Outliers: Implementing RAG with MedQuAD for Enhanced Medical Question Answering

Soumya Chowdary Daruru, Abishek Vanam, Susrutha Kanisetty

1. Introduction

In today's digital age, navigating through vast repositories of medical knowledge can be daunting, often resulting in misinformation or confusion. Even the modern LLMs do have only general information and nothing in specific. In response to this challenge, we propose to implement a Retrieval-Augmented Generation (RAG) model leveraging the MedQuAD (Medical Question Answering Dataset) to enhance medical question-answering systems by inducing problem specific knowledge into the models. We aim to do this by integrating advanced NLP techniques like Regular Expressions, Tokenizations, Contextual Embeddings, Regressive Language Generators and Question answering with domain-specific retrieval mechanisms.

We used the MedQuAD dataset as the information source for our RAG system. The data is preprocessed and chunked using NLTK libraries and is embedded into vector space and stored in tensors. Queries to LLMs are used to retrieve relevant context from our source data using these semantic embeddings and pass the prompts to our LLM. The Question Answering pipeline is built to handle the entire end to end flow from taking the user Question and generating the Answer based on the domain-specific knowledge, Medical data in our case. The main technical challenge we attempted to solve with this is for efficient retrieval of relevant context from the Information Source and also the augmentation of Prompt with the context to get fine grained and specific answers to user queries instead of generalized answers which the current LLMs give. Our goal is to enable the model to work on local data and prevent hallucinations by providing factual information sources.

2. Background

RAG is built on top of 3 main ideas:

1. Retrieval of the information from multiple reliable information sources.
2. Augmenting of the Prompt that is passed to the LLM such that it contains both the user query and the relevant contextual information.
3. Generation of the response using an LLM that produces accurate and satisfactory results.

To understand the application, one needs to know how the information retrieval system works by transforming the user query into vector space, searching over the documents using indexing, different semantic search algorithms and ranking algorithms. One also needs to understand the working of transformers and LLMs and how to work with them to get best results. Knowledge of Prompt Engineering to get the most out of LLMs capabilities is also important.

3. Data

3.1 IR Source Dataset

The project utilized the MedQuAD Medical Question Answering Dataset, which comprises 47,457 medical question-answer pairs derived from 12 National Institutes of Health (NIH) websites representing diverse medical fields. Each question-answer pair in the dataset is accompanied by extensive annotations in XML format designed to support various Natural Language Processing (NLP) and Information Retrieval (IR) tasks. However, for the purposes of our project, we extracted only the questions and their corresponding answers, concatenating them into a single text file to serve as the basis for our Information Retrieval System.

3.2 Evaluation Dataset

To assess the performance of our Information Retrieval System, we employed the question and gold standard answers from the TREC-2017 LiveQA medical task. This test collection contains 2,479 judged answers, each assigned one of four judgment scores indicating the quality of the response: 1 (Incorrect), 2 (Related), 3 (Incomplete), and 4 (Excellent). Each question in this evaluation set was associated with multiple answers. This approach allowed us to rigorously compare the generated responses of our system

against a gold standard across a variety of medical topics and answer qualities.

Following table shows the descriptive statistics of Source and Test data:

Statistic	Value
Total Word Count	6150231 (6.1M)
Total Token Count(Roberta Tokenizer)	8268264 (8.2M)
Average QA Pair Word Length	151.6
Average Answer Sentence Length	9.26
# of Answers with Rating 1, 2, 3, 4	1436, 678, 223, 142

4. Methods

We started with concatenating all the question answer pairs from the MedQuAD dataset and saving it as a PDF file which acts as a source of Information Retrieval. We used the "fitz" python library to read the PDF and divide it into pages. We then used regular expressions to have a consistent space pattern between sentences and used the"spacy" library with sentencizer to split the page text into sentences. Then all the sentences are chunked to create 10 sentences per chunk. Chunk size of 10 sentences is chosen to accommodate Question text and average sentence length of Answer text and this ensured the average token length of 285 for better fitting into our embedding model. This ends our pre-processing steps where the final chunks of whole data is available.

Next we used the concept of Embeddings to represent these chunks in the vector space, we first used TF-IDF from scikit-learn to create matrices. TF-IDF creates embeddings only based on word frequencies and does not consider the semantic relationships between the words. To overcome this we used BERT to embed our text chunks into the vector space. To get better sentence-level embeddings and for faster semantic similarity search, we replaced BERT with an Pre-trained Model : all-mpnet-base-v2 from Sentence Transformers library.

Model Card - Implementing RAG with MedQuAD for Enhanced Medical Question Answering

Model Details
Retrieval-Augmented Generation System for Enhanced Medical Question Answering. Generates medical domain answers using the LLM based on question and the augmented retrieved context from the information source. *Algorithms*: Embedding generation with TF-IDF, BERT, Sentence Transformers; Question answering with Gemma autoregressive LLM.

Intended Use
This model is designed to provide answers to medical-related queries using a context-aware information retrieval system. This model is not intended for providing diagnoses or medical advice for treatment.

Factors
Not specifically modeled to any group; the system uses medical data that is universal but may inherently contain biases based on the data sources. Designed to perform in typical academic or research settings.

Metrics
Semantic Similarity Scores , BLEU Scores , Rouge Scores(Rouge-1,2,L) (F1 score, precision, recall), and manual evaluation on answer relevance. Cosine (Semantic) similarity scores between generated and reference answers. Comparison of different embedding methods (TF-IDF, BERT, Sentence Transformers).Comparison between different prompts sent to the LLM.

Evaluation Data
TREC-2017 LiveQA medical task dataset. To assess performance of the information retrieval system against a gold standard of medical queries and rated answers. Each answer is rated by the experts as(1 - irrelevant, 2 - related,3 - related not complete, 4 - excellent)

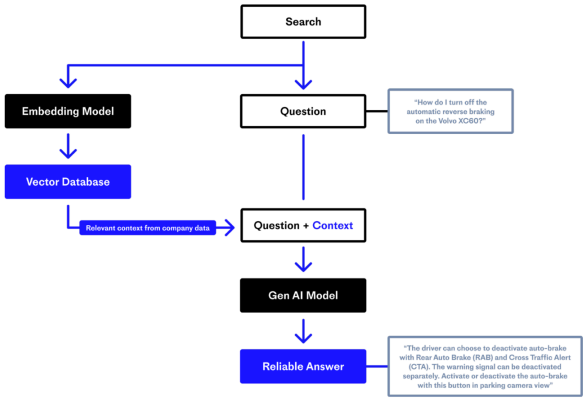
Quantitative Analyses
Performance comparison between different embedding models and the final chosen model across various types of medical queries. Performance comparison of different prompts and a final chosen prompt.

Training Data
MedQuAD Medical Question Answering Dataset. The dataset includes a broad range of medical questions and answers derived from reputable sources, making it a strong information source.Q&As span 37 types, including Drugs, Treatment, Diagnosis, and Side Effects.

Ethical Considerations
The model's training on data from NIH websites may not represent all demographics equally. Further studies are needed to assess any biases.

Caveats and Recommendations
The model should not be used as a sole source for medical decision-making.Consider expanding the training data to include more diverse medical sources to improve coverage and fairness.

Now with the embeddings in place, we started with the questions from TREC-2017 LiveQA medical task dataset, we directly passed the query to the generator model without any additional context for Baseline 1. For generating the response we are using an autoregressive LLM Gemma model from google with the model name : “google/gemma-2b-it”. This direct generation only with query was not performing well because of not passing in any context. Next we compared the query first with our TF-IDF embeddings, retrieved the relevant context documents and passed it to the generator which is our Baseline 2. As an enhancement 1 we have used BERT embeddings and repeated the process and got better results. For enhancement 2 we have used Sentence Transformers embeddings and got better results than BERT.



Source : https://www.pinecone.io/learn/retrieval-augmented-generation/

Now as the final step we did prompt augmentation(prompt engineering) useful for Question Answering where we have given specific instructions in the prompt text to use the context and produce better results. We have given a few example question and answer pairs and asked the model to generate it in a similar way and this improved the performance a bit more. All the evaluation results for each of these enhancements are included in the results section. The assumptions we made were about the MedQuAD dataset having relevant information about the questions it will be asked or assuming the information is present in the pre-trained model parameters. It is possible that the model might not have seen that data as medical data could be confidential and the model might not be trained on all the data that is needed.

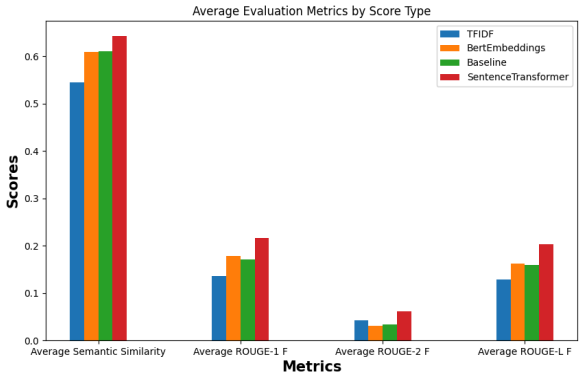
5. Results

We evaluated four different Retrieval-Augmented Generation (RAG) models along with three different prompting strategies for the best performing model. The models assessed include TFIDF , BertEmbeddings, Baseline (direct LLM use without retrieval augmentation), and SentenceTransformer Embeddings. Performance metrics included Semantic Similarity Score, BLEU Score, and ROUGE (ROUGE-1 F, ROUGE-2 F, ROUGE-L F). Scores are calculated comparing the generated answer with the reference answer in the evaluation dataset.

5.1 Model Evaluation

The SentenceTransformer model outperformed all other models. The Baseline model, which does not utilize external information, fell short compared to the SentenceTransformer model. This suggests that more complex embedding strategies like those used in the BertEmbeddings and SentenceTransformer models provide better context integration and response quality, than traditional methods like TFIDF.

Fig:Comparison of metrics across different models

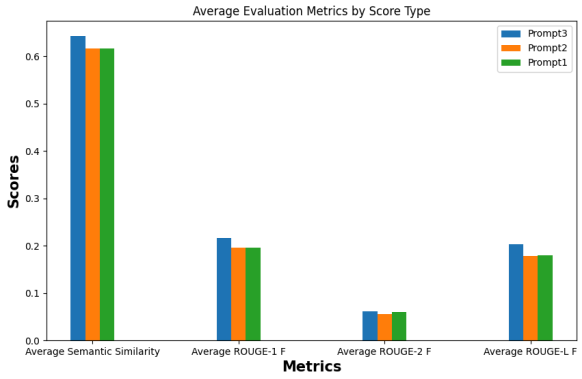


5.2 Prompt Strategy Evaluation(With Sentence Transformer Embedding)

Further analysis was conducted on the SentenceTransformer model using three different prompting strategies to optimize response generation. Prompt 3, which instructed the model to use additional retrieved information to produce explanatory answers, led to the highest scores across all metrics. Prompt 1 and Prompt 2, which used less detailed instructions, resulted in slightly lower scores.

Prompt 1	User query: {query}. Use the information to generate answers. {context}.
Prompt 2	{query} {context}
Prompt 3	User query: {query}. Use the following additional information retrieved from local information storage to enhance the answers. Make sure the answers are explanatory. {context}. Answer:

Fig:Comparison of different prompts



5.3 **Accuracy**

By analyzing the expert ratings and semantic similarity scores, for responses generated using the Sentence Transformer embedding model and reference answer, provides a clear view of how the model's generated answers align with the expert expectations(answers).

Table: Similarity scores grouped according to ratings

Rating	Mean	Std	50%	75%	Max
1	0.26	0.17	0.21	0.38	0.82
2	0.42	0.17	0.42	0.55	0.86
3	0.53	0.17	0.56	0.65	0.86
4	0.64	0.14	0.66	0.76	0.85

In the evaluation dataset for a single question we have multiple reference answers which cover ratings from 1- 4. We are generating the response using Sentence Transformer embeddings and augmented prompt and comparing them with rating 4(Excellent) and rating 1 (Irrelevant) reference answers , separately .

Rating 4 (Excellent): Has 142 responses with highest average similarity score of 0.643. And has narrower standard deviation of 0.143. This highlights consistent high quality generated answers.More than 50% of the generated answers match with the expectations of the expert, with a minimum 0.67 similarity score.

Rating 1(Irrelevant): "Irrelevant" category offers unique insights into the performance. Has 1436 responses, with an average semantic similarity of 0.262. For Rating 1 (Irrelevant), the 75th percentile value is 0.381, meaning that 75% of the responses considered irrelevant have a cosine similarity score below this value. This outcome is not a flaw but rather a demonstration of the model's proficiency. Typically, for irrelevant reference answers that do not align with the question's intent, a low similarity score is desired.

Fig: Correlation between Semantic Similarity and Ratings

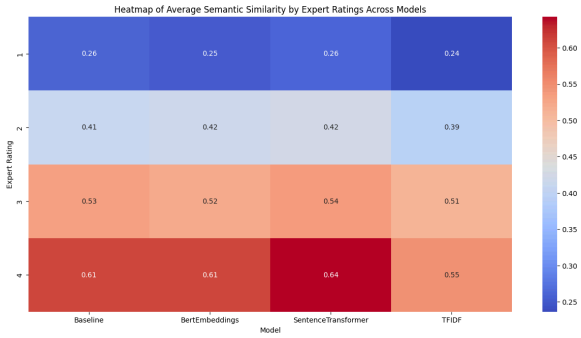
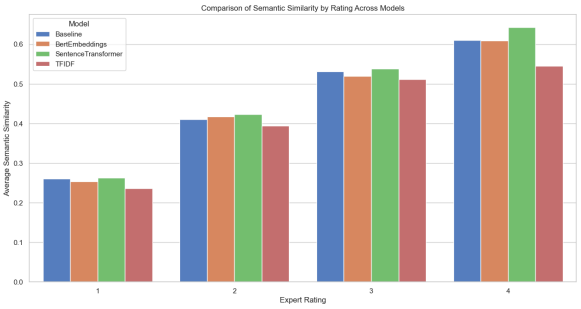


Fig: Scores vs Ratings across different models



5.4 Insights

Effectiveness and Importance of Techniques: The superior performance of the SentenceTransformer model underscores the effectiveness of advanced embedding techniques and retrieval augmentation. Advanced embeddings capture nuanced semantic relationships, enhancing the quality of generated responses. Retrieval augmentation, in contrast to the baseline model which lacks external information, significantly improves response relevance and factual accuracy by leveraging context-rich external data.

Impact of Prompt Design and Embedding Chunk Size: The evaluation of different prompting strategies revealed that detailed prompts, particularly those asking for explanatory answers using additional information (as in Prompt 3), lead to the highest performance metrics. This indicates that specific instructions in prompts effectively guide the model in utilizing available information. Moreover, handling large embedding chunks posed challenges; excessively large chunks often diluted the specificity necessary for accurate information retrieval, resulting in mismatches or failures in response generation. This highlights the need for optimal chunk sizing to maintain focus on relevant query aspects and ensure high-quality outputs.

6. Conclusion

In conclusion, the implementation of a Retrieval-Augmented Generation (RAG) model using the MedQuAD dataset for enhanced medical question answering has demonstrated significant improvements in generating accurate and contextually relevant responses. The advanced embedding techniques, particularly those enabled by the SentenceTransformer model, have proven critical in capturing nuanced semantic relationships by elevating the quality of responses. Moreover, the integration of detailed prompt engineering further enhanced the model's ability to utilize retrieved information effectively, resulting in higher accuracy and consistency in the generated answers. The findings underscore the importance of retrieval augmentation and advanced NLP techniques in overcoming the limitations of general LLMs such as producing generalized answers and hallucinations.

7. References

1) https://github.com/abachaa/MedQuAD/tree/master
2) https://github.com/abachaa/LiveQA_MedicalTask_TREC2017/tree/master
3) https://github.com/glicerico/medquad-scraper
4) https://www.pinecone.io/learn/retrieval-augmented-generation/
5) https://huggingface.co/docs/huggingface_hub/
6) Ben Abacha, A., & Demner-Fushman, D. (2019). A Question-Entailment Approach to Question Answering. BMC Bioinformatics, 20(1), 511:1–511:23. https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-3119-4
7) https://chatgpt.com/