FinFeed RAG ChatBot/Executive Summary

Team Members: Aryama Singh, Diliya Yalikun, Korel Gundem, Nazanin Komeilizadeh, Roberto Nunez.
GitHub Link: FinFeed RAG ChatBot

Objective/Overview:

This project aims to develop an AI assistant that efficiently aggregates current news related to finance, economy, and politics from YouTube news channels within a specified timeframe. The assistant allows users to inquire about recent economic news and receive responses through a Retrieval-Augmented Generation (RAG) system. The system simultaneously presents a dynamic sentiment graph for each context relevant to the query and examines public opinions derived from YouTube comments.

Data Collection and Preprocessing:

1. **Downloading Videos as Audio Files:** Using YouTube Data API v3 we downloaded videos as audio files from YT channels. The following provides some examples: Yahoo Finance, Bloomberg Television, The Economist, Financial Times, Reuters, The Wall Street Journal, Al Jazeera English
2. **Converting Audio to Text:** The audio files were transcribed into text using OpenAI's Whisper model, a state-of-the-art automatic speech recognition system capable of transcribing and translating multiple languages.
3. **Enriching Data with Metadata:** We extracted video URLs, titles, channel names, publication times, and top 50 comments using YouTube Data API v3. This metadata was stored in a dataframe along with the transcribed text files.
4. **Preprocessing Text Data:** We removed English stop words and punctuation from the text files to focus on content-bearing words, reducing file size and facilitating the chunking of longer transcripts.
5. **Splitting Long Documents:** Using LangChain's recursive text splitter, we divided lengthy transcriptions into smaller chunks at paragraph breaks, maintaining semantic coherence.
6. **Vectorizing Text Chunks:** Each text chunk was vectorized using OpenAI's embeddings model to create vector embeddings for efficient processing.
7. **Pinecone Index:** The vectorized chunks were stored in a Pinecone index, a cloud-native vector database for high-dimensional vectors, enabling efficient similarity search and retrieval.
8. **Enriching Data with Metadata:** We enriched the data with associated YouTube metadata (channel name, video title, publication date, and comments), enhancing the accuracy and relevance of responses generated by the language model.

Modeling Approach:

In our modeling approach, chaining and context transmission to the LLM model is crucial for generating precise responses. After preprocessing and vectorizing text chunks, embeddings are stored in a Pinecone vector database. Upon receiving a user query, we use cosine similarity to identify and rank the most relevant text chunks efficiently.

These top-matching chunks are then chained together to form a cohesive context. LangChain, a framework for building applications with language models, facilitates this process by seamlessly integrating different components and ensuring efficient data flow.

The curated context is then sent to our LLM model, GPT-3.5 Turbo, chosen for its advanced natural language capabilities. By providing the model with rich and relevant input, we ensure accurate and contextually

appropriate responses. LangChain and the vector database work together to maintain a dynamic and responsive system, meeting high standards of information accuracy and relevance.

Evaluation:

To evaluate our models, we asked GPT-3.5 Turbo to generate 50 finance-related questions. We then used these questions to benchmark GPT-3.5 Turbo (our baseline) and our specialized model, FinFeed. Next, GPT-4o acted as an impartial evaluator, using a specific prompt to choose the most relevant answers between the two models. This comparison allowed us to determine which model—GPT-3.5 Turbo or FinFeed—provided the most accurate and relevant responses.

This process helped us benchmark our specialized model against a strong baseline, ensuring effective financial information delivery. The plot below shows the comparison between GPT-3.5 Turbo and FinFeed.

Results:

Using the above modeling approach and framework, this chatbot delivers the most current information in financial fields with proper citations. It also provides users with dynamic sentiment graphs for each context (news) based on the prompted query and analyzes the sentiments associated with public opinions on that news. We also developed a web app that interacts with users based on their queries. The chatbot's responses are then derived from the latest financial news.

Challenges & Possible Solutions:

1. **Transcription:** Transcribing audio files took a long time since we were doing it locally using only our CPU. We used async IO, a concurrent programming technique that allows us to handle multiple tasks.
2. **Upserting metadata:** Pinecone doesn't provide a straightforward function to get vector ids. We had to collect all ids and match it with the source transcription file to update it with YT metadata.
3. **Prompt templates:** Prompting is a crucial component in creating a RAG system. Since our system involved multiple LLMs, we employed various prompt techniques for different tasks. We experimented extensively with techniques such as Zero-Shot, One-Shot, and Chain-of-Thought prompting. Ultimately, Chain-of-Thought prompting yielded the best performance, particularly for sentiment analysis.
4. **Chaining of Multiple LLMs:** Chaining multiple LLMs in a RAG system introduced several challenges, such as increased latency, inconsistent outputs, and compounded errors. To mitigate this, parallel processing and optimizing model efficiency were employed. Inconsistent outputs were addressed by implementing robust validation and consensus mechanisms. Compounded errors escalated through the chain, necessitating error correction protocols and redundancy checks to ensure the reliability and accuracy of the final output. We are still working on this.

Future Directions:

Since our data collection from YouTube encompasses a wide variety of topics, our framework can be adapted to different subjects by altering its knowledge base, and a multimodal approach can be integrated at the initial data processing step to enhance the product's sophistication. We can further augment our model by:

- Making our current Chatbot multimodal by incorporating images generated from the videos and updating their embeddings to Pinecone index, based on transcription timestamps.
- Including an audio assistant to take input questions as audio and also output the generated answers as audio.
- Using BERTopic to cluster documents of similar topics together
- Using SERPs (Scrape Google Search Results) based on the user question, to augment the context data.