AIE4 Midterm Challenge Report

Submitted by Daniel Gutierrez
gutierrez.daniel.v@gmail.com
September 24, 2024

Task 1:

1. Describe the default chunking strategy that you will use.
    a. The default chunking strategy will be using the text splitter and chunking by length. This is the most common and basic form of chunking. Using the RecursiveCharacterTextSplitter with a chunk size of 750, overlap of 25 and length function to len this is a great first approach to chunking pdf documents. This chunker will automatically chunk at punctuation to not miss any information and an overlap of 25 is a safety measure to not miss any critical contextual information. After examining each of the documents and doing a quick measurement on paragraph length and document organization a chunk size of 750 is a safe number. It may seem large but it might turn out to be useful
2. Articulate a chunking strategy that you would also like to test out.
    a. The alternative chunking strategy that I would like to test out is chunking by paragraph or section based on the document's structure. Based on the initial review of the documents, a paragraph chunker might have an advantage because this is a government document and the wording is concise and straight to the point in each section and paragraph.
3. Describe how and why you made these decisions
    a. As stated in the first two questions, an initial review of both documents was completed. Because we are dealing with PDF reports, a basic chunking method may be all that is needed for this type of RAG system. A alternative chunking strategy may be implemented based on the structure of the two documents but i believe the changes would have a significant effect on the evaluation results that will follow.

Task 2:

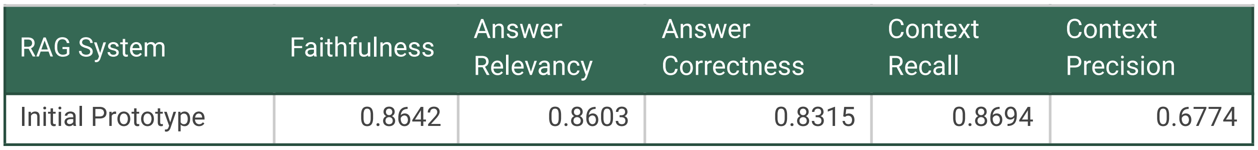
1. Build a prototype and deploy to a Hugging Face Space, and create a short (< 2 min) loom video demonstrating some initial testing inputs and outputs.
    a. Link to Prototype:
        i. https://huggingface.co/spaces/dgutierrez/aie4_midterm
2. How did you choose your stack, and why did you select each tool the way you did?
    a. This is a basic RAG system for document-based question-and-answer retrieval. I used fundamental building blocks to quickly create an easy-to-evaluate prototype as a proof of concept.
        i. Chunking: TextSplitter
        ii. Embedding Model: OpenAi/text-embedding-3-small
        iii. Vector Store: Quadrant
        iv. UX: Chainlit

v. Deployment: Hugging Face Spaces

Task 3:

1. Assess your pipeline using the RAGAS framework including key metrics faithfulness, answer relevancy, context precision, and context recall. Provide a table of your output results.

Initial RAG POC Evaluation Results:



RAG System	Faithfulness	Answer Relevancy	Answer Correctness	Context Recall	Context Precision
Initial Prototype	0.8642	0.8603	0.8315	0.8694	0.6774

2. What conclusions can you draw about performance and effectiveness of your pipeline with this information?
    a. The system performs well in terms of faithfulness, answer relevancy, and context recall, meaning it retrieves mostly correct and relevant information. However, the lower context precision score indicates it sometimes retrieves too much unnecessary information, which might reduce efficiency or clarity in the answers.

Task 4:

1. Swap out your existing embedding model for the new fine-tuned version. Provide a link to your fine-tuned embedding model on the Hugging Face Hub.
    a. Link to Fine-Tuned BGE model
        i. https://huggingface.co/dgutierrez/midterm_finetuned_bge
2. How did you choose the embedding model for this application?
    a. At this time, fine-tuning the OpenAI embedding model isn't an option. However, from previous assignments, I've had experience fine-tuning the Snowflake Arctic model, which yielded promising results. After researching similar models, I discovered the BAAI/bge-base-en-v1.5, which has received positive feedback from the AI community, with numerous attempts at fine-tuning it. It's also relatively lightweight, with only 109M parameters, making it a good candidate for experimentation in this type of project.That said, I believe fine-tuning this model may not provide significant benefits because the context of this project is largely available to the public, and the information isn't particularly niche.

Task 5:

1. Test the fine-tuned embedding model using the RAGAS frameworks to quantify any improvements. Provide results in a table.

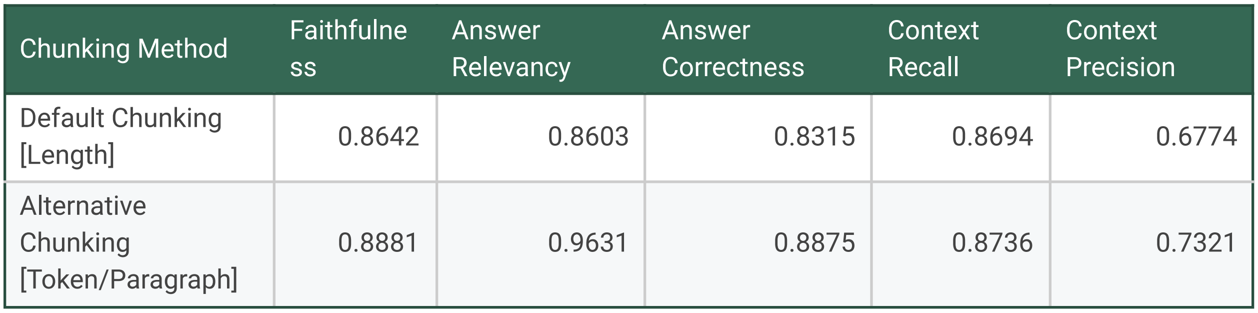
Three base models were evaluated against the Fine-Tuned BAAI model.

Model	Faithfulness	Answer Relevancy	Answer Correctness	Context Recall	Context Precision
OpenAi/ text-embedding-3-small	0.8761	0.9211	0.7805	0.8345	0.8028
Snowflake/ snowflake-arctic-embed-m	0.8102	0.9665	0.6901	0.9229	0.7764
BAAI/ bge-base-en-v1.5	0.8713	0.9222	0.7428	0.8345	0.8028
Fine-Tuned BAAI/ bge-base-en-v1.5	0.7918	0.9186	0.7244	0.9229	0.7764

Summary of Fine Tuning RAGAs Results

1. **Best for Faithfulness and Relevancy**: The OpenAI/text-embedding-3-small model performs well in keeping answers faithful to the source and highly relevant.
2. **Best for Recall**: The Snowflake/snowflake-arctic-embed-m model excels at retrieving the most relevant information but suffers in correctness and faithfulness.
3. **BAAI/bge-base-en-v1.5** (original and fine-tuned) offers a balance between these two models but shows that fine-tuning didn't bring significant improvements across most metrics, and in some cases, slightly worsened performance.

2. Test the two chunking strategies using the RAGAS frameworks to quantify any improvements. Provide results in a table.



Chunking Method	Faithfulness	Answer Relevancy	Answer Correctness	Context Recall	Context Precision
Default Chunking [Length]	0.8642	0.8603	0.8315	0.8694	0.6774
Alternative Chunking [Token/Paragraph]	0.8881	0.9631	0.8875	0.8736	0.7321

Comparison and Summary of Chunking RAGAs Results

1. **Faithfulness**: The Alternative Chunking method scores better, meaning it generates answers more closely aligned with the retrieved content.
2. **Answer Relevancy**: The Alternative Chunking is vastly superior, providing more relevant answers compared to the default method.
3. **Answer Correctness**: Again, Alternative Chunking performs much better, producing more accurate responses.
4. **Context Recall**: Both chunking methods perform similarly, with Alternative Chunking having a slight edge in retrieving the relevant information.
5. **Context Precision**: Alternative Chunking outperforms the default method, retrieving less irrelevant information, though there’s still room for improvement.

Conclusion

The Alternative Chunking [Token/Paragraph] method clearly outperforms the Default Chunking [Length] method in almost every metric. The results suggest that chunking by tokens or paragraphs leads to more faithful, relevant, and correct answers, while also slightly improving context recall and significantly enhancing context precision. Therefore, Alternative Chunking would be the better choice for improving overall performance in your RAG system.

3. The AI Solutions Engineer asks you “Which one is the best to test with internal stakeholders next week, and why?”
    a. To enhance the performance of our RAG system, I recommend using the OpenAI/text-embedding-3-small model in combination with the Alternative Chunking [Token/Paragraph] method. This combination provides the best balance of relevance, accuracy, and retrieval precision for our project, leading to more reliable and actionable results.There are further experimentations that we can assess in the future; however I don’t see any major changes having a major impact. For example the time and cost of fine tuning a model is overkill for this type of application. I would explore one or two more chunking strategies and move on.

Task 6:

1. What is the story that you will give to the CEO to tell the whole company at the launch next month?
    a. The world has changed a lot in just one year. With the launch of OpenAI's popular app, ChatGPT, the way we handle information has been transformed. Things like writing emails, creating photos, copying faces, mimicking personalities, and even coding can now be done in seconds. This raises important questions: What can AI do, and more importantly, what should AI do? It's a bit like a scene from the movie *Jurassic Park*, where a scientist wonders if bringing dinosaurs back is a good idea. The big question was: Just because you can, does it mean you should? People often fear change, and with AI being such a powerful and disruptive technology, it's natural for people to have many questions. To help with this, we've created a tool to answer your questions about the future of AI, how it may be regulated, and what policies might shape it. AI has the power to greatly improve lives, but like any tool, it can also be misused. This application is here to keep you informed about the future of AI, providing you with clear and helpful information on how AI might impact our world.
2. There appears to be important information not included in our build, for instance, the 270-day update on the 2023 executive order on Safe, Secure, and Trustworthy AI. How might you incorporate relevant white-house briefing information into future versions?
    a. We can easily fetch information using a specialized agent that checks for important and relevant information coming off the White House website or any other government site. Also if we have access to the document we can easily update it into our pdf folder where it will be chunked and embedded during the next launch of the application.

Final Deliverables:

1. A public link to a written report addressing each deliverable and answering each question.
    a. AIE4_Midterm_Challenge_Report_Daniel_Gutierrez
    b. Also pdf version in repo
2. A public link to any relevant GitHub repo
    a. https://github.com/dgutierrez24/Midterm
3. A public link to the final version of your application on Hugging Face
    a. https://huggingface.co/spaces/dgutierrez/aie4_midterm
4. A public link to your fine-tuned embedding model on Hugging Face
    a. https://huggingface.co/dgutierrez/midterm_finetuned_bge
5. Loom Video with Demo
    a. https://www.loom.com/share/3b49971c498149bf982e86b735d03ecf?sid=1b0c6031-a9c3-4290-8d97-cbd31fa905c5

Some Sample Questions to ask the Prototype:

From the "Blueprint for an AI Bill of Rights":

1. What measures are proposed in the AI Bill of Rights to ensure data privacy for individuals?
2. How does the AI Bill of Rights address the issue of AI accountability in decision-making systems?
3. What are the specific protections mentioned for marginalized communities against AI-driven discrimination?
4. What guidelines does the AI Bill of Rights provide for obtaining informed consent in the use of AI technologies?
5. How does the AI Bill of Rights propose to handle errors or biases in automated decision-making systems?

From the "NIST AI 600-1" document:

1. What are the key challenges NIST identifies in ensuring the trustworthiness of generative AI systems?
2. How does NIST recommend mitigating risks associated with the deployment of large language models (LLMs)?
3. What steps does the NIST AI 600-1 suggest for improving transparency in generative AI outputs?
4. How does the NIST framework propose to handle risks related to confabulation in generative AI systems?
5. What are the recommended actions in the NIST framework to manage security risks in generative AI?