

THE UNIVERSITY OF HONG KONG
DEPARTMENT OF COMPUTER SCIENCE 2024 – 25



COMP4801/FITE4801
Final Year Project
Interim Report

Group 24079
Lee Jong Seung (3035555547)
Lee Changjin (3035435840)
Kim Taehyun (3035741330)

Topic
LLM-based Real-time Personalized Financial News Notification System

1. Project Background

The financial market is highly dynamic, with news acting as one of the main drivers behind price fluctuations. A recent example that highlights the influence of news is Nvidia CEO Jensen Huang's comments on quantum computing. Huang made a remark at CES on January 8th, 2025 and stated quantum computing is still 15-30 years away from commercialization, which was contrary to the market's optimism [1]. This statement contributed to negative market sentiment, leading to a sharp decline in quantum computing-related stocks such as Rigetti Computing (RGTI, -45%), IonQ (IONQ, -39%), D-Wave Quantum (QBTQ, -36%) and Quantum Computing (QUBT, -43%) just in a single trading day.

Another example of news-driven market movement stems from macroeconomic changes. On January 10th, 2025, the release of consensus beating December U.S. job growth and the University of Michigan consumer sentiment drop negatively influenced the market [2], resulting in a tumble in major U.S. indices (Dow Jones -1.63%, S&P 500 -1.54%, Nasdaq -1.63%). However, many retail investors struggle with analyzing the news due to a superficial understanding of the theoretical relationships between various macroeconomic figure announcements and these price movements.

Despite the critical role of analyzing financial news in investment processes, retail investors often lack the time to process and analyze the overwhelming inflow of information. According to a web traffic study by Fintext [3], financial news readers typically browse only 3 to 4 pages daily, spending approximately 30 seconds to a minute on each. This equates to just 2 to 4 minutes per day devoted to reading financial news, which may result in incomplete understanding and poor investment decisions, potentially leading to financial losses.

While many existing financial news platforms offer news curation services to some extent, they typically rely on basic keyword filtering and lack the deeper contextual understanding required to tailor news specifically to an individual's portfolio, risk profile, or investment objectives. However, recent advancements in large language models (LLM) present new possibilities for real-time, personalized news delivery. LLM can process and comprehend complex language and can be fine-tuned with financial data to extract relevant information, predicting which news is most likely to impact a user's portfolio. By leveraging LLM, a news delivery system can help users stay ahead of market trends, make informed

decisions quickly, and engage with the model to ask questions for further clarity. This offers a valuable opportunity to close the information gap between retail investors and financial institutions, enabling faster and more informed investment decisions.

2. Project Objectives

To address the challenges faced by retail investors, this final-year project ultimately aims to narrow the information asymmetry in the investment process and educate users on fundamental investment knowledge. Leveraging fine-tuned LLM and aggregating reliable data sources such as Refinitiv and Newscatcher API, the project seeks to deliver actionable insights and improve the decision-making process for retail investors.

To achieve the primary objective, the team proposes a personalized real-time financial news notification application that leverages LLM and Natural Language Processing (NLP) to deliver highly relevant and time-sensitive information to investors. This application aims to enable retail investors to make informed decisions quickly and effectively by delivering summarized news updates and insights. The proposed solution will:

1. Delivers real-time, personalized news alerts tailored to an individual's portfolio, ensuring that only relevant updates are provided. By fetching live news data and sending notifications via third-party platforms like Discord, the system keeps users promptly informed of market events impacting their investments.
2. Offers detailed summaries and in-depth analysis of each news article, highlighting key points and assessing potential impacts on stock prices to save time and facilitate quick responses. This enables users to stay ahead of market shifts without having to read through lengthy articles. For beginners, some financial concepts or terminology might be unclear, so LLM provides further explanations.

The platform differentiates itself from existing products through several key features. The use of a Keyword Generator LLM and Embedding Search method ensures more precise matches between news content and user portfolios. The system personalizes summaries based on user proficiency and risk appetite, enhancing relevance and ensuring tailored content. Additionally, users receive clear explanations of the LLM's decision-making process, making it more transparent and educational. The summaries provide just enough detail to highlight key points about stock movements, ensuring easy comprehension. By integrating these

features, the platform empowers retail investors with timely, actionable information, helping them make informed decisions and improve their understanding of financial markets.

However, there will be some challenges posed by the complexity of implementing such a system. Firstly, supporting high-quality similarity search goes beyond simple keyword matching. Relevancy matching must accurately capture both the direct and indirect impacts of news on stock prices, which requires relevancy calculation to properly assess the nuanced effects of each news piece. Secondly, the flawless integration of multiple models into the system is crucial. This includes the seamless operation of polling agents for news fetching, various LLMs for analysis, and embedding models for similarity search. Ensuring all components work together efficiently without latency or errors will be a significant challenge. Lastly, it is essential to employ fine-tuned LLMs and optimize the embedding search for the specific use case due to the nature of the project. The model should be specialized in finance and education, therefore the team either needs to fine-tune by ourselves or look for existing pre-trained models. Optimizing the embedding search will involve iterative testing on a large dataset and qualitative analysis to ensure the system delivers accurate, relevant results consistently.

3. Project Methodology

This section introduces the methodologies utilized in the project. It details the data source and APIs, system architecture, and various system components to achieve personalization and real-time news summary and analysis delivery.

3.1 Data Source and APIs

The project will utilize NewsCatcher API [4] to retrieve stock information, including stock name, industry classification, and relevant keywords and tags describing the company and its activities. The decisive advantage of NewsCatcher API, compared to other finance information providers, is that it provides endpoints to retrieve real-time news updates along with headlines and full content.

3.2 System Architecture

The system will have various components (see Figure 1 below) to construct a personalized and real-time financial news notification system.

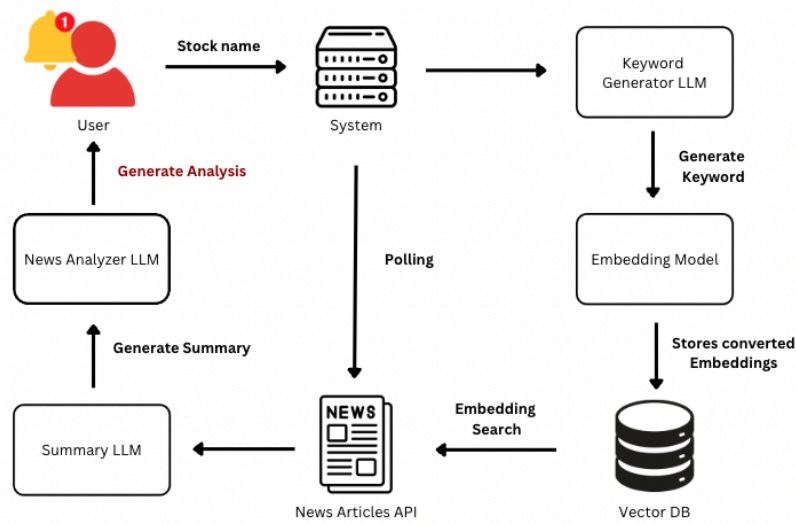


Figure 1. *System Architecture Design*

The system generates the embeddings from the user-provided stock information and stores them in the vector database. A polling agent extracts news articles, and LLMs analyze them to create personalized insights, delivering real-time updates to users through SNS channels.

The system's personalization begins with user input processing. Upon registration, users can provide stock names which will be used for personalized news article selection. Then, the system will utilize Refinitiv API to fetch stock information, including stock price, previous close price, trading volume, etc. Once the system has stock names, the Keyword Generator LLM will expand the data by generating additional related keywords that reflect potential news narratives. The generated keywords and sentences will then be converted into embeddings and saved into a vector database. The vector database will be utilized for efficient embedding search against news articles to filter only the relevant ones.

To achieve real-time delivery, a polling agent, a background batch process running asynchronously from the main backend, will continuously poll NewsCatcher API at short intervals (about 1 minute as below) to extract news articles. The system will compare the embeddings of the recently fetched article with the stored embeddings of user-specific keywords. This process selects only the users whose stock data is relevant to the extracted

article. Then, the Summary LLM and News Analyzer LLM will generate a summary and user-specific stock impact analysis and deliver them to users through SNS channels.

3.3 Real-Time News Polling Agent

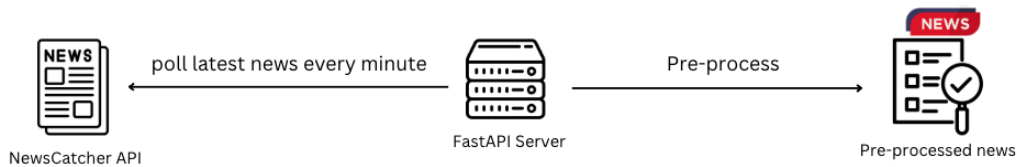


Figure 2. *News Polling Agent*

A key feature of the project is to deliver the stock summary and analysis as close to real-time as possible. To achieve this, the system continuously polls the data source using FastAPI's asynchronous CronJob to minimize the latency and impact on the main backend server. The polling interval is set to 1 minute but may be adjusted depending on the frequency of news article updates.

3.4 News Relevancy Calculation - Keyword Generation LLM

Embeddings are numerical representations of words or sentences in vector space, capturing the semantic relationships between text data [5]. Due to their ability to distinguish semantic nuances, embeddings are frequently utilized in embedding search, also known as semantic search, to determine the similarity between sentences or whole documents.

The system leverages embedding search with cosine similarity to determine the relevancy between the user input and the news articles. Although there are multiple aspects of data that can be retrieved from a news article, such as title, content, and publisher, the only data provided as the user input is the stock name. Using only the stock name for embedding search might work to some degree, especially when the news article directly contains the stock name. However, the team suspects that the stock name itself contains too little context, and it might not be enough to capture the indirectly related news articles as the embedding search tries to capture the semantic relations. For example, suppose the user provides Nvidia as a stock name, and a news article talks about the surge in the demand for AI chips in the market. Then, it's reasonable to conclude that the article is relevant to Nvidia. However,

simply vectorizing ‘Nvidia’ for embedding search will likely give suboptimal results. This argument is supported in an experiment tested with Hugging Face’s all-MiniLM-L6-v2 embedding model where the cosine similarity score between ‘Nvidia’ and ‘AI chip market growth’ was only 0.193, while the score between ‘Nvidia’ and NVideo’, an intuitively unrelated word, yielded 0.641. This demonstrates the limitation of embedding models to capture the indirect relationships and nuances deducted from industry and domain knowledge.

In order to empower the model to capture those intricacies, the system includes the Keyword Generator LLM that generates 5 to 10 example keywords that are likely to appear in actual news articles and reflect domain intricacies related to the provided stock name. For example, if the user provided ‘Nvidia’, then the Keyword Generator LLM might produce keywords such as ‘Data center demands’, ‘Surge in AI chip demands’, or ‘Gaming industry trends’ - industries that have a high correlation with Nvidia. Those keywords, in addition to the original stock name, will be converted into vectors to enhance the embedding search performance. Yet, challenges still remain in that similarity is a rather subjective concept—for instance, an article about the flight industry might not seem directly relevant to Nvidia at first glance, but it actually could be due to the industry’s use of AI technology. As such, achieving meaningful results requires reasonable data labeling and qualitative analysis to refine the system’s ability to capture nuanced connections.

3.5 News Relevancy Calculation - News Data

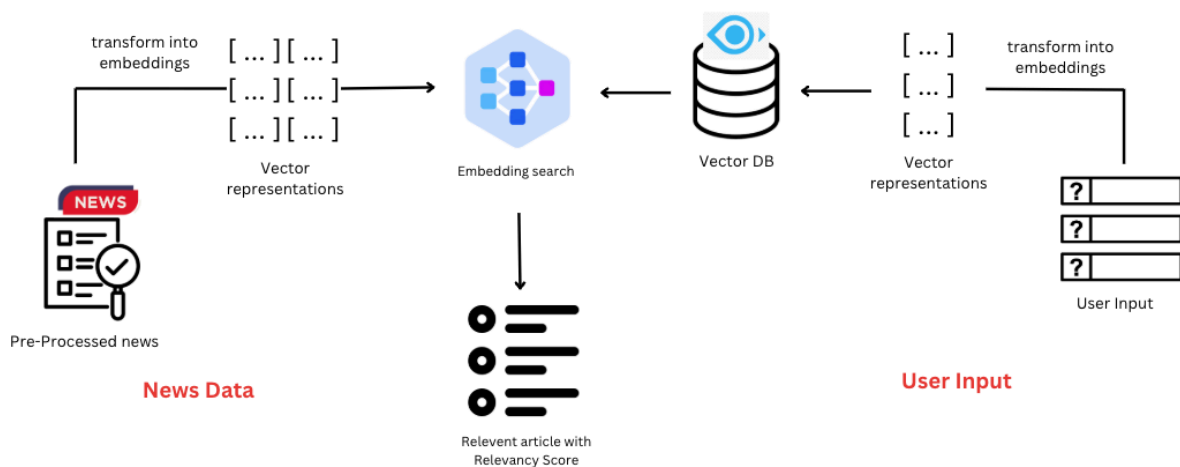


Figure 3. *Extraction of Embeddings from News Data*

On the other hand, the news data must also be processed and converted into vector embeddings to calculate the relevancy score. One intuitive method is using only the title for embedding search. Although this method decreases latency, it contains too little context to deliver thorough information about the article. Another approach is converting the whole article body into embeddings. This approach would utilize richer information than the previous but it might introduce the semantic dilution problem where important keywords and sentences might be diluted due to the lengthy text.

In order to avoid the dilution problem and maintain rich information, the system utilizes the following two approaches. The first is to generate a brief summary from the article and convert it to embeddings, effectively avoiding semantic dilution. In the second approach, the system first slices the article body into multiple chunks. Then, embeddings will be extracted from individual chunks in parallel, leveraging multithreading. Finally, the maximum relevancy score will be selected as a result. This approach also minimizes semantic dilution by shrinking the text size and increases the efficiency of the process.

3.6 News Summary Generator LLM & Stock Impact Analysis LLM

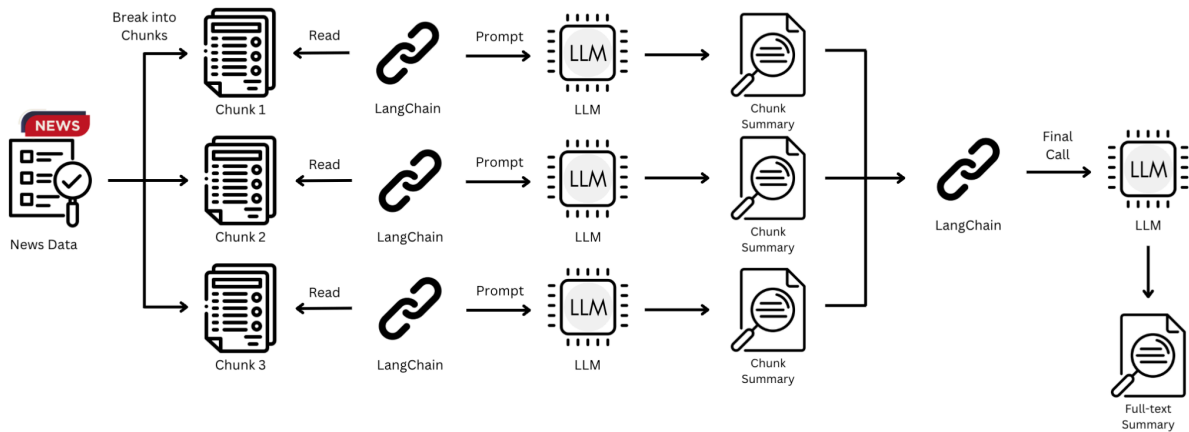


Figure 4. *News Summary Generator LLM*

After the newly fetched article is determined as relevant to user input, the News Summary Generator LLM will generate a concise summary. In order to optimize the efficiency, the news article will be sliced into multiple chunks, and the LLM will generate a summary for each individual chunk in parallel using multithreading to avoid time-consuming

API calls. Finally, based on those chunk summaries, the LLM will produce a final summary which will be delivered to the user.

Finally, a stock impact analysis, such as the potential stock price movement, will be generated along with the summary. An example of the analysis might be ‘This might be a strong indicator that the price would go up in the near future due to increased revenue projections following the company’s announcement of a major contract with a leading distributor in the AI chip market’.

3.7 Development Tools and Infrastructure

The primary backend system of the project will be implemented using FastAPI. Compared to other frameworks, FastAPI offers a powerful asynchronous capability to efficiently handle long-running API calls. In addition, FastAPI is a Python-based framework, as the majority of machine learning frameworks are also provided with Python SDK. OpenAI is chosen as the LLM provider due to its excellent performance and robust ecosystem. The frontend will be developed with React, chosen for its efficiency, reusability of components, and strong ecosystem for building dynamic web applications.

The project’s infrastructure will be hosted on Amazon Web Services (AWS) for its first-class scalability and reliability. Moreover, Milvus will be used as the vector database due to its superior performance metrics. It provides the highest 2,406 queries-per-second rates and offers the lowest computation latency(1 ms), outperforming alternatives such as Pinecone and Qdrant [6].

4. Preliminary Results & Discussions

4.1 Progress & Preliminary Result

This section discusses the progress made so far, including preliminary results (Section 4.1), challenges encountered, and potential solutions (section 4.2).

4.1.1 Frontend Development

First, the frontend demo has been developed, including the sign-up and sign-in page, a page for entering the user's stock preferences, and a dashboard page. Figure 5 below shows the stock input page that receives stock ticker input from the users they wish to track.

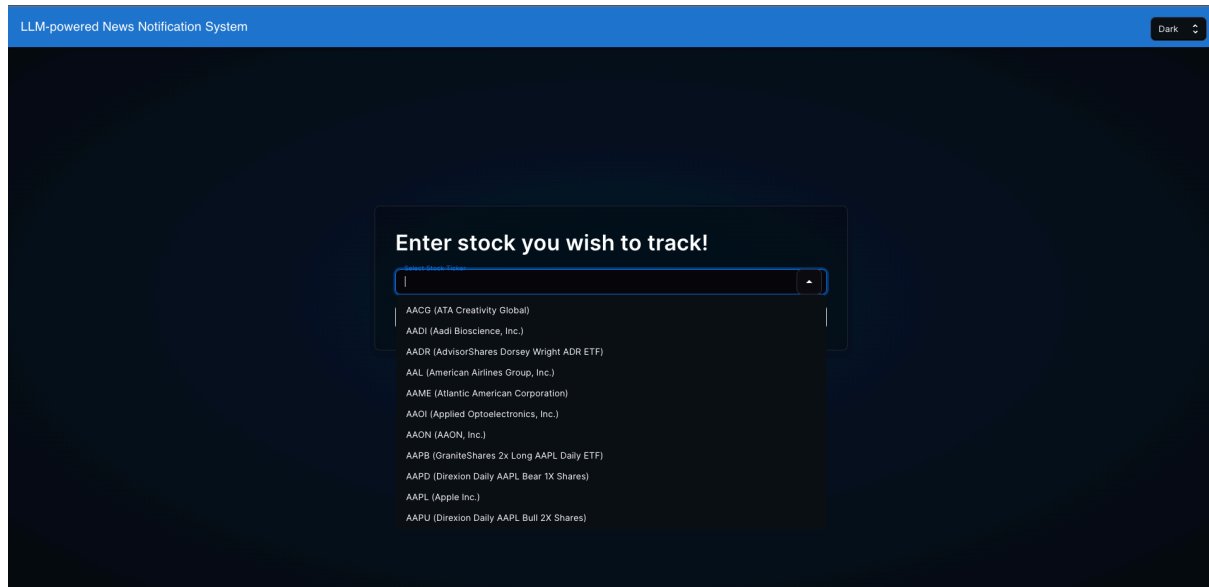


Figure 5. *Stock data input page*

When the users input stock tickers, it calls our backend's API to save their stock preferences, generate embeddings based on the stock name, and save them into vector databases.

After the user input page, the user will be directed to the dashboard page illustrated in Figure 6 below.

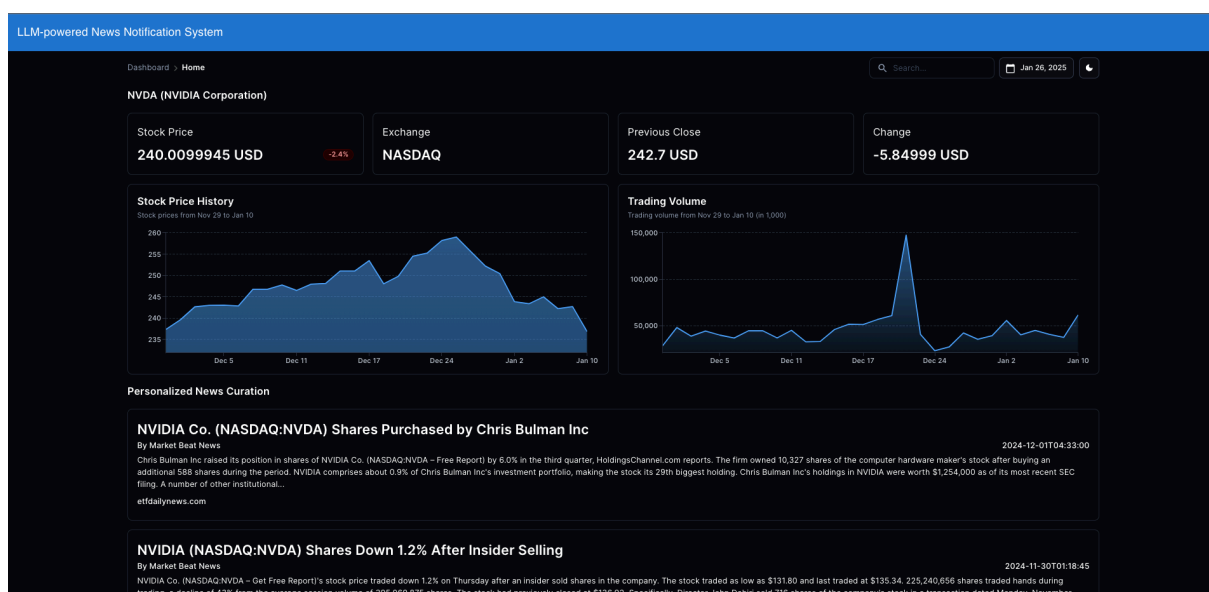


Figure 6. *Dashboard page*

The dashboard page consists of basic stock information, including the current stock price, stock price history, trading volume, etc. These details serve as an additional metric for tracking stock performance, complementing the news displayed in the lower section of the dashboard.

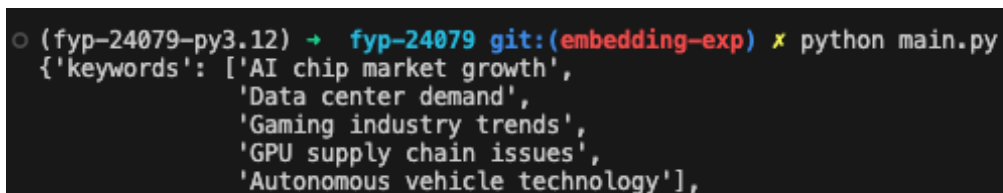
In the lower section of the dashboard, the “Personalized News Curation” section provides only the news relevant to the user’s stock preference by performing an embedding search between stock-related keywords in the vector database and real-time news collected by the polling agent.

4.1.2 Polling Agent

For the news articles fetched to display on the dashboard page, the NewsCatcher API was utilized to first retrieve all the latest news. To fetch news articles at regular intervals, a data collection function was developed to gather all news articles from the past 1 minute using the NewsCatcher API, filtering only English news from the top 500 popular sources. This function will be integrated with a polling agent to feed news data. Currently, an initial version of a polling agent has been developed utilizing a cron job in Python to simulate news polling with news articles collected by this function beforehand.

4.1.3 Keyword Generator LLM

Among the number of LLMs mentioned in Section 3, Keyword Generator LLM has been developed. This component plays a key role in enhancing embedding search results between news articles and users’ stock preferences to determine the similarity between them. Figure 7 below illustrates the result of testing the Keyword Generator LLM using an example input of “Nvidia.”



```
(fyp-24079-py3.12) → fyp-24079 git:(embedding-exp) ✖ python main.py
{'keywords': ['AI chip market growth',
              'Data center demand',
              'Gaming industry trends',
              'GPU supply chain issues',
              'Autonomous vehicle technology'],
```

Figure 7. *Results of the Keyword Generator LLM.* The Keyword Generator LLM generates potential keywords to expand user inputs, with an example input of “Nvidia.”

When the user inputs their preferred stock name, such as “Nvidia,” the Keyword Generator LLM expands the input by generating five to ten keywords, such as “AI chip market growth” and “Data center demand,” that are likely to appear in the actual news article relevant to Nvidia. The results suggest that the Keyword Generator LLM can generate contextually accurate keywords and sentences, indicating its potential to enhance relevant financial news filtering via the embedding search. The results also align with the prompts provided to the LLM, which is instructed to generate five realistic and relevant keywords likely to appear in financial news articles. A strong emphasis was also provided in the prompts to encompass diverse aspects such as industry trends, market dynamics, and technological advancements for more comprehensive keyword generation. These generated outputs will then be converted to embeddings and used to enhance the embedding search accuracy to filter the relevant articles to the user input, which will be carried out in subsequent steps.

4.1.4 Embedding Search Model

An initial version of the embedding search model has been developed as a preliminary result. News relevancy calculation was implemented based on the full text of news articles and single queries such as stock names like “Nvidia” or semantically meaningful ones such as “news relevant to Nvidia’s stock movement.” To calculate the relevancy score and return the news that is relevant to the stock, the cosine similarity between the embeddings of queries and news full-text embeddings have been calculated, and the list of stock tickers that exceeds the threshold value of 0.6 was determined as relevant, and returned as an embedding search result.

As a result, simply using the stock name embedding as a query (e.g., “Nvidia”) resulted in less accuracy in determining relevant news, whereas semantically meaningful queries (e.g., “News relevant to Nvidia stock movements”) gave better performance. Figure 8 below shows the result of the embedding search with an example query of “news relevant to Nvidia’s stock movement.”

```
[idx: 2] NVIDIA Co. (NASDAQ:NVDA) Shares Purchased by Chris Bulman Inc
Distance: 0.6096170544624329

[idx: 3] NVIDIA (NASDAQ:NVDA) Shares Down 1.2% After Insider Selling
Distance: 0.6731924414634705
```

Figure 8. Results of embedding search between single queries and news full-text.

News relevant to Nvidia's stock prices shows a moderate similarity score of about 0.6, suggesting relevancy (see Figure 8 above). Nevertheless, those directly relevant news still do not show strong relevancy (i.e., similarity score greater than 0.8).

```
[idx: 5] Boston heads to Toronto for conference showdown
Distance: -0.03863076493144035

[idx: 1] Spurs take on the Grizzlies on 3-game losing streak
Distance: 0.07161866873502731
```

Figure 9. Embedding search results that are determined as irrelevant.

On the other hand, irrelevant news accurately resulted in a low similarity score of below 0.1 (see Figure 9 above). The results suggest that embedding search could be further improved, especially in terms of determining relevant ones.

4.2 Challenges & Potential Solutions

Throughout embedding search testing and results, a few limitations with embedding search against the full article content have been identified.

First, the embedding search results showed only moderate relevancy scores between 0.5 and 0.6, even for genuinely relevant articles. This could be attributed to embedding dilution caused by long full-text article content, as lengthy text often includes less impactful information related to stock prices, reducing the focus on sections relevant to stock price changes.

To enhance the accuracy of embedding search, the following approaches are proposed:

1. News Contents Pre-Processing: Experimenting with various pre-processing techniques can help mitigate embedding dilution by long articles. The following pre-processing methods will be considered:
 - a. Summarization: Compress articles into concise summaries focusing on more important information.
 - b. Chunking: Split articles into smaller and meaningful segments for individual embedding.
 - c. Weighted scoring: Give higher scores if the article contains the exact stock name.
2. Threshold Optimization: Determining a reasonable relevancy threshold to classify articles accurately would be crucial. To find such a threshold, testing and analysis with more various types of news will be required.
3. Enhancing Search Queries: The Keyword Generator LLM will be integrated to generate a richer set of search queries. For example, the LLM generates N keywords that are related to a single stock ticker. If more than K keywords show a strong relevancy score to the given news article, the article can be more confidently classified as relevant to that stock. This approach can enhance the accuracy by expanding the scope of queries beyond a single keyword query.

On the other hand, challenges remain in determining the ground truth for the news relevancy to a specific stock. The current preliminary results are limited to testing “Nvidia” with selected news articles that were qualitatively reviewed and deemed relevant to “Nvidia” by our team. To enhance embedding search and quantitatively evaluate our embedding search performance, finding labeled news data would be crucial yet challenging.

A potential solution is finding the news articles labeled with stock tickers by authoritative sources, such as financial news platforms, along with past stock price data. By matching stock price changes with the associated news articles, it may be possible to construct a labeled dataset to improve and validate the embedding search performance and even be used for the News Analysis LLMs that will be developed in subsequent steps.

5. Upcoming Schedules

After the inception phase, the team is now in phase 2 and will be focusing on the development of LLM and enhancing the embedding search models. The detailed schedules are as follows.

Phase 2: Elaboration (18 Jan, 2025 - 20 Apr, 2025)

1 Jan - 31 Jan, 2025

- Frontend development
- Backend development
- Implementation of polling agent
- Implementation of the embedding search model

1 Feb - 28 Feb, 2025

- Frontend development
- Backend development
- Enhancing embedding search model
- Implementation of News summary LLM

1 Mar - 31 Mar, 2025

- Implementation of Stock Analyzer LLM
- Test and experiment with the Stock Analyzer LLM (summary generation LLM & stock price impact analysis NLP model)

1 Apr - 20 Apr, 2025

- Whole System Integration & Deployment
- Integration test of the overall system
- Continuous testing and enhancement of the language models of the system

Phase 3: Construction (21 Apr, 2025 -)

21 Apr - 30 Apr, 2025

- Preparation of final presentation and project exhibition

The immediate next step would be finding appropriate methods to get the labeled ground truth news articles to enhance embedding search model accuracy and fine-tuning Large Language Models.

5. Citation

[1] J. Wittenstein, “Quantum Computing Stocks Drop as Nvidia CEO Jensen Huang Sees Use Years Away,” *Bloomberg.com*, Jan. 08, 2025.

<https://www.bloomberg.com/news/articles/2025-01-08/quantum-computing-stocks-drop-as-nvidia-ceo-sees-use-years-away> (Accessed Jan. 8, 2025).

[2] J. M. Cherian, S. Gupta, and C. Mandl, “Wall Street ends lower as blowout job data spooks traders,” *Reuters*, Jan. 11, 2025. Available:

<https://www.reuters.com/markets/us/futures-drop-caution-ahead-key-payrolls-data-2025-01-10/>

[3] “Who Reads Finance News? Traffic and User Behaviour,” *FinText*, Feb. 21, 2023.

<https://www.fintext.io/case-studies/benchmarking/who-reads-financial-news-web-traffic-and-user-behaviour/> (Accessed Nov. 26, 2024).

[4] "News API: Search Global News Data for Insights and Analysis,

"<https://www.newscatcherapi.com>. [Online]. Available:

<https://www.newscatcherapi.com/docs/v3/documentation/get-started/overview>. (Accessed: Oct. 15, 2024).

[5] J.-T. Huang et al., "Embedding-based Retrieval in Facebook Search," in *Proc. 26th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining (KDD '20)*, pp. 2553–2561, Aug. 2020. doi: 10.1145/3394486.3403305.

[6] "Picking a vector database: a comparison and guide for 2023," benchmark.vectorview.ai.

[Online]. Available: <https://benchmark.vectorview.ai/vectordbs.html>. (Accessed: Oct. 15, 2024).