Paper ML07

Biomedical Question Answering with Large Language Models

Matthew Moody, Hastings, UK

ABSTRACT

Question answering (QA) systems have the potential to improve biomedical research and the quality of clinical care by providing users with the latest and most relevant evidence.

Usually, the input of a QA system is a question, potentially alongside a knowledge base containing terminologies and a corpus of scientific publications; and the output contains the answer(s) to the question. Explanation is important, i.e. by showing where the answers come from and the reasoning path to the answer.

In this paper, I will explore a biomedical QA dataset and implement a QA system through the use of Language Models, utilizing key techniques such as Retrieval Augmented Generation (RAG), and I will explore how the explanation of an answer can be improved by providing provenance to a publication.

Keywords—Question answering, Natural Language Processing, Large Language Models

PLANNING

INTRODUCTION

The purpose of this work is to evaluate the use of Large Language Models (LLMs) for answering user-provided questions by extracting relevant information from a pool of resources and using the information found to provide an answer.

LLMs such as ChatGPT, BERT, and Llama have already achieved significant success in question answering (QA) tasks on general corpora, which contain data of multiple modalities, including text, images, audio, and video [1].

Following the success of the LLMs in QA tasks on general data domains, several biomedical-specific LLMs have been developed by either building and training new models from scratch or fine-tuning existing pre-trained LLMs with biomedical data. Such models include Med-PaLM 2 and BioMedLM [2]. These QA systems can be used to assist clinical decision support, create medical chatbots and facilitate consumer health education [3].

However, while LLMs have been proven to have very useful capabilities, they also come with limitations, and there are challenges and risks involved, particularly in the biomedical industry, where the health of patients can be impacted and therefore accuracy is of paramount importance.

The main area of focus for this project will be to investigate the answers provided by the LLMs, and to look to identify cases where a model has 'hallucinated' an answer to a question. That is, producing content which seems plausible but is not actually correct. Currently, LLMs tend to lack transparency in terms of explaining the thought process for generating a response. This project will aim to fine-tune an existing model so that it can explain how it has produced a response, by referring to the information source within the pool of information initially provided. By challenging the model to give this explanation, it would make it much easier for an end user to differentiate the hallucinated answers from the accurate answers.

RESEARCH CONTEXT

Extensive work has taken place on the topic of LLMs and how they can be used for QA tasks. [1] provides a general overview of biomedical QA systems. Its findings point towards the fact that ChatGPT-related tools are effective in addressing medical QA challenges, encompassing a variety of modes of information.

This paper starts with a discussion of unimodal models in general QA, and how they have progressed over time, briefly discussing the advances achieved by models such as UniLM (2019) [4], which employs a shared Transformer network to achieve unified modelling and achieved a strong performance on various natural language generation datasets including the CoQA (Conversational Question Answering) [5] generative question answering dataset. The review continues to discuss different models and the enhancements they provided over recent years, right up to models such as Llama and ChatGPT which is indicative of a substantial evolution in a short space of time. Llama constitutes a suite of foundational language models [6], with parameters ranging from 7bn to 65bn. These models are trained on publicly available datasets. ChatGPT [7] has demonstrated the ability to consider previous questions and responses when answering questions, and to produce responses which are contextually appropriate. It operates without access to

external information sources, and generates responses based on the abstract relationships between words within a neural network.

The discussion then moves towards how these general models have been fine-tuned for the case of biomedical QA. One of the main differences is that biomedical literature contains unique concepts and terminologies which wouldn't typically be found in standard data domains. Therefore, resources such as MedQA and PubMed are often utilized in biomedical QA. This area of QA has also seen huge advancements with the introduction of models such as BioBERT [8] (2019) and BioLinkBERT [9] (2022). BioBERT fine-tuned a single BERT to a pre-trained biomedical language representation model, adapting the word distribution to biomedical corpora in a variety of text mining tasks, outperforming previous state of the art models on biomedical QA (12.24% MRR (Mean Reciprocal Rank – see section on **Methods & Experiment Design** for further information) improvement). BioLinkBERT can capture links within documents, such as hyperlinks and citations, which allows for information across multiple resources to be linked. MedPaLM2 [10] and PMC-Llama [11] have significantly reshaped the biomedical QA field and allow for generation of high-quality responses. MedPaLM2 does this by harnessing the power of Google's LLMs, becoming the first model to perform at an 'expert' level on the MedQA dataset, with an 85% accuracy. PMC-Llama was designed specifically for the biomedical domain, developed through data-centric knowledge injection and comprehensive fine-tuning for alignment with domain specific instructions, and has even outperformed ChatGPT on some biomedical QA benchmarks.

The paper also discusses models for handling multimodal resources, but since the dataset used for this project consists only of text data, we will not investigate this part of the paper in detail.

AIMS & OBJECTIVES

The aim of this project is to implement a biomedical QA system with the ability to explain how it has arrived at an answer to a question, by referring to the information source the model has used. The aim would be to reduce hallucination of responses by the LLM and help the end user to identify cases where a response has been hallucinated. The approach used in this project will be to provide the LLM with links to a selection of webpages containing information required to answer a series of questions from a QA dataset, along with some question & answer pairs which will be used to train the model. Further questions are then provided to the trained LLM as prompts to test the model's ability to select the appropriate webpages from those provided which will allow it to answer each question correctly. This technique is known as Retrieval Augmented Generation (RAG) [12].

The objectives of the project can be further refined as follows:

Retrieval Quality: This project will measure the quality of the information retrieved from the biomedical knowledge base using RAG. Metrics such as Hit Rate, MRR, and NDCG (Normalized Discounted Cumulative Gain) will be used to measure retrieval quality [13]. Section **Methods & Experiment Design** contains detailed descriptions and formulae for these metrics.

Response Generation Quality: The model will also be assessed on its ability to generate relevant and accurate answers based on the information retrieved. Metrics to measure coherence, relevance to the context, grammatical correctness, and accuracy against a gold standard answer will be used to assess the model.

DATA & RESOURCES

The model will be trained on a subset of the BioASQ 9b Training dataset [14], and then tested on a different subset of that dataset. The dataset consists of biomedical questions, resources which could be useful in answering the question, and ideal answers to the questions. The dataset contains questions required four different types of answer: Yes/no, Factoid, List, Summary. While the data is publicly available, there are still some terms and conditions of use. As part of the licence for using this data, sharing and adaptation of the material is permitted under the condition that appropriate credit is given to the data source, a link to the licence is provided [15], and any changes made to the data are indicated. The data resources used for developing the datasets were accessed courtesy of the U.S. National Library of Medicine, subject to the terms outlined in [16].

Llama-2 will be used as the baseline model for this project so access to this will also be required. Llama-2 is a family of pre-trained and fine-tuned LLMs released by Meta AI in 2023, capable of natural language processing (NLP) tasks including text generation and programming code. The Llama-2 API can be accessed for free, and a pipeline will be set up in Python for prompting the API. The Python coding will be carried out using Google Colab.

METHODS & EXPERIMENT DESIGN

The general approach will be as follows:

1. Training the Model: Using an existing biomedical QA training dataset, and an existing pre-trained LLM, train the LLM using Retrieval Augmented Generation (RAG) to create a trained version of the LLM.

2. Model Testing: The fine-tuned model is tested by being given prompts to which it provides an answer, and an explanation for how it arrived at the answer, along with the 5 most relevant resources from the supporting webpages provided. The following performance metrics will be measured:

Hit Rate: The percentage of the resources chosen by the model to support an answer to a question which are correct as per the QA dataset. HR = Number of Hits / Total Number of Resources Returned.

MRR (Mean Reciprocal Rank): A measure of the ability of the model to rank the supporting resources correctly so that relevant resources are closer to the top of the list of chosen resources.
MRR = $\frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i}$, where N is the number of questions, and $rank_i$ is the rank of the first relevant webpage for question i.

NDCG (Normalised Discounted Cumulative Gain): Measures the ability of the model to rank the 5 resources returned in the most efficient order possible (i.e. all relevant items before non-relevant items).

NDCG = $\frac{DCG}{IDCG}$, where DCG = $\sum_{i=1}^{K}\frac{rel_i}{log_2(i+1)}$ and IDCG = DCG where every webpage is already ranked in the most efficient order. In the equation for DCG, K = the number of webpages for the model to return, and reli is the relevance of webpage i (1 if relevant, 0 otherwise).

Accuracy: How often the models provide correct responses. Accuracy = Number of Correct Responses / Total Number of Responses. This will be done for Yes/No, List, and Factoid questions where a correct answer can be easily identified.

ROUGE score: Measure of how well a model generates text that is like reference texts. This measure will be particularly useful when assessing answers to questions requiring a summary.

F1-score: Measures a model's balance between precision and recall. Precision is the ratio between the number of words shared between the response and the ideal answer, and recall is the ratio between the number of shared words and the total number of words in the ideal answer. The formula is given by F1-score = 2 (precision * recall) / (precision + recall).

VISUALISATION OF PROJECT

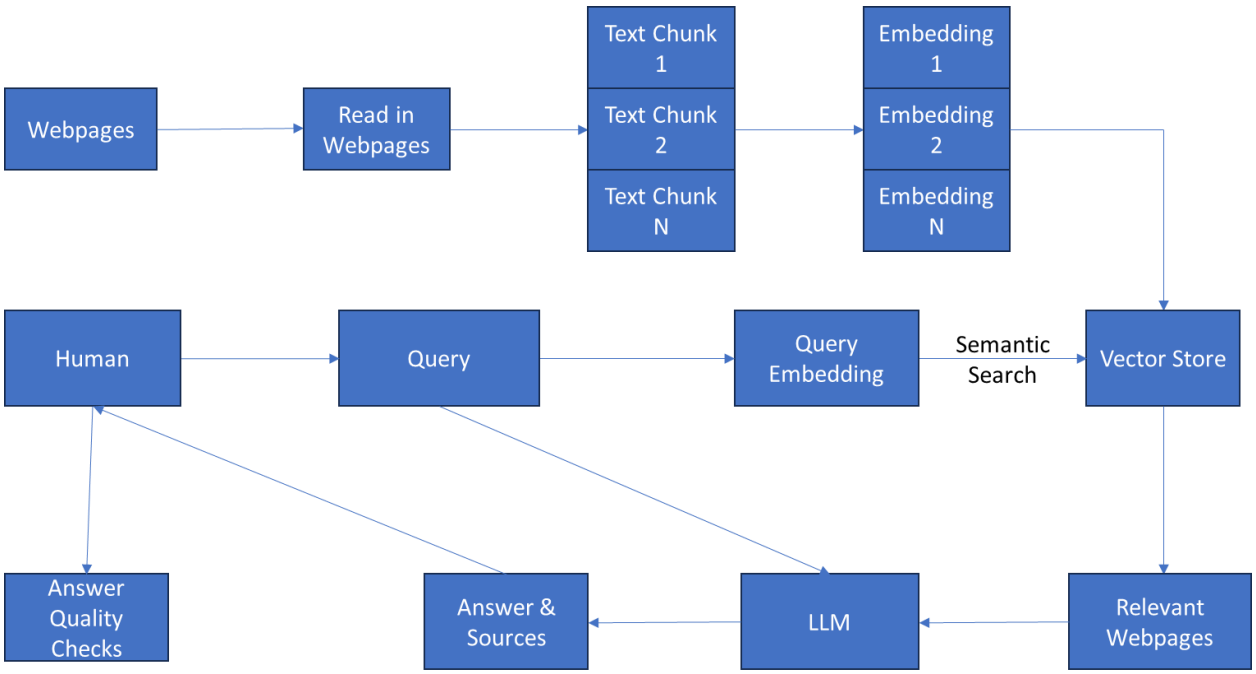

Figure 1

DATA GOVERNANCE & ETHICS

Since this project will be using a publicly available dataset as input, the conditions outlined in [15] and [16] will need to be followed. This licence states that the user is free to share and adapt the material provided that the licence is followed. The output data created will consist of information on the accuracy of the model responses, which will be stored in Excel spreadsheets backed up to a personal OneDrive location. While no personal or business sensitive data is used in this project, it will still be important to be able to explain how a model has arrived at a response, as if the models were to be used in decision making then there would be an impact on public health further down the line.

RISK ASSESSMENT

Possible risks which could affect the successful completion of this project are as follows:

Risk: BioASQ training datasets may need updating so that the model can be trained in how to explain a response.

Mitigation: Factor in time at the start of the project for modifying the target responses if necessary.

Risk: Sending data to a centralized cloud leaves the user open to loss of IP.

Mitigation: Llama-2 can be used privately offline, which makes its use promising for corporate organizations. Hence it was chosen as the base model for this project.

Risk: This project requires access to a biomedical QA data domain and using company data would pose a risk for my company.

Mitigation: A publicly available dataset from the BioASQ website will be used for this project to avoid the need to put proprietary datasets at risk of exposure.

CONCLUSION

Overall, this project will look to build upon the existing research into the use of LLMs within the biomedical industry, leveraging the models in a way that they can explain answers to questions in a more human-like way. This will make it easier for users to identify cases where the model has hallucinated an answer, and therefore make the model a more reliable tool to support decision making within the industry.

PROJECT EXECUTION PHASE

INTRODUCTION

The project was coded in Python, where a RAG system was created using Llama-2_1, Langchain$_2$ and ChromaDB$_3$. This system will allow the user to ask questions about the webpages which will be fed into the system in an upcoming step, without needing to further finetune the LLM. Upon receiving a question, the model will automatically carry out a retrieval step to fetch the most relevant webpages in its database for answering that question, before providing the response.

Notes:

1. Llama-2 is the LLM which was used for this experiment, from Meta. It is pretrained and fine-tuned with 2tn tokens and 7-70bn parameters, making it one of the most powerful open-source models.

2. Langchain is a framework designed to simplify the creation of applications using LLMs.

3. ChromaDB is a vector database, that is a database which organizes data through high-dimensional vectors.

APPROACH

LLMs have proven their ability to understand context and provide accurate answers for QA tasks. However, while they can provide very good answers to questions about information on which they are trained, they can hallucinate when the topic is about 'unknown' information, e.g. was not included in the training data. RAG allows for external resources to be combined with the LLM. The main two components of a RAG system are a retriever and a generator.

Retriever: A system which can encode our training data so that the relevant parts can be easily retrieved upon querying. This encoding is done using text embeddings, that is a model trained to create a vector representation of the information. For this experiment, the ChromaDB vector database was used.

Generator: The Llama-2 LLM was used for the generator part, from the Kaggle Models collection.

This system is orchestrated by Langchain, which allows for the receiver-generator to be created in one line of code.

ISSUES

There were a few issues to deal with during the writing of the code to implement this model:

1. I initially tried to run the code using a Central Processing Unit (CPU) hardware accelerator but encountered an error when attempting to load the *bitsandbytes* package, which would be used to enable the use of 4-bit quantization in my model, which would maintain the performance of the model at a much smaller memory cost compared to a regular 32-bit quantization.

   Solution: Switch to using a Graphics Processing Unit (GPU) hardware accelerator on which *bitsandbytes* could be loaded successfully.

2. Changing to using the GPU caused another problem further down the line when it came to running the model on the QA dataset. Google Colab imposes usage limits on the availability of the GPUs as they are very expensive

resources. Therefore, when I initially attempted to read in the entire QA dataset, which contains 3742 questions, I had already reached my usage limit long before the execution of the code was completed.

Solution: To get around this problem, I have split up the dataset into sets containing 100 questions each (80 training and 20 test questions), which the model is able to manage within the usage limits. A total of 1000 questions (10 sets of 100 questions) are used in the experiment. This still allows for assessment of the model over a relatively large sample of questions. The information retrieval system can still be tested as the relevant information for each question in a set is read into the model, which still needs to retrieve the relevant webpages for each individual question, and the assessment of the response generation system is unaffected.

3. The function used to scrape the text from the webpages for the information retrieval system was originally retrieving all text from the webpages, including metadata and links to other resources, which could potentially affect the ability of the model to utilize the most useful information from each resource when answering the questions.

   Solution: A function was created to only retrieve the section of text titled 'Abstract' from each webpage, as this section is the only section required from each resource for this experiment. If sections with a different title were to be required, the function could be updated to allow for this.

RESULTS – EXPERIMENT I

Table 1

	Number of Questions	Hit Rate	MRR	NDCG	Accuracy	ROUGE Score – F1-score
Yes/No (Training)	223	91.48%	1.000	0.994	77.58%	0.442
Yes/No (Test)	57	85.61%	0.982	0.980	77.19%	0.261
List (Training)	178	89.21%	1.000	0.992	70.98%	0.512
List (Test)	45	84.00%	0.917	0.934	39.30%	0.264
Factoid (Training)	225	83.64%	0.993	0.985	61.33%	0.422
Factoid (Test)	51	72.16%	0.813	0.845	33.33%	0.313
Summary (Training)	174	87.24%	1.000	0.993	N/A	0.507
Summary (Test)	47	84.68%	0.922	0.919	N/A	0.330
Overall (Training)	800	87.85%	0.998	0.991	69.86%[1]	0.466
Overall (Test)	200	81.60%	0.910	0.921	51.43%[1]	0.291

1. Over Yes/No, List, and Factoid questions only.

Table 1 shows the results for the first experiment. The results for each measure can be interpreted as follows:

- Hit Rate: This shows the percentage of resources chosen by the retriever which are relevant to the questions being asked, as per the dataset. Of the 1000 resources chosen in total (200 questions * 5 resources per question), ~81.6% of them were relevant to the question being asked in the response generation phase.

- MRR: The Mean Reciprocal Rate (MRR) is a measure of the efficiency of the retriever. Each of the 5 retrieved resources for each question are ranked in order of relevance. The higher the first relevant item is ranked (1 being the highest possible ranking), the higher (better) then rank reciprocal will be for that question, i.e. if the first resource is relevant then the reciprocal will be 1. If none of the 5 chosen resources are relevant, then the rank will be 0. The MRR is the average rank reciprocal over all questions, which will be a number between 0 (worst) to 1 (best). In this case, we have a score of 0.91, which indicates that the model has performed well in choosing relevant resources to support answers and ranked them highly within the 5 chosen resources for each question.

- NDCG: Normalized Discounted Cumulative Gain (NDCG) is a measure of how well the model has ranked the 5 chosen resources for each question in relation to the ideal ranking (with all relevant resources before non-relevant resources). For each resource, if it's relevant to the question, it is given a score of $1/\log_2(i+1)$, where i is its ranking (1-5), and 0 if it's not relevant. In the best possible case, DCG = IDCG, in which case NDCG = 1. If all resources are not relevant to the question, then NDCG = 0. In this case, we have an average score of 0.921 across all questions, meaning that the model has performed very well in ranking the chosen resources correctly, with a score close to 1.

- Accuracy: Accuracy is an assessment of the ability of the generation part of the model to produce correct answers to the questions. This was assessed differently depending on the type of question being asked.

    - For Yes/No questions, the model was deemed to have answered the question correctly if the exact answer (Yes or No) appeared as the first word in the response provided by the model, with all words changed to lower case to remove the case sensitivity issue. Using this measure of accuracy, the model achieved a 77.19% success rate from 57 test questions. While performing reasonably well, there is room for improvement, and as part of the second experiment, some of the responses to these questions will be reviewed to determine the issue and whether the assessment method should be altered.

    - For List questions, the model response will be searched for each of the terms contained in the exact answer from the QA dataset. For example, if on a List question there are 4 terms in the exact answer and the model response contains 3 of them, then the accuracy score will be ¾ = 75%. The model achieved a score of 39.3% on this metric across 45 test questions, so again spot checks on some of the responses will be performed to determine possible reasons for why the accuracy is so low for this experiment.

    - For Factoid questions, where the exact answer is one word/term, the model response is checked for this term and is given a score of 1 if the term is found, otherwise 0. The model performs at an accuracy of 33.33% across 51 test questions, which again will be investigated to determine whether the model can perform better with some modifications.

    - For Summary questions, since there isn't an 'exact' answer, and the answers are in more of a free text format, the accuracy measure isn't as suitable and instead an alternative measure will be used to assess the performance of the model.

- ROUGE Score – F1-score: The F1-score is a measure of the harmonic mean of the precision and recall of the model. Precision concerns the ability of the model to produce responses of a similar length to the ideal answer, while recall measures the number of words in common between the model response and the ideal answer. The F1-score is a number between 0 and 1, where 1 is the best possible score. Therefore, this measure is suitable for questions without an exact answer, though we have measured this score across all questions in the experiment. A score of 0.33 across the 47 Summary test questions, and 0.291 across all test questions indicates that again this model could perform better.

ANALYSIS

While the model has performed well in terms of how it retrieves relevant information depending on the question, there is room for improvement in the accuracy of the responses. We looked at some of the reasons for the relatively poor results for accuracy metrics in this section.

We first looked at the Yes/No test questions. Out of the 13 wrong responses returned by the model, these can be broken down as follows:

- Hit Rate < 100% or NRR < 1 or NDCG < 1 – 4 responses which can be attributed to failure by the retriever part.

- First word of model response isn't Yes or No – 9 responses which can be attributed to model responses not in correct format.

Next, we looked at the List questions, in which 38 of the 45 test questions were not answered entirely correctly by the model:

- Hit Rate < 100% or NRR < 1 or NDCG < 1 – 13 responses which can be attributed to failure by the retriever part.
- Multiple extracts chosen from the same information source by the retriever – 17 cases.
- Retriever has chosen 5 distinct relevant resources, and the generator hasn't answered the question perfectly – 8 cases.

In some of these 8 cases, the model returns some of the terms in the list, and the percentage of these terms correctly returned can be broken down by question as follows:

Percentage Range	Number of Questions
75%+	0
50-<75%	2
25-<50%	4
<25%	2

Some spot checks on individual responses have revealed a few cases where the model has referred to the correct term, but not worded it in the same way as the QA dataset, i.e. 'loss of memory' was worded as 'severe cognitive impairment, especially concerning memory' by the model, which has caused the performance of the model to be marked down. In another case, the word 'anaemia' was spelt 'anemia' by the model, which meant that the model was marked down for accuracy.

For the Factoid questions, there were 34 incorrect responses out of 51 test questions, of which 17 could be attributed to non-relevant resources being returned by the generator. For a further 8 questions, the same information source was chosen multiple times by the retriever. This leaves 9 'incorrect' responses where 5 distinct relevant sources were chosen. Again, a recurring issue with the responses deemed as 'incorrect' is that either a word has been spelled slightly differently, or a phrase worded differently, i.e. 'The study of protein-protein interactions in a physiologically relevant developing context' vs 'to study protein-protein interactions in a physiologically relevant developing context' or 'small-cell lung cancer' vs 'small cell lung cancer'. This indicates that the model is performing better than these accuracy metrics show, and the challenge will be to (a) better represent how the model is performing and (b) investigate was in which both the retriever and generator parts of the model can be improved.

Due to time and budget constraints, it isn't possible to re-generate the responses in this experiment, but we can still investigate how the model's performance has been evaluated and make some modifications.

In terms of measuring the accuracy, one thing which could be done is to look closer at the F1-score metric. Currently, the metric has no scores higher than 0.33 for any type of test question. Something which could be done is to remove stop words from both the QA dataset answers and model responses. This may help with recognizing responses such as 'The study of protein-protein interactions in a physiologically relevant developing context' as meaning the same as 'to study protein-protein interactions in a physiologically relevant developing context', and therefore being correct. This could be particularly useful for the List and Factoid questions, as well as the Summary questions, as removing stop words will ensure more of a focus on the keywords for each question, to check whether the model is on the right track.

An option for the Yes/No questions is to use a classifier to analyze the sentiment of the responses which don't specify 'Yes' or 'No', so that the accuracy can be more easily assessed for these responses.

RESULTS – EXPERIMENT II

Table 2

	Accuracy	F1-score
Yes/No (Training)	89.70%	0.448
Yes/No (Test)	94.74%	0.233
List (Training)	N/A	0.544
List (Test)	N/A	0.260
Factoid (Training)	N/A	0.438
Factoid (Test)	N/A	0.313
Summary (Training)	N/A	0.507
Summary (Test)	N/A	0.287
Overall (Training)	N/A	0.480
Overall (Test)	N/A	0.272

After implementing a classifier to determine whether the responses to the Yes/No questions fit into the 'Yes' or 'No' category, the accuracy of the model for these questions has increased from 77.19% to 94.74%, which indicates that some of those responses where the words 'Yes' or 'No weren't specifically mentioned can in fact be classified to one or the other.

On the other hand, removing the stop words hasn't had a major impact on the free text answer evaluation, as average scores are down from 0.291 to 0.272, and Factoid questions are the only type to maintain their performance level, at 0.313.

CONCLUSION & NEXT STEPS

There were two main areas of focus for this project, retrieval quality and response generation quality. The model has performed very well across all metrics for the former, and while the quality of the Yes/No question answering has improved considerably in the second experiment, the general standard of the response generation still has considerable room for improvement before the full end goal of the project can be achieved.

Possible ways in which the project could be developed to improve the quality of responses are as follows:

1. Increase the number of resources to be chosen by the retriever. Some questions in the QA dataset have more than 5 relevant resources, and this limit was chosen to ensure that the token limit in Google Colab wouldn't be reached too quickly. The downside of this was that many resources weren't considered for each question, which may have had an impact on the quality of responses.

2. Set a condition that the resources chosen by the retriever must be distinct, i.e. from different webpages. This will ensure a maximal variety of content is being used to formulate responses to the questions.

3. Provide additional instructions depending on the type of question, in the cases where the question in the QA dataset doesn't already do this. For example, on Yes/No questions, specifically tell the model to answer with 'Yes' or 'No'.

Moving forwards, it would also be ideal to be able to use the full pool of resources for all questions in the QA dataset, instead of having to split it up into 100 questions at a time. However, this depends on the availability of tokens, or on taking out a paid subscription with Google Colab.

REFERENCES

[1] Q. Li, L. Li, and Y. Li, Developing ChatGPT for Biology and Medicine: A Complete Review of Biomedical Question Answering. Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong SAR, China, 2024. [Online]. Available: 2401.07510 (arxiv.org)

[2] S. Tian, Q. Jin, L. Yeganova, P. Lai, Q. Zhu, X. Chen, Y. Yang, Q. Chen, W. Kim, D. Comeau, R. Islamaj, A. Kapoor, X. Gao, and Z. Lu, Opportunities and challenges for ChatGPT and large language models in biomedicine and health. Briefings in Bioinformatics, 2024, 25(1), 1-13. [Online]. Available: https://doi.org/10.1093/bib/bbad493

[3] Q. Jin, Z. Yuan, G. Xiong, Q. Yu, H. Ying, C. Tan, M. Chen, S. Huang, X. Liu, and S. Yu, Biomedical Question Answering: A Survey of Approaches and Challenges. ACM Computing Surveys, 2022. [Online]. Available: https://arxiv.org/pdf/2102.05281

[4] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H. Hon, Unified Language Model Pre-training for Natural Language Understanding and Generation. Microsoft Research, 2019. [Online]. Available: 1905.03197 (arxiv.org)

[5] S. Reddy, D. Chen, C.D. Manning, CoQA: A Conversational Question Answering Challenge. TACL, 2019. [Online]. Available: https://paperswithcode.com/paper/coqa-a-conversational-question-answering

[6] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. Lachaux, T. Lacroix, B. Roziere, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, LLaMA: Open and Efficient Foundation Language Models. Meta AI, 2023. [Online]. Available: 2302.13971 (arxiv.org)

[7] OpenAI, Introducing ChatGPT. OpenAI, 2023. [Online]. Available: Introducing ChatGPT | OpenAI

[8] J. Lee, W. Yoon, Sungdong Kim, D. Kim, Sunkyu Kim, C. Ho So, and J. Kang, BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 2019. [Online]. Available: OP-CBIO190693 1..7 (arxiv.org)

[9] M. Yasunaga, J. Leskovec, and P. Liang, LinkBERT: PretrainingLanguage Models with Document Links. Stanford University, 2022. [Online]. Available: 2203.15827 (arxiv.org)

[10] K. Singhal, T. Tu, J. Gottweis, R. Sayres, E. Wulczyn, L. Hou, K. Clark,S. Pfohl, H. Cole-Lewis, D. Neal, M. Schaekermann, A. Wang, M. Amin, S. Lachgar, P. Mansfield, S. Prakash, B. Green, E. Dominowska, B. Aguera y Arcas, N. Tomasev, Y. Liu, R. Wong, C. Semturs, S. Sara Mahdavi, J. Barral, D. Webster, G. Corrado, Y. Matias, S. Azizi, A. Karthikesalingam, and V. Natarajan, Towards Expert-Level Medical Question Answering with Large Language Models. Google Research, DeepMind, 2023. [Online]. Available: 2305.09617 (arxiv.org)

[11] C. Wu, W. Lin, X. Zhang, Y. Zhang, Y. Wang, and W. Xie, PMCLLaMA: Towards Building Open-source Language Models for Medicine. Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai, China, Shanghai AI Laboratory, Shanghai, China, 2023. [Online]. Available: 2304.14454 (arxiv.org)

[12] J. Liu, Building production-ready rag applications. AI Engineer Summit, 2023. [Online]. Available: summarize.tech summary for: Building Production-Ready RAG Applications: Jerry Liu

[13] I. Nguyen, Evaluating RAG Part I: How to Evaluate Document Retrieval. Deepset, 2023. [Online]. Available: https://www.deepset.ai/blog/rag-evaluation-retrieval

[14] BioASQ Participants Area – Datasets for task b. [Online]. Available: http://participants-area.bioasq.org/datasets/

[15] ATTRIBUTION 2.5 GENERIC Deed. [Online]. Available: CC BY 2.5 Deed | Attribution 2.5 Generic | Creative Commons

[16] National Library of Medicine Terms and Conditions. [Online]. Available: National Library of Medicine Terms and Conditions (nih.gov)

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name: Matthew Moody
Email: matthewmoody@ymail.com
Website: linkedin.com/in/matthew-moody-8992aa168