IFML Retrieval Augmented Generation Models

Michelle Ding
Department of Computer Science
The University of Texas at Austin
michelle.ding01@utexas.edu

Yuan Lu
Department of Computer Science
The University of Texas at Austin
yl38457@utexas.edu

J Whorley[*]
Department of Computer Science
The University of Texas at Austin
jwhorley@cs.utexas.edu

Alexandros G. Dimakis[*]
Department of ECE
The University of Texas at Austin
dimakis@austin.utexas.edu

Adam Klivans[*]
Department of Computer Science
The University of Texas at Austin
klivans@utexas.edu

Abstract

Retrieval Augmented Generation (RAG) is a framework for enhancing the accuracy and reliability of generative AI models with facts fetched from external sources. In this project, we attempt to improve the accuracy of RAG models past classical retrieval algorithms, specifically by building a RAG model trained for the purpose of answering questions about IFML NeurIPS papers.

1 Introduction

1.1 Challenges

Retrieving embeddings from a vector store knowledge base is known to be computationally expensive, not to mention physically costly. Thus, one major challenge of RAGs is scalability in terms of dealing with larger datasets. In addition, it is common to initially see the retrieval component of a RAG select irrelevant or incorrect information that can lead to inaccurate responses or even hallucination with seemingly fabricated data. Fine-tuning RAG models to account for specific tasks such as accommodating images, equations, or metadata can also be a challenge. Adapting these models to new domains or changing requirements to meet the standards of our goal often require more knowledge of the system which we attempt to tackle in this project.

1.2 Our Setup and Goals

RAG models are highly versatile in producing responses that are more contextually relevant to the user's input by leveraging retrieved information with an LLM and embedding store. This creates a highly enticing option of using RAGs as an alternative to classical retrieval algorithms. Training our model to match the accuracy and correctness of responses in classical retrieval systems like Google PageRank is a challenge. In our project, we seek to address these challenges, and also provide a practical system that can be run online, by developing and fine-tuning our model to achieve our most optimal accuracy level. Our baseline implementation uses free, available software, which consists of Qdrant embeddings and Hugging face LLMs. Knowing what we wanted to base our LLM off of,

[*] Supervising faculty.

we then consulted various literature and documentation to select a series of tests to run and use the resulting metrics to steer in the direction of what worked best. As for RAG architecture, we ended up using a re-ranking addition, shift to Llama-Index, and the VoyageAI vector database configurations.

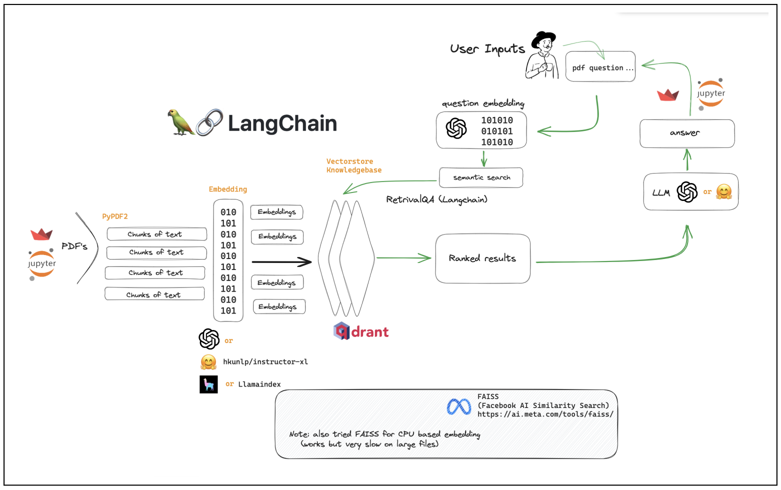

Figure 1: Our workflow

2 Evaluation

2.1 Retrieval Evaluation

To ease the burden of creating the eval dataset in the first place, we rely on synthetic data generation [1]. Specifically, we use LlamaIndex's `generate_question_context_pairs` function to evaluate our model on a set of auto-generated queries and ground truths.

However, knowledge of all the relevant documents does not solve the difficulties related to the fact that documents can be relevant to different degrees, and this problem is solved by ranking. We create a two-stage pass for retrieval. First, we use a VoyageAI embedding-based retrieval with a high top-k value in order to maximize recall and get a large set of candidate items. Then, we use the `bge-reranker-large` reranker-based retrieval to further dynamically select the nodes that are actually relevant to the query.

Embedding	Reranker	Hit Rate	mrr
OpenAI	WithoutReranker	0.893805	0.702212
OpenAI	bge-reranker-base	0.938053	0.786504
OpenAI	bge-reranker-large	0.938053	0.795870
Voyage	bge-reranker-base	0.933628	0.782080
Voyage	bge-reranker-large	0.933628	0.804941

Table 1: Embeddings and Reranker Evaluations from 10 Sample Documents

We derived these metrics by running a customized script of Embedding and Reranker combinations[9]. Of these, our baseline and unoptimized model lies in the first row, using OpenAI embeddings with no Reranker and a default `gpt-3.5-turbo` OpenAI LLM. From there, we upgraded to OpenAI priced models, and experimented with different modifications. Specifically, out of all synthetic queries, we had the highest accuracy with the last line, our current model. In parallel with the synthetic generation, we ran ~70 handmade queries[2]. While each query was not able to be analyzed on a fine-tuned scale, we use the culmulative statistics to obtain a broad overview of the results. More details on the response evaluation using our current model will be mentioned in the following section.

Additionally, we ran 5 harder hand-made queries to evaluate the behavior of our model on a fine-tuned scale. Among those, we had single and multiple document identification queries, T/F questions, summarization questions, and more complex queries such as document-hopping questions and comparison questions. The results and analysis of these queries are mentioned more in the error analysis section.

2.2 Response Evaluation

To summarize, we ran our model across 6 different configurations (Data embedding model, Vector Database, User Input Embedding Model, LLM, Reranker):

1. `OpenAI ada-2`, `Qdrant`, `OpenAI ada-2 ,` `google/flan-t5-large; temperature = 0.3, max_length = 2048`, `No`
2. `OpenAI ada-2`, `Qdrant`, `OpenAI ada-2`, `OpenAI (from langchain.llms)`, `No`
3. `OpenAI ada-2`, `Qdrant`, `OpenAI ada-2`, `OpenAI and metadata`, `No`
4. `VoyageAI\verbLlamaIndex+`, `LlamaIndex`, `OpenAI and title/abstract only`, `No`
5. `VoyageAI\verbLlamaIndex+`, `LlamaIndex`, `gpt-3.5-turbo from OpenAI`, `Yes`
6. `VoyageAI\verbLlamaIndex+`, `LlamaIndex`, `gpt-3.5-turbo from OpenAI and ChatBot`, `Yes`

Evolution of Configurations: We first experimented with different methods for embedding. A study[8] observed that the best performance typically arises when crucial data is positioned at the start or conclusion of the input context. Thus, we initially hosted a dozen or so documents on our mounted google drive and manually added title, dates, and author text to the beginning of each document, chunking each document into paragraphs (size=512, 1024). However, under our configuration settings the model still did not show noticeable improvement. We then switched to pulling documents from the web, which did not impact performance. We next experimented with different learning models. We initially used Hugging Face's `google/flan-t5-large; temperature = 0.3, max_length = 2048` and later OpenAI. While the former was free, among the three supported dataset libraries in Hugging Face, none were related to the category of Document QA and hence did not perform well. As for the embedding architecture, we found that VoyageAI outperformed Qdrant as a vector store knowledge base, and replaced it. Finally, we use the VoyageAI embeddings to transform a user query into question embeddings and run it through the LlamaIndex document retriever (with $k = 10$ documents) and further a reranker (with $k = 5$ documents) to retrieve the appropriate results. We run this through another OpenAI chatbot to produce a single English answer, and output the list of retrieved documents alongside it. All code and results were ran and displayed on Colab.

We have summarized the results of running each model through our 70 handmade queries. below.

Table 2: Single Paper Prompts

Model	AC	IR	WA	Grand Total	Accuracy
Hugging Face	7	1	5	13	53.85%
OpenAI	12	0	1	13	92.31%
OpenAI + metadata	9	1	4	14	64.29%
Voyage + Reranker	12	0	0	12	100.00%
Voyage + Reranker + ChatGPT	11	0	1	12	91.67%
Voyage + title/abstract only	2	0	1	3	66.67%
Grand Total	53	2	12	67	79.10%

Using our current model (`Voyage+Reranker+ChatGPT`), we note a few trends in the underlying data. In particular, the first run of our model was able to differentiate authors and identify dates, which were historically challenging problems for our other LLMs. Although we have not proven this holds across all runs, we noticed that the addition of metadata in the Reranker allows us to access fields exclusive to the data, such as date, title and author. We plan to conduct further extraction of metadata, as described in our future work section. Next, the ChatBot sometimes gave incorrect statements, even

Table 3: Across Paper Prompts

Model	AC	IR	WA	Grand Total	Accuracy
Hugging Face	0	0	10	10	0.00%
OpenAI	0	4	7	11	0.00%
OpenAI + metadata	2	6	4	12	16.67%
Voyage + Reranker	0	0	4	4	0.00%
Voyage + Reranker +ChatGPT	1	0	3	4	25.00%
Voyage + title/abstract only	8	0	4	12	66.67%
Grand Total	11	10	32	53	20.75%

Table 4: Single Paper Summarization Prompts

Model	AC	IR	WA	Grand Total	Accuracy
Hugging Face	1	0	8	9	11.11%
OpenAI	9	0	0	9	100.00%
OpenAI + metadata	7	0	1	8	87.50%
Voyage + Reranker	7	0	1	8	87.50%
Voyage + Reranker + ChatGPT	7	0	1	8	87.50%
Voyage + title/abstract only	0	0	0	0	0.00%
Grand Total	31	0	11	42	73.81%

Table 5: Single Paper General Author Prompts

Model	AC	IR	WA	Grand Total	Accuracy
Hugging Face	0	0	5	5	0.00%
OpenAI	8	0	1	9	88.89%
OpenAI + metadata	6	0	3	9	66.67%
Voyage + Reranker	11	0	16	27	40.74%
Voyage + Reranker + ChatGPT	11	0	16	27	40.74%
Voyage + title/abstract only	4	0	9	13	30.77%
Grand Total	40	0	50	90	44.44%

hallucinating on some queries (as seen in the analysis section) when the correct documents were already retrieved. Like our other LLMs, our current model worked best for single document queries. The spreadsheet for which all our LLMs were evaluated from past to present will be linked below.

2.2.1 Fine-Tuning Results and Attempted Modifications

Our model currently returns a fixed k documents. Because we are querying over a test set of 10 documents[3], this does not negatively impact our accuracy, but in the future we would like to expand our model to try other modifications. Currently, there are supported methods in LlamaIndex for returning a threshold value, although trying this is predicted to perform worse for retrieved documents since the range of values can vary greatly. We also previously attempted to modify code in the k-neighbors algorithm to return a dynamic set of documents based on clustering. Although the current black-box implementation of returning a fixed $k = 5$ documents from reranking in LlamaIndex works for us, we have a couple ideas for modifications for the future. For example, to select the optimal number of documents for a larger test set, we can consider subtracting 0.5 units from the optimal score, and returned the set of documents within that range. In the context of beam search, this is more optimal than setting a threshold, which can vary across specific queries.

One notable feature is that currently our model returns the top 5 documents with highest text similarity; this may include surfacing repeated documents if there are multiple hits in a single document.

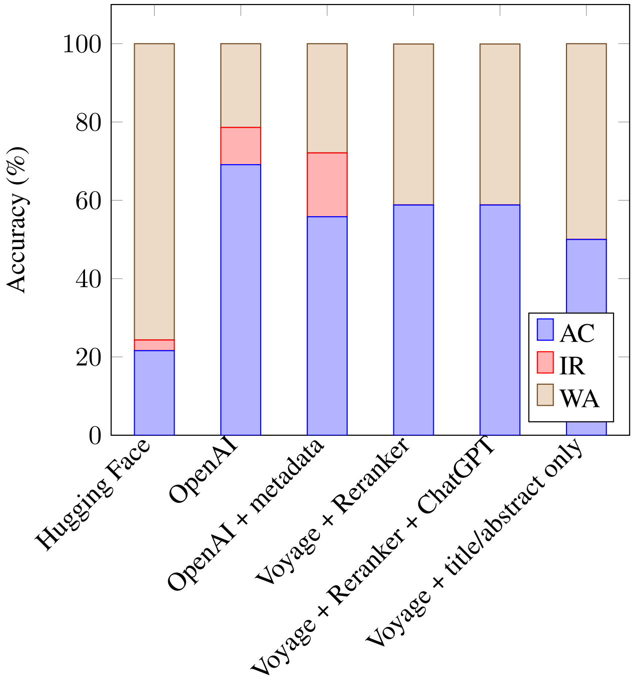

Figure 2: Human assessments for 70 Question Generation Tasks

3 Error Analysis

3.1 Approach

The model outputs the top k pieces of text (a passage in a document) that are most relevant to the query, together with the score. Therefore, a relevant metric to use for evaluating accuracy is the mean reciprocal rank (MRR).

$$\mathrm{MRR} = \frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{\mathrm{rank}_i}$$

For a single query, the reciprocal rank is $\frac{1}{\mathrm{rank}}$, where rank is the rank position of the first relevant document (if no correct answer was returned in the query, then the reciprocal rank is 0). The metric MRR take values from 0 (worst) to 1 (best).

We could also use the simpler form of accuracy. If the most relevant text is ranked to be the highest, then we label the response as correct, otherwise we label it as incorrect and finally calculate the percent accuracy.

In addition to evaluating the quality of the answers themselves, we run these queries on classical retrieval systems as per our original goal. Note that while we aim to compare the two, training our model to answer questions about specific NeurIPS papers is placed in a different context than running them on classical web engines such as Google, which stores more information and may sometimes return extraneous data.

To test the structure of our model, we designed and tested different types of fine-tuned queries: T/F fact-checking questions, single and multiple document queries, summarization or implication, and more complex queries.[5]

The model is able to perform quite well on most single paper queries, such as identifying objectives, extracting details, and asking for related topics. We then aimed to test the model's ability with answering queries involving multiple papers, and more general and conceptual questions. The model has some limitations with this type of queries, as described below.

3.1.1 Analysis

We based our handmade queries on a selection of single paper and multi-paper topics, the results and analysis of which are shown below. Inside the parentheses we include the zero-based rank of the most relevant text, title of the paper, and the score. We denote `rr` for the reciporcal rank without the mean, in relation to one specific query.

Query 1 Who are the authors (primary) in “DATACOMP: In search of the next generation of multimodal datasets”?

Model result 1 $\mathrm{rank} = 1, \mathrm{rr} = 1$ (0 datacomp: In search of the next generation of ... 0.995665)

Then we have some more complex queries that the model doesn’t perform well on:

Query 2 What key experiments were made in the paper that discussed PointMaze?

Model result 2 $\mathrm{rank} = 1, \mathrm{rr} = 1$ (0 f-Policy Gradients: A General Framework for Go... 0.508841)

For single paper queries, the model performs quite well and has a mmr close to 1. We then transitioned to evaluating accuracy. With our `VoyageAI + Reranker + ChatGPT` model, this gave a 91.67% accuracy.

We get 0.6 mmr on the 10 multi-paper queries that we ran. Since we are running on just 10 papers, which is a relatively small dataset, the model usually either gets the right paper at the top 1 or 2 rank, or not at all. We also analyzed the accuracy, which is 50%.

Next, we picked out a few more complex queries that the model fails to give a satisfactory answer on. These are the limitations of our model that we described.

(1) Not working very well with multi-paper queries.

Query Which papers are on optimization from IFML?

Classical response

> - A Survey of Optimization Methods from a Machine Learning Perspective, Shiliang Sun, Zehui Cao, Han Zhu, Jing Zhao.
> - Seminar: Finite-Sum Coupled Compositional Optimization: Theories and Practical Applications

Analysis The classical query returns papers outside of our databases, and includes irrelvant results like seminars.

Model response

> - Paper: "Finite-Time Logarithmic Bayes Regret Upper Bounds" - Score: 0.033854008
>
> - Paper: "Finite-Time Logarithmic Bayes Regret Upper Bounds" - Score: 0.025931083
>
> - Paper: "’f-Policy Gradients: A General Framework for Goal Conditioned RL using f -Divergences" - Score: 0.0023583106
>
> - Paper: "f-Policy Gradients: A General Framework for Goal Conditioned RL using f -Divergences" - Score: 0.00053475477

Analysis For this query, the model is actually correct. We generally observe that the model tends to output different parts of the same paper for the top k most relevant matches, which occurs if there are more than one hit per paper. The scores are also very low, making the results less credible.

(2) Not picking up more important details of the question.

Query Have Alexandros G. Dimakis and Constantine Caramanis worked on the same paper? If so, which ones?

Classical response

> - Constantine Caramanis
> - Full list of publications · Constantine Caramanis
> - Constantine Caramanis - Home - ACM Digital Library
> - Alexandros Dimakis University of Texas at Austin | UT

Analysis The classical query has limited ability in being able to identify the two authors in conjunction.

Model response

> Alexandros G. Dimakis and Constantine Caramanis have worked on the paper "Consistent diffusion modelsMitigating sampling drift by learning to be consistent".

Analysis Wrong answer. Alexandros G. Dimakis and Constantinos Daskalakis worked on this paper, but we asked for Constantine Caramanis, which has different last names. The model has a problem with distinguishing similar names. We conclude that the cosine similarity was too weak to surface the true output.

(3) Limitation: cannot do inference / summary very well.

Query What area of research does Adam Klivans work on in general? Give some examples of paper he worked on and their summaries?

Classical response

> - `https/www.cs.utexas.edu > people > faculty-researchers`
>   “His research interests lie in machine learning and theoretical computer science, in particular, Learning Theory, Computational Complexity, Pseudorandomness, ..."

Analysis The classical query is able to answer the first part of the question. However, it does not provide sufficient text to answer the subsequent parts of the question.

Model response

> Adam Klivans’ area of research is not mentioned in the given information.

Analysis We revisit this error from A5 error analysis, under the category of Incomplete Context. The top 5 surfaced texts are from Ambient Diffusion, a correctly surfaced paper. However, the model’s top matched text has a score of 0.000324. It doesn’t know how to answer the query because it cannot find textual evidence for the author. This is because the model is unable to see the big picture context; when the context is split during conversion, the model fails to generalize the meaning of section headers and titles.

Query I want to learn about the latest researches on image processing. Which papers talk about them?

Classical response

> - NeurIPS 2023 Papers
> - A Guide to NeurIPS 2022 — 10 Topics and 50 Papers You ...

Analysis The classical query does not provide specific papers on Image Processing.

Model response

> - Paper: "Image Reconstruction without Explicit Priors" by Angela F Gao, Oscar Leong, He Sun, and Katherine L Bouman. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (page 1).

Analysis The highest scored output is only 0.006395. The aforementioned paper in the response also comes from the reference section, not from the original a paper itself or any other paper in the database. This shows our model cannot distinguish the importance of certain papers from others.

4 Conclusion and Future Work

4.1 Issues

To recap, our model was able to outperform most of our previous models, and gave promising output on topic-based questions as well as single-document queries. Our analysis shows that the biggest issue of our model currently lies in lossy data conversion, and we should look into methods that capture meaning from PDF to text. As for other issues, we initially ran into some trouble with funding and OpenAI tokens, but resolved this by directing payment to the IFML account.

4.2 Next steps

The next steps are broadening our test set and adding additional functionality to our code. In the future, we would like to be able to parse metadata; for example, using libraries in LlamaIndex can help with title and date identification [6]. To add more context, we are also interested in extending to multimodal LLMs by training on images in addition to text [7]. We plan on hosting a prototype of our model on a live server within the near future.

5 Supplementary Material

Please see code and references, attached externally.

References

[1] “Retrieval Evaluation." Retrieval Evaluation - LlamaIndex 0.9.11.
docs.llamaindex.ai/en/latest/examples/evaluation/retrieval/retriever_eval.html. Accessed 3 Dec. 2023.

[2] Link to Google Drive Analysis Spreadsheet

[3] https://arxiv.org/pdf/2310.06794v1.pdf
https://arxiv.org/pdf/2310.06794v1.pdf
https://arxiv.org/pdf/2305.19256.pdf
https://arxiv.org/pdf/2302.09057.pdf
https://arxiv.org/pdf/2305.11765.pdf
https://arxiv.org/pdf/2307.01178.pdf
https://arxiv.org/pdf/2310.12979.pdf
https://arxiv.org/pdf/2304.14108.pdf
https://arxiv.org/pdf/2307.00619.pdf
https://arxiv.org/pdf/2306.09136.pdf

[4] https://arxiv.org/pdf/2005.11401.pdf

[5] Link to Manually Generated Queries Document

[6] Link to Metadata Processing Software

[6] Link to Image Processing Software

[7] https://arxiv.org/abs/2307.03172

[8] Script for running Embeddings and Rerankers