader range of relevant chunks that we can pass to the LLM for final response generation, thus generating a knowledge-rich answer.

The selected technique for the research is Multi Query. First the user input a question to the system. The question is then converted to an embedding with the same model that was used for embedding the medical data. We use a LLM Model to use the original question as reference and convert that into three questions with different wording but same semantically meaning. This give our LLM a large

pool to get data from as the style of question asked can differ from person to person. Then similar questions are found based on the semantic search and then the top search questions related to the question asked along with the meta data is then given to LLM. The LLM then formed the answer based on the question and relevant answer and meta data.

The Multi Query process also involves the LLM to feed upon the answer that it formed and consider the decision it made that reached upon the answer and send a query to itself a form a more coherent answer and return the answer to the question.

5 Implementation

The med Rag Framework was implemented on a web application called the medInfo. Screens and prototype of the web application were designed using Adobe XD. React that is a web frontend framework used to carry out the application development. React was setup on visual studio code. The screens were exported to visual studio code. JavaScript and python were used to implement the application logic. The Ollama framework was used to load the LLM Model and the embedding model. A python library Lang chain was used to make the rag pipeline and the self-retrieval and chain of thought process. Some of the UI of applications such as the chat screen is shown in **fig 3**

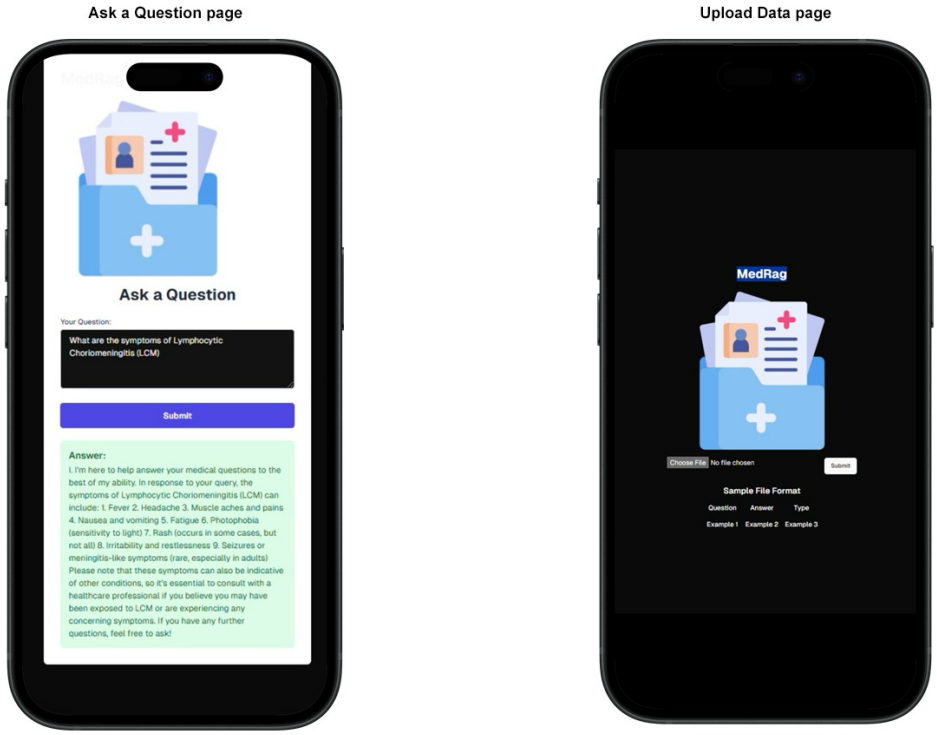

Fig 2: MedInfo AI UI

The medical question answer is pre-processed first and then converted into the embedding with the meta data and then only the questions are embedded data is then saved into with the meta-data. The model file is saved within the Ollama context and called with the Ollama library provided in python.

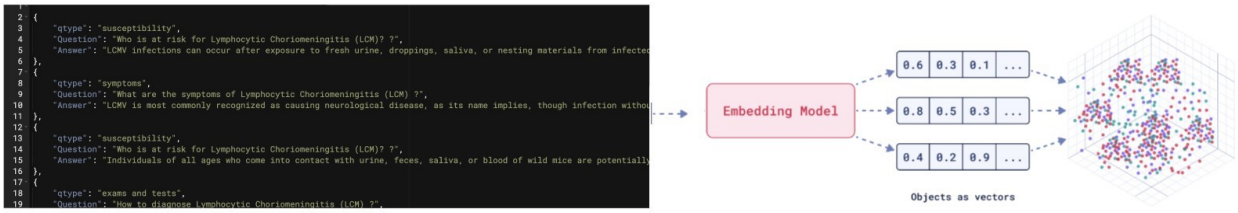

Fig 4: embedding Implementation

A vector database is then used to save the embedding. An LLM model is then used to retrieve the data, parse the relevant information with the meta data and give a response to the user.

6 Results and Discussion:

The aim of this experiment is to compare different RAG techniques Multi-Query, Chain of Thoughts, RAG Fusion and Hyde RAG pipeline with LLaMA3, GeMMa2, Mistral and GPT Large Language Models on the MedRAG Dataset. Through the use of Retrieval Augmentation, five pretrained LLM models were used with different RAG techniques to generate answer that is semantically correct.

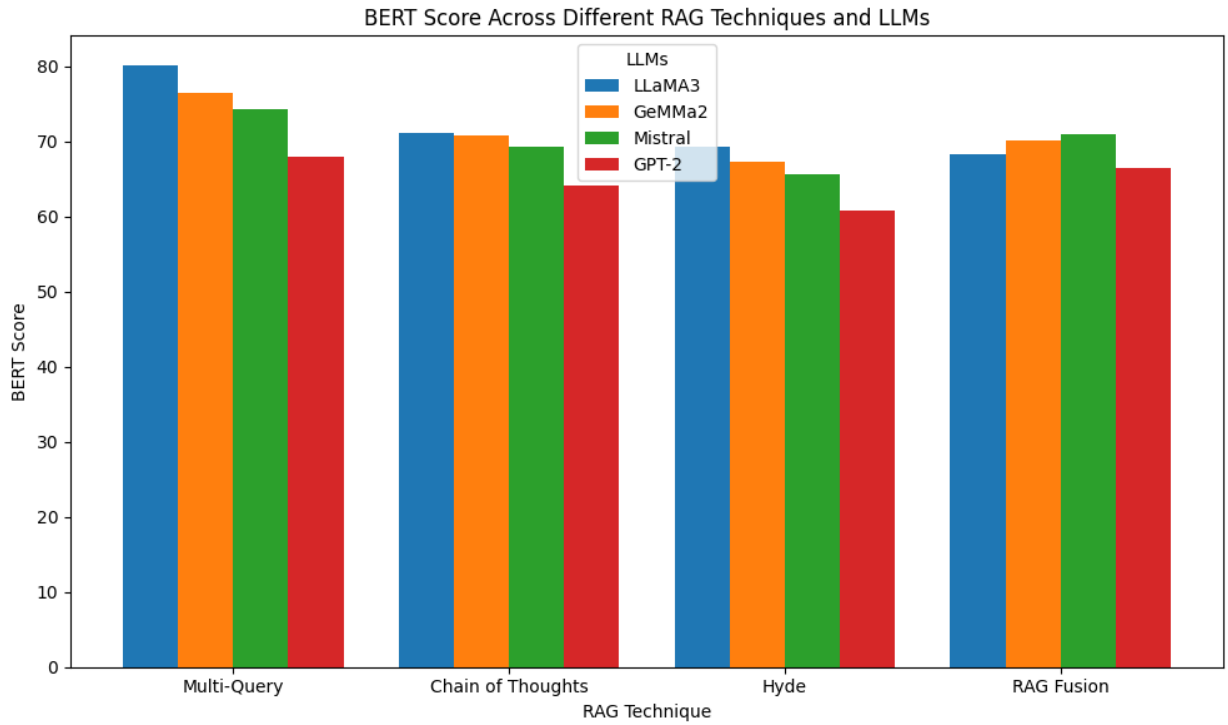

Fig4: RAG technique comparison with BERT Score

Fig 4: Shows the comparison of 5 LLMs based on Bert Score. The results indicate that LLaMa3 with the use of Multi Query RAG technique accurately generated answers with a BERT score of 78%. The BERT score combines precision, recall, and F1 score and indicates the overlap of n-grams, word sequences, and word pairs between the generated answer and the reference answer. This indicates that the generated text is semantically similar to the source text, which is the actual answer to the question.

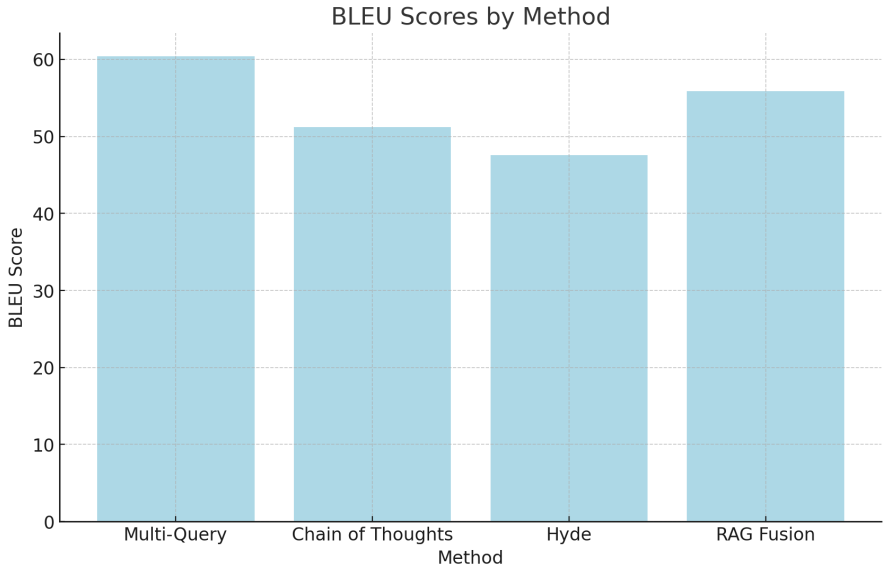

Fig 5: RAG Technique Comparison by BLEU Score

Fig5 shows comparison of 4 RAG techniques using LLaMa3 Model using the BLEU score. Bilingual Evaluation Understudy is an automatic translation and human-created reference translations of the same source. So, for our experiment it indicates how much the LLM imitates the style of the source answer. The results show that with 59% of BLEU score for multi query RAG technique the LLM was able to generate 80 percent of the semantically correct text this shows the diversity of writing style of LLM.

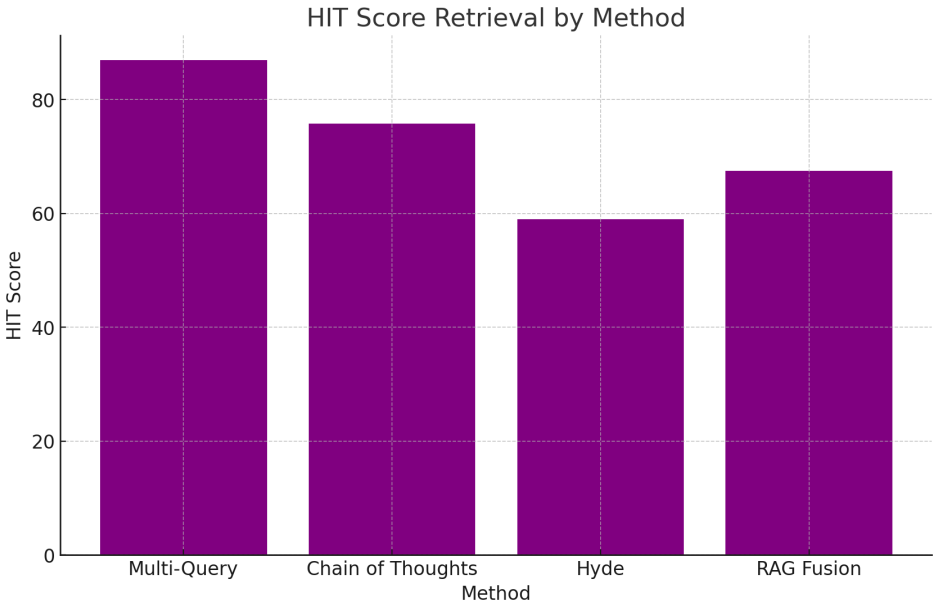

Fig 6: HIT score comparison of RAG framework

A comparison of Retrieval Hit across the four RAG techniques is shown in Fig 6. Retrieval Hit shows how good the RAG technique is in retrieving information from the vector score by cosine similarity. And the result shows that the proposed Multi query RAG technique is significantly better than the other RAG techniques with an HIT score of 83%. This indicates that Multi Query RAG technique is good at finding relevant information from the source information and this will result in generating contextually accurate answers.

The characteristics of RAG framework for Medical Questionnaires indicates that

- Multi Query RAG shows promise in generating contextually accurate answers
- RAG Fusion shows promised in less computing power.

7 Conclusion and Future Work:

The aim of this research was to investigate to what extent a RAG-based framework using LLMs can be able to generate a contextually aware system that can generate correct answers without hallucinations. The research proposes a framework that combines Multi Query RAG technique with a Large Language model and Medical Question Answer knowledge base. The results demonstrated that the RAG technique Multi Query combined with LLM Models can generate semantically accurate information without hallucinations, achieving an 82% BERT score, which is 7% higher than the state-of-the-art Chain of Thoughts approach and RAG Fusion is performs better where computing power is the limitation. A limitation of this study was the diversification of the medical dataset.
This research can potentially enhance the generation capabilities of Large Language models and reduce the hallucination problem in LLMs especially in areas where critical thinking is important like medical field. This work can be improved by using high quality dataset and pretraining the model on the same dataset so that we can combine RAG Fusion technique with Multi Query. This will help us make the LLM compute efficiently and get real time data of patients. This can help professors and medical students to ask questions and improve the speed of diagnosis without the worry of hallucinations in LLMs.

8 References:

1. Yu, W. 2022. Retrieval-augmented generation across heterogeneous knowledge. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop, pp. 52-58. Available at: https://doi.org/10.18653/v1/2022.naacl-srw.7 [Accessed 10 Dec. 2024]
2. Naveed, H., Khan, U.A., Qiu, S., Saqib, M., Anwar, S., Usman, M. and others, 2023. A comprehensive overview of large language models. Available at: https://doi.org/10.48550/arXiv.2307.06435 [Accessed 10 Dec. 2024]
3. Keivalya (2023) MedQuad-MedicalQnADataset [Online]. Available at: https://huggingface.co/datasets/keivalya/MedQuad-MedicalQnADataset (Accessed: 10 Dec 2024).
4. Levonian, Z., Li, C., Zhu, W., Gade, A., Henkel, O., Postle, M. and others, 2023. Retrieval-augmented generation to improve math question-answering: trade-offs between groundedness and human preference. In: NeurIPS'23 Workshop on Generative AI for Education. Available at: https://doi.org/10.48550/arXiv.2310.03184 [Accessed 10 Dec. 2024]
5. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N. and others, 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In: NeurIPS. Available at: https://doi.org/10.48550/arXiv.2005.11401 [Accessed 10 Dec. 2024]
6. Shuster, K., Poff, S., Chen, M., Kiela, D., Weston, J., 2021. Retrieval augmentation reduces hallucination in conversation. Available at: https://doi.org/10.48550/arXiv.2104.07567 [Accessed 10 Dec. 2024]
7. Glass, M., Rossiello, G., Chowdhury, M.F.M., Naik, A.R., Cai, P., Gliozzo, A., 2022. Re2G: Retrieve, rerank, generate. In: NAACL. Available at: https://doi.org/10.48550/arXiv.2207.06300 [Accessed 10 Dec. 2024]
8. Cormack, G.V., Clarke, C.L.A., Buettcher, S., 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, pp. 758-759. Available at: https://doi.org/10.1145/1571941.1572114 [Accessed 10 Dec. 2024]
9. Bruch, S., Gai, S., Ingber, A., 2023. An analysis of fusion functions for hybrid retrieval. ACM Transactions on Information Systems, 42(20), pp.1-35. Available at: https://doi.org/10.1145/3596512 [Accessed 10 Dec. 2024]
10. Raudaschl, A.H., 2023. Forget RAG, the future is RAG-Fusion. Towards Data Science. [Online] Available at: [URL] [Accessed 10 Dec. 2024]
11. Infineon Developer Community, [n.d.]. [Online] Available at: https://community.infineon.com/ [Accessed 10 Dec. 2024]
12. XENSIV™ – sensing the world sensor solutions for automotive, industrial, consumer and IoT applications, [n.d.]. [Online] Available at: https://www.infineon.com/cms/en/product/sensor/mems-microphones/ [Accessed 10 Dec. 2024]
13. Power MOSFET - Infineon Technologies, [n.d.]. [Online] Available at: https://www.infineon.com/cms/en/product/power/mosfet/ [Accessed 10 Dec. 2024]
14. Raudaschl, A.H., [n.d.]. RAG-Fusion: The next frontier of search technology. [Online] Available at: https://github.com/Raudaschl/RAG-Fusion [Accessed 10 Dec. 2024]
15. Camara, A. and Barrera, F.R., [n.d.]. RAGElo. [Online] Available at: https://github.com/zetaalphavector/RAGElo [Accessed 10 Dec. 2024]

16. Es, S., James, J., Espinosa-Anke, L., Shockaert, S., 2023. RAGAS: Automated evaluation of retrieval augmented generation. Available at: https://doi.org/10.48550/arXiv.2309.15217 [Accessed 10 Dec. 2024]
17. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V., 2019. Roberta: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692.
18. He, K., Mao, R., Lin, Q., Ruan, Y., Lan, X., Feng, M. and Cambria, E., 2023. A survey of large language models for healthcare: from data, technology, and applications to accountability and ethics. arXiv preprint arXiv:2310.05694.
19. Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D. et al., 2022. Emergent abilities of large language models. arXiv preprint arXiv:2206.07682.
20. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E.H., Le, Q.V., Zhou, D., 2022. Chain of thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems. Available at: https://openreview.net/forum?id=VjQlMeSB_J [Accessed Date].
21. Liu, J., Zhou, P., Hua, Y., Chong, D., Tian, Z., Liu, A., Wang, H., You, C., Guo, Z., Zhu, L. et al., 2024. Benchmarking large language models on cmexam-a comprehensive Chinese medical exam dataset. Advances in Neural Information Processing Systems, 36.
22. Cai, Y., Wang, L., Wang, Y., de Melo, G., Zhang, Y., Wang, Y. and He, L., 2024. Medbench: A large-scale Chinese benchmark for evaluating medical large language models. Proceedings of the AAAI Conference on Artificial Intelligence, 38(16), pp.17709-17717.
23. Yang, R., Liu, H., Zeng, Q., Ke, Y.H., Li, W., Cheng, L., Chen, Q., Caverlee, J., Matsuo, Y. and Li, I., 2024. Kg-rank: Enhancing large language models for medical QA with knowledge graphs and ranking techniques. arXiv preprint arXiv:2403.05881.