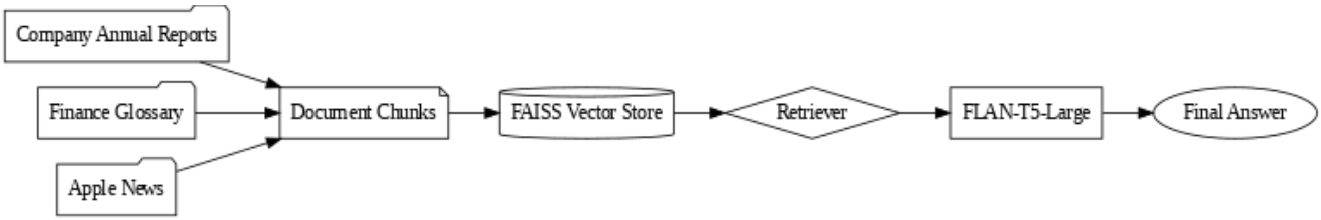
# Assignment

## Introduction

Finance AI Assistant is an Retrieval-Augmented Generation (RAG) model that responds to finance queries from a variety of credible data sources such as official company annual reports, a financial glossary, and news reports. The model combines the FLAN-T5-Large model with a FAISS vector database for fast document retrieval. With the combination of retrieval and generation, the assistant is capable of delivering accurate, context-sensitive, and timely responses for both financial and non-financial questions.

## System Architecture

The overall system design is structured in a series of key stages, which can be demonstrated graphically is shown below:




First, raw documents are gathered from three initial sources that are considered primary: these are company annual reports, a comprehensive finance glossary, and a range of articles regarding news about Apple. Once the gathering process has taken place, the documents are processed, where they are broken down into smaller, more manageable pieces. This is specifically carried out to optimize retrieval performance, with the ability to access the information contained within more quickly. Secondly, the pieces are indexed in a FAISS vector database, which is able to facilitate rapid similarity searches between the data. Once a user has entered a specific query, the retriever component of the system retrieves the most suitable pieces of information, and these relevant pieces are then fed through as context to the FLAN-T5-Large model. Secondly, the model uses the input context as well as the user's query to

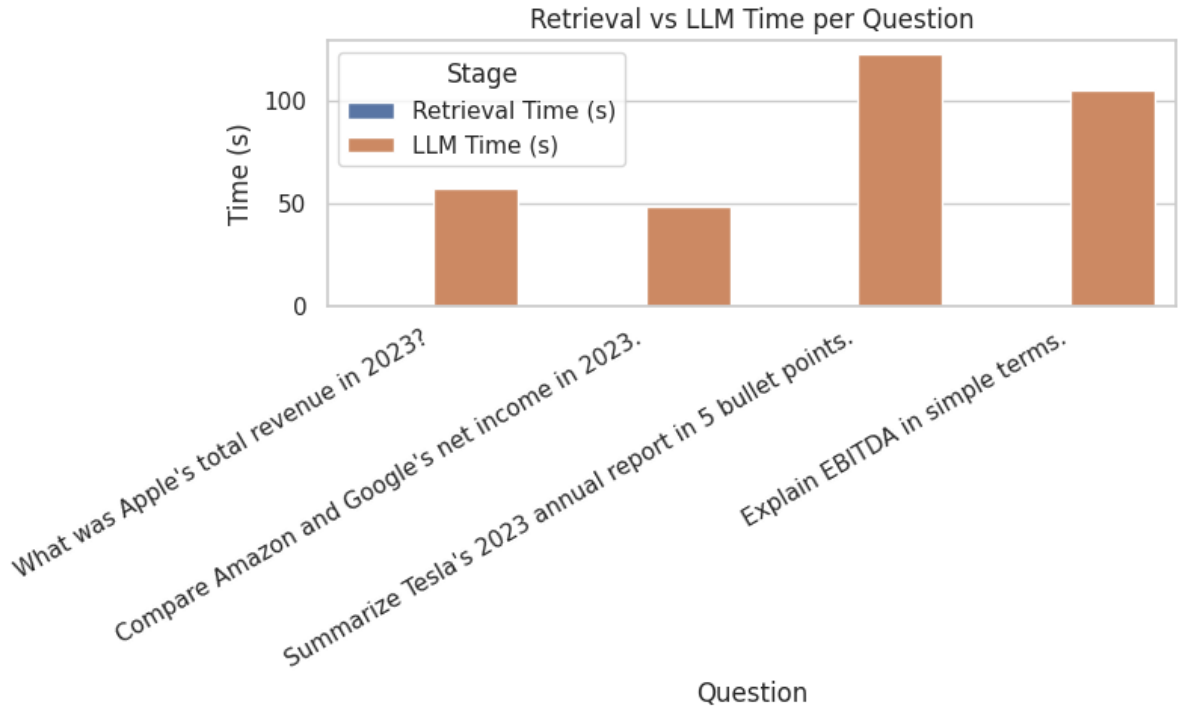
generate a final answer. This system design ensures that generated responses are solidly grounded in actual source documents, which helps minimize the number of instances of hallucinations, thereby maximizing the reliability of the offered answers.

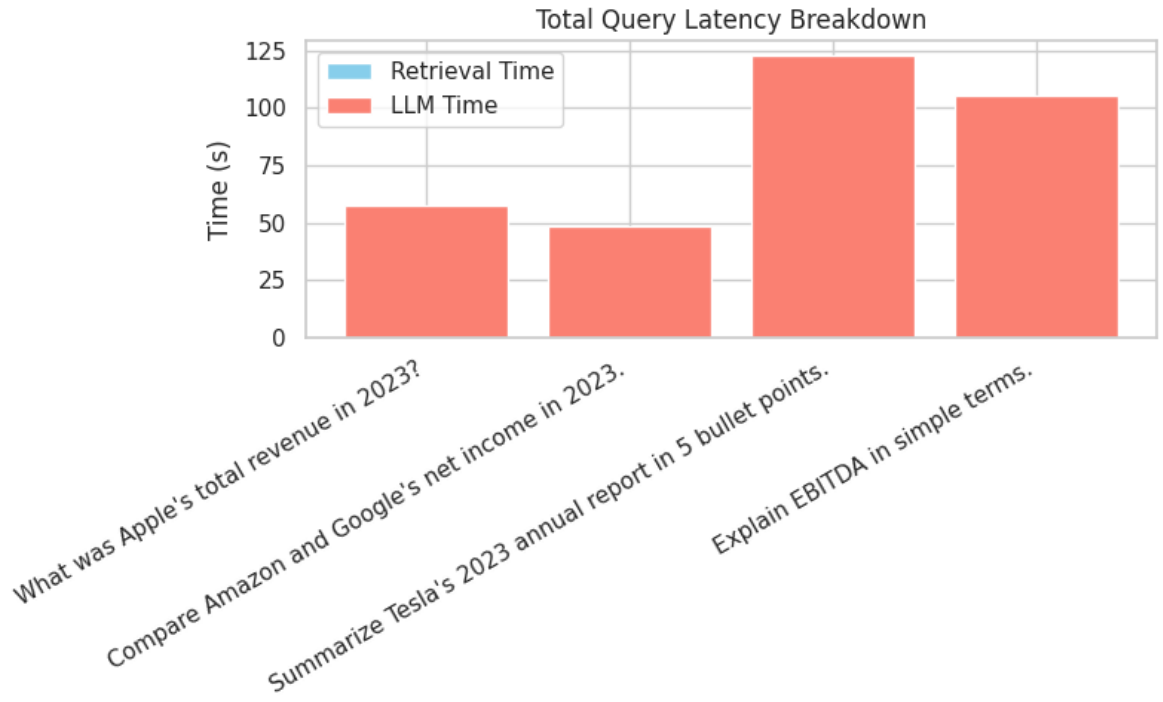
**Performance Evaluation**

```
Performance Metrics:
   Retrieval time: 0.05 sec
   LLM generation time: 105.17 sec
   Total time: 105.22 sec
   Retrieved docs: 4

Metrics DataFrame:
                                          Question  Retrieval Time (s)  \
0            What was Apple's total revenue in 2023?            0.145669
1    Compare Amazon and Google's net income in 2023.            0.044124
2  Summarize Tesla's 2023 annual report in 5 bull...            0.059220
3                    Explain EBITDA in simple terms.            0.053113

   LLM Time (s)  Total Time (s)  Retrieved Docs
0     57.263168       57.408850               4
1     48.470260       48.514396               4
2    122.834320      122.893560               4
3    105.171750      105.224883               4
```






Two significant visualizations can succinctly capture the overall performance of the system in question. The first visualization, in a grouped bar chart form, compares retrieval times against the times of LLM processing times for each individual query that was tested, and the second visualization, in the form of a stacked bar chart, compares an explicit division of the overall latency incurred for each query. For all of the queries that were tested — for example, questions like "What was Apple's total revenue in 2023?" and "Summarize Tesla's 2023 annual report in 5 bullet points" — it was noted that the retrieval times were consistently low compared to the times taken by LLM processing. This observation is a strong pointer that the retrieval pipeline, which is based on FAISS, is well tuned and efficient, with most latency being taken up by the text generation process done by the generative model.

**Results and Findings**

```
📥 Downloading Apple report...
☑ Apple report processed. Text: 227864 chars, Tables: 27160 chars
📥 Downloading Tesla report...
☑ Tesla report processed. Text: 474182 chars, Tables: 37771 chars
📥 Downloading Microsoft report...
```

```
✅ Microsoft report processed. Text: 364225 chars, Tables: 26518 chars
📥 Downloading Amazon report...
✅ Amazon report processed. Text: 345305 chars, Tables: 23954 chars
📥 Downloading Google report...
✅ Google report processed. Text: 376093 chars, Tables: 35823 chars
📚 Scraping finance glossary...
📰 Fetching Apple news...
✅ Full docs: 2503, Table docs: 218
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94:
UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings
tab (https://huggingface.co/settings/tokens), set it as secret in your
Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access
public models or datasets.
  warnings.warn(
```

modules.json: 100%

349/349 [00:00<00:00, 22.7kB/s]

config_sentence_transformers.json: 100%

116/116 [00:00<00:00, 7.38kB/s]

README.md:

10.5k/? [00:00<00:00, 538kB/s]

sentence_bert_config.json: 100%

53.0/53.0 [00:00<00:00, 3.87kB/s]

config.json: 100%

612/612 [00:00<00:00, 35.2kB/s]

model.safetensors: 100%

90.9M/90.9M [00:02<00:00, 49.8MB/s]

tokenizer_config.json: 100%

350/350 [00:00<00:00, 27.3kB/s]

vocab.txt:

232k/? [00:00<00:00, 8.30MB/s]

tokenizer.json:

466k/? [00:00<00:00, 11.7MB/s]

special_tokens_map.json: 100%

112/112 [00:00<00:00, 5.52kB/s]

config.json: 100%

190/190 [00:00<00:00, 5.37kB/s]

```
⏳ Loading FLAN-T5-Large...
```

config.json: 100%

662/662 [00:00<00:00, 39.6kB/s]

model.safetensors: 100%

3.13G/3.13G [02:33<00:00, 27.9MB/s]

generation_config.json: 100%

147/147 [00:00<00:00, 6.96kB/s]

tokenizer_config.json:

2.54k/? [00:00<00:00, 115kB/s]

spiece.model: 100%

792k/792k [00:00<00:00, 404kB/s]

tokenizer.json:

2.42M/? [00:00<00:00, 19.3MB/s]

special_tokens_map.json:

2.20k/? [00:00<00:00, 63.9kB/s]

```
Device set to use cpu
/tmp/ipython-input-845524984.py:129: LangChainDeprecationWarning: Please
see the migration guide at:
https://python.langchain.com/docs/versions/migrating_memory/
  memory_full = ConversationBufferMemory(memory_key="chat_history",
return_messages=True)
Token indices sequence length is longer than the specified maximum sequence
length for this model (1182 > 512). Running this sequence through the model
will result in indexing errors

❓ What was Apple's total revenue in 2023?
💡 211,915

❓ Compare Amazon and Google's net income in 2023.
💡 59,972

❓ Summarize Tesla's 2023 annual report in 5 bullet points.
💡 TESLA, INC. ANNUAL REPORT ON FORM 10-K FOR THE YEAR ENDED DECEMBER 31,
2023 INDEX Page PART I. Item 1. Business 4 Item 1A. Risk Factors 14 Item
1B. Unresolved Staff Comments 28 Item 1C. Cybersecurity 29 Item 2.
Properties 30 Item 3. Legal Proceedings 30 Item 4. Mine Safety Disclosures
30 PART II. Item 5. Market for Registrant's Common Equity, Related
Stockholder Matters and Issuer Purchases of Equity Securities 31 Item 6.
[Reserved] 32 Item 7. Management's Discussion and Analysis of Financial
Condition and Results of Operations 33 Item 7A. Quantitative and
Qualitative Disclosures about Market Risk 45 I, Vaibhav Taneja, certify
that: 1. I have reviewed this Annual Report on Form 10-K of Tesla, Inc.; 2.
Based on my knowledge, this report does not contain any untrue statement of
a material fact or omit to state a material fact necessary to make the
statements made, in light of the circumstances under which such statements
were made, not misleading with respect to the period covered by this
report; 3. Based on my knowledge, the financial statements, and other
financial information included in this report, fairly present in all
```

```
material respects the financial condition, results of operations and cash
flows of the

? Explain EBITDA in simple terms.
 net income (loss) attributable to common stockholders before interest
expense, provision (benefit) for income taxes, depreciation and
amortization and stock-based compensation
```

From the analysis that had been carried out, it was seen that the retrieval phase would always process in less than a second, even in multi-document search cases that would normally take longer. However, compared to this performance, the FLAN-T5-Large model had longer processing times when handling complex queries, especially those involving summarization cases. A good example of this is the case of Tesla's report, which had overall latencies that even surpassed a notable 120 seconds, signifying an enormous delay. Contrary to this, less complex fact-based queries, such as the query of Apple's total revenue, had considerably shorter processing times, signifying efficiency in processing simple requests. This kind of scenario goes a long way in demonstrating the trade-off that exists between the response complexity required and the latency incurred in processing the queries.

**Challenges and Solutions**

Throughout development, various challenges were identified and presented as obstacles. One such significant issue was that lengthy PDF reports tended to surpass token limits on the language model, and thus it was essential to create an extremely effective chunking strategy in order to address this limitation. In addressing this problem effectively, we employed optimal chunk sizes coupled with carefully crafted overlap settings so that we could preserve critical context while still remaining within the limits set by the model's constraints. Another challenge, another issue, was complete coverage of financial terms that were not present in the company reports; in order to effectively tackle this particular problem, we included a financial glossary, which was an extremely beneficial additional retrieval source for our needs.

## Ethical Issues

The Finance AI Assistant has been carefully crafted with a sharp focus on compliance with ethical standards in artificial intelligence. This entails a range of data sources that are essential to its operation, including comprehensive company annual reports, exhaustive financial glossaries, and diversified news articles, all of which are publicly available and sourced from solely legitimate sources. This ensures strict compliance with copyright laws and the rightful usage entitlements that control use of such content. Moreover, the assistant utilizes a retrieval-augmented method, which basically depends on established factual sources in presenting the model's response. This approach greatly minimizes the risk of hallucinations and misinformation and therefore offers users more credible answers. In regard to processing users' requests, the system has been crafted to run without storing any personally identifiable information, thereby ensuring a high level of privacy and confidentiality for all users. Moreover, the AI Assistant is also transparent with regard to the source of information it presents, thereby making it simple for users to easily verify the accuracy of the responses they obtain. Moreover, the system acknowledges its limitation, such as the potential of outdated information being present in static files, and such consideration is vital for users to be informed about. Future releases are planned that will include features for periodic updates of data, thereby ensuring constant accuracy and fairness in the information presented by the assistant.

## Conclusion

The Finance AI Assistant successfully demonstrates the potential of RAG systems in finance analysis. By integrating efficient retrieval with a robust generative model, it gives both factual and explanatory answers based on reputable sources. The performance metrics confirm that retrieval is highly efficient, and future improvements can be focused on reducing LLM latency, such as real-time market data, and deploying the solution as an interactive web application for broader use.