Retrieval Augmented Generation for Pittsburgh and CMU

Alex Fang
alexfang

Aviral Agrawal
avirala

Dhatchi Kunde Govindarajan
dkundego

Abstract

Large Language Models (LLMs) are trained on trillions of tokens, enabling them to learn the syntax and semantics of natural language and also memorize facts. However, newer data is generated every second and the models have to be continually fine-tuned on this data to keep up. To avoid the huge cost of fine-tuning, we have Retrieval Augmented Generation (RAG) systems. We want to use open-source LLMs to answer targeted questions about Carnegie Mellon University (CMU) and Pittsburgh city. This project attempts to improve the performance of LLMs, such as Llama-3.1-8B-Instruct, roberta-base-squad2, and Mistral-7B-Instruct-v0.1, on CMU and Pittsburgh domain queries using a (RAG) system. Our project investigates how integrating a RAG system into the LLM query pipeline can help an LLM to better perform on domain-specific queries. We found that the LoRA-fine-tuned Llama-3.1-8B-Instruct model with the FAISS retriever performed the best out of all of our models with a F1-score of 0.6777, which is 2.88 times better than the baseline Llama model without RAG. https://github.com/Aviral-Agrawal/11711_RAG_A2/tree/main

1 Introduction

Large language models (LLMs), such at GPT, Llama, and Mistral have shown to be effective for tasks such as question-answering (Touvron et al., 2023). However, there are some notable accuracy issues with LLMs. Such situations may commonly arise when LLM systems are queried about recent events that occurred after the cutoff date for training or when the LLM is asked about private company data that it may not have access to through training on publicly available data. To address such knowledge cutoff and learning failures, RAG systems can alleviate these issues by providing the LLM with access to documents that were not included in their training corpus. This report outlines our tests of building an LLM-based RAG system using LangChain, Torchtune, and several different LLMs from HuggingFace, as to be discussed in 3.3. In our experiments, we evaluated several configurations: a base Llama model without a retrieval-augmented generation (RAG) system, a fine-tuned Llama model with RAG, and two additional language models, Mistral, and RoBERTa. For the RAG system, we tested two retrieval methods: FAISS and BM25.

2 Data Creation

2.1 Compilation of Data Sources

To create our data set, we compiled information from Pittsburgh Cultural websites, Pennsylvania government and cultural websites, CMU websites, and other pages broadly related to Pittsburgh and CMU. For a full list of websites, please check the scraped data folder on our GitHub linked in the abstract.

2.2 Data Extraction

To extract data from the websites in our data sources, we wrote a Python script to scrape the data of each of the websites, as well as their subpages and links, recursively. We used tools such as Beautiful Soup, PyPDF, and Selenium to scrape HTML responses, PDFs, and dynamic web pages, respectively. Initially, we scraped all the data off of each webpage. However, this led to noise in the documents as they contained text unrelated to the body of the documents themselves, such as website headers. Therefore, we decided to try to reduce the noise to a certain extent by only scraping the title and body of each webpage. For PDFs, we still scraped the entire PDF as there wasn't as much noise. The .gov websites had SSL encryp-

Model	Context Length	Model Size	Purpose
Llama-3.1-8B-Instruct	4096	8 Billion	Chat, Summarization, etc.
roberta-base-squad2	512	125 Million	QA
Mistral-7B-Instruct-v0.1	8192	7 Billion	Dialogue, Summarization, Reasoning

Table 1: Model Details

tion issues, therefore, those were scraped without verification. The CMU events website also dynamically loaded their data, which led to the need for Selenium and ChromeDriver to scrape the data.

2.3 Dataset Creation and Annotations

To create our training and testing datasets, we fed each scraped document to meta.ai, which runs the open-source Llama-3.1-405B model, via API calls to generate 5 question-answer pairs along with the context of where the answer was found. This was done for the entire document database to create a varied dataset regarding question types and content. After minor output processing, we create a split with 10,634 training examples and 1,000 test examples. The amount of data we annotated was decided by the fine-tuning compute budget, open-source LLM querying budget, and comparison to the amount of data other open-source datasets had. From the overall dataset, two members then hand-annotated 100 examples and measured inter-annotator agreement (IAA) using exact match, precision, recall, and F1-score, achieving 0.32, 0.71, 0.73, and 0.791, respectively. These IAA scores are relatively high, showing that there isn't much subjectivity in the labels, meaning that the model should be able to come to a definite answer.

3 System Details

3.1 Model Details

We tested several models, namely: Llama-3.1-8B-Instruct (Dubey, 2024), Llama-3.1-8B-Instruct LoRA-fine-tuned on our generated training dataset (Hu et al., 2021), roberta-base-squad2(Liu et al., 2019), and Mistral-7B-Instruct-v0.1 (Jiang et al., 2023). The model details and variations are shown in Table 1 and described more in section 3.3.

3.2 RAG System

Our RAG system was developed using the Langchain framework. The first step of developing the system requires us to index and store all of our scraped documents. To accomplish this, we first use the RecursiveCharacterTextSplitter to chunk our documents into chunks of 1024 characters with 128 character overlaps. We originally chose to create chunks of 2000 characters, but we deemed the context to be too long since the RAG system is returning the top four most similar documents. Then, we use the nomic-ai/nomic-embed-text-v1 model from the MTEB leaderboard due to its small size, dense embeddings, and fast generation. We then created a vector database using FAISS (Facebook AI Similarity Search) for efficient similarity search and clustering of dense vectors. We chose FAISS over ChromaDB because it is optimized for dense vector similarity search and grouping.

Asides from FAISS, we also test out BM25, which is a sparse vector retrieval algorithm to test the difference in performance for sparse versus dense retrieval methods. The reason we decided to do this is to see if the performance-to-compute tradeoff is worth it since dense retrieval is more computationally expensive than sparse retrieval.

3.3 Models and Variations

As previously stated, we tested 4 models/variations: Llama-3.1-8B-Instruct, Llama-3.1-8B-Instruct LoRA-fine-tuned on our generated training dataset, roberta-base-squad2, and Mistral-7B-Instruct-v0.1. We choose the Llama-3.1-8B-Instruct as our baseline model. We chose the Llama and Mistral models due to their high performance on text generation tasks, which made us wonder if it is possible to transfer their capabilities over to the task of question-answering. Additionally, we decided to perform LoRA fine-tuning on the Llama model, which is our baseline model, to see if we can instruction-tune the model on CMU and Pittsburgh domain-specific data for QA. We also chose the roberta-base-squad2 model as it had been fine-tuned on the Stanford SQuAD2.0 dataset, which is specifically trained on question-answer pairs, including unanswerable questions, for the task of Extractive Question Answering, which is exactly the task we had at hand.

As previously mentioned, the roberta model is fine-tuned on context-based question-answering,

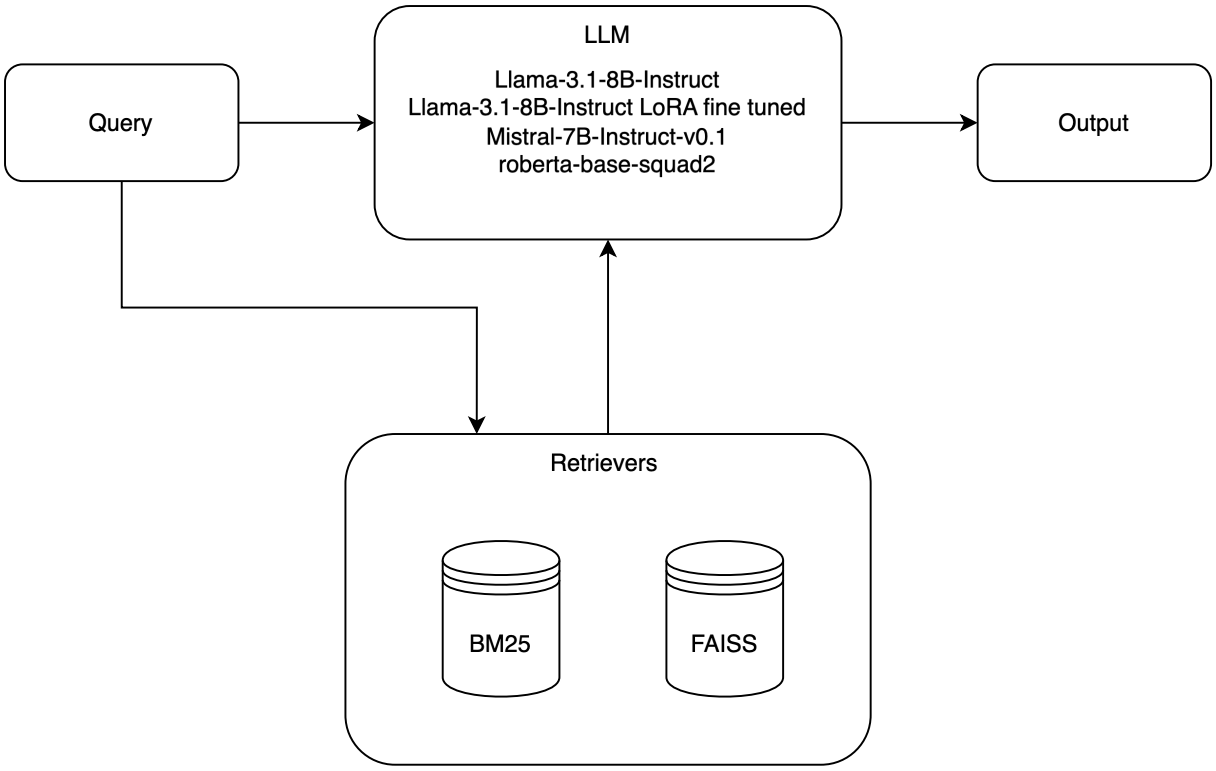

Figure 1: Overview of RAG + LLM system.

however, the Llama and Mistral models have typically been tuned for text-generation. In order to boost their capabilities, we use few-shot prompting to improve the LLMs' performance. The few-shot prompt is also aimed at making the model outputs more concise and directed. Our prompt looks as follows:

```
prompt = [
    {"role": "user", "content": "You are a question-answering machine. Your task is to return a json object. \
        Do NOT give any explanation and you do not need to answer in full english sentences. \
        ONLY answer the question on the context given. \
        If you don't know the answer, say 'I do not know'."},

    {"role": "assistant", "content": "Understood, I will exactly follow the instructions and answer the question accordingly."},

    {"role": "user", "content": "{\
            'context': 'France is renowned for its rich culture, diverse landscapes, and historical significance, making it one of the most visited countries in the world. Paris, the capital city, is famous for its art, fashion, and architectural marvels like the Eiffel Tower and Notre-Dame Cathedral. From the scenic vineyards of Bordeaux to the sunlit beaches of the French Riviera, France offers a unique charm and beauty at every corner.'\
            'question': 'What is the capital of France?',\
            'answer': ''\
        }"},

    {"role": "assistant", "content": "{\
            'question': 'What is the capital of France?',\
            'answer': 'Paris'\
        }"},

    {"role": "user", "content": f"{{\
            'context': '{context}',\
            'question': '{question}',\
            'answer': '',\
        }}"}
]
```

To dive deeper into the aspects of the prompt, we first let the model know about the kind of role it needs to play for the task at hand. Furthermore, we restrict the model for giving any wordy explanations and even encourage it to not use full English sentences. We then give the model concrete examples of how we want it to function given a sample

Model	F1 Score	Exact Match (EM)	Precision	Recall
Llama no RAG	0.2354	0.1270	0.2468	0.2477
Llama_BM25	0.5655	0.4000	0.5481	0.6094
Llama_FAISS	0.6690	0.4550	0.6369	**0.7395**
Llama_FT_BM25	0.5202	0.3950	0.5116	0.5310
Llama_FT_FAISS	**0.6777**	**0.4980**	**0.6660**	0.7012
roberta_BM25	0.4482	0.3130	0.4826	0.4451
roberta_FAISS	0.5447	0.3920	0.5858	0.5352
Mistral_BM25	0.4778	0.3250	0.4556	0.6159
Mistral_FAISS	0.5588	0.3770	0.5241	0.7286

Table 2: Performance Metrics for baseline model, model variations and RAG retriever variations.

input. Finally, we give the model the actual input (question along with the context) and expect the model to return a json string. The motivation behind making the model return a JSON string is to guide the model towards a structured output. This structure would further encourage the model to stick to the given instructions.

4 Results

For each of our model outputs on our test data, we calculated the F1-score, precision, recall, and exact match scores against the ground truth reference answers using the HuggingFace evaluate package with SQuAD. These results are reported in table 2.

To further analyze our results, we calculated the p-value, with a threshold of 0.05, for our top and worst models to check for statistical significance. In calculating the p-value of the F1-scores for the baseline closed-book Llama model and for the Llama-FT models, we observe a p-value of 2E-10 and a p-value of 0.17 respectively. These values show that the results are statistically significant. Similarly, we observe a p-value of 0.17, 1E-10, and 1E-9 for Llama_FAISS and Llama_FT_FAISS, Llama_FT_FAISS and roberta_FAISS, and Llama_FT_FAISS and Mistral_FAISS respectively.

From these results, we then choose the top models to generate answers for the final testing dataset. Our final outputs are from the Llama_FT_FAISS, Llama_FAISS, and Mistral_FAISS models.

5 Analysis

From Table 2, we can compare the retrieve-and-augment strategy vs closed-book use of our model. We implement baseline model, the Llama-3.1-8B-Instruct model in the closed book setting. For the Llama model in closed-book setting, we get 46.7% "I do not know" responses from the model. On the other hand, from the retrieve-and-augment strategy with the same model, we get only 5.7% "I do not know" answers. This goes to show how significant the boost in performance is using the context that we can mine from the retrieve-and-augment strategy.

From Figure 2, we can see some examples of outputs from the various models that we implement for several different questions. The baseline Llama model performs the worst with all incorrect answers. The Llama_FAISS model and the Llama_FT_FAISS model perform similarly as the best two models. Then, the next two best models are Roberta_FAISS and Mistral_FAISS, which perform similarly. Overall, these results are representative of the quantitative performance from Table 2.

Expanding on the analysis, table 3 highlights how different models stack up across question types—Factual, Descriptive, Procedural, and List Answer. The Llama_FT_FAISS model emerges as the top performer, especially for Factual and Procedural questions. With fine-tuning and FAISS retrieval, this model handles Pittsburgh and CMU-specific details much better than the closed-book versions.

One thing that stands out in the Descriptive category is the lower F1 scores across the board, even for models with retrieval. Interestingly, Roberta_BM25 scores relatively well here, hinting that sparse retrieval (like BM25) might actually work better for general queries, whereas dense retrieval with FAISS seems to benefit questions that lean more heavily on specific, in-depth content.

In short, models with retrieval not only score higher F1s but also reduce the “I do not know” responses, making retrieval-augmented setups as the

Question	GT Answer	Llama	Llama+RAG	Llamafinetune+ RAG	Roberta+RAG	Mistral+RAG
Who is the current Council President of the Pittsburgh City Council?	**R. Daniel Lavelle**	I do not know	R. Daniel Lavelle	R. Daniel Lavelle	Beth Pindilli	Rebecca Dyas
What city is nicknamed the "Pittsburgh of the South"?	**Birmingham, Alabama**	Chattanooga	I do not know	I do not know	Da 'Burgh	Albuquerque, New Mexico
What did the PHLF establish in 1966?	**The Revolving Fund for Preservation**	The Western Pennsylvania Historical Society	The Revolving Fund for Preservation	The Revolving Fund for Preservation	Revolving Fund for Preservation	The Revolving Fund for Preservation
What did Bob Regan do in the late 1990s and early 2000s?	**He cataloged Pittsburgh's steps.**	I do not know	I do not know	I do not know	supply-side (monetarist) economic policies	Bob Regan was a member of the.....
What agreement did the Pittsburgh Agreement replace?	**Cleveland Agreement**	The Munich Agreement	The Cleveland Agreement of October 22, 1915	The Cleveland Agreement of October 22, 1915	Cleveland Agreement	Cleveland Agreement

Correct answers
No hallucinations
Hallucinations

Figure 2: Some outputs from the system

Model	Factual F1	Descriptive F1	Procedural F1	List Answer F1
Llama_ft_faiss	0.889	0.4285	0.6806	0.6815
Mistral_faiss	**0.9444**	0.2031	0.4889	0.5250
Llama_faiss	0.9028	0.4007	0.5756	**0.6988**
Roberta_bm25	0.4450	**0.6417**	0.3929	0.4000
Llama_base	0.4444	0.1875	0.2500	0.3455
Llama_ft_bm25	0.5000	0.4100	0.4712	0.5748
Roberta_faiss	0.6111	0.4539	0.3929	0.4308
Mistral_bm25	0.8333	0.2479	0.2364	0.4768
Llama_bm25	0.6667	0.4451	**0.7102**	0.6156

Table 3: F1 Scores by Category for Each Model (Best Scores Bolded)

bet choice when it comes to handling specialized, domain-focused queries. Overall, these findings back up our previous results and show the effectiveness of retrieval for better accuracy on targeted tasks.

6 Conclusion

Our experiments highlight the effectiveness of utilizing retrieval-augmented systems with Large Language Models to increase the accuracy of the models for domain-specific tasks such as answering questions about CMU and Pittsburgh. Comparing the performance of base Llama model without RAG to that of RoBERTa, a much smaller model, with RAG, portrayed how great of an improvement RAG system can have on LLMs, and can make smaller LLMs outperform larger ones.

While both Llama-Faiss and Llama-Fine-tuned-Faiss showed strong performance, the gains from fine-tuning were relatively minor. Due to limited compute resources, we could not fine-tune the model for more than a few epochs. With the current level of fine-tuning, we observe greater conciseness and relevance in responses from the LLM. However, since the model adheres more strictly to the input context, we also observe a higher frequency of the model answering "I do not know", for questions it wasn't able to answer given the context and prompt. The given context could be lacking information necessary to answer due to issues with such as poor context retrieval for that particular question. Although not beneficial for this task, this behavior might be better suited for applications that require high confidence in responses.

Future work could focus on a classification system that identifies the type of each query, assigning

it to the best-suited model based on category performance. Such an approach could effectively utilize each model's strengths, optimizing both accuracy and resource efficiency in real-world use.

References

Abhimanyu Dubey. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.