Imperial College London

Department of Computing

Retrieval Augmented Generation for Domain Specific Knowledge

Author:
Ameen Izhac

Supervisor:
Pedro M. Baiz V.

Submitted in partial fulfillment of the requirements for the MEng degree in MEng Computing of Imperial College London

June 2024

Abstract

Until recently, chatbots have only been capable of simple tasks. Recent advancements in Natural Language Processing (NLP) and Large Language Models (LLM) have facilitated the application of LLMs to a wide range of advanced applications not possible before. Along with the introduction of Retrieval Augmented Generation, LLM aptitude for Open Domain Question Answering (ODQA) has largely transformed.

This project works towards the creation of a chatbot capable of domain specific question answering. Taking the Anatomy domain as case study, the necessary aspects of a competent chatbot are investigated and evaluated. The findings reveal important characteristics of a knowledgeable chatbot including that an estimated lower bound on parameter size is required for sufficient answering ability, at which point even niche medical terms are well understood by an LLM, and therefore supplementary information provided must be more instructive than definitions. When using Retrieval Augmented Generation (RAG), conciseness, the absence of irrelevant information in retrieval documents, as well as the absence of poor quality documents are all important factors. Other techniques such as fine-tuning with Low Rank Adapters (LoRA) are found to have little effect, and are discussed with respect to the cause of their downfall.

Contents

Chapter 1

Introduction

1.1 Overview

Recent developments in the field of Natural Language Processing (NLP) and Large Language Models (LLM) has opened the door to chatbots being integrated into multitude of applications where previously the technology was not at an advanced enough level to be useful. One particular area is Open Domain Question Answering (ODQA) in which a user can prompt a chatbot about any topic and receive an informative response.

One limitation of ODQA however is that it is prone to portray erroneous responses in a confident manner, not indicating to the questioner that the information could be incorrect. While for some applications this is not an issue, there are times more accuracy is required. One recent technique that has been introduced to help improve the reliability of chatbot responses is called Retrieval Augmented Generation (RAG) covered in the background section of this report.

In this case, this project endeavors towards the development of a chatbot designed for question answering about a narrow domain. While there are models that specialise in particular fields such as Google's Med-PaLM, The scale of the domain focused on in this project needed to be something smaller than an entire discipline such as Medicine, but rather something that can be contained within a few hundred pages. To that end, the Anatomy domain is used as a case study in this project.

The motivation for this project comes from the frequent task of querying comprehensive documents for specific details, which is a necessary and common task in many professions, for example lawyers cross checking legal information, engineers checking technical specifications or energy compliance officers reading EU energy regulation documents.

For this reason, the Anatomy domain of choice in this project is not the topic of interest, but rather it is to be used as a tool for experimentation, as what is most important is that the technique should be transferable to other domains.

With the use of RAG, and the simplifications that arise from limiting the problem to a narrow domain, this work strives to produce a chatbot who's responses can be deemed reliable (avoid hallucinations). This work investigates a number of techniques used to improve LLMs above their base performance, applies them, and takes learnings from the experiments. Different approaches are used to improve results based on either techniques from the literature, or by devising a unique approach.

To evaluate the chatbot, popular evaluation metrics relevant to the task are considered and selected based on which are most suited for measuring the objective. Metrics used in existing literature are targeted to enable comparison to existing work and relative evaluation of the approach taken.

1.2 Objectives

The development of a chatbot designed for question answering about a narrow domain consists of a number of different aspects. The objective of this project is to focus on a selection of those aspects as outlined by the following points.

- To improve a language model beyond its base capabilities to answer questions requiring domain specific knowledge by supplementing external knowledge using RAG and fine-tuning techniques.
- To evaluate a language models knowledge capabilities on the Anatomy domain, using exact match for the evaluation of multiple choice questions.
- To compare the benefit of using RAG and fine-tuning techniques for improving language model domain specific question answering ability by comparing multiple choice evaluation scores resulting from each technique.

1.3 Ethical Issues

One of the reasons language models have become so powerful in recent years is due to their vast number of parameters. This is especially true for autoregressive Models that produce natural language responses, and that is the type of model that will be needed for this project. It can be expected that models used will be at least hundreds of millions of parameters. Training Language Models with such large number of parameters inevitably requires large amounts of energy and therefore contributes to the release of a significant amount of carbon emissions.

Not only does training on significant amounts of data cause environmental concerns, but there is also questions around what specific data is being trained on. When collecting enormous amounts of data for pre-training a language model, it can be very difficult to be selective as the process is largely automated. The automation of this process can lead to the collection of data whose use is ethically questionable or legally problematic. An article written by the New York Times reports the New York Times themselves suing OpenAI for copyright infringement [28]. In another case [22], a number of unnamed programmers sued GitHub for training on their code to create Microsoft Copilot, an AI programming assistant tool. Such cases highlight the legal and ethical complexities of training such models, and as a consequence there has been much discussion about introducing regulations around AI, with the EU introducing 'The AI Act' in 2024.

Beyond copyright concerns, indiscriminate collection of data has resulted in prejudice, stereotypes, and societal biases being adopted by language models. For example, a language model may exhibit bias on the likelihood of different genders assuming different professional roles, or represent certain races as more likely to be involved in crime than others.

There are also issues around what will happen if LLMs become capable of fulfilling roles that are currently done by humans potentially making a number of peoples jobs redundant. This has led to discussion about concepts such as "universal basic income" [1], which considers a situation in which states pay their citizens a universal stipend due to there not being enough work for people to do.

One of the ethical concerns of LLMs that is especially relevant to this project is that when using RAG, the document being queried may detail information about some important information (such as financial regulations, energy regulations etc). If the LLM hallucinates and it's response is believed by the user, it could lead to incorrect information being used in possibly serious contexts which could also include legal issues.

One counter to this problem is to have models reference the passages they retrieve from the knowledge source directly alongside the generated response. This allows the user to double check correctness. Whilst this method is more guaranteeing of integrity, there remains a trade of between transparency and exploiting the power of language models to produce more tailored and informative responses.

It is also a concern that Language Models are used for bad rather than good. Currently it is very difficult to distinguish a response generated by a computer and one from a human. This can facilitate many negative applications such as cheating in examinations, spreading false information, and impersonation.

1.4 Project Structure

The background section of the project covers important theory in NLP, practical techniques, and evaluation methods. The next chapter, Below 1 Billion Parameters, explores what can be achieved with a relatively small language model. The cost and time of experimenting with a language model scales with its parameters, so it is ideal to work with smaller models if possible. However the chapter reveals that the capabilities of models of this size are limited, and even with enhancements it is evident that effective domain specific question answering necessitates a larger model. The following chapter goes an order of magnitude up in size and undergoes a variety of experiments revealing the benefit of applying different methods to enhance the model, and the final chapter concludes the work conducted and talks about future directions.

Chapter 2

Background

This section first covers some of the important breakthroughs and progressions made in the field of NLP and LLMs that form the foundations of the research leading up to RAG, as well as much of the underlying implementations of the algorithms and models used in this project. Afterwards RAG is introduced, followed by looking at some work that has been built on top of RAG. Finally, this section looks at some of the evaluation metrics that are prominent in the field and relevant to this project.

Table 2.1: Summary of Papers

Paper	Year	Citations	Models	Authors	Domain	Results	Pros and Cons
RAG for Knowledge Intensive NLP Tasks [33]	2020	1018	Seq2seq, DPR	Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. - Facebook, UCL, NYU	RAG	**BLEU Jeopardy**: RAG-Tok - 17.3 BART - 15.1 **MSMARCO**: RAG-Seq - 40.8 BART - 38.2	Introduces RAG field, provides code and details underlying mathematical theory.
BERT [18]	2018	82,856	BERT	Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova - Google	Pre-training	**QQP**: GPT(1) - 70.3 BERT - 72.1 **MNLI**: GPT(1) - 86.7/85.9 BERT - 86.7/85/9 **F1**: 91.8/93.2	Presents innovative new pre-training technique. Pre-training is not reproducible within reasonable costs.
Attention is all you need [5]	2017	96,297	Transformer	Ashish Vaswani, Noam Shazeera Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin - Google	Training	**BLUE EN-DE**: ConvS2S Ensemble - 26.36 Transformer - 28.4 **FLOPs**: Deep-Att + PosUnk - 1E+20 Transformer - 2.3E+19	Transformed the field of NLP with new core language model architecture replacing RNNs. Made much larger context sizes possible.
A Survey on Retrieval-Augmented Text Generation [15]	2022	12	N/A	Huayang Li, Yixuan Su, Deng Cai, Yan Wang, Lemao Liu	RAG Survey	N/A	Concise brief on work building on RAG. Lacks metrics for comparison of techniques.
Improving RAG Models for Open Domain QA [39]	2022	15	N/A	Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayak- kara	RAG	**F1**: RAG-original-QA+R - 12.12, 16.46, 24.62 RAG-end2end-QA+R - 19.57, 23.7, 37.96 **EM**: RAG-original-QA+R - 3.66, 8.62, 14.21 RAG-end2end-QA+R - 8.32, 14.08, 25.95	Introduces performant method for improving embedding and retrieval quality. Opened door to further ideas around improving retrievers. Complex implementation requirements.
Improving language models by retrieving from trillions of tokens [37]	2021	471	Retrieval-Enhanced Transformer (Retro)	Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, et al. - Deep Mind	RAG	**QA Accuracy**: REALM(2020) - 40.4 DPR(2020) - 41.5 RAG(2020) - 44.5 RETRO - 45.5	Shows improved performance by using very large scale retrieval source. Expensive to apply or attempt to reproduce.
Benchmarking Large Language Models in Retrieval-Augmented Generation [20]	2023	8	N/A	Jiawei Chen, Hongyu Lin, Xianpei Han, Le Sun	Benchmarking	N / A	Considers important factors for quality RAG system. Doesn't include some popular open source models such as Llama.
Retrieval-Augmented Generation for Large Language Models: A Survey [46]	2023	5	N / A	Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, Haofen Wang	RAG Survey	N / A	Informative overview diagram including a timeline. Provides an idea of the ordering of technique introduction.
QLoRA: Efficient Finetuning of Quantized LLMs [42]	2023	299	LLaMA, T5	Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, Luke Zettlemoyer	Fine-tuning Efficiency	**Elo**: ChatGPT - 966 ± GPT-4 - 1384 ± Guanaco 65B - 1022 ± **GLUE**: BF16 - 88.6 QLoRA FP4 - 88.6	Makes access to very large models possible with significantly lower hardware requirements. Reduces performance to some extent.
LoRA: Low-Rank Adaptation of Large Language Models [12]	2021	4292	RoBERTa, DeBERTa, GPT-2, GPT-3	Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen	Fine-tuning Efficiency	**BLEU**: GPT-2 Finetuning - 68.2 ± GPT-2 LoRA - 70.4 ± **ROUGE-L**: GPT-2 Finetuning - 71.0 ± GPT-2 LoRA - 71.8 ±	Creates a cost effective way of partial fine-tuning. Can save, reuse and swap fine-tuned parameters. Paper criticized by reviewer for over claimed benefit.

2.1 Relevant Progress in NLP

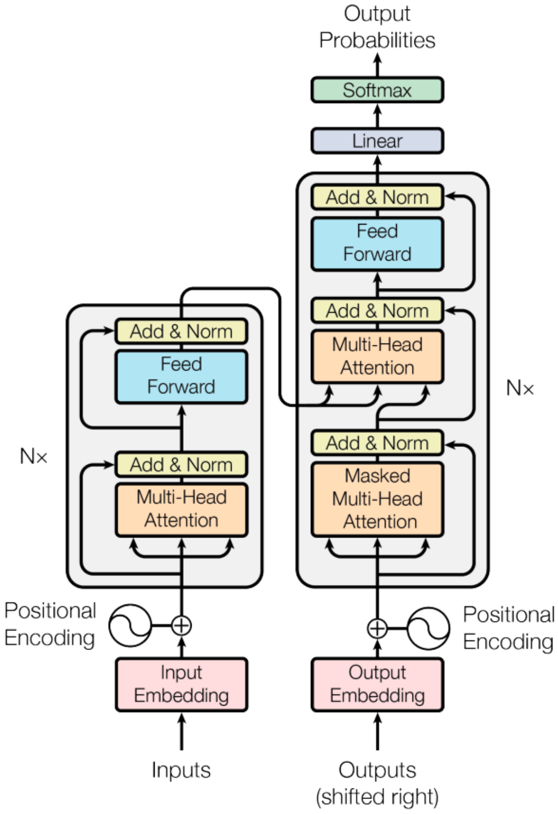


Figure 2.1: Transformer Architecture. Image Source - Vaswani et al. [5]

Vaswani et al. [5] (2017) introduced a completely new architecture for Neural Networks called the transformer, which is distinguished for its use of a mechanism termed ”attention”. In this architecture, in the encoder, an attention vector (a vector of values comprising of weights assigned to words) weighs the relevance of each word in the current sequence, with respect to the current token in focus when learning the meaning of a word. And in the decoder, an attention vector is used to weigh how important each of the input tokens, and of the tokens generated so far are for generating the next token. As sequences are processed, the model maintains a hidden state representing its current understanding, and uses this hidden state to determine the attention vector. Multi-Head Attention uses multiple attention vectors and concatenates their results to learn different relationships between tokens.

The transformer will be the core architecture operating all of the models used throughout this project.

Devlin et al. [18] (2018) introduced bidirectional transformers. Before the BERT model, models often learnt the meaning of a word in a sentence based on the words prior to it and these would be called (unidirectional transformers). BERT introduced the idea of bidirectional transformers, that use both the words before a word and the words after it to understand the meaning of that word. They did this by masking some words in a sentence and having the model predict the masked words based on the context. Another training method they used was providing a model with two sentences and having it predict whether or not one sentence followed the next in a given source text.

2.2 RAG

The primary area of NLP that this project will focus on is RAG. Prior to RAG, question answering NLP models would either search documents for the most relevant text excerpt, extract and return it to the questioner, or instead generate a response based on the information stored in the model's parameters. The problem with extracting text excerpts is that the model is limited in how it can answer a question, and as has become evident in recent times with chatbots like ChatGPT, the problem with using model parameters to generate a response is that the response is often unreliable. Lewis et al. [33] introduced RAG which uses a combination of the two prior mentioned techniques to generate flexible informative answers based on an information source.

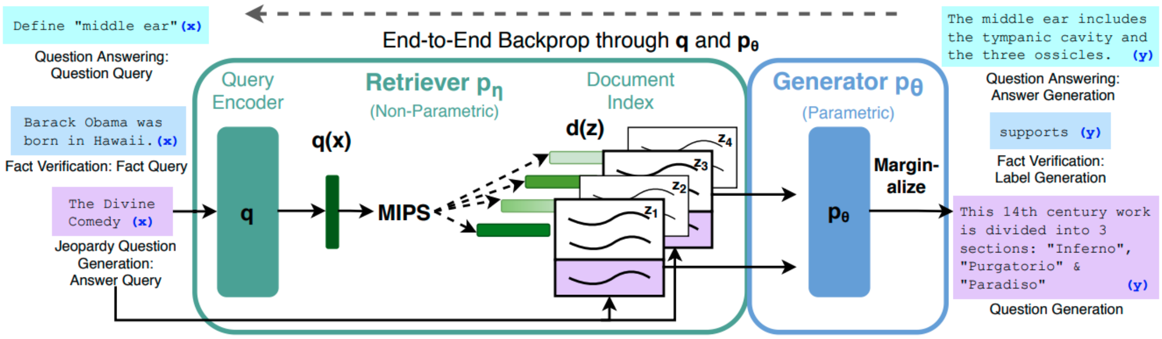

Define "middle ear" (x)
Question Answering: Question Query
Barack Obama was born in Hawaii. (x)
Fact Verification: Fact Query
The Divine Comedy (x)
Jeopardy Question Generation: Answer Query
End-to-End Backprop through q and pθ
Query Encoder
q
q(x)
Retriever pη
(Non-Parametric)
MIPS
Document Index
d(z)
z4
z3
z2
z1
Generator pθ
(Parametric)
pθ
Margin-alize
The middle ear includes the tympanic cavity and the three ossicles. (y)
Question Answering: Answer Generation
supports (y)
Fact Verification: Label Generation
This 14th century work is divided into 3 sections: "Inferno", "Purgatorio" & "Paradiso" (y)
Question Generation

Figure 2.2: RAG Original Architecture Lewis et al. [33]

In Lewis et al. [33] (2020), the information source used is Wikipedia. They create and store a knowledge base of information as an abstract vector form representation from Wikipedia discretised as ”documents” where each document is 100 words of information. When a user inputs a prompt, the prompt is embedded into vector representation and fed to a retriever model which compares the input prompt to documents in the knowledge base using Maximum Inner Product Search (MIPS) to return the most relevant documents. A seq2seq model is then used to predict the sequence of words that will form the response, however where a regular seq2seq model would only consider the prompt as well as the so far generated response in predicting the next word, the seq2seq model here also considers the documents returned by the retriever to guide the generation of each word. The paper used two approaches, one that would use the same documents to generate the entire response, and another that would use multiple documents, switching between them for the prediction of each next token.

2.3 Advancements in RAG

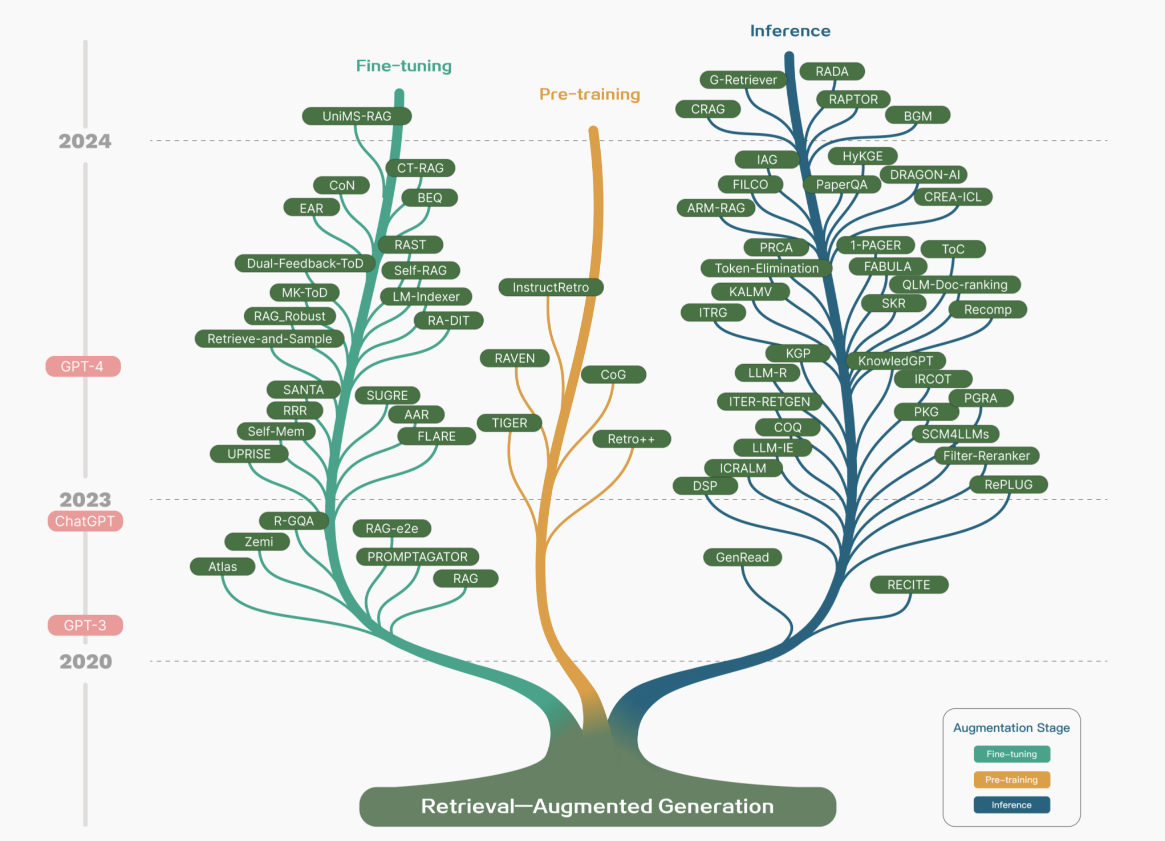


Figure 2.3: Summary Timeline of RAG Field [46]

Borgeaud et al. [37] (2021) is research conducted by DeepMind that investigates the effects of using a very large scale retrieval database for RAG. They found by doing this they can achieve impressive results comparable to GPT-3 with 25 times less parameters. While this work is most likely out of the scope of this project in terms of computing resources, it demonstrates that RAG can significantly improve performance given a strong relevant retrieval database. Fortunately, the retrieval database required for this project can be significantly smaller, given that the scope of the model being developed is limited to a particular field.

Li et al. [15] (2022) provides a summary of the progress in RAG up to 2022. The paper explains that there are three main kinds of knowledge bases used by RAG models. The most common is a Training Corpus, which is the labeled set of training data that the model itself was trained on. Another type of knowledge base is External data, which is labeled data that the model was not trained on (which is good for domain adaptation). And finally Unsupervised Data, which is a corpus of large scale unlabeled data.

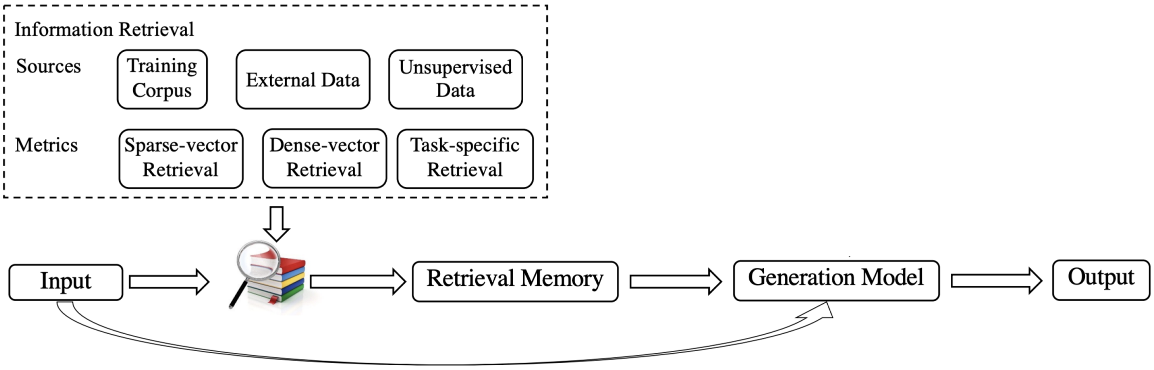


Figure 2.4: RAG Information Retrieval [15]

There are also three common methods for evaluating document relevance for retrieval. Sparse Vector Retrieval is a method that compares the input prompt with the candidate text where both are embedded as sparse vectors (where most elements are 0 and few are 1 representing the syntactic meaning of the words). Dense Vector Retrieval in which retrieval is conducted by comparing vectors in dense form that represent the semantic meaning of the text. Finally, Task Specific Retrieval tries to get the model itself to learn how to best evaluate a document for retrieval. Future work will most likely focus on dense vector retrieval due to the more common need for semantic similarity, but Sparse Vector Retrieval may still be useful in applications such as exact word match.

Once the documents for augmenting the response are selected, there are multiple methods for integrating the documents into the response. Data augmentation simply concatenates the vector prompt with the fetched document which is simple and has proved to work well. Attention mechanisms use attention to find the most relevant parts of the retrieved documents. And Skeletal Extraction takes the most relevant parts of a document and masks the rest. Future work may consider applying these methods selectively, as there are advantages to each. Attention is very powerful but as this project will show, there is also benefit to removing irrelevant information as Skeletal Extraction does by masking. Perhaps a combination of these methods is something worth being explored.

Retrieved documents are often prompt response pairs, and other times just regular informative passages. The relevance of the documents retrieved is very important because documents that are similar to the prompt yield better performance, but documents too dissimilar cause the model to perform worse than if no document was used. This could explain the benefit found in tailoring a RAG system to a particular domain [37], as this narrows the focus of the knowledge base and perhaps improves the relevancy of retrieved documents.

2.4 Domain Specific RAG

Kalurachchi et al. [39] (2022) investigates the benefits of limiting the domain of a RAG model. The original RAG model uses a model for encoding the query, another model for encoding the documents of the knowledge base, and a generator model. The original RAG only fine-tunes the query encoder and the generator, and mentions that fine tuning the document encoder is unnecessary. In this paper, they also fine-tune the document encoder and call it RAG-end2end (end-to-end trainable) because it helps the model differentiate between different domain specific knowledge bases.

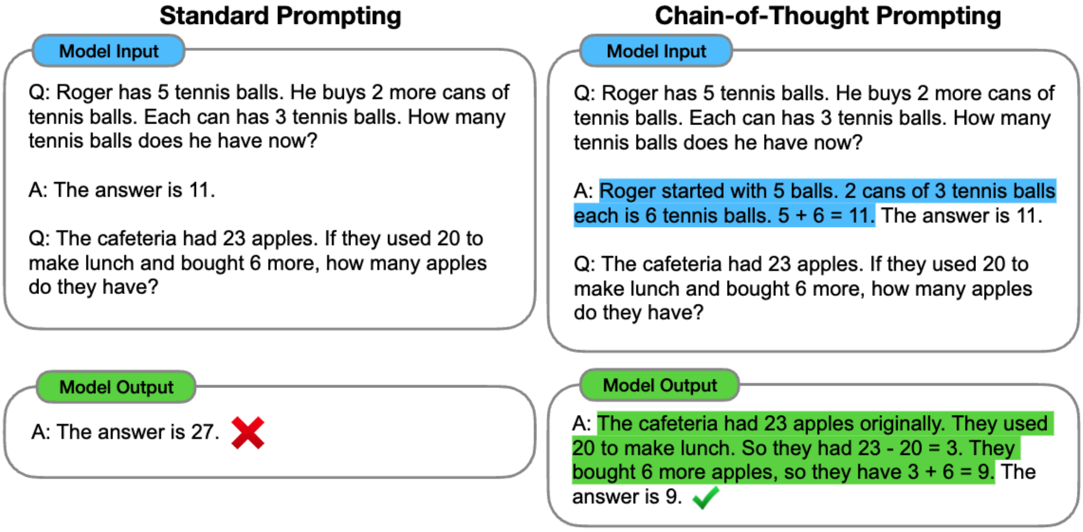
One additional training method used in this paper is statement reconstruction. This is where they would provide the model with a prompt, allow it to retrieve relevant documents, then they would try to predict the prompt based solely on the documents retrieved. This was under the assumption that it would force the model to gain more domain specific knowledge.

2.5 Prompt Engineering

Prompt engineering is the easiest technique for improving the performance of a language model and arguably the most important for certain tasks [21]. Language models are incredibly powerful at leveraging the information they are provided, but if they are provided with garbage, it does not matter how powerful the model is, it cannot produce something of value.

Brown et al. [43] introduces an innovative technique for improving performance via prompt engineering. The authors found that providing a model with one or a few example prompts and responses can help improve performance without any need for fine-tuning or alternative methods. It also showed that the number of examples made a difference. In the paper they investigate the benefit of both One-Shot and Few-Shot prompts which supplement one or many examples respectively with many usually being in the range of 10 - 100. By applying this technique, they find that in some cases Few-Shot prompting can lead to results comparable to fine-tuning methods.

Chain of Thought is another prompt engineering strategy that has been applied to LLMs. Wei et al. [19] demonstrates state of the art accuracy achieved simply by encouraging the model to respond with the reasoning behind its answer. Similar to Few-Shot, chain of thought provides the model with an example question and answer, except in this case, the answer is not simply stated, but prompt provides a series of reasoning steps taken to reach the answer. An example from the paper can be seen in Figure 2.3.



Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A: The answer is 27. ✗

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Figure 2.5: Chain of Thought Prompting [19]

Whilst this method demonstrated impressive results, there has also been concern about whether the reasoning the model provides in its response is reflective of the actual reasoning behind the model's response [29]

2.6 Low Rank Adaptation

Low Rank Adaptation (LoRA) [12] is a technique introduced to support the fine-tuning of large language models. LoRA extends a number of supplementary parameters on top of a models original parameters proportional to a 'rank' parameter. The supplementary parameters can be trained independently by freezing the model's original parameters resulting in significant savings in memory. It is possible to train up to 10,000 times less parameters and reduce the required GPU memory by 3 times. It was found that LoRA can provide performance results similar to that of the results achieved via the full fine-tuning of all a models parameters and provides additional benefits such as the ability to attach or detach the LoRA parameters or even switch between different trained sets of LoRA parameters fine-tuned for different tasks.

2.7 Quantized Low Rank Adaptation

Quantised Low Rank Adaptation (QLoRA) [42] takes model parameters down to a 4bit representation using a newly introduced datatype called NormalFloat along with other techniques such as double quantisation. With this, they report the ability to train a 65 billion parameter model on a single 48GB GPU while preserving the performance of 16-bit fine-tuning.

A significant benefit of LoRA and QLoRA is that they make some of the largest models that were previously only accessible to those with large compute power, accessible to a wider audience.

2.8 Evaluation Metrics

BLEU Papineni et al. [26] is a metric originally designed for machine translation evaluation but has become a prominent metric for more general tasks in NLP. It uses a formula to estimate a score for how close a models output was to the reference output(s). Formally it is calculated as

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2.1}$$

Where p_n is the geometric average of the modified n-gram precisions with n-grams up to length N and weights w_n. A practical example of how it would be calculated is as follows. Suppose the model was expected to output ”The sky is blue” but instead it output ”Red is the sky”. First we calculate precision as $common\ unigrams/total\ unigrams$ in the model output (a unigram is word or token such as a full stop). In our case $3/4 = 0.75$. Then we calculate a brevity penalty BP as $min(1, words\ in\ generated\ text/words\ in\ reference)$. In our case $4/4 = 1$. Then the BLEU score is $BP * \exp(log(precision))$. In our case $exp(log(0.75)) = 0.8825$.

Thibault et al. [32] introduces BLEURT, in which an LLM acts in itself as a metric for evaluating other LLMs. BLUERT takes advantage of language models understanding for words, and uses this to propose a solution to the difficult problem of accounting for synonyms or similar language to a reference without the need for exact word matches as is common for many metrics in NLP. The authors took millions of sentence pairs from Wikipedia and randomly deleted some words in each sentence and replaced them with others. They then trained the model to predict the similarity of sentences. This is especially beneficial for evaluating how semantically accurate a model is as a pose to syntactically.

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [10] is an evaluation metric useful for comparing summaries. Summaries may be difficult to evaluate with metrics such as BLEU due to the amount of variance in how a quality response may manifest. ROUGE measures the amount of overlap between the language models output and the reference summaries in terms of word sequences, word pairs, and n-gram terms. There is different ROUGE metrics, namely ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S. ROUGE-N is calculated as

$$\frac{\sum_{S\in\{\text{ReferenceSummaries}\}}\sum_{\text{gram}_n\in S}\text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{S\in\{\text{ReferenceSummaries}\}}\sum_{\text{gram}_n\in S}\text{Count}(\text{gram}_n)}$$

Where n is the length of the n-gram, and $gram_n$ and $Count_{match}(gram_n)$ are the maximum number of n-grams co-occuring between a candidate summary and the references. However the statistical evaluation of the quality of a summary is not a simple task, and there are many subtleties that ROUGE-N does not account for. For this reason ROUGE-L provides a measure of the longest common subsequence, ROUGE-W is similar to ROUGE-L but rewards summaries preserving original word order, and ROUGE-S accounts for gaps between words in matching bi-grams.

Whilst some of the metrics used for open answer evaluation are necessarily quite involved, for closed answer questions it is still possible to use simpler metrics such as precision, recall and F1 score. Much of the research in the medical field, such as Google's Med-PaLM [24] makes use of multiple choice questions for evaluation. Whilst text to text language models do not output discrete values for multiple choice selection, exact match may be used, and that is what is used in this project. With appropriate prompt engineering, a language model can be instructed to consistently match one of the options. One of the reasons this is a particularly good metric for evaluation, and possibly why it is so popular is because it is very unambiguous. Each question is either correct or incorrect, there are no concerns around considering synonyms or other wording. Furthermore exact match is very simple to implement and does not require using libraries as is more convenient for more complex evaluation metrics.

Chapter 3

Below 1 Billion Parameters

In this section I look at how well a (relatively) small language model performs on a range of tasks, allowing us to better understand its characteristics. Different tasks allow us to learn about the different strengths and weaknesses of a model, and based on this we can analyse whether the model is suitable for the given task of domain specific question answering. I experiment with four different datasets evaluating the models performance on each and considering what each tells us, and then come to a conclusion.

3.1 Model Choice

I use Google's flan-t5-base model with 248M parameters as my initial model. FLAN-T5 is based on the original T5 model but is trained on more than 1000 additional tasks [17] making it better in almost every respect. I chose this model firstly because it is an autoregressive model, i.e. it is a sequence to sequence model which are designed for taking text as input and producing text as output, which is necessary for my objective to produce a chatbot for question answering. Secondly, the size of the model is ideal. It is not too large to the extent that it is too computationally expensive to train and evaluate, yet it is still large enough to be able to produce intelligent responses.

3.2 Evaluation Approach

I select four datasets of different tasks and evaluate the performance of the model. The evaluation metrics used in this section are BLEU, BLEURT, Exact Match, precision, recall, F1 and ROUGE. In this case, Exact Match calculates the fraction of times the model predicts the reference exactly, and F1 is the typical harmonic mean of precision and recall, calculated at the token level.

3.3 CoQA

3.3.1 Data Profile

The first dataset I use to evaluate the base model is Stanford's Conversational Question Answering (CoQA) Challenge dataset. This is a dataset comprising of a context story, and then a conversation about the context in the form of a series of questions and answers. The objective of the CoQA dataset is to measure the ability of a model to understand a piece of text and answer questions that appear in a conversation about it.

The graphs in figures 3.1 - 3.4 show the distribution of the length for the contexts, questions, and answers excluding the upper 0.1% outliers.

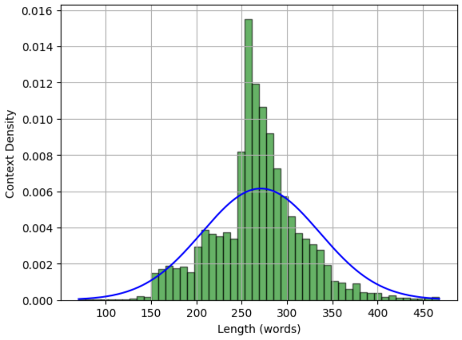


Figure 3.1: CoQA Context Lengths

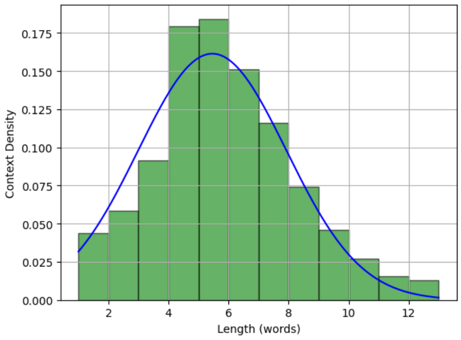


Figure 3.2: CoQA Question Lengths

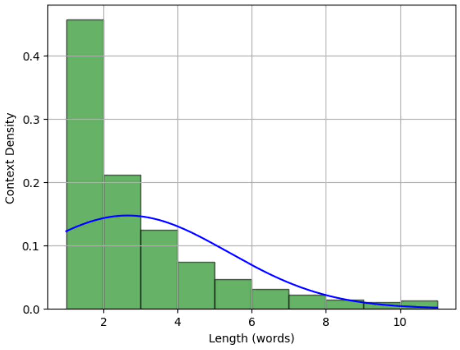


Figure 3.3: CoQA Answer Lengths

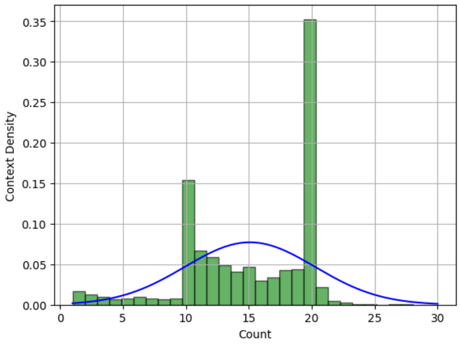


Figure 3.4: CoQA QA Pairs Per Context

We can see that the question and answers are relatively short, with answers frequently being just one or two word responses. Furthermore, the number of questions per context is generally few, as most samples are concise focused conversations. Consequently the comprehension required on the part of the question and answer is not particularly difficult. The main difficulty in the task is related to understanding the context for which the length is more substantial. The short questions and answers paired with a long context are beneficial for isolating the model's ability to comprehend long text. The following is a sample of some questions and answers about a context taken from the dataset:

```
who believe he has good fortune?
Wang Jiaming
where is he educated?
Beijing Chenjinglun High School
is he testing?
Yes
```

This small sample is taken directly from the dataset, and as is evident, there exist some minor grammatical mistakes. Nevertheless, the quality of the dataset otherwise is generally good and should not cause a problem for evaluation.

3.3.2 Evaluation

To process the data for evaluation, for each context story, I create a number of samples to use for evaluation by using each question and answer as an individual sample.

The train set consists of 7.2k items, which manifests as close to 100k training samples given the number of question answer pairs per context. The results can be seen in Table 3.1.

Table 3.1: CoQA Evaluation

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEURT	EM	precision	recall	F1
base	0.50	0.45	0.40	0.35	0.044	0.42	0.71	0.74	0.69
fine-tuned	**0.68**	**0.64**	**0.59**	**0.54**	**0.38**	**0.56**	**0.84**	**0.82**	**0.82**

It appears without fine-tuning, the model is capable of understanding context and correctly answering questions about it, however the only result that is significantly low is BLEURT at 0.044. After inspecting the responses, it seems the explanation for this is that whilst the model achieved many high scores for BLEURT, when it was wrong, it could be very far off, yielding very negative BLEURT scores. For example the model responds with ”farmer’s orange paint” as a pose to ”the farmer”.

Fine-tuning appears to have corrected this issue as we see the BLUERT score increases by 18% to a much more reasonable level, and fine-tuning also improved results across the board. The BLEU score demonstrates that the model is able to recall mostly correct wording from the context even when not matching exactly, and in most cases the output matches the reference exactly. It is fair to conclude from this that the model is capable of answering simple questions about a context, but we will also require it to have further capabilities, as explored with the next datasets.

3.4 SQuAD

The next dataset I use is Stanford’s Question Answering (SQuAD) [34] Challenge dataset. I use version 1 of this dataset which consists of a context, a question, and an answer where the answer is an excerpt from the context. For each context and question, I create a single sample by appending the question to the context with the answer as the label. Note what differentiates SQuAD to CoQA is that the answer is to be found in the context where as in CoQA, the model is expected to come up with the answer on its own. For this reason, it is expected that the model will perform better on SQuAD. The results of evaluating on SQuAD with and without fine-tuning can be seen in Table 3.2.

Table 3.2: SQuAD Evaluation

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEURT	EM	precision	recall	F1
base	0.80	0.80	0.74	0.61	0.78	0.76	0.90	0.74	0.89
fine-tuned	**0.85**	**0.82**	**0.75**	**0.62**	**0.80**	**0.80**	**0.92**	**0.90**	**0.90**

The performance on SQuAD is clearly much superior to that of CoQA, likely due to the fact that the task is now only to identify the answer from the context rather than produce it. Fine-tuning shows only a 4% increase in Exact Match score, however smaller improvements are expected at close to 100% Exact Match score. Human performance on the SQuAD dataset achieved an Exact Match score of 86.831 and an F1 score of 89.452. Our model scores similarly, only 6.8% behind on Exact Match and in fact only just outperforming on F1 score when considering our fine-tuned version. With an average context length of 120 words per context in the SQuAD dataset, we can confidently say the model is capable of reading comprehension within this range.

3.5 CNN and Daily Mail Dataset

The CNN and Daily Mail dataset [25] consists of articles from both news organization paired with summaries for the articles. Applying flan-t5 to this task was a simple case of prompting with the following structure

”Summarize the following: {Article}”

with the summary as the label. The results of evaluating on the CNN and Daily Mail Dataset with and without fine-tuning can be seen in Table 3.3.

Table 3.3: CNN and Daily Mail Evaluation

Model	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-LSUM	BLEURT
base	**0.31**	**0.16**	**0.26**	**0.26**	**-0.78**
fine-tuned	0.30	0.15	**0.26**	**0.26**	-0.85

The model performs poorly across the board, and discernibly fine-tuning had absolutely no positive impact on the model’s performance. As an example, here is one Flan-T5 summary compared to the reference summary:

Flan-T5:

```
"It was just like everyone has been saying: It was just like out of
the movies
```

Reference:

```
NEW: "I thought I was going to die," driver says . Man says pickup
truck was folded in half; he just has cut on face . Driver: "I
probably had a 30-, 35-foot free fall" Minnesota bridge collapsed
during rush hour Wednesday .
```

Seeing the summary comparison explicitly we can understand why the evaluation turned out so poorly. Flat-T5 doesn’t even correctly capture the subject matter. Without need for in depth scrutiny, we can undoubtedly draw that the Flan-T5 is not suited for summarisation.

3.6 MedMCQA

The Medicine Multiple-Choice Question Answering (MCQA) [4] dataset consists of over 194,000 exam questions taken from the AIIMS and NEET PG examinations.

Each question provides an explanatory answer as well as four possible short answers, usually one or two words. Some questions are multi-choice and some single, so for simplicity, I filtered the questions to only the single choice.

As T5 is a sequence to sequence model, in order to have it predict a multiple choice answer, I needed to structure the prompt in a manner that would standardize the model's outputs. I follow the example from Singhal et al. [17] in which they prepend the question with the following:

```
Instructions: The following is a multiple choice question about medical
knowledge. Solve it in a step-by-step fashion, starting by summarizing
the available information. Output a single option from the four options
as the final answer. Question:
```

This is followed by the question and each answer option prefixed with (A) (B) (C) and (D). This yielded about a 62% success rate for getting the model to output in the expected format. Table 3.4 shows the result of the evaluation on MedMCQA.

Table 3.4: MedMCQA Evaluation

Model	EM
base	0.16
random choice	0.28
fine-tuned	0.34
fine-tuned (extensive)	**0.37**

After evaluating on a sample of the validation dataset without any fine-tuning, it was clear that the model is guessing randomly. Given there are four options, random guessing should yield an Exact Match score of 0.25. Taking into account that only 62% of the answers were correctly formatted, we should expect an Exact Match of 0.25 x 0.62 = 0.155, which rounds to exactly the model's Exact Match score.

After fine-tuning on the training dataset, the model still performs poorly, but we see two improvements. Firstly, it seems the model has fully understood the format of the answer that should be produced, but secondly, the model has begun to show some sign of understanding of the questions as it now achieves an Exact Match score of 0.34. After conducting a binomial hypothesis test, we can be confident at the 5% level of significance that 0.34 is not due to chance, suggesting that the application of fine-tuning was successful in augmenting the model with knowledge it didn't have before.

3.7 Discussion

Based on the model's performance on the metrics used, we can arrive at some conclusions. Firstly, from the CNN and Daily Mail Dataset, we realise the model is certainly not powerful enough to perform a task such as summarizing to a good standard. From the CoQA dataset, it is evident that the model is capable of somewhat accurately answering closed questions about a context that it is provided with, and in fact when the model is only tasked with selecting an answer to a question as in SQuAD it performs much better. And finally, we see the model is capable of going from random choice performance, to a distinguishable level of understanding of medical related questions.

For a more illustrative demonstration of the model's capability, take the following prompt:

```
Q: Can Geoffrey Hinton have a conversation with George Washington? Give
the rationale before answering.
```

The model's response was:

```
George Washington was born in 1789. Geoffrey Hinton was born in 1789.
```

The model seems to have grasped that both persons year of birth is relevant, and there is some attempt at composing the two pieces of information, but it's factual knowledge is incorrect and it has failed to realize how to use both pieces of information in order to coordinate an answer.

While the model's performance on SQuAD was acceptable, our task of question answering on a specific domain is more complex. The task of question answering perhaps falls somewhere between the difficulty of the CoQA and the CNN and Daily Mail Dataset. For this reason, this concludes that the flan-t5-base model does not have sufficient capability to perform the task of domain specific question answering. Given the Flan-T5 architecture and pre-training is state of the art, the shortcoming here is due to the parameter count. This calls for exploring a larger model.

Chapter 4

Beyond 1 Billion Parameters

While Google's Flan-T5 model was not powerful enough for the projects requirements, Meta conveniently released Llama 3 [16] just a few weeks ago from the time of writing. Llama 3 was trained on over 15 trillion tokens. Both 8 billion and 70 billion parameter models were released, and it has demonstrated especially impressive performance significantly outperforming Google's Gemma [41] and Mistral AI's [3] similar size 7 billion parameter models on a number of tasks. While Llama 3 70B is too large to train cost effectively, with some adjustment, Llama 3 8B can be trained rather cost effectively within a Google Colab notebook. For these reasons, I continue my experiments using this model. The initial version of the model I will use will be the ”unsloth/llama-3-8b-Instruct-bnb-4bit” model from Hugging Face. This is a quantised 4 bit version of the model making it trainable with less memory.

4.1 Performance Criteria

The domain of Medicine is larger than the size of domain intended by this project. For that reason, it is wise to choose a sub domain within Medicine. An attractive choice is the Anatomy domain. This is because it is a very much knowledge focused subject, and after some research it appears the Anatomy knowledge for medical school examinations is usually limited to a few hundred pages by the standard of a number of texts books on the subject [14] [2]. For this reason, Anatomy will be used as a case study domain for this project.

There is benefit in using more than one subdomain such as cross validation and reliability of results. However due to the length of time and cost required to run inference, there becomes a trade off between the number of subdomains explored and how much experimentation is possible for each subdomain. This project will focus on only one subdomain allowing for suitable analysis depth with the given constraints, and future work may build on this with other subdomains, allowing for comparison and cross validation.

In the medical domain, the predominant method for performance evaluation is multiple choice questions. This is ideal for medical questions given Medicine is very much a knowledge based field, and it is also suitable for the purpose of creating a domain specific knowledge chatbot, for which knowledge is important, whilst also being a key focus of this project.

As most research targets the Medicine field in general as a pose to specifically Anatomy, there are only a couple evaluation datasets specifically on the subject of Anatomy. There exists MMLU-Anatomy, as well as MedMCQA, a generic Medicine dataset with an Anatomy split. I will prioritise these two datasets for Anatomy centered evaluation, and also include some generic Medicine datasets for comparison to existing research. Each dataset comprises of 4 option multiple choice questions, except for PubMedQA which comprises yes / no questions. The count was chosen on the basis of the counts used in the Google Med-PaLM-2 paper [23] whilst also trying to maintain a consistent count per dataset, yet MMLU-Anatomy differs as there exist no more than 135 samples in the dataset. The datasets I have selected are summarised in Table 4.1 and an overview of the question and answer lengths are shown in Figures 4.1 - 4.8.

Table 4.1: Multiple-choice question evaluation datasets.

Dataset	Count	Description
MedMCQA	234	Indian medical entrance exams questions [4]
MedMCQA-Anatomy	234	Indian medical entrance exam questions (Anatomy only) [4]
MedQA (USMLE)	234	General medical knowledge from US medical licensing exam [11]
PubMedQA	234	Provided PubMed abstract Closed-domain question answering [35]
MMLU-Anatomy	135	Anatomy multiple-choice questions [8]

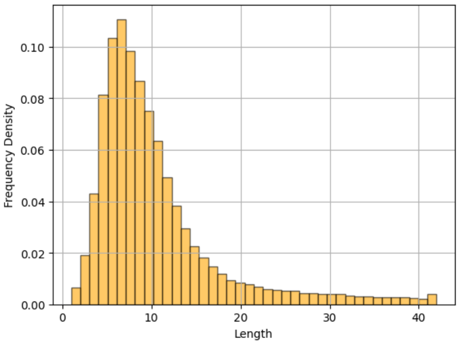


Figure 4.1: MedMCQA Question Lengths

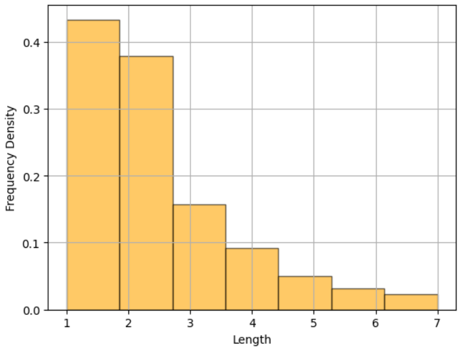


Figure 4.2: MedMCQA Answer Lengths

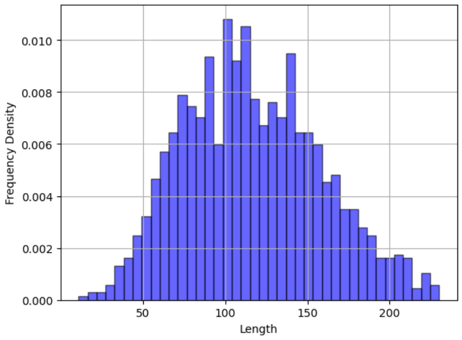


Figure 4.3: MedQA Question Lengths

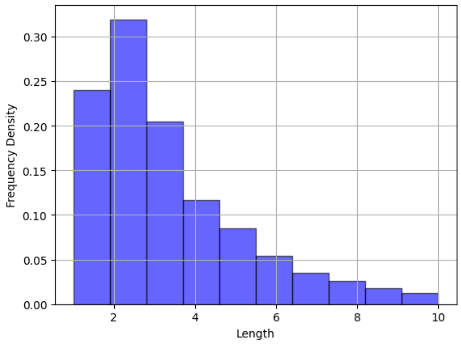


Figure 4.4: MedQA Answer Lengths

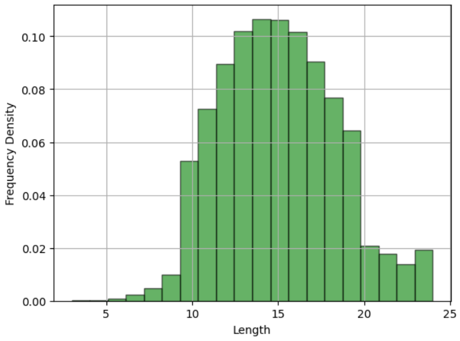


Figure 4.5: PubMedQA Question Lengths

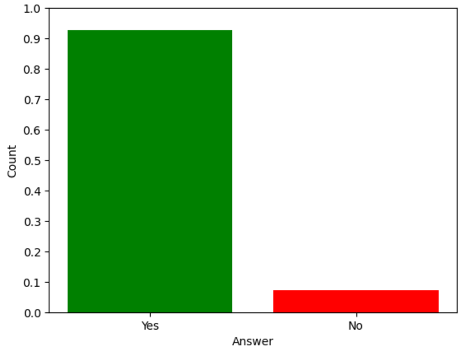


Figure 4.6: PubMedQA Yes No Count

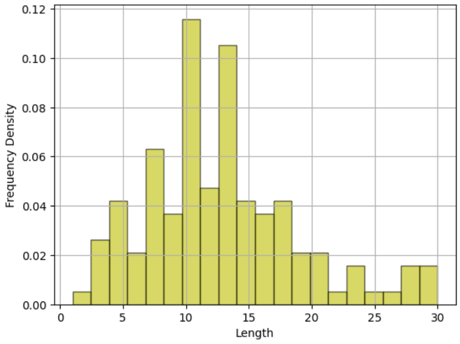


Figure 4.7: MMLU Question Lengths

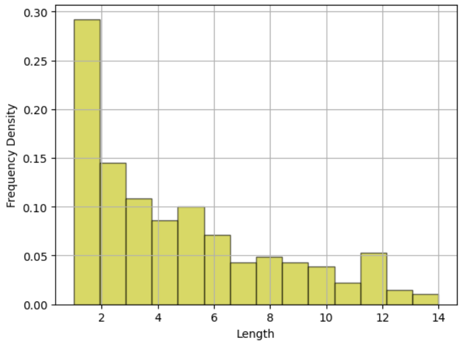


Figure 4.8: MMLU Answer Lengths

The questions for each dataset are generally short, except for MedQA who's question lengths are much more pronounced. The reason for this is because the questions each introduce the context of a patient before asking a related question. Consequently, the MedQA dataset gives an insight into the model's comprehension abilities, and there is no elaboration expected in the response, for each of the multiple choice questions are most commonly 1-gram or 2-gram terms. Notably, the balance of yes to no in the answers for PubMedQA is heavily skewed towards yes. This implies that fine-tuning on PubMedQA data may not be wise, as the model may simply learn this bias. However for other techniques such as RAG, this does not cause an issue as the model has no prior knowledge of the distribution of the answers.

4.2 Initial Performance

The initial performance of Llama 3 8B on each dataset is shown in Table 4.2. The metric used is the exact match score and for reference, the performance of Google's state of the art 340B parameter Med-PaLM 2 model from 2023 is shown in comparison. The model's performance is reasonable given the large difference in parameter size and given that this is prior to any tailoring to the medical domain.

Table 4.2: Initial Benchmarks

Dataset	Llama 3 8B base	Med-PaLM 2
MedMCQA	0.462	**0.723**
MedMCQA-Anatomy	**0.534**	-
MedQA (USMLE)	0.513	**0.865**
PubMedQA	0.722	**0.818**
MMLU-Anatomy	0.474	**0.844**

Notice the MedMCQA dataset limited to only Anatomy yields some 7% increase in performance. This could perhaps be due to the factual based nature of the Anatomy domain. It is also interesting that the difference in performance on PubMedQA is much less distinct than the other datasets. This could be because given that PubMedQA is a yes no task, there is more opportunity for the models to get the answer correct due to random choice.

4.3 Strategy

To improve the model, there is a number of options that were discussed in the background section, each with different expected benefits and time expectations. Generally, with language models there are four architectural patterns for tailoring to a specific task, each of different complexity and computational requirement as summarized in Figure 4.9. Whilst Figure 4.9 summarises the different techniques at disposal at a high level, there is a variety of ways to apply each technique.

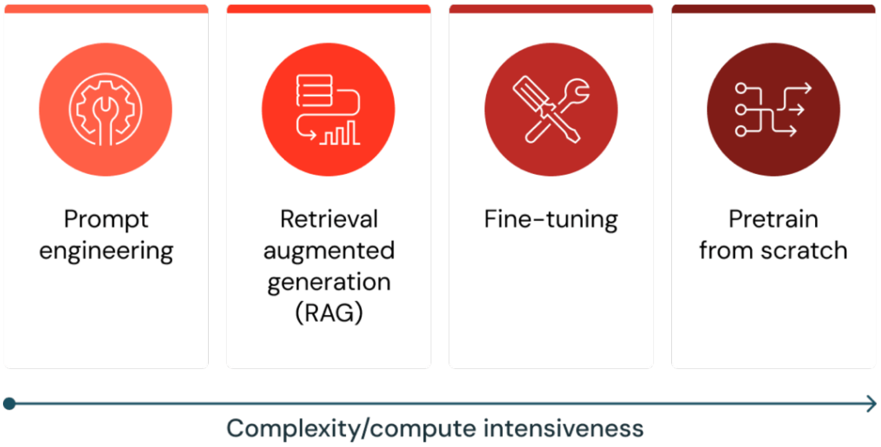

Prompt engineering
Retrieval augmented generation (RAG)
Fine-tuning
Pretrain from scratch
Complexity/compute intensiveness

Figure 4.9: Databricks LLM Architectural Patterns

Prompt engineering can be as simple as modifying the instruction in the prompt to be more clear, and this simple change can significantly improve results. However there exist more advanced techniques as discussed in the background such as Few-Shot [43] in which the prompt is supplemented with examples, and chain of thought [19] which supplements the prompt with a number of reasoning steps. With little effort and carefully crafted prompts, such methods have shown to yield significant performance improvement [6]. Before attempting any more involved methods, it is in our time, cost, and performance based interests to explore these routes first.

There are also various methods for fine-tuning models such as Direct Preference Optimisation [36] allowing stable performant fine-tuning without the need for significant hyper parameter-tuning, and for larger models where updating all parameters is more computationally expensive, techniques such as LoRA [12] and QLoRA [42] can be applied to reduce the amount of parameters that need to be updated which in turn reduces computation and memory requirements.

Aside from the aforementioned methods mentioned, there are many other tricks and quirks at our disposal that can be applied to help improve performance, and we may selectively explore these.

4.4 Prompt Engineering

The main objective I see prompt engineering achieving for multiple choice selection is preventing wrong predictions due to mismatch. Subtle changes in how the prompt is structured can throw the model off and cause it to get an answer wrong due to formatting where to a human reader, it would be obvious that the model selected the correct answer.

4.4.1 Rephrasing

The number of responses from the base results that failed due to mismatch are shown in Table 4.3. My initial prompt, the same prompt used to obtain the base results, was of the following structure:

```
{Question} {Option A} {Option B} {Option C} {Option D} Respond with
the correct choice from the list above.  Do not include any explanation.
```

Table 4.3: Mismatch by Dataset

Dataset	Mismatch
MedMCQA	0.137
MedMCQA-Anatomy	0.081
MedQA (USMLE)	0.128
PubMedQA	0.000
MMLU-Anatomy	0.215

A couple alternatives attempted were:

```
Respond with only the answer term from the provided options and do
not provide any explanation.
```

```
Only one of the provided options is correct. State the correct answer
from the provided options.
```

These changes didn't have significant impact. In some cases the score was brought down to 0. This was to be expected in cases where a nuance in the prompt caused the model to consistently incorporate a misformatting, such as prepending the answer with the letter label of the correct answer. However as long as the model understood the output format, switching between these prompts had no impact on performance other than fluctuations of some 1% in the positive or negative for different tasks.

I did however find one subtle change that yielded substantial benefit. I simply added the word ”verbatim” to the initial prompt keeping everything else the same:

```
{Question} {Option A} {Option B} {Option C} {Option D} Respond with the
correct choice from the list above verbatim. Do not include any
explanation.
```

This resulted in improvements of 3-4% on MedMCQA, MedMCQA-Anatomy and MedQA. Other terms, ’exactly’ and ’word for word’, were not able to achieve any better improvement. This however is interesting as it suggests a prompt doesn't necessarily need to be very elaborate, but carefully chosen wording is especially important. What this implies is that there is a high level volatility in the semantic meaning of a sentence provided to a language model unlike when speaking with humans. As is also observed later in the RAG section, irrelevant information is consequential for language models, and this could also be due to high attention being given to irrelevant words.

4.4.2 Re-prompting

In a further attempt to correct mismatches, I cross checked the model's output against the 4 options and if the answer did not match any of them, I prompted the model to try again. The re-prompt also required some adjustment for example to prevent the model from apologizing or including dialogue other than the answer, nonetheless after some adjustments, the resulting prompt was of the following form:

```
Your response does not exactly match one of the choices from the list.
Do not apologise or include any text other than one of the options
from the list verbatim without any label. Here are the options gain
```

Re-prompting a single time was successful in reducing the proportion of mismatch. It seems if the initial prompt was too ambiguous, after trying and failing, the model could use together the prompt, and the refusal of its previous attempt as guidance for the second attempt. This encouraged taking this technique further. The technique was applied as a loop, i.e. re-prompting the model indefinitely until it responded with one of the options, but in some cases the model would continue to respond with a mismatch. Of course it is possible to set a threshold, but in this case the difference between one re-prompt and more than that did not justify the overhead.

4.4.3 System Prompt

An alternative method that perhaps could be expected to provide more significant benefit is a system prompt. This is a short statement that instructs the model on its role so that it interprets the prompt from the perspective of that role. System prompts are simply pre-pended to the prompt, using special tokens to identify it, for example when using Llama 3, system prompts are provided in the prompt following special tokens $< |start\_header\_id| > assistant < |end\_header\_id| >$. The following are some of the system prompts[1] that were attempted:

[1] The final prompt should read "multiple choice question", however this grammatical mistake was made during the experiment, therefore it is left as it was used for integrity.

```
You are a medical professional. Answer the question truthfully.

You are tasked with answering multiple choice questions about Medicine.

You are a multiple choice answer selector. When prompted with a
multiple question, you respond by stating the correct option verbatim
with no other text.
```

It seemed the model was not reacting much at all to the system prompts. Perhaps a reason for this is that the task is well defined, and system prompts are more beneficial when the specification of the task is less clear. It may also be that the instruction is already well specified within the user prompt, and the system prompt is just regurgitating the user prompt, therefore the system prompt is not providing any useful information.

4.4.4 Few-Shot

Next I attempted Few-Shot [43] using 1, 3 and 5 examples in the prompt. This required leveraging Llama's prompt template to create an artificial history of conversation in which the model is prompted and responds correctly to a series of questions. For brevity, an instance of using 1 example is shown here, whilst 3 and 5 are shown in section 1 of the appendix:

```
<|start_header_id|>user<|end_header_id|>

Chronic urethral obstruction due to benign prismatic hyperplasia can
lead to the following change in kidney parenchyma
Hyperplasia
Hyperophy
Atrophy
Dyplasia<|eot_id|><|start_header_id|>assistant<|end_header_id|>

Atrophy<|eot_id|><|start_header_id|>user<|end_header_id|>
```

This also caused the model to achieve a score of 0. Upon reviewing the responses, it appears the model is still able to identify and understand the question, but Few-Shot confuses the model on the format of the output. The reason Few-Shot was not of benefit may be similar to that of system prompting. It may be that these techniques are most impact when the task requires understanding a more involved instruction, which is not the case for multiple choice selection.

Few-Shot for each of 1, 3 and 5 examples performed worse than Zero-Shot. It appears the model mismatch is just as low which indicates Few-Shot is at least not worsening the output format, however Few-Shot makes more mistakes. The reasoning for this is very likely that Few-Shot is instructive in clarifying the task, but it is not instructive in performing the task. This is because the examples provided are not selected by their semantic similarity as in RAG, they are simply example questions of the upcoming task, so they do not provide any information to help answer the question. For this reason, since mismatch is already low, there becomes little for Few-Shot to do.

The updated performance after prompt engineering taking the best of rephrasing, re-prompting, system prompting and Few-Shot, is shown in Table 4.4, as well as Mismatch in Table 4.5.

Table 4.4: Prompt Engineering Benchmarks

Dataset	Llama 3 8B base	Llama 3 8B (PE)
MedMCQA	0.462	**0.517** (5.5%)
MedMCQA-Anatomy	0.534	**0.585** (5.1%)
MedQA (USMLE)	0.513	**0.573** (6.0%)
PubMedQA	**0.722**	**0.722** (0.0%)
MMLU-Anatomy	0.474	**0.578** (10.4%)

Table 4.5: Prompt Engineering Mismatch

Dataset	Llama 3 8B	Llama 3 8B (PE)
MedMCQA	0.137	0.060
MedMCQA-Anatomy	0.081	0.017
MedQA (USMLE)	0.128	0.009
PubMedQA	0.000	0.000
MMLU-Anatomy	0.215	0.037

4.4.5 Prompt Engineering Review

In conclusion, optimizing prompts is crucial for exploiting a language models abilities. Without it, a model could indeed be capable of a task but its ability to perform the task does not manifest due to poor instruction. However it is also the case that past providing a clear instruction, further prompt engineering was of no particular benefit. This is understandable in this case given the simplicity of the task. The difficulty is not in understanding the question, but in knowing the answer, and after providing a prompt that was clear enough for the model to output it's response in the correct format, attempting anything further would only cause confusion and corrupt the output structure.

4.5 RAG

This section explores making use of a vector database to retrieve relevant information to help the model in responding to the question. The Pinecone database is used for storing vectors, which is a popular cloud service managing vector storage, retrieval and similarity search.

4.5.1 Question Answer Knowledge Base (RAG-1)

The first method attempted was creating a knowledge base using only the train set of MedMCQA. One option was to simply concatenate the question and answer, however as has been demonstrated in Wei et al. [19], providing the reasoning for an answer can be especially helpful, so rather than using the correct multiple choice option, the answer explanation field of the MedMCQA dataset is concatenated. This experiment is referred to as RAG-1, and after running the experiment with and without the prompt engineering (PE) strategy from the previous section, vastly different results were observed. These can be seen in Table 4.6.

Table 4.6: RAG-1 Benchmarks

Dataset	Llama 3 8B base	Llama 3 8B (PE)	Llama 3 8B (RAG-1)	Llama 3 8B (PE + RAG-1)
MedMCQA	0.462	0.517 (5.5%)	0.274 (-18.8%)	**0.585** (12.3%)
MedMCQA-Anatomy	0.534	0.585 (5.1%)	0.303 (-23.1%)	**0.632** (9.8%)
MedQA (USMLE)	0.513	0.573 (6.0%)	0.359 (-15.4%)	**0.581** (6.8%)
PubMedQA	**0.722**	**0.722** (0.0%)	0.427 (-29.5%)	0.436 (-28.6%)
MMLU-Anatomy	0.474	**0.578** (10.4%)	0.296 (-17.8%)	**0.578** (10.4%)

What was certainly unexpected was the negative impact on performance without prompt engineering. The mismatch was the main issue, being more than 50% of samples in some cases. After printing the prompt along with the outputs and manually inspecting, the problem appeared to be that the retrieved context was misleading the model from the correct output format as it seemed to loose focus on the instruction to only respond with the correct answer, and focused instead on copying the format of the retrieved context.

For example, the RAG knowledge base consists of a question, followed by four options, with each option prefixed by a letter identifier, followed by the explanation of the correct answer. In some cases, the model understood that it is expected to output the answer label, or the correct answer prefixed by a label.

Applying RAG-1 (comparing PE + RAG-1 with PE), the performance for MedMCQA, MedMCQA-Anatomy and MedQA each improved by up to 7%, yet PubMedQA suffered significantly dropping down almost 30%. This is possibly due to the fact that PubMedQA is the only yes no benchmark whilst the retrieved question answer documents are multiple choice. To understand the remaining results further, the questions that became correct, incorrect, and the questions that were unchanged due to RAG, were recorded for MedMCQA. These can be seen in Figure 4.10.

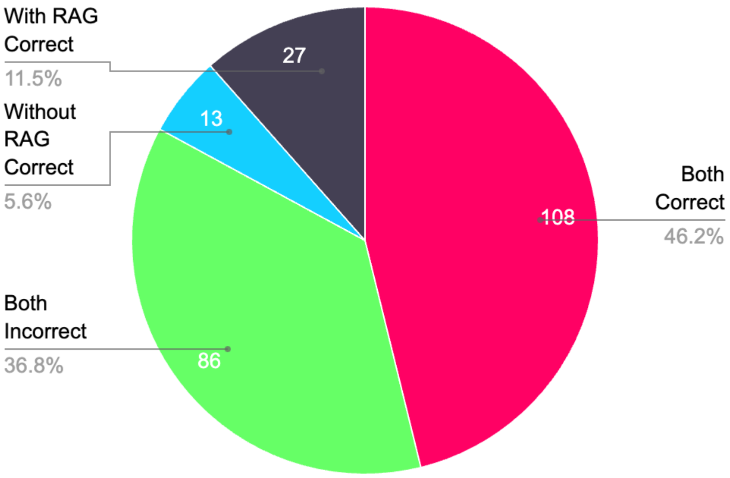


Figure 4.10: RAG-1 on MedMCQA [5]

The breakdown shows that whilst RAG did succeed in aiding the correct answering of 11.5% of the questions that would have otherwise been answered incorrectly, there is a significant 5.6% of questions that are now being answered incorrectly due to the introduction of RAG. This was a peculiar observation, and the next experiment discovers a potential reason as to what could be causing this.

Dataset Anomalies

Whilst assessing some of the prompts along with the model's answers, there were some anomalous cases. One of which was the following:

```
Buccinator is pierced by all of the following except:
```

```
Labial branch of facial nerve
Buccal branch of mandibular nerve
Parotid duct
Molar mucous glands

Respond with the correct choice from the list above verbatim. Do not
include any explanation. You may use the following information only
if it is helpful:
All of the following structures pierce the buccinator muscle except -
(A) Parotid duct, (B) Molar glands of the cheek (C) Buccal branch of
facial nerve (D) Buccal branch of the mandibular nerve. Answer with
explanation: Structures piercing the buccinator are :-


Parotid duct
Buccal branch of mandibular nerve
Four or five molor mucous glands lying on the bucco-pharyngeal
fascia around the parotid duct.
```

RAG Answer:

```
Labial branch of facial nerve
```

Correct Answer:

```
Buccal branch of mandibular nerve
```

Looking at the question the model is tasked with answering and the question retrieved as context, both appear to be asking about the same thing. This is a positive sign indicating the knowledge base contains a wide enough variety of questions to enable it to retrieve relevant context, however it seemed in this case, the model wasn't to blame. Comparing the original question with the retrieved context question, it is clear that they contradict each other giving different answers to the same question.

After the consultation of a Medicine student, it looks as though it is not unusual to come across contradictory knowledge for some niche subjects in Medicine. This perhaps attributes some weakness to the MedMCQA dataset, and also the knowledge base which consists of questions from the MedMCQA dataset. Fortunately these cases are not many, but counting or removing them would require an expert on Medicine to validate thousands of medical questions. Given this, it must be accepted as a weakness to the dataset, but this remains a realisation to be taken forward.

4.5.2 Textbook Knowledge Base (RAG-2)

The next knowledge base applied was an Anatomy text book. This approach will be referred to as RAG-2. A text book as a source of knowledge is especially ideal as it is a source of knowledge designed specifically for the purpose of teaching knowledge about Anatomy. The text book used here is Netter's Clinical Anatomy, Fourth Edition [14].

One issue is that with Anatomy being a very visual subject, a consequence of this is that much of the text is referring to diagrams in the text book. Neither is it consistently arranged in an ideal structure. It was decided to accept that this will mean some documents in the knowledge base will be irrelevant, but this is also mitigated by the similarity search which should effectively filter out such documents from the knowledge base by not selecting them.

A second problem is that by parsing a PDF of the text book into documents for a knowledge base using code, there is no way of ensuring the split points to create documents are at semantically meaningful points. Another issue is that some information may be lost if a sentence is split across two documents.

To prevent loosing information at split points in the text, 100 word documents are created with overlap of 50 words. I.e. words 0 - 100 form the first document, and words 50 - 150 form the second document and so on. To deal with the context of retrieved documents not starting or ending at semantically meaningful points, the documents were fed to the GPT-3.5 Turbo API for cleaning. GPT was provided with each document with the following prompt prepended:

```
The following is a piece of text scraped from an Anatomy text book.
It may be containing missing information and start and end in the
middle of a sentence. If the text speaks about a figure or an image,
try to instead explain what it is mentioning in a way that does not
require access to the image. Respond with a clean form of the
information suitable for use as a document in a RAG Anatomy knowledge
base:
```

The entire book was parsed using the aforementioned method for 4,440 documents and uploaded them to the Pinecone vector database. The model performance was then evaluated retrieving from this knowledge base. This approach was much less successful than RAG-1. There was little performance improvement, and in fact some performance reduction on some metrics.

In an attempt to discover the reasoning behind this, the responses with and without the RAG-2 approach, were written to a text file. After manually inspecting in which cases RAG-2 answered the question correctly whilst not using RAG-2 didn't, and vice versa, it appeared that while the retrieved information was indeed beneficial to some questions, it was in fact detrimental to others.

In some cases, RAG-2 helped retrieve documents that clearly addressed the question. For example, the following is a question where RAG-2 helped the model arrive at the correct answer where it didn't answer correctly before:

```
Which of the following is unpaired bone of facial skeleton:

Nasal
Lacrimal
Inferior nasal concha
Vomer
```

Retrieved document:

```
...The vomer bone is unpaired and contributes to the septum...
```

Correct RAG answer:

```
Vomer
```

The relevant part of the context is shown, in which it can be seen the context clearly addresses the question assisting the model in arriving at the correct answer.

However in other cases, RAG-2 would retrieve a document comprising of correct information, however alluding to one of the other options. For example, the following is a question where RAG-2 caused the model to answer incorrectly, where it answered correctly before

```
Root value of cremaster reflex is

L1, L2
L2, L3
S1, S2
S3, S4
```

Retrieved document:

```
C3, C4, C5, C6, C7, C8, T1, T2, L5, L4, S1, L2, L3, S2, S3 refer to
different levels of the spinal cord....
```

Correct answer:

```
L1, L2
```

RAG answer:

```
L2, L3
```

Notice that the retrieved document is relevant to the question, but it appears to have thrown the model off as it mentions only L2 and L3, which compose an incorrect option, but it doesn't mention L1. Whilst the document isn't necessarily wrong, it is misleading to the model.

Therefore RAG-2 was not a successful experiment, but from it, we can hypothesise about the necessary the characteristics of retrieved information. From inspection, it seems that documents need to directly address the question, and that information in the retrieved document that doesn't directly address the question, even if not incorrect, can in fact be harmful to the model. Setty et al. [38] mentions the issue of irrelevant information retrieval, however, what is considered irrelevant may be ambiguous. Testing this hypothesis could involve the manual inspection of many retrieved documents by human reviewers, to score each document's relevance, and then measure whether this correlates with the evaluation score. The negative impact of irrelevant information could also explain why retrieving entire questions in the RAG-1 experiment would cause the model to answer incorrectly. For this reason, it may be more beneficial to provide the model with shorter documents. In this way, it will be easier to avoid retrieving irrelevant information, as the retrieval process will be more fine grained. This leads to the next RAG method, providing definitions.

4.5.3 Providing Definitions (RAG-3)

Providing definitions will be referred to as RAG-3. We can investigate the hypothesis from the RAG-2 experiment by creating a knowledge base consisting of definitions for each of the 1-gram answer options, and observing whether there is any change. To do so, we should first record the performance of each n-gram individually. Since the RAG-1 experiment was the most successful, RAG-3 is be applied on top of RAG-1. Table 4.7 takes the results from the RAG-1 experiment and shows the results split by n-gram.

Table 4.7: n-gram

Dataset	1-gram	2-gram	3-gram	4-gram	5-gram
MedMCQA	0.556	**0.619**	0.560	0.500	0.538
MedMCQA-Anatomy	0.686	0.621	**0.759**	0.458	0.625
MedQA (USMLE)	0.595	**0.634**	0.483	0.625	0.529
MMLU-Anatomy	**0.909**	0.591	0.333	0.444	0.400

The MedMCQA Anatomy benchmark is used to investigate the benefit this technique before applying it to the other datasets. The dataset was parsed to extract all the answer options, GPT-3.5 was used to generate definitions, and then these were stored as a dictionary in a JSON file. Definitions were generated using the following prompt:

```
Provide a short definition for the following medical term:
```

The first time this experiment was ran, there was some noise due to the dictionary having picked up words like ”None” or ”All” and provided definitions for these words. The dictionary was manually edited to remove such instances. However, even after running the experiment with the refined dictionary, this RAG technique was causing more questions to be answered incorrectly than correctly.

This was unexpected since previous experiments had suggested shorter more relevant information would be of benefit. In attempt to understand this better, example prompts and responses were inspected in the cases where providing definitions produced an incorrect response to questions that were previously being answered correctly without definitions. What was found to be one potential problem was that often the dictionary consisted of definitions for only a subset of the options, and this seemed to be leading to bias. After considering only the options where definitions for all four options were provided, it transpired that there was in fact no difference in the model’s answers. However using a different prompt, there was a discrepancy found in only one question, which was the following:

```
If there is absence of precursor cell of an organ with the subsequent
non development of the organ, what is the condition is called as?

Agenesis
Aplasia
Atresia
Atrophy

Respond with the correct choice from the list above verbatim. Do not
include any explanation. You may use the following definitions only
if they are helpful:

Agenesis: Agenesis is a developmental disorder characterized by the
absence or underdevelopment of a body part or organ.

Aplasia: Aplasia is the absence or underdevelopment of an organ or
tissue in the body.

Atresia: Atresia refers to a condition where a normal opening or
passage in the body is blocked or absent.

Atrophy: Atrophy is the wasting away or reduction in size of a bodily
```

```
tissue or organ due to lack of use, disease, or injury.
```

Correct Answer:

```
Agenesis
```

Answer with definitions provided:

```
Aplasia
```

The definitions provided were checked to be correct, however the issue seems to be down to a very subtle nuance in the definition of two of the answer options. As was found before, for some niche subjects of medicine, there is a lack of clear definition. This being the second case of a question affected by ambiguous definition suggests that this issue is somewhat significant.

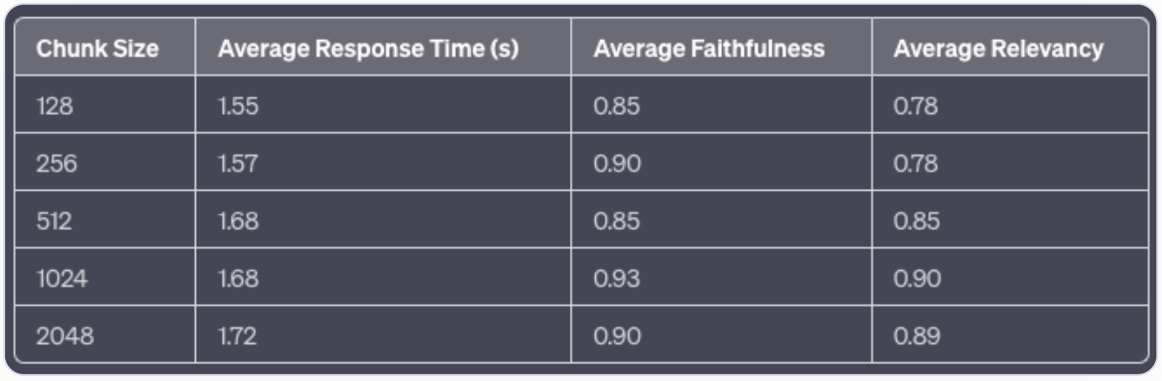
Based on the RAG-3 experiment, it seems providing definitions was of negligible benefit or detriment, suggesting the model already has a good enough understanding of complex Anatomy terms, and considering the other experiments, the retrieved information needs to be informative with respect to the question in order to be of any assistance to the model. This leaves the best performing method being to provide full question and answer pairs, knowing that whilst in most cases question answer pairs help the model answer the question, it still suffers the risk of misleading the model due to occasionally including irrelevant information.

4.5.4 LlamaIndex (RAG-4)

Whilst the Anatomy text book (RAG-2) did not outperform the question knowledge base (RAG-1), it was left to to see the effect of making variations to the knowledge base structure. Note that throughout this section, the number of experiments had to be limited due to API costs, so some experiments that would have been ideal to have for reference and comparison been left out.

LlamaIndex is a tool that streamlines much of the process for creating and using RAG knowledge bases. LlamaIndex is used to create multiple knowledge bases of Anatomy text book information with different configurations. For the generation model, GPT-3.5-Turbo is used. With the cost of accessing the OpenAI API and the number of experiments needed to be carried out, this experiment is limited to the MedMCQA-Anatomy benchmark, which is suitable given the Anatomy knowledge base, and to further reduce the number of experiments, all experiments are conducted using the same prompt engineering strategy found to be best in the Prompt Engineering section.

LlamaIndex defines a ’chunk’ unit which it uses to split documents. Figure 4.11 shows the results of an experiment conducted by LlamaIndex for the best performing chunk size for each document



Chunk Size	Average Response Time (s)	Average Faithfulness	Average Relevancy
128	1.55	0.85	0.78
256	1.57	0.90	0.78
512	1.68	0.85	0.85
1024	1.68	0.93	0.90
2048	1.72	0.90	0.89

Figure 4.11: Performance by Chunk Size LlamaIndex [40]

It is evident that chunk size is significant, and perhaps knowledge bases of different domains are more tailored to different size chunks.

With this, the variations made to the knowledge base structure are the following: the length of each document, the amount of overlap between consecutive documents, and the number of documents used to create the knowledge base. It was envisioned that each of these may have different affects. Setty et al. [38] mentions, if an answer spans over a few different sections in a document, it may not have the most semantically similar chunks, and a retriever may not retrieve the most ideal chunk. More overlap may help combat this as it enables the retriever to select information in a more fine grained manner. The benefit of longer documents is that they provide more context but potentially more irrelevant information. Using multiple documents could have benefits of allowing us to see whether different books covering the same information can help fill gaps, for example where one book may have text referring to a diagram, another may cover that same information as a pure text passage. Furthermore it may prove beneficial to include different perspectives on the same information in the knowledge base.

The text books used for this experiment were the following: Netter’s Clinical Anatomy [14], Grant’s Atlas of Anatomy - Anne M R Agur and Arthur F. Dalley [2], Cunningham’s Manual of Practical Anatomy: Upper and Lower Limbs [27], Clinical Anatomy [13], Human Anatomy and Physiology [30].

Table 4.8: Anatomy Text Books

Book	Pages	Author	Application	Comments
Netter's Clinical Anatomy	588	John T. Hansen	Medicine Students	Good general coverage of Anatomy. Focuses on basics. Includes USMLE style questions.
Grant's Atlas of Anatomy	871	Anne M R Agur and Arthur F. Dalley	Medical Students	Popular text book, renowned for pedagogy, however is an Atlas, therefore includes many figures.
Cunningham's Manual of Practical Anatomy: Upper and Lower Limbs	312	Rachel Koshi	Medical and Surgical Students	Simplified language and relevant, however focuses is on upper and lower limbs.
Clinical Anatomy	439	Harold Ellis	Students and Junior Doctors	Popular text book with good generality, and useful separation between figures and text.
Human Anatomy and Physiology	418	Nega Assefa and Yosief Tsige	Nursing Students	Aimed towards Nursing Students rather than Medical students.

Aside from changing the knowledge base, the effects of the parameter k was also considered, which dictates the number of documents fetched as context. As there are now four parameters, the number of combinations are many, so with time and computational cost in mind, the combination of parameters are chosen selectively for each experiment.

I decided to first consider the impact of the number of books used for the knowledge base. I used the default chunk size of 1024, chunk overlap of 200 and the default k value of 2 for retrieval. I assumed that the impact of using different books would not be depended on the other parameters, so I did not vary any other parameters for this experiment. Table 4.8 shows the results of using one to five different books as a knowledge base.

Table 4.9: Performance by Number of Books

Number of Books	MedMCQA-Anatomy
1 book	**0.590**
2 books	0.560
3 books	0.521
4 books	0.560
5 books	0.543

Netter's Clinical Anatomy is used as the knowledge base of 1 book. It seems from the results that providing the same information from different perspectives was of little benefit as using a single book yielded the best result, and in fact the pattern looks to be that the more books, the worse the performance. Since the knowledge base consisting of all 5 books contains each other knowledge base as a strict subset, the problem cannot be a lack of information, rather the retrieved content is no longer as good quality.

This finding is interesting, because it would be expected that the performance with multiple books could not possibly be worse than one book, because the the knowledge base from the 1 book experiment is a subset of the knowledge base in each of the other experiments. What this suggests is that increasing the documents in the knowledge base doesn't necessarily improve retrieval, rather the more documents in the knowledge base, the less accurate the retriever becomes. Perhaps this means that a requirement of the knowledge base is that it must be clean, i.e. not only must the knowledge base consist of high quality documents, but specifically *only* high quality documents, with the absence of poor quality documents as a requirement in itself, as this experiment has shown that supplementing a knowledge base with further documents whist preserving the original documents can be detrimental to the knowledge base.

As it is expected that the effect of the number of books is likely be independent of other parameters, the remaining experiments will be conducted with a single book.

To analyse the impact of chunk overlap, chunk size is kept constant at the default value 1024, and then overlaps are applied from size 25, increasing by a factor of 2 up to 400. The results of this experiment are shown in Table 4.9.

Table 4.10: Performance by Chunk Overlap

Chunk Overlap	MedMCQA-Anatomy
25	0.581
50	0.581
100	0.564
200	0.598
400	0.585

The Chunk Overlap doesn't show any particular pattern. The worst performing was actually the median size of 100, and either side of that, the performance increased. The range between the smallest and largest value is 0.034, suggesting the influence of overlap is little. Perhaps this is due to the chunk size of 1024 being large enough so that the majority of the documents content is self contained. It is expected that within an Anatomy text book, much of the knowledge will be short facts and definitions, so with little overlap, the majority of sentences manage to be self contained within a single document.

Next the impact of varying the chunk size is investigated. Knowledge bases are created using chunk sizes 128, 256, 512, 1024, and 2048, the same chunk sizes used in the LlamaIndex experiment. *k* is kept fixed at 2, but for chunk overlap, it is the proportion that needs to be maintain. Therefore for chunk size 1024, the default chunk overlap of 200 is used, as well as using overlap 25, 50, 100, and 400 fore each of the other chunk sizes respectively, in order to maintain a constant ratio between chunk size and overlap. Table 4.10 shows the results of this experiment.

Table 4.11: Performance by Chunk Size

Chunk Size	MedMCQA-Anatomy
128	0.585
256	0.575
512	0.568
1024	0.590
2048	0.598

These results are inconsistent with those produced by LlamaIndex in finding that chunk size 2048 was the best. However it is possible that this is simply due to random error due to the difference being very subtle. The range of the values is 0.03, similar to chunk overlap. Based on the little impact made by chunk size and chunk overlap, it is presumed that either the questions for which the book contains relevant information are already being answered correctly, and modifying the structure of the knowledge base does not help because the required knowledge for the incorrect questions is missing, or perhaps the retriever is missing important context. This may be a good direction for future work.

Modifying *k* should help to test the latter. Having found varying context size to not be very effective, it is inferred that increasing or decreasing the amount of context taken from the *same* portion of the document is not going to make much difference. For that reason, modifying *k* will expose what benefit taking shorter context, but from a *different* portions of the document can have. To do this, chunk size is set to 128 and overlap is set to 25 as in the prior experiment, and values of *k* from 1 to 10 are run for evaluation. The results of this experiment are shown in Table 4.11.

Table 4.12: Performance by Number of Documents

k	MedMCQA-Anatomy
1	0.590
2	**0.615**
3	0.521
4	0.577
5	0.573
6	0.526
7	0.560
8	0.573
9	0.568
10	0.603

Looking at *k* value 2 alone, it may seem like this is a good value, however in context of the other *k* value results, it appears there is no strict pattern, and in fact it is more likely that the difference is due to error. A possible explanation for this is that the retrieved contexts are retrieved independently of each other. I.e. where there may be benefit in extracting two different excerpts from different parts of the Anatomy book that complement each other, this opportunity is lost due to isolated retrieval.

Another consideration is that there may only be a specific number of documents that are relevant to each question, but since *k* is fixed, it may be that for some questions, more documents are retrieved than are needed, and for others, less. We have seen how extra information that is not directly relevant can be problematic, and retrieving insufficient information will similarly fail to correct a question.

4.5.5 RAG Review

Taking the observation that *k* had little impact, alongside the fact that neither did chunk size, overlap, or the number of documents, it is presumable that the number and length of documents is not an issue, and rather incorrect questions are suffering from other problems. After assessing a random sample of the incorrect outputs and identifying what emerged to be the reason the model was failing to benefit from the retrieved context, the following is what was gathered:

- A question from an exercise section is retrieved without the answer
- The passage retrieved refers to a figure
- The context is relevant to the subject but not the question

After considering the evaluation results, in conjunction with manually reviewing the model responses and the retrieved information from the knowledge base, a few conclusions are reached.

Firstly, for multiple retrieved documents, the retrieved documents should take into context the other documents. For instance, consider the first case in which the retrieved document was a question exercise from the book, as is common in many text books, the answers to the questions are provided in a separate section, so there was no document in the knowledge base that contained both the question and answer together. Had the retrieval of the second document taken the first document into account, retrieving the answer to the question would have been an obvious choice, but instead the second document retrieved was unrelated to the question of the first document. Note, a trade off is that having each document retrieved in context of the other retrieved documents would mean that retrieval could not be implemented in parallel, but for small K, doing so would not be too costly.

In terms of not directly addressing the question, the PDF was manually searched in pursuit of better context in an effort to find out whether the retriever is to blame, but doing so revealed that in some cases, the retriever simply didn't have many quality options. It appears sometimes the idea that is required from the knowledge base may be present in the book, but not as a self contained passage as needed to create a document to capture the idea.

A disadvantage of using an Anatomy textbook as a knowledge base is the pollution of documents with references to figures and other information outside the scope of the context. This isn't an issue when using MedMCQA as a knowledge base in which the question answer pairs are self containing, however question answer pairs also have limitations such as being a spontaneous source of information, i.e. comprehensiveness is not an objective of a question set as it is with a textbook.

4.6 Fine-tuning

Fine-tuning Llama 3 8B is not as straight forward as it was to fine-tune Google's Flan-T5 model, because Flan-T5 is a 248M parameter model, where as Llama 3 8B has more than 30 times the number of parameters. For this reason, fine-tuning the entire model is not feasible within the scope of this projects cost, however we can still take advantage of LoRA adapters to fine-tune a lightweight extension to the parameters. It is still necessary to load the model parameters for inference, but this is also made easier with the use of QLoRA as it provides a much smaller, quantised form of the model. Finally, a library called Unsloth is used which has been shown to provide memory savings of 13.7% to 73.8%.

4.6.1 LoRA Fine-tuning

The LoRA paper demonstrates that even an rank value as low as 1 can provide performance improvement, however in the fine-tuning experiments conducted ranks between 16 and 256 are used as these values are more standard practice. It is also common to use an alpha value of twice the rank which is adopted here. Aside from this, the learning rate is modified as well as the effective batch size, and a various number of training samples are used. All experiments use the AdamW 8 bit optimizer.

Finetuning on the entirety of each dataset was not feasible because the topics range across all areas of Medicine, and therefore the amount of information the model would have to learn would amount to the entirety of the Medicine domain, and without a significant amount of time and cost, this would perhaps prevent the model from being able to learn anything useful. To ensure the model had a chance of learning, this experiment is limited to focus only on the Anatomy domain.

As it wasn't clear how much the model would be able to learn from fine-tuning LoRA parameters, training was applied initially to what appeared to be the most easy task, which was fine-tuning on the train set of the MedMCQA-Anatomy dataset, and evaluating on it's validation set. This is because if the model cannot learn from its own train set, then the issue is not the data, and the model shouldn't be able to learn from other data. A summary of the fine-tuning experiments on the MedMCQA-Anatomy dataset are seen in Table 4.12.

Table 4.13: Fine-tuning Benchmarks

Effective Batch Size	Samples	Learning Rate	LoRA rank	LoRA alpha	Exact Match
-	-	-	-	-	0.585
128	4,000	2e-4	32	64	0.585
128	4,000	2e-4	64	128	0.585
128	4,000	2e-4	128	256	0.585
128	4,000	2e-4	256	512	0.585
128	3,500	1e-5	256	512	0.585
128	3,500	3e-5	256	512	0.585
128	4,000	5e-5	256	512	0.585
128	4,000	7e-5	256	512	0.585
128	4,000	9e-5	256	512	0.585
64	3,800	4e-5	64	128	0.585
64	3,800	1e-4	64	128	0.585
64	3,800	3e-4	32	64	0.585
128	4,000	5e-4	128	256	0.585
256	5,000	6e-4	128	256	0.585
128	4,000	2e-4	16	32	0.585
128	8,000	2e-4	16	32	0.585
128	14,560	2e-4	16	32	0.585

As is evident, there was absolutely no change in any of the fine-tuning experiments. To see whether the model was learning anything at all I tried evaluating on the train set itself and found only an increased score of 0.02.

4.6.2 Fine-tuning Review

This section is cut short due to lack of direction, but the results are discussed further as well as possible explanations.

The lack of impact of fine-tuning was quite a surprise at first. There are 14,560 samples in the train set of MedMCQA-Anatomy. Evaluating on over one quarter of these saw no change. Adjusting the rank, alpha, and learning rate whilst keeping the number of samples the same found no respite either. Presuming the issue may be that the amount training data was not sufficient encouraged the effort to fine-tune on the entire dataset. If doing so yielded just a small change, it would have revealed that what the model needed was more data, and then experiments could be repeated with different parameters, however still no difference was found.

After looking into perhaps why these results were observed, there seemed to be much talk in the NLP community around LoRA modifying style as a pose to teaching new knowledge. Gekhman et al. [47] suggests that full fine-tuning a model with new information other than what it was trained on can be successful in teaching the model that new information, but new information is learned significantly slower than information consistent with the model's knowledge, and whilst it is eventually possible to teach new information, doing so can make the model more likely to hallucinate. It seems the more 'raw' the training, the more learning the model does in the process of training (i.e. pre-training being the most raw in which the model learns the most, then fine-tuning, then LoRA at which point some suggest no learning takes place at all). In other words, the less intrusive the training, the more subtle the impact.

Considering the full fine-tuning of Flan-T5, we were able to see significant improvement as the model initially scored indistinguishably from random choice, yet after fine-tuning the results showed significant improvement. It is suspected this is due to the fact that all the parameters were fine-tuned which gives credit to the idea that more 'raw' training is necessary for learning.

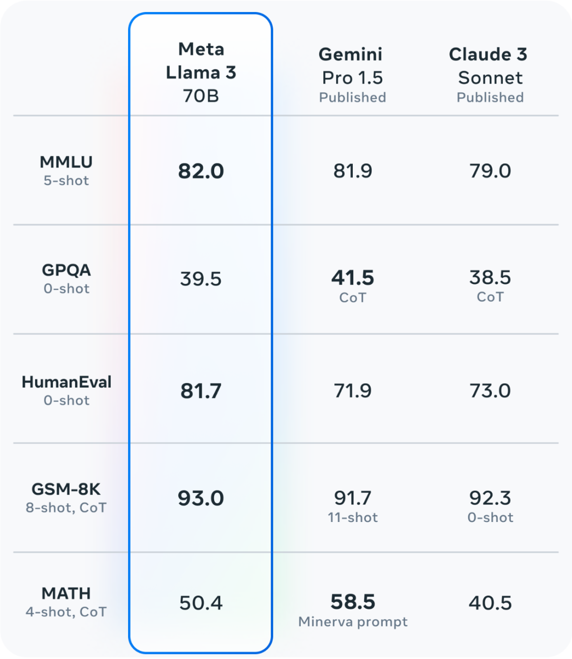
Whether any improvement at all can be achieved with LoRA fine-tuning is a question to be answered, but from this experiment, it appears the amount of improvement, if not none, is very little. From a theoretical perspective, of course whether a model 'knows' a piece of information has no definitive answer; since model parameters are continuous values, whether a model 'knows' or doesn't 'know' a piece of information is a question of how strongly the representation of that information is ingrained into its parameters. Perhaps for questions the model answered incorrectly, the model's full parameters in fact do compose the information but it's representation is weak and can be thrown off. From an empirical perspective, it is easy to have ChatGPT correct itself even when it is correct in some cases, but if you try to get ChatGPT to correct itself on 2 + 2, it will not do so.

Finally, models learn from experiencing examples multiple times in different contexts. It may be that it is more manageable to teach a model 'style' rather than new information, and this is also suggested by Zhou et al. [7]. The reason may be because if the objective is for the model to learn the style of MedMCQA questions, then every sample in the dataset will contribute towards learning that, however if the objective is for the model to learn that "the Cranial accessory nerve does not carry proprioception from the head and neck", then there may only be a hand full of questions in the dataset that speak about proprioception, and perhaps only 2 or 3 that mention the interaction with the head and neck. This could be another possible explanation as to why it is more difficult to teach information than it is to teach style.

4.7 Getting a Second Opinion

The next method attempted could be considered some form of prompt engineering, but incorporates multiple models. Xu et al. [45] looks at automated sequential prompting of a model from a single prompt to produce a single response, which is a technique used elsewhere, for example in the underlying algorithms of libraries such as LlamaIndex. However given creative methods for prompting a single model have proven successful, it would be interesting to learn whether there could be some benefit in prompting different models and composing a response from their combined effort.

The idea is similar to the idea of mixture of experts [31] in which models are trained to contain different groups of parameters that are tailored to specific tasks. However it appears even large language models amongst themselves exhibit signs of being experts in different fields. Comparing Llama 3 70B with Gemini 1.5, despite the fact that Llama overall is much better, there is still an evident trade off between Llama and Gemini when considering Mathematical ability and on GPQA [9], a benchmark intended to consist of extremely difficult multiple choice questions. One might interpret their differences as Gemini being more capable at STEM and Llama more articulate and knowledgeable.



	Meta Llama 3 70B	Gemini Pro 1.5 Published	Claude 3 Sonnet Published
MMLU 5-shot	**82.0**	81.9	79.0
GPQA 0-shot	39.5	**41.5** CoT	38.5 CoT
HumanEval 0-shot	**81.7**	71.9	73.0
GSM-8K 8-shot, CoT	**93.0**	91.7 11-shot	92.3 0-shot
MATH 4-shot, CoT	50.4	**58.5** Minerva prompt	40.5

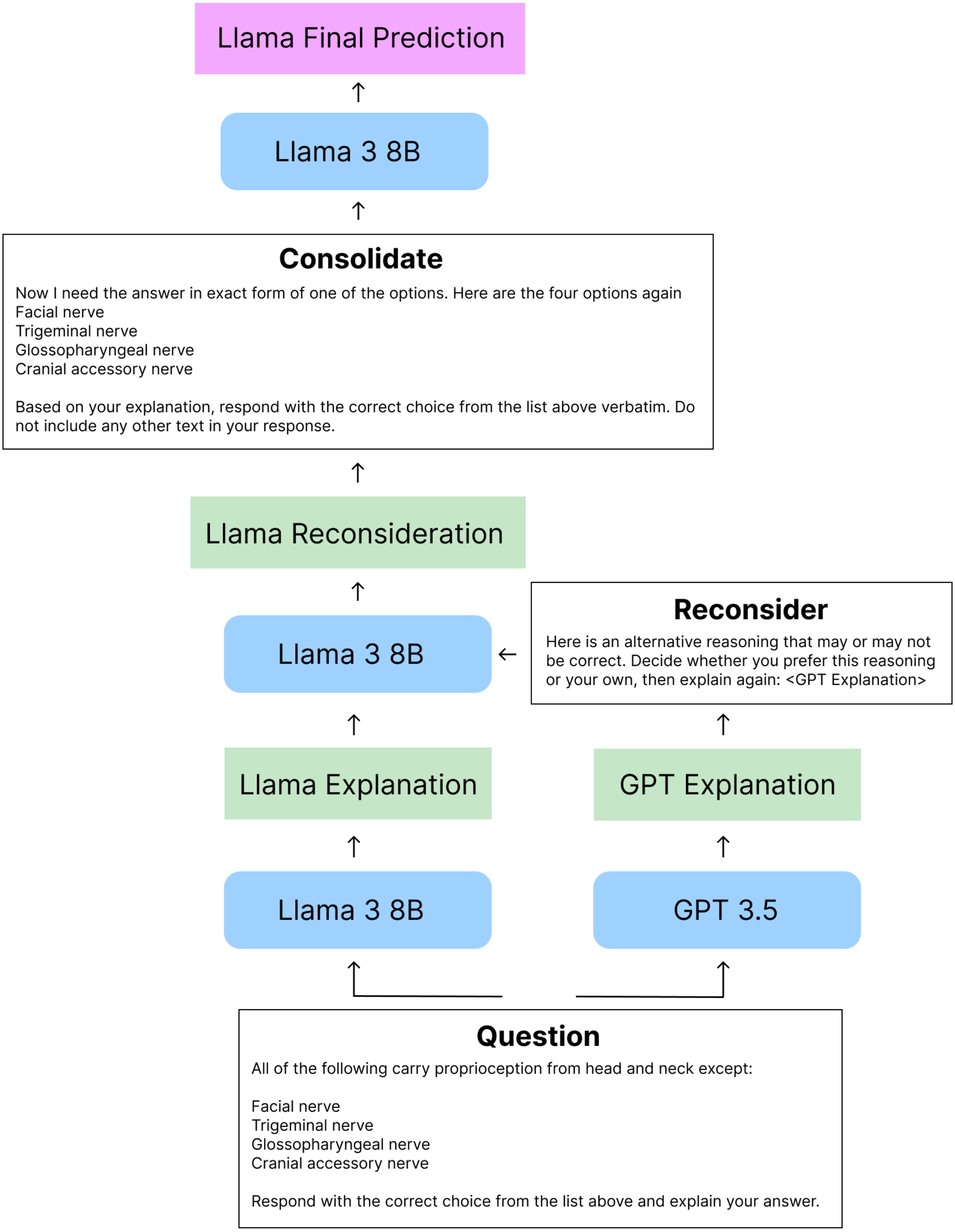
Figure 4.12: Llama 3 Benchmarks [44]

As well as the fact that different LLMs are better in different areas, it can be noticed that when pushing the boundaries of an LLMs capabilities, it becomes very hesitant, and at a little questioning, it is very easy to have the model contradict itself. This opened the question as to whether it is possible to exploit these instances in which the model is hesitant, by providing a second opinion from another model and seeing whether it helps the model reach the right answer.

The two models used for this experiment were Llama 3 8B, and GPT-3.5-Turbo. Note the only difference between the Llama model used in this experiment and the one used previously is that the version used in this experiment is not quanitsed, i.e.

the parameters are 16 bit bfloats as a pose to 4 bit. Groq is an API service that provides free prompting access to a number of open source models including Llama 3 8B which is what is used to prompt Llama here. GPT-3.5-Turbo is accessed via the OpenAI API and no RAG knowledge base is used.

As for the prompting structure, it is illustrated in Figure 4.13.



Consolidate

Now I need the answer in exact form of one of the options. Here are the four options again
Facial nerve
Trigeminal nerve
Glossopharyngeal nerve
Cranial accessory nerve

Based on your explanation, respond with the correct choice from the list above verbatim. Do not include any other text in your response.

Reconsider

Here is an alternative reasoning that may or may not be correct. Decide whether you prefer this reasoning or your own, then explain again: <GPT Explanation>

Question

All of the following carry proprioception from head and neck except:

Facial nerve
Trigeminal nerve
Glossopharyngeal nerve
Cranial accessory nerve

Respond with the correct choice from the list above and explain your answer.

Figure 4.13: Second Opinion Prompting Structure

Figure 4.13 shows the prompting structure with Llama taking the final decision. The experiment was repeated identically but in reverse in which GPT takes the final decision. The former is referred to as ’Llama 3 8B (SO)’ and the latter ’GPT 3.5 (SO)’.

The results are shown in Table 4.13.

Table 4.14: Second Opinion Benchmarks

Dataset	GPT 3.5	GPT 3.5 (SO)	Llama 3 8B	Llama 3 8B (SO)
MedMCQA	**0.500**	0.449	0.449	0.487
MedMCQA-Anatomy	0.521	0.551	0.543	**0.603**
MedQA (USMLE)	**0.645**	0.564	0.526	0.598
PubMedQA	0.449	0.808	**0.821**	0.701
MMLU-Anatomy	0.644	0.563	0.570	**0.667**

Taking into account the results as a whole, there is no consistent pattern. By looking at what proportion of the questions only one model got right and the other wrong, we can get an estimate how frequently the second opinion managed to correct an incorrect answer. To do this, the set difference between both models individual answers was used to calculate the percentage of questions the models differed on. This value is then compared to each models performance gain after applying the second opinion technique. These statistics can be seen in table 4.14.

Table 4.15: Second Opinion Success Rate

Dataset	GPT 3.5 SO Gain	Llama 3 8B SO Gain	Disagreement
MedMCQA	-0.1%	3.8%	29.1%
MedMCQA-Anatomy	3.0%	6.0%	34.2%
MedQA (USMLE)	-8.1%	7.2%	29.1%
PubMedQA	35.9%	-12.0%	44.9%
MMLU-Anatomy	-8.1%	9.7%	17.1%
Mean	4.5%	2.9%	30.9%
Standard Deviation	7.3%	7.7%	4.0%

4.7.1 Second Opinion Review

Taking into account the mean for GPT Gain and Llama Gain, both are positive, however when considered in context of the high the standard deviation, it suggests there is no consistent gain.

It is interesting to see however that the biggest difference in both cases is made on PubMedQA, which also has the biggest proportion of disagreement. Looking at Table 4.13, it appears the majority of this disagreement is due to GPT 3.5's poor performance on PubMedQA as compared to Llama which scores highly. When combining the models, Llama's performance dropped by 12%, suggesting Llama is somewhat unconfident in its knowledge and is easily coerced by GPT 3.5. Likewise, it appears GPT 3.5 is well aware of its lack of knowledge and easily accepts answers from Llama given it's score improves by 35.9%.

Due to the high variance, it is difficult to decisively say whether getting a second opinion is beneficial, but in either case it is evident that there is a lack of ability in recognizing credible alternative answers on the part of both models. Considering the amount of disagreement compared with the amount of performance gain resulting from that, we see little change, only 4.5% and 2.9%, whilst the average disagreement is 30.9%. I.e. despite a large opportunity for collaboration, only a little of that opportunity is exploited.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This project worked towards the development of a chatbot for domain specific question answering. The domain of Anatomy was taken as case study in this work, in pursuit of being able to replicate the work for alternative domains. Initially experiments with Flan-T5, a relatively small size 248M parameter model, found that such few parameters were not capable of satisfying the project objectives, at which point a sufficiently capable Llama 3 8B model was equipped and iterated upon.

The first objective was the improvement upon a language models base capabilities to answer questions requiring domain specific knowledge. This objective was met to various extents via the application of different techniques. The method that achieved the most improvement used question answer pairs from the MedMCQA dataset as a RAG knowledge base. Whilst this method achieved the most improvement, understanding the reasoning for this and considering what made it superior to other techniques assisted in achieving the third objective, namely to compare techniques for improving domain specific knowledge. Considering the downfall of other options for the knowledge base such as word definitions or the use of one or multiple Anatomy text books, helped us to understand that models of substantial size (at least over 8 billion parameters in this case) generally have a strong understanding for the meaning of words. It was also learnt that more elaborate context is needed to be of benefit, and that irrelevant information is not easily dismissed, and can in fact be detrimental. Finally, the second objective was achieved by evaluating the model throughout the application of each technique, using a selection of datasets commonly used in NLP research focused on the Medicine domain.

5.2 Future Work

In finding question answer pairs as the most performant technique, possible future work could consider a dynamic approach to developing a models knowledge. Specifically, this approach would store users questions that could not be answered. Subsequently, subject experts would provide ideal answers, and these question answer pairs would be appended to the knowledge base.

Many of the issues encountered experimenting with RAG were due to the information in the knowledge base being designed for other purposes. Given sufficient time and cost, it would be interesting to see future work explore the manual creation of a knowledge base, specifically designed for RAG. Doing so could ensure documents are designed to be relevant, concise, and informative.

Furthermore, whilst long answer capabilities are necessary for a chatbot, this was not an aspect covered in this project. Future work may consider the application to a another domain lending itself to long answer evaluation, specifically looking to prioritise factually correctness and preventing hallucination. When considering long answer evaluation, fine-tuning becomes more relevant as style and answer structure becomes important. It would be intriguing to see whether the hypothesis made here around the effects of fine-tuning would be reflected.

Finally, it is important to be conscious that the field of NLP and RAG is a rapidly advancing one. It is highly plausible that new language models and knowledge augmentation techniques will come to light in the near future, and may provide new paths towards domain specific knowledge enhancement.

Appendix

.1 Few-Shot

1-shot:

```
<|start_header_id|>user<|end_header_id|>

Chronic urethral obstruction due to benign prismatic hyperplasia can
lead to the following change in kidney parenchyma
Hyperplasia
Hyperophy
Atrophy
Dyplasia<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

```
Atrophy<|eot_id|><|start_header_id|>user<|end_header_id|>
```

3-shot:

```
<|start_header_id|>user<|end_header_id|>

Chronic urethral obstruction due to benign prismatic hyperplasia can
lead to the following change in kidney parenchyma
Hyperplasia
Hyperophy
Atrophy
Dyplasia<|eot_id|><|start_header_id|>assistant<|end_header_id|>

Atrophy<|eot_id|><|start_header_id|>user<|end_header_id|>

Which vitamin is supplied from only animal source:
Vitamin C
Vitamin B7
Vitamin B12
Vitamin D<|eot_id|><|start_header_id|>assistant<|end_header_id|>

Vitamin B12<|eot_id|><|start_header_id|>user<|end_header_id|>

All of the following are surgical options for morbid obesity except -
Adjustable gastric banding
Biliopancreatic diversion
Duodenal Switch
Roux en Y Duodenal By pass<|eot_id|><|start_header_id|>assistant<|end_header_id|>

Roux en Y Duodenal By pass<|eot_id|><|start_header_id|>user<|end_header_id|>
```

5-shot:

```
<|eot_id|><|start_header_id|>user<|end_header_id|>Chronic
urethral obstruction due to benign prismatic hyperplasia can
lead to the following change in kidney parenchyma
Hyperplasia
Hyperophy
Atrophy
Dyplasia<|eot_id|><|start_header_id|>assistant<|end_header_id|>

Atrophy
```

```
<|eot_id|><|start_header_id|>user<|end_header_id|>Which vitamin
is supplied from only animal source:
Vitamin C
Vitamin B7
Vitamin B12
Vitamin D<|eot_id|><|start_header_id|>assistant<|end_header_id|>

Vitamin B12

<|eot_id|><|start_header_id|>user<|end_header_id|>All of the
following are surgical options for morbid obesity except -
Adjustable gastric banding
Biliopancreatic diversion
Duodenal Switch
Roux en Y Duodenal By pass<|eot_id|><|start_header_id|>assistant<|end_header_id|>

Roux en Y Duodenal By pass

<|eot_id|><|start_header_id|>user<|end_header_id|>Following endaerectomy
on the right common carotid, a patient is found to be blind in the right
eye. It is appears that a small thrombus embolized during surgery and
lodged in the aery supplying the optic nerve. Which aery would be blocked?
Central aery of the retina
Infraorbital aery
Lacrimal aery
Nasociliary aretry<|eot_id|><|start_header_id|>assistant<|end_header_id|>

Central aery of the retina

<|eot_id|><|start_header_id|>user<|end_header_id|>Growth hormone has its
effect on growth through?
Directly
IG1-1
Thyroxine
Intranuclear receptors<|eot_id|><|start_header_id|>assistant<|end_header_id|>

IG1-1<|eot_id|><|start_header_id|>user<|end_header_id|>
```

Bibliography

[1] By Rebecca Hasdell JULY 2020. WHAT WE KNOW ABOUT UNIVERSAL BASIC INCOME A CROSS-SYNTHESIS OF REVIEWS. 2020.

[2] John Charles Boileau Grant A. M. R. Agur, Arthur F. Dalley. Grant's Atlas of Anatomy. 2013.

[3] Arthur Mensch Chris Bamford Albert Q. Jiang, Alexandre Sablayrolles. Mistral 7B. 2023.

[4] Malaikannan Sankarasubbu Ankit Pal, Logesh Kumar Umapathi. MedMCQA : A Large-scale Multi-Subject Multi-Choice Dataset for Medical domain Question Answering. 2022.

[5] Google Ashish Vaswani, Noam Shazeera Niki Parmar. Attention is all you need. 2017.

[6] Nicolas Langrené Shengxin Zhu Banghao Chen, Zhaofeng Zhang. Unleashing the potential of prompt engineering: a comprehensive review. 2023.

[7] Puxin Xu Srini Iyer Chunting Zhou, Pengfei Liu. LIMA: Less Is More for Alignment. 2023.

[8] Steven Basart Andy Zou Dan Hendrycks, Collin Burns. Measuring Massive Multitask Language Understanding. 2020.

[9] Asa Cooper Stickland Jackson Petty Richard Yuanzhe Pang David Rein, Betty Li Hou. GPQA: A Graduate-Level Google-Proof QA Benchmark. 2023.

[10] Marina del Rey. https://aclanthology.org/W04-1013/. 2004.

[11] Nassim Oufattole Wei-Hung Weng Di Jin, Eileen Pan. What Disease does this Patient Have? A Large-scale Open Domain Question Answering Dataset from Medical Exams. 2020.

[12] Phillip Wallis-Zeyuan Allen-Zhu Edward J. Hu, Yelong Shen. LoRA: Low-Rank Adaptation of Large Language Models. 2021.

[13] Harold Ellis. Clinical Anatomy: Applied Anatomy for Students and Junior Doctors. 2006.

[14] John T. Hansen. Netter's Clinical Anatomy. 2019.

[15] Deng Cai Yan Wang Lemao Liu Huayang Li, Yixuan Su. A Survey on Retrieval-Augmented Text Generation. 2022.

[16] Gautier Izacard Xavier Martinet Hugo Touvron, Thibaut Lavril. LLaMA: Open and Efficient Foundation Language Models. 2023.

[17] Shayne Longpre Barret Zoph Hyung Won Chung, Le Hou. Scaling Instruction-Finetuned Language Models. 2022.

[18] Kenton Lee Google Jacob Devlin, Ming-Wei Chang. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2018.

[19] Dale Schuurmans Maarten Bosma Jason Wei, Xuezhi Wang. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. 2022.

[20] Xianpei Han Le Sun Jiawei Chen, Hongyu Lin. Benchmarking Large Language Models in Retrieval-Augmented Generation. 2023.

[21] Tahmineh Mohati Maleknaz Nayebi Jiho Shin, Clark Tang. Prompt Engineering or Fine Tuning: An Empirical Assessment of Large Language Models in Automated Software Engineering Tasks. 2023.

[22] Travis Manfredi Joseph R. Saveri, Cadio Zirpoli. UNITED STATES DISTRICT COURT NORTHERN DISTRICT OF CALIFORNIA SAN FRANCISCO DIVISION COMPLAINT CLASS ACTION DEMAND FOR JURY TRIAL . 2023.

[23] Juraj Gottweis Rory Sayres Karan Singhal, Tao Tu. Towards Expert-Level Medical Question Answering with Large Language Models. 2023.

[24] Tao Tu S. Sara Mahdavi Karan Singhal, Shekoofeh Azizi. Large Language Models Encode Clinical Knowledge. 2022.

[25] Edward Grefenstette Lasse Espeholt Karl Moritz Hermann, Tomáš Kočiský. Teaching Machines to Read and Comprehend. 2015.

[26] Todd Ward Kishore Papineni, Salim Roukos and Wei-Jing Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. 2002.

[27] Rachel Koshi. Cunningham's Manual of Practical Anatomy: Upper and lower limbs. 2017.

[28] Ryan Mac Michael M. Grynbaum. The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work. 2023.

[29] Ethan Perez Samuel R. Bowman Miles Turpin, Julian Michael. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. 2023.

[30] Yosief Tsige Nega Assefa. Human Anatomy and Physiology. 2003.

[31] Krzysztof Maziarz Andy Davis Quoc Le Geoffrey Hinton Jeff Dean Noam Shazeer, Azalia Mirhoseini. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. 2017.

[32] Thibault Sellam Dipanjan Das Ankur P. Parikh. BLEURT: Learning Robust Metrics for Text Generation. 2020.

[33] Aleksandra Piktus et al. Facebook UCL NYU Patrick Lewis, Ethan Perez. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. 2020.

[34] Konstantin Lopyrev Percy Liang Pranav Rajpurkar, Jian Zhang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. 2016.

[35] Zhengping Liu William W. Cohen Qiao Jin, Bhuwan Dhingra. PubMedQA: A Dataset for Biomedical Research Question Answering. 2019.

[36] Eric Mitchell Stefano Ermon Rafael Rafailov, Archit Sharma. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. 2023.

[37] Jordan Hoffmann et al. Deepmind Sebastian Borgeaud, Arthur Mensch. Improving language models by retrieving from trillions of tokens. 2021.

[38] Eden Chung Natan Vidra Spurthi Setty, Katherine Jijo. Improving Retrieval for RAG based Question Answering Models on Financial Documents. 2024.

[39] Suranga Nanayakkara Tharindu Kaluarachchi, Rajib Rana. Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering. 2022.

[40] Ravi Theja. Evaluating the Ideal Chunk Size for a RAG System using LlamaIndex. 2023.

[41] Robert Dadashi Surya Bhupatiraju Thomas Mesnard, Cassidy Hardin. Gemma: Open Models Based on Gemini Research and Technology. 2024.

[42] Ari Holtzman Luke Zettlemoyer Tim Dettmers, Artidoro Pagnoni. QLoRA: Efficient Finetuning of Quantized LLMs. 2023.

[43] Nick Ryder Melanie Subbiah Tom B. Brown, Benjamin Mann. Language Models are Few-Shot Learners. 2020.

[44] Meta Official Website. Introducing Meta Llama 3: The most capable openly available LLM to date. 2024.

[45] Nebojsa Jojic Weijia Xu, Andrzej Banburski-Fahey. Reprompting: Automated Chain-of-Thought Prompt Inference Through Gibbs Sampling. 2023.

[46] Xinyu Gao Kangxiang Jia Jinliu Pan Yuxi Bi Yi Dai Jiawei Sun Qianyu Guo Meng Wang Haofen Wang Yunfan Gao, Yun Xiong. Retrieval-Augmented Generation for Large Language Models: A Survey. 2023.

[47] Roee Aharoni Matan Eyal Zorik Gekhman, Gal Yona. Does Fine-Tuning LLMs on New Knowledge Encourage Hallucinations? 2024.