**MSBA 6461 Advanced AI for NLP**

**Assignment 2**

**RAG-powered Daily Financial News Summarizer with Q&A System**

**Executive Summary**

This project presents a small-scale implementation of a **Retrieval-Augmented Generation (RAG) pipeline** integrated into a **Streamlit web application**. The application automates the process of **fetching, clustering, and summarizing the day’s most important financial news**, delivering concise highlights and enabling interactive Q&A.

The system begins by scraping financial RSS feeds from **CNBC, Yahoo Finance, Investing.com,** and **Google News**. Retrieved articles are filtered for relevance using domain keywords (e.g., “market”, “inflation”, etc) and then clustered using KMeans on embedding vectors generated by OpenAI’s text-embedding model. For each cluster, the app **uses LLM to generate a topic headline and a concise English summary**. Representative article links are included to support further reading.

To ensure timely insights, the application caches fetched articles with their publication timestamps. **Only articles from the current day are used**, guaranteeing that summaries reflect fresh market events. The app also incorporates a lightweight RAG component: users can pose follow-up questions, and their queries are matched to the most relevant topic cluster using semantic similarity. **The best-matched topic’s content is then used to generate an answer via the LLM**.

Built with reproducibility in mind, the entire workflow — from data retrieval to summarization and Q&A — is **implemented in Python** and **packaged in a Streamlit UI**. The codebase is designed for easy extension and future integration with more robust vector databases or production-grade retrieval engines.

This project demonstrates how modern LLM technologies, combined with classical NLP pipelines and interactive dashboards, can transform unstructured market data into actionable insights for both casual investors and financial professionals.

**Project Objective**

The goal of this project is to implement a small-scale **Retrieval-Augmented Generation (RAG) system** that automatically:

1. Collects the latest financial news articles each day.
2. Generates concise summaries of the five most important topics.
3. Allows users to ask follow-up questions about these topics and receive context-aware answers.

This demonstrates how RAG can combine real-time data retrieval with large language models (LLMs) to provide up-to-date, contextually relevant responses.

**Technologies Used**

* **LLM**: OpenAI’s **o4-mini** model via the OpenAI Python API.
* **Embeddings**: OpenAI’s **text-embedding-3-small** to encode news articles and user queries.
* **RAG pipeline**:
  + **Retrieval**: Vector search over daily news topic embeddings.
  + **Augmentation**: Passing retrieved context into the LLM prompt.
* **Clustering**: KMeans to group related news into topics.
* **Frontend UI**: **Streamlit** to display summaries and provide a Q&A interface.
* **Data sources**: RSS feeds from **CNBC, Yahoo Finance, Investing.com**, and **Google News**.

**Data Pipeline Overview**

1. **Fetch News**:
   * Parse RSS feeds for today’s published articles.
   * Download full article content (if possible) with caching to avoid redundant HTTP requests.
   * Store article text, title, source, and publish time.
2. **Filter & Cluster**:
   * Embed articles with OpenAI embeddings.
   * Filter for finance-relevant keywords.
   * Cluster articles into 5 key market topics using KMeans on embeddings.
3. **Summarization**:
   * For each cluster, send representative articles to the LLM with a prompt instructing it to generate a short headline and 2-3 sentence summary.
   * Use regex & rules to extract headline and summary cleanly from LLM output.
4. **Q&A (RAG)**:
   * User question is embedded.
   * Similarity search retrieves the most relevant topic.
   * The question + topic context is combined in a prompt to the LLM to generate an answer.

**Key Implementation Details**

* **Cache Management**: Articles are cached with publication timestamps to ensure content freshness and avoid reprocessing outdated articles.
* **Embedding-based RAG**:
  + Each topic has an embedding representing combined cluster content.
  + Cosine similarity is computed between user question embedding and topic embeddings.
* **Prompting Strategy**:
  + Summary prompt instructs the LLM to avoid repeating article numbers/titles.
  + Answer prompt includes question and the selected topic context for accurate responses.

**Challenges & Learnings**

* **Parsing & Caching**: Managing dynamic content freshness was critical; articles were filtered by exact publish date and timestamps were compared with the cache.
* **LLM Response Variability**: Sometimes the model returned unexpected formats (missing headlines, duplicated summary labels), so regex and fallback logic were needed.
* **Cluster Consistency**: Checking each cluster’s thematic coherence with a second LLM call helped ensure better topic grouping, but increased latency.
* **Embeddings Quality**: Short news articles produced noisy clusters; combining title + partial content and using PCA or re-clustering could further improve.

**Future Improvements**

While this project demonstrates a functional prototype of a RAG-based financial news summarization and Q&A system, several enhancements could further improve performance, reliability, and user experience:

* **Use Vector Database**: Replace the in-memory embedding comparison with a proper vector database (e.g. Pinecone) to efficiently store, index, and query a large volume of historical news embeddings.
* **Improve Clustering Robustness**: Incorporate advanced clustering techniques such as HDBSCAN or hierarchical clustering with automatic cluster number detection to adapt better to varying amounts of daily news.
* **Dynamic Keyword Filtering**: Replace the current static keyword list with an NLP-based relevance classifier or named entity recognition to more accurately filter finance-related articles.
* **User Personalization**: Allow users to specify industry sectors (e.g., tech, energy) or keywords of interest (e.g., stock, bonds, commodities, foreign exchange) to receive personalized topic summaries.
* **Multi-language Support**: Extend the system to summarize and answer questions in multiple languages to reach a broader global audience.
* **Expand News Sources, Including Paid Feeds**: Integrate additional reputable and specialized news sources—including premium or subscription-only feeds—to access more precise, comprehensive, and exclusive information, which can significantly enhance the quality of generated summaries and topic insights.

**Results**

* A web-based application (built in Streamlit) displays 5 key market highlights for the day.
* Users can ask follow-up questions of the key news, such as:
  + *“What were the major banks mentioned in topic 2?”*
  + *“Why did stocks rally today?”*
* The system responds with concise, contextually accurate answers backed by retrieved news articles.

**Conclusion**

This project successfully demonstrates a practical small-scale RAG system that augments LLMs with daily retrieved data to provide updated summaries and relevant Q&A.  
It highlights how retrieval mechanisms can keep LLM outputs grounded in current events, solving one of the key limitations of static knowledge in language models.

**Repository & Deployment**

* Codebase: (provide your GitHub repo link if applicable)
* Local deployment instructions included via streamlit run app.py.