

THE UNIVERSITY OF HONG KONG

FITE4801/COMP4801

Final Report

Kim Taehyun (3035741330)

21 April 2025

Topic

TradeInbox: LLM-based Real-time Personalized Financial News Notification System

Abstract

In today's fast-paced financial markets, promptly identifying relevant information is crucial, yet retail investors are often overwhelmed by information overload due to limitations in existing news platforms that lack tailored articles and immediate predictive analytics. To address these limitations, a financial news notification system was designed to streamline news processing by delivering real-time, personalized news articles with tailored analyses curated for each user's portfolio. The project aims to reduce news processing time and aid decision-making of retail investors while serving the educational purpose of enhancing their financial literacy. The system utilizes a Keyword Generator LLM combined with embedding search to capture both directly and indirectly related news articles to a selected stock. Multiple prompt-engineered large language models (LLMs) have been implemented to generate concise summaries, key metrics, sentiment analysis, and multi-level stock impact analyses (Beginner, Intermediate, Expert). Finally, concise notifications are delivered via Discord. The results demonstrate a successful end-to-end implementation, with experiments validating the effectiveness of the filtering approach using embedding search with the Keyword Generator LLM, and initial user testing showing strong satisfaction (>70%) regarding promptness, personalization, education, and comprehension. This work presents a viable LLM-based solution for personalized financial news, with future work focusing on fine-tuning models and integrating user portfolio APIs.

Acknowledgments

I would like to express my gratitude to our final-year project supervisor, Dr. Chow Ka Ho, for his expertise, guidance, and support throughout the project. I also want to thank our CAES9542 lecturer, Mr. Gagandeep Singh, who gave invaluable lessons on writing technical reports. Many thanks to my teammates for their contribution and commitment to the project. Lastly, I am also grateful to friends and seniors who shared their insights and feedback on the project.

Table of Contents

1. INTRODUCTION	1
2. PROJECT BACKGROUND	2
3. METHODOLOGY	3
3.1 Data Source & APIs	3
3.2 System Architecture	7
3.2.1 Keywords Generator LLM	8
3.2.2 Embedding Model	10
3.2.3 Polling Agent	12
3.2.4 Summary LLM	14
3.2.5 Key Metrics LLM	15
3.2.6 Sentiment Analysis LLM	17
3.2.7 News Impact Analysis LLM	20
3.2.8 Delivery via Discord Notification	23
3.3. Feedback for Embedding Search Results	23
3.4 Engineering choices	24
3.5 Database Design	26
4. RESULTS & DISCUSSION	27
4.1 Frontend Result	27
4.1.1 Sign-up page	28
4.1.2 Sign-in page	29
4.1.3 Stock Input page	29
4.1.4 Dashboard Page	31
4.1.5 News Detail Page	33
4.1.6 Real-time Notification via Discord Channel	34
4.2 Results & Experiment on Embedding Search and Keyword Generator LLM	35
5. CONCLUSION & FUTURE WORKS	38
5.1 Conclusion & Findings	38
5.2 Future Works	39

List of Figures

Figure 1. API call function to retrieve news data using the NewsCatcher API	4
Figure 2: News data example retrieved from the NewsCatcher API	5
Figure 3: Stock-related data example from Refinitiv API	5
Figure 4. Entering the stocks input page	6
Figure 5. Dashboard page with data from Refinitiv API at the left side	7
Figure 6. System Architecture - Overview of the LLM-based Real-time Personalized Financial News Notification System	7
Figure 7. Prompt given to the Keyword Generator LLM	10
Figure 8. Embedding search system design.	11
Figure 9. Polling agent workflow.	12
Figure 10. News Detail page showing Summary, Key Metrics, Sentiment Analysis and Stock Impact analysis	14
Figure 11. News Summary LLM Prompt	15
Figure 12. Key Metrics LLM Prompt	17
Figure 13. Sentiment Analysis LLM prompt	19
Figure 14. News Impact Analysis LLM prompt	22
Figure 15. Stock detail page with feedback loop feature.	23
Figure 16. Development tools used: FastAPI, MySQL, React, Milvus, HuggingFace, and OpenAI, in that order.	25
Figure 17. Entity-Relationship Diagram of the system database schema	26
Figure 18. Sign-up page	28
Figure 19. Sign In page	29
Figure 20. User stock input page with stock lists	30
Figure 21. User stock input page with stocks selected	30
Figure 22. Initial dashboard page design	31
Figure 23. Revised Dashboard page	32
Figure 24. Choosing a stock among multiple stocks	33
Figure 25. News Detail Page	34
Figure 26. Discord Notification for Nvidia stock	35
Figure 27. Results of Keywords generated from the Keyword Generator LLM	36

List of Tables

Table 1. Results of Embedding Search against keywords generated_____	37
--	----

1. INTRODUCTION

In today's fast-paced financial markets, promptly identifying and tracking appropriate information is crucial for all market participants, such as investors, traders, and financial institutions. Among the variety of information, one of the critical components is news articles that significantly impact stock prices and require immediate response. Research has shown that stock prices are significantly influenced by unanticipated financial news and its sentiment, highlighting the importance of tracking and analyzing it [1]. Hence, it is crucial for investors to identify and track relevant news to their portfolio and make an informed decision by analyzing the news to minimize risks. However, the high volume, speed, and real-time nature make it difficult for retail investors to process that information and make prompt investment decisions [1]. Those challenges span from staying up-to-date and constantly checking news websites for updates to cherry-picking the most relevant information from vast amounts of data and then analyzing it to make a decision.

Recent advancements in LLMs have opened new possibilities for real-time data analysis and processing vast amounts of information. By utilizing these models with tailored prompt engineering, it becomes possible to filter only the most relevant news based on a user's portfolio and predict its potential impact. Hence, the team proposed a financial news delivery system that can quickly and concisely deliver only relevant information, along with a predictive analysis feature, to support better investment decisions and educate retail investors. This project's news delivery system leverages LLMs to offer enhanced personalization, enabling users to quickly access relevant news, make informed decisions, and directly ask the LLMs any questions about the news analysis.

This project aims to achieve two primary objectives. The first objective is to provide efficient financial news processing and decision-making cycles for investors by increasing the density of the information consumed. Achieving this involves reducing the time spent identifying, reading, and analyzing entire news articles while increasing the number of articles consumed for better decision-making, ultimately reducing information asymmetry between retail and professional investors. The second objective is to educate retail investors by enhancing their understanding of market dynamics, helping them make more informed investment decisions.

Key contributions of this work will be listed as follows: (1) a prompt-engineered Keyword Generator LLM that together with embedding search, captures both direct and indirect

relevance signals; (2) an embedding-search feedback loop that adapts each user's relevance criteria over time; and (3) a multi-tiered LLM analysis engine delivering summaries, sentiment scoring, and impact explanations at beginner, intermediate, and expert levels.

Accordingly, the main deliverable will be a website with two key features with a real-time notification feature. First, a dashboard where user can input their stock tickers they wish to track and see their portfolio overviews. Second, a list view of relevant news articles selected by our LLMs, with a summary and dedicated analysis of each article. Lastly, it will have a chatting function with our fine-tuned LLM model that analyzes the articles so that users can ask questions and gain deeper insights into the analysis and prediction of the impact on the price. Apart from the website features, it will offer real-time notifications via WhatsApp or email with a one-line summary of relevant news articles, along with a prediction of stock price changes. The scope of the project focuses on the US stock market, with real-time news articles collected from the top 500 popular news resources from the NewsCatcher API.

The remainder of this report is organized as follows: Section 2 provides more detailed project backgrounds. Section 3 provides detailed methodologies including the system architecture, LLM component implementation, and development environment used. Next, Section 4 discusses experiments and results of the project. Finally, Section 5 presents the conclusion of the report, along with the desired future work.

2. PROJECT BACKGROUND

Currently, several major platforms (e.g., Seeking Alpha, Yahoo Finance) provide portfolio tracking and real-time alerts on major breaking news and stock price changes [2]-[4]. Seeking Alpha's Portfolio Digests deliver a daily email summary of news tied to each user's portfolio [2], while Yahoo Finance enables users to create custom watchlists and receive push notifications for breaking news and earnings reports via its mobile app and web interface [4]. However, they often provide basic filtering or keyword matching for relevant news or do not provide immediate predictive analytics to help guide investment decisions or educate investors based on the news. Although these platforms offer some expert analyses, it is mostly understandable to experts with some financial literacy.

These limitations lead to two key challenges that this project aims to address for retail investors. First, it is overwhelming for retail investors to spend time identifying relevant news in a flood of information and reading those articles thoroughly to make an investment decision without fully understanding their implications [1]. Second, the challenge lies in most retail investors' limited time and resources. A web traffic study showed that financial news readers visit three to four pages on news platforms each day and spend an average of 30 to 60 seconds on each page [5]. This suggests that retail investors trade with limited information, which could lead to poor investment decisions. Given these challenges, this project hypothesizes the need for a financial news delivery system that can quickly and concisely deliver only relevant information, along with a predictive and educational analysis feature, to support better investment decisions and foster financial literacy.

3. METHODOLOGY

This section discusses the implementation details of the project. It includes the data source and APIs used for retrieving real-time financial news data (Section 3.1), the system architecture with a detailed explanation of each component, including multiple large language models (Section 3.2), and the feedback loop for embedding search results (Section 3.3). The development environment and tools chosen for the project are discussed in Section 3.4, and the section concludes with implementation details on database design in Section 3.5.

3.1 Data Source & APIs

The project primarily uses the NewsCatcher API [6] to retrieve real-time global news articles data in JSON format. The API provides ready-to-use news from over 90,000 news sources worldwide. It serves as a crucial baseline for our project by providing a real-time news source that will be retrieved every 10 minutes. This interval was selected after load-testing, considering the number of news items incoming every minute, striking a balance between real-timeness and efficiency of the API calls.

```

def news_load():
    datetime_now = datetime.now(pytz.timezone('UTC')) + timedelta(minutes=-10)
    datetime_str = datetime_now.strftime("%Y/%m/%d %H:%M:%S")

    all_articles = newscatcherapi.get_search(
        q = '*',
        lang = 'en',
        countries = 'US',
        from_ = datetime_str,
        published_date_precision = 'full',
        ranked_only = True,
        to_rank = 500,
        sort_by = 'date'
    )

    return all_articles['articles']

```

Figure 1. *API call function to retrieve news data using the NewsCatcher API*

The news retrieval function in Figure 1 above calls the NewsCatcher API with various filter options applied. It retrieves articles published within the last 10 minutes, according to our 10-minute polling interval. Then it filters out news in English, published in the United States only. Also, to prevent users from getting overwhelmed by too many news articles and prevent duplicate news, the function retrieves data from only the top 500 news websites ranked by traffic. These basic filters enable our system to serve only necessary and credible sources to users and to process them by our system components. Nevertheless, the function is configured to retrieve all news articles without any keyword filtering. As this option only provides a basic keyword match between the news and the keyword input, the parameter `q="*"` is set to include every source. Since our system aims to capture both direct and indirect matches between stock and news articles, the actual filtering to find relevant news articles for a specific stock will be conducted in a subsequent implementation under the embedding search part. Finally, this function will be called by the polling function, which will be discussed in the next sub-section.

```
[
  {
    "title": "Nasdaq Sell-Off: After Losing Nearly $800 Billion in Market Cap, Is Nvidia Stock a Buy Anymore? History Offers a Clear In",
    "author": "Adam Spatacco",
    "published_date": "2025-03-31 13:30:00",
    "published_date_precision": "full",
    "link": "https://www.msn.com/en-us/money/topstocks/nasdaq-sell-off-after-losing-nearly-800-billion-in-market-cap-is-nvidia-stock-a-",
    "clean_url": "msn.com",
    "excerpt": null,
    "summary": "On Jan. 6, shares of semiconductor powerhouse Nvidia (NASDAQ: NVDA) closed at an all-time high of $149.43. At the time,",
    "rights": "msn.com",
    "rank": 125,
    "topic": "news",
    "country": "US",
    "language": "en",
    "authors": "Adam Spatacco",
    "media": null,
    "is_opinion": false,
    "twitter_account": null,
    "score": 6.638991,
    "_id": "7c78d4b97c1b11df69e69a4da7781ad0"
  },
]
```

Figure 2: *News data example retrieved from the NewsCatcher API*

Figure 2 shows an example result from the NewsCatcher API call. The API provides various metadata, including title, published date, link to the original news, excerpt, and full content of the news etc. Fields such as published data and title will be served directly to the user for their reference. Mainly, the full content of the news, under the 'summary' field, will be processed further for summarization and comprehensive analysis.

On the other hand, another data source—the Refinitiv API has been utilized. Refinitiv API was used to fetch stock-related information, including stock ticker names, company names, exchanges, market capitalizations, and more. Figure 3 below illustrates an example of NYSE constituent data fetched from the Refinitiv API.

	Instrument	TR.CommonName	TR.ExchangeName	TR.CompanyMarketCap	TR.TRBCBusinessSector	3 Month Total Return
0	FN.N	Fabrinet	NEW YORK STOCK EXCHANGE, INC.	7974.583213	Technology Equipment	-7.003891
1	TGLS.N	Tecnoglass Inc	NEW YORK STOCK EXCHANGE, INC.	3727.667037	Cyclical Consumer Products	15.744247
2	HPPN	Hudson Pacific Properties Inc	NEW YORK STOCK EXCHANGE, INC.	427.934054	Real Estate	-36.610879
3	PLOW.N	Douglas Dynamics Inc	NEW YORK STOCK EXCHANGE, INC.	545.712331	Industrial Goods	-13.314197
4	GDOT.N	Green Dot Corp	NEW YORK STOCK EXCHANGE, INC.	572.29202	Banking & Investment Services	-9.137489
...
2065	KLCN	Kindercare Learning Companies Inc	NEW YORK STOCK EXCHANGE, INC.	2099.796384	Personal & Household Products & Services	-31.879066
2066	INGM.N	Ingram Micro Holding Corp	NEW YORK STOCK EXCHANGE, INC.	4553.268016	Software & IT Services	-21.178862
2067	ECG.N	Everus Construction Group Inc	NEW YORK STOCK EXCHANGE, INC.	3351.412879	Industrial & Commercial Services	34.183673
2068	GEAR.N	Revelyst Inc	NEW YORK STOCK EXCHANGE, INC.	1122.818085	Industrial Goods	1.210526
2069	JACS_u.N	Jackson Acquisition Company II	NEW YORK STOCK EXCHANGE, INC.	231.38	Investment Holding Companies	0.399202

Figure 3: *Stock-related data example from Refinitiv API*

Among these data, stock ticker (e.g., AAPL) with a company name (e.g., Apple Inc.) has been used in the 'entering stock input page' (See Figure 4 below.), to serve a list of stocks — allowing users to search for the stocks they hold or wish to track.

TradeInbox

Light

Enter stocks you wish to track!

Select Stock

Type to search...

AAON (AAON, Inc.)

AAPB (GraniteShares 2x Long AAPL Daily ETF)

AAPD (Direxion Daily AAPL Bear 1X Shares)

AAPL (Apple Inc.)

AAPU (Direxion Daily AAPL Bull 2X Shares)

AAXJ (iShares MSCI All Country Asia ex Japan ETF)

ABAT (American Battery Technology Company)

ABCL (AbCellera Biologics Inc.)

ABCS (Alpha Blue Capital US Small-Mid Cap Dynamic ETF)

ABEO (Abeona Therapeutics Inc.)

ABL (Abacus Life, Inc.)

ABLLL (Abacus Life, Inc.)

Figure 4. *Entering the stocks input page*

Moreover, stock exchange, sector, real-time stock prices, previous close, real-time stock price, and trading volume history were used to supply data in the left part of the dashboard. (see Figure 5 below).

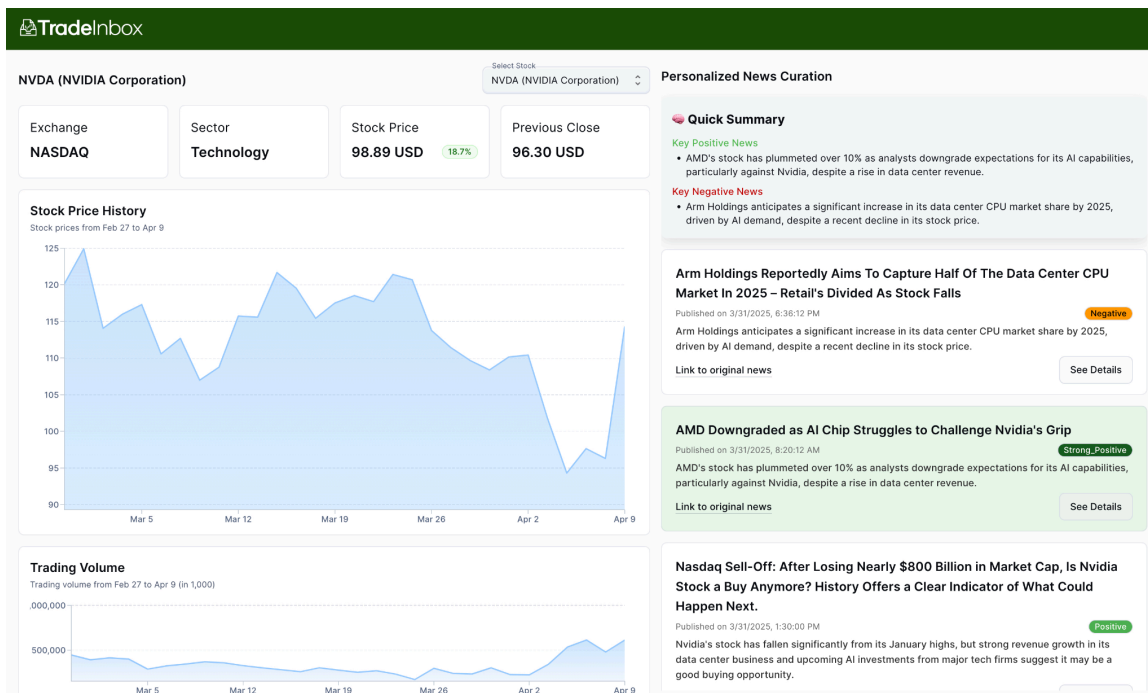


Figure 5. Dashboard page with data from Refinitiv API at the left side

3.2 System Architecture

The project consists of several system components designed to provide a personalized and real-time financial news notification system with summaries and personalized analysis, as shown in Figure 6 below.

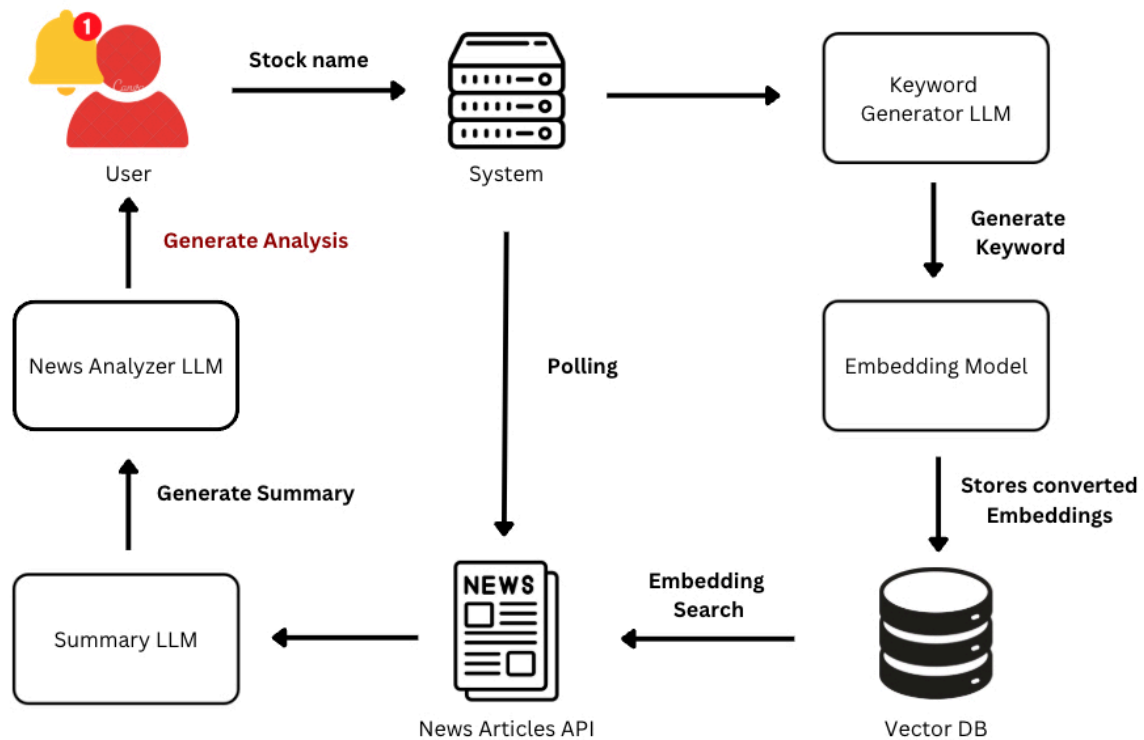


Figure 6. System Architecture - Overview of the LLM-based Real-time Personalized Financial News Notification System

As most of the components use LLM, prompt engineering was adopted because it tailors a general-purpose OpenAI GPT model to specialized finance tasks without the time and cost of fine-tuning. Carefully designed role instructions and a few-shot learning were implemented, and ensured responses follow a strict JSON schema.

3.2.1 Keywords Generator LLM

First, a Keywords Generator LLM has been implemented to serve as a pre-processing step to find relevant news articles tailored to individual users' stocks. Upon user registration, users are directed to provide stock names to the backend system for personalization. When stock

names are provided, the keywords generator LLM expands the data by generating 10 relevant keywords that may reflect potential news narratives. For example, if a user provides 'Coca-Cola' stock, the generated example keywords may include 'beverage industry trends,' 'sugar tax impact,' or 'global supply chain challenges.' Those keywords are generated to improve the process of finding relevant news articles for a better personalized experience through the embedded search. Especially, generating example keywords enables the system's embedding search model to go beyond a direct literal string match between the keyword (e.g., "Nvidia") and the news title or content. It allows embedding search against news content and all the keywords generated, enabling indirect search between them. The Keyword Generator LLM was instructed with a tailored prompt as below. Hence, in total, eleven keywords are generated per stock—ten from the Keyword Generator LLM and one from the ticker/company name itself, providing both direct and indirect semantic captures.

Context: I am building a system that uses embedding search to determine if a news article is relevant to a specific stock (e.g., 'Coca-Cola'). The goal is to provide real-time stock impact analysis. While exact name matching catches direct mentions, the system must also identify articles indirectly related to the stock to be effective. For example, news about the 'beverage industry trends' or a major competitor like 'PepsiCo earnings' could impact 'Coca-Cola' and should be considered relevant.

Your Objective: Generate a set of 5-10 diverse, semantically rich keywords and short phrases related to the provided stock name ({{\$stock_name}}). These terms are critical for improving embedding-based relevance detection for news articles. They will be embedded and used to search against news article embeddings.

Key Requirements for Generated Terms:

1. Capture Indirect Relevance: Go beyond the company name. Think about concepts, entities, and topics frequently discussed in relation to or affecting the target stock. Consider:

- Industry/Sector: Broader market trends, sector-specific regulations (e.g., "soft drink industry", "consumer staples sector").
- Major Competitors: News about direct rivals (e.g., "PepsiCo results", "Keurig Dr Pepper competition").

- Key Products/Brands: Significant product lines or brands owned by the company (e.g., "Sprite sales", "Dasani water", "Fanta marketing").

- Supply Chain/Partnerships: Major suppliers, distributors, or significant partnerships (e.g., "bottling partners", "sugar prices impact").

- Relevant Macro Factors: Economic or social trends strongly impacting the specific company/sector (e.g., "health trends beverages", "commodity cost inflation").

- Key Personnel (Less Common but possible): Sometimes news focuses on impactful executives (e.g., "CEO James Quincey statement").

2. News-Oriented: The terms should be phrases or concepts likely to actually appear in financial or business news articles related to the stock or its ecosystem.

3. Semantically Rich: Each term should represent a distinct, meaningful concept relevant to the stock's performance or perception.

4. Concise yet Effective: Aim for brevity (1-4 words typically), but prioritize capturing the concept accurately for embedding over extreme shortness.

5. Diverse: Cover different types of relationships (competitor, industry, product, macro factor, etc.).

Task: Based on the user-provided stock name below, generate 5-10 relevant terms meeting the above criteria.

Input: User-provided Stock name: {{\$stock_name}}

Output Format Constraint:

You MUST return ONLY a valid JSON object containing a single key "keywords" with a list of strings as its value. Do NOT include any explanations, apologies, or introductory text outside the JSON structure.

Your JSON Output format must be

```
{ "keywords":  
  
  ["xxx", "yyy", ...]
```

```
}
Your JSON Output:
```

Figure 7. Prompt given to the Keyword Generator LLM

The prompt above emphasises three core ideas. Firstly, it explicitly instructs the model to capture indirect relevance by listing industry trends, competitors, supply-chain partners, and macro factors, ensuring broader coverage. Secondly, it enforces output to be a strict JSON schema, which allows the following pipeline to parse results reliably without extra validation. Thirdly, diversity and brevity are mandated so that each keyword adds a distinct and unique semantic value.

3.2.2 Embedding Model

Next, the system converts these keywords into embeddings, which are then stored in the Milvus Vector Database for later use in embedding search to filter relevant financial news articles. Embeddings represent words or sentences as numerical vectors that capture the semantic relationships within text data [7]. Those embeddings are widely employed in embedding search due to their ability to capture and distinguish semantic nuances. This approach enables the determination of similarities not only between individual sentences but also across entire documents, making it a powerful tool for extracting relevant information. The Keywords Generator LLM mentioned in Section 3.2.1 enhances the stock data by expanding it into a more comprehensive dataset, enabling more robust embedding and thorough analysis. The figure 8 below illustrates a detailed embedding search process.

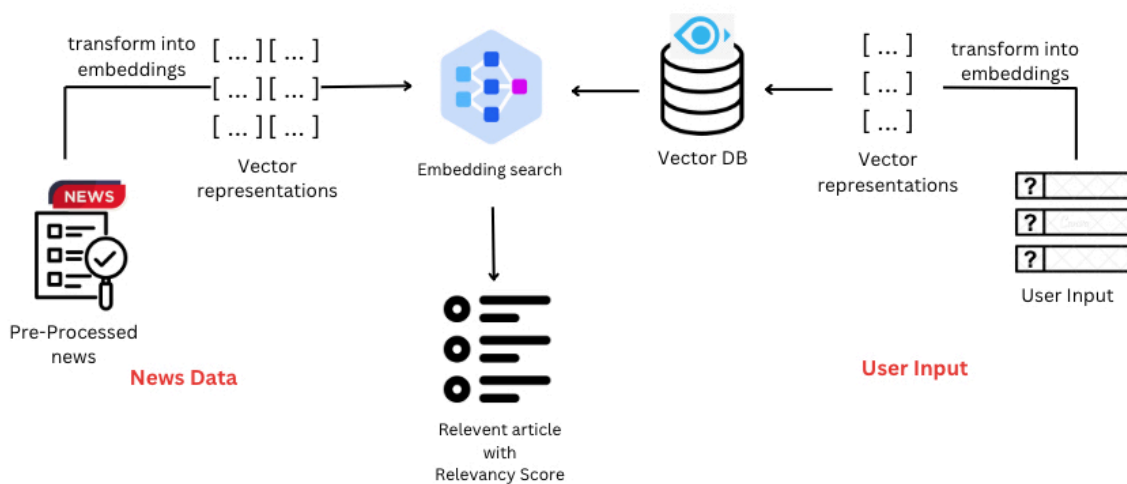


Figure 8. *Embedding search system design.*

On the right side under “User Input”, the 11 generated keywords from the Keyword Generator LLM discussed in the previous subsection are stored in the Milvus vector database. On the left side, the news article’s full content provided by the polling agent is converted into vector embeddings in real-time. Then, the embedding search is conducted between each news article and the keywords stored in the vector database. The cosine similarity of two embedding groups is calculated to obtain a similarity score, and only news articles whose highest similarity score exceeds our pre-defined threshold value of 0.4 are selected as relevant articles to a matched keyword.

The threshold value of 0.4 was selected as a result of qualitative analysis on the results, where the patterns show that irrelevant news articles such as ‘Spurs take on the Grizzlies on 3-game losing streak’, showed an extremely low similarity score of 0.016 when compared against the keyword embeddings of Nvidia Corporation. On the other hand, news directly related to Nvidia such as “Nasdaq Sell-Off: After Losing Nearly \$800 Billion in Market Cap, Is Nvidia Stock a Buy Anymore?” showed a moderate similarity score between embeddings of Nvidia Corporation. Although it did not show a strong similarity score, upon multiple observations, our team has safely deduced that even directly related news did not show strong relevancy trends due to embedding dilution from lengthy articles' full-text. Crucially, the value was chosen to be 0.4 to capture indirectly relevant news that did not mention “Nvidia” at all. For example, news such as “Arm Holdings Reportedly Aims To Capture Half Of The Data Center CPU Market In 2025” showed a similarity score of 0.437. Hence, setting the threshold of 0.4 filters out obvious irrelevant news while still admitting both direct and indirect news articles.

The matched news article that exceeds the threshold score is considered relevant to a specific stock and stored in the database with a matched keyword. It is then forwarded to the News Summary LLM and Stock Analyzer LLM for further analysis of the news article.

2.2.3 Polling Agent

Simultaneously, a polling agent continuously polls the financial news from the NewsCatcher API with an interval of 10 minutes. Figure 9 below illustrates the detailed workflow of the polling agent.

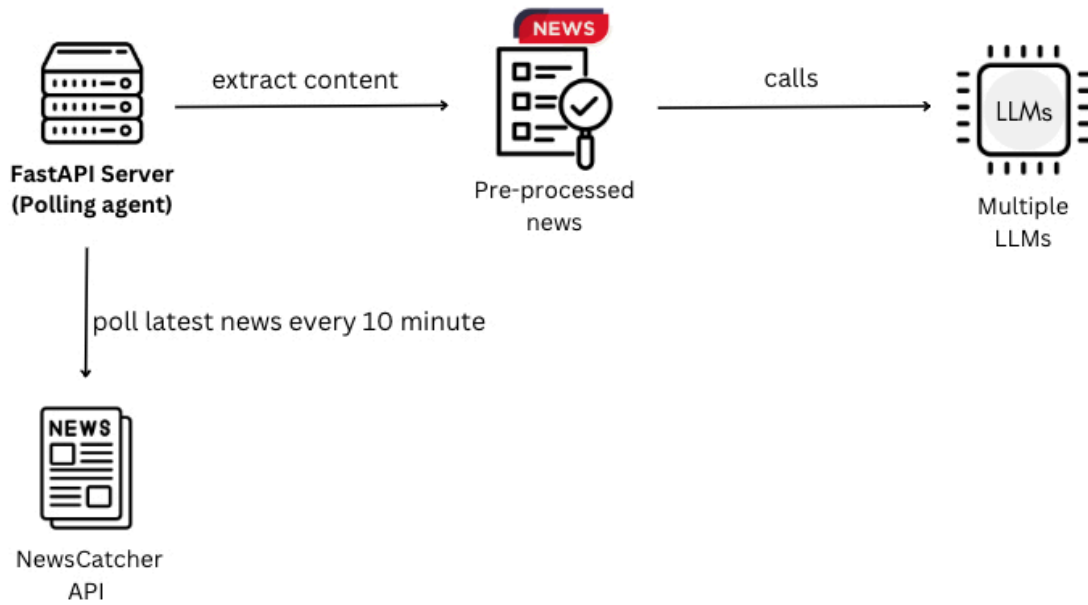


Figure 9. *Polling agent workflow.*

The main FastAPI server runs a polling agent implemented by the APScheduler library's BackgroundScheduler to poll news every 10 minutes asynchronously without blocking the main thread. Then, the agent pre-processes it by extracting news content from the full JSON response and converts the news content into embeddings. After retrieving news articles, the articles undergo a series of tasks, starting from pre-processing news to extract only relevant data (i.e., news title & content) from the full JSON result to generate embeddings of each news article, perform embedding search, and passing it to subsequent Large Language Models to generate various analysis. Since those multiple LLM APIs exert heavy network I/O which may cause significant delay if processed sequentially, Python's asynchronous functions through `asyncio` and `aiohttp` were utilized to process each task concurrently.

In the following subsections, multiple LLMs will be introduced that are used to feed data for the news detail page, as shown in the figure below.

Nasdaq Sell-Off: After Losing Nearly \$800 Billion in Market Cap, Is Nvidia Stock a Buy Anymore? History Offers a Clear Indicator of What Could Happen Next.

Published on 2025. 3. 31. 1:30:00 PM

Summary

Nvidia's stock, which reached an all-time high of \$149.43 in January, has seen a significant decline, dropping its market capitalization to \$2.9 trillion due to various economic factors and competition from emerging AI startups. Despite this sell-off, Nvidia's data center revenue surged by 142% year-over-year, driven by strong demand for its GPUs, particularly in AI infrastructure. Analysts suggest that the stock is currently undervalued and poised for recovery, especially with substantial investments in AI infrastructure from major tech companies. Given its historical resilience and growth prospects, now may be an opportune time to invest in Nvidia shares.

Key Metrics

- Nvidia's market cap decreased to \$2.9 trillion, down \$800 billion from its previous high.
- Data center revenue reached \$115 billion, up 142% year over year.
- Fourth quarter data center revenue was \$35.6 billion, nearly double from the prior year.
- Nvidia stock has risen by 615% since November 30, 2022.
- Forward P/E ratio is 26.7, 47% off its three-year high.

Sentiment Analysis

Overall Sentiment: Positive

Despite the recent sell-off and a significant drop in market cap, Nvidia's strong revenue growth in its data center business, particularly in AI infrastructure, indicates robust future prospects. The company's forward P/E ratio suggests it is undervalued compared to historical averages, and the planned investments from major tech players in AI further bolster its growth potential. Historical trends show that Nvidia has rebounded from sell-offs, making this a strategic buying opportunity.

Stock Impact Analysis

Nvidia's stock has dropped significantly from its peak of \$149.43 to around \$2.9 trillion in market value due to various factors like new tariffs and competition from a Chinese AI startup. However, Nvidia's earnings show strong growth, especially in its data center business, which is crucial for AI technology. Major companies like Apple and Amazon are investing heavily in AI, which bodes well for Nvidia's future. Despite the recent sell-off, the stock is considered a good buy because it is currently cheaper than usual compared to its earnings.

Beginner

Intermediate

Expert

Figure 10. News Detail page showing Summary, Key Metrics, Sentiment Analysis and Stock Impact analysis

3.2.4 Summary LLM

First, to save users' time reading through the full content of the selected article, the application was designed to provide a concise summary of the news article by implementing the Summary Generator LLM. The prompt given to the model is as below.

```
[%system%]

You're a professional journalist and an expert in summarizing a news article
into a concise and high-quality summary.

Your Task: Generate a 3-4 sentences summary from the given news article
content from any topic. In addition, generate a short ONE-SENTENCE summary.


[IMPORTANT]

Note that the generated keywords and examples MUST NOT BE too general. IT
MUST BE RELATED TO the provided user input.

{

  "summary": "xxx",

  "one_sentence_summary": "yyy"

}

YOU MUST RETURN ONLY JSON. DO NOT INCLUDE ANYTHING ELSE IN YOUR RESPONSE.

News Article Content: {

    {$article_content}

}

Your JSON Output:
```

Figure 11. *News Summary LLM Prompt*

The prompt directs the system to generate a summary in a concise and high-quality manner, and emphasizes it to be related to the provided user input. It also enforces returning the summary in JSON format for storage in the database and further processing. Moreover, the `one_sentence_summary` is generated, allowing users to scan through a short one-line summary on the dashboard and choose what to read further.

3.2.5 Key Metrics LLM

Followed by the summary, key metrics from the news article were extracted by the large language model. This allows users to scan through quantitative key metrics, including numerical values and financial indicators relevant to the company or market mentioned that may impact the stock price. This serves as a useful tool especially for experienced investors, who can quickly grasp the current topics discussed in the article through the factual, quantitative data. Figure 12 below illustrates the prompts given to implement Key Metrics LLM.

To guide the model further, our system employed few-shot learning where an example input-output pair is embedded in the prompt under [%user%] and [%assistant%], feeding the full article text and the desired JSON output. These exemplars teach the LLM the required extraction pattern without additional fine-tuning, reducing formatting errors and ensuring the model's consistency.

```
[%system%]

You're a professional financial data extraction assistant and an expert in
analyzing news articles and generating key financial metrics in the news
article. Focus on numerical values related to the company's financial
performance, stock movement, and market reactions.

Your Task: Extract key financial metrics from the provided financial news
article. Focus on numerical values and financial indicators relevant to the
company or market mentioned.

[IMPORTANT]

Note that the key metrics MUST NOT BE generated on your own. It must be
consolidated from the user input. Restrict your each response into one short
concise sentence. Return 1 to 5 key metrics, ensuring each is short and
concise while covering all critical financial indicators. If there's no key
metrics, return empty list in JSON format given.

[%user%]
```

Article: NVIDIA Co. (NASDAQ:NVDA - Get Free Report)'s stock price traded down 1.2% on Thursday after an insider sold shares in the company. The stock traded as low as \$131.80 and last traded at \$135.34. 225,240,656 shares traded hands during trading, a decline of 43% from the average session volume of 395,069,875 shares. The stock had previously ...

[%assistant%]

{

"key_metrics": [

"Stock traded as low as \$131.80; last price at \$135.34; previous close was \$136.92.",

"Volume: 225,240,656 shares, 43% below average of 395,069,875 shares.",

"Insider transaction: 716 shares sold at \$142.00 for \$101,672.00; director's holding decreased by 3.47% to 19,942 shares.",

"Quarterly earnings: Revenue of \$35.08B vs. estimate of \$33.15B; EPS reported at \$0.81, beating the \$0.69 consensus by \$0.12.",

"Valuation: Market cap of \$3.39T; P/E ratio of 54.41; Debt-to-Equity ratio of 0.13; Beta of 1.66.",

"Corporate action: Board authorized a \$50B stock repurchase program."

]

}

Extraction Criteria:

1. Company Performance: Revenue, Net Income, Earnings Per Share (EPS), Year-over-Year (YoY) or Quarter-over-Quarter (QoQ) changes, Operating Profit, Gross Margin, etc.
2. Stock Market Metrics: Stock price changes (e.g., % increase or decrease), pre-market or after-hours movement, analyst target price updates, trading volume, market capitalization, etc.
3. Financial Ratios: Price-to-Earnings (P/E) ratio, Debt-to-Equity (D/E) ratio, Dividend Yield, Free Cash Flow, etc.

```
4. Macroeconomic Indicators (if applicable): Interest rate impact, inflation
rate, GDP growth, unemployment rate.

[%user%]

YOU MUST RETURN ONLY JSON. DO NOT INCLUDE ANYTHING ELSE IN YOUR RESPONSE.

Return the key metrics as concise, well-formatted short sentences.


Article: {{$content}}

Your JSON Output:
```

Figure 12. *Key Metrics LLM Prompt*

The prompt also contains a system message that defines the model’s role as a “professional financial data-extraction assistant” and explicit instructions limiting each response to 1–5 concise sentences in JSON format, ensuring consistent and accurate outputs.

3.2.6 Sentiment Analysis LLM

Followed by the Key Metrics LLM, a Sentiment Analysis LLM has been implemented. The following figure illustrates the prompt given to the large language model.

```
[%system%]

You're a professional financial expert who specializes in sentiment analysis
of a financial news article related to a provided stock name.


The sentiment analysis includes the following items.


1) One of [Strong Negative, Negative, Neutral, Positive, Strong Positive].

    - where strong negative means the given news article or the contents of the
news article are high indicators of the given stock's price will fall down in
a future.

    - where strong positive means the given news article or the contents of the
news article are high indicators of the given stock's price will go up in a
future
```

- Neutral means that it's not reasonable or there's no significant indicator of the stock price from the news article.

2) Insightful, factual, reasonable, logical, detailed analysis & rationale & proofs & evidence of your claim from item (1).

Do not merely give meaningless, verbose, no-depth, abstract reasons. Give DEFINITIVE PROOFS OR EVIDENCE OR RATIONALE to your claim in a structured manner.

3) Also, provide a "SENTIMENT SCORE" ranging from [-5 ~ 5] (inclusive) to indicate how positive/negative the given news is.

- 5 means Strong positive, and -5 means Strong Negative, and 0 means Neutral.

- Give it as a floating-point number.

Make sure that your sentiment analysis output is CONCISE AND INSIGHTFUL. It should not be verbose and long. Must be around 3~4 sentences depending on the situation.

In addition, you will be provided a specific stock name that you must analyze the impact with.

Your sentiment analysis of the news article must be done in consideration of this given stock name.

YOU MUST RETURN ONLY JSON. DO NOT INCLUDE ANYTHING ELSE IN YOUR RESPONSE.

You must return response in the following "JSON" format (only JSON)

```
{  
  
  "sentiment": "xxx", // item (1) - either Strong_Negative, Negative,  
Neutral, Positive, Strong_Positive  
  
  "sentiment_score": xx,  
  
  "analysis": "..." // item (2)
```



```
}

Given stock name: {{$stock_name}}

News Title: {{$title}}

News Content: {{$content}}

Your JSON Output:
```

Figure 13. *Sentiment Analysis LLM prompt*

According to the prompt above, the model receives the stock name, news title, and news full content to generate 3 outputs. Firstly, it generates a `sentiment` label ranging from [Strong Negative, Negative, Neutral, Positive, Strong Positive] scale that captures the overall tone of the article toward the stock, letting users a to gauge the sentiment at a glance. Secondly, `sentiment_score`, which is a numerical floating-point value between -5 and 5 that quantifies the sentiment. The quantified value will be used later in the dashboard UI to deliver “Quick Summary”, where the news articles are divided into positive and negative news and sorted in descending order to provide the headline of the most impactful news article, which means having highest/lowest sentiment score. Lastly, a concise sentiment analysis of the news article is generated in relation to the specific stock name given. The model was mandated to generate only insightful, factual, reasonable, logical, detailed analysis & rationale & proofs & evidence in generating the analysis, to deliver only impactful analysis to the user. Combined with a `sentiment` tag, the analysis was instructed to be concise and insightful to prevent users from spending too much time reading the analysis but grasp the takeaway quickly.

3.2.7 News Impact Analysis LLM

Lastly, the News Impact Analysis LLM has been implemented with a curated prompt (see Figure 14 below). It goes beyond the sentiment analysis by focusing on generating an analysis on the actual possible impact of the news article on the stock's price.

```
[%system%]
```

You are a stock market news analysis agent that evaluates how news impacts stock prices, catering to retail investors with three expertise levels (easy, intermediate, expert).

The news could be discussing any kind of topic, but related to the stock. The key task is to find an impact to stock price of this news, and possibly provide a logical explanation behind the stock price prediction in English.

You will also be given a stock name. Create your stock impact analysis report towards the given stock.

<Requirements>

1. Summary of the news is not compulsory, mainly discuss the implication to the stock price and logical explanation behind it.
2. If the news has negligible impact to stock price then you can just give some logical explanation of why it is not important.
3. For the analysis, please find below instructions for your reference. It would be better if you can provide industry-specific viewpoints, as well.
4. ****Easy****: Explain all industry-specific/technical/financial terms (e.g., P/E ratio, EBITDA) in simple language. Give full definition of the terms below the analysis. better
5. ****Intermediate****: Assume basic financial knowledge; skip explanations for common terms (e.g., dividends, market capitalization). Better to give some explanation of complex technicals below the analysis.
6. ****Expert****: Use advanced knowledge (e.g., discounted cash flow, beta volatility and many others) and financial ratios without much explanations.

<Output Format>

Please return a JSON format like the following:

```
{
"easy": "...
"intermediate": "...
"expert": "...
}

[%user%]

Example 1: Biotech FDA Approval

Stock: BioPharma Inc.

News Content:

BioPharma Inc. receives FDA approval for its Alzheimer's drug, projects $1.2B
in peak annual sales, and sets a 12-month price target of $85. A short seller
report warns of trial data inconsistencies.

[%assistant%]

{
"easy": "BioPharma Inc. got approval from the FDA (U.S. drug regulators) for
its Alzheimer's treatment, which it expects to generate $1.2 billion per
year. Analysts predict the stock could reach $85 within a year, but a
critical report claims some test results might be unreliable. Risks include
competition from larger drugmakers and high R&D costs (money spent developing
new drugs).",
"intermediate": "BioPharma's FDA approval supports a bullish $85 PT (35x
P/E), but a $300M per quarter cash burn raises dilution risks. Pipeline
catalysts include a Parkinson's drug entering Phase 2 trials.",
"expert": "BioPharma's Alzheimer's drug approval (FDA label includes broad
indication) drives PT to $85 (DCF: WACC 12%, $1.2B peak sales, 55%
probability-adjusted). Short seller claims on trial data heterogeneity
```

```
(p=0.07 in subgroup) may limit near-term upside. With a cash runway of six
quarters at the current burn rate, an equity offering (15-20% dilution) is
likely. EV/sales at 5x vs. sector 7x reflects pipeline overhang."
}

[%user%]

YOU MUST RETURN ONLY JSON. DO NOT INCLUDE ANYTHING ELSE IN YOUR RESPONSE.

Stock: {{$stock}}

News Content: {{$content}}

Your JSON Output:
```

Figure 14. *News Impact Analysis LLM prompt*

It is prompted to generate the news analysis in 3 different difficulty levels - Easy, Intermediate, and Expert. Each analysis includes a prediction of how the news article might impact the stock price. A crucial part of this model is that it generates this three-tiered analysis to serve users who may have different financial literacy. Some users may not recognize financial terms that appear in the news article such as the P/E ratio and understand why it matters, while others can grasp them easily. Therefore, to serve our objective in educating novice retail investors, the model is instructed to provide three different levels of analysis. Novice retail investors can build financial knowledge through the explanation on industry-specific/technical/financial terms (e.g., P/E ratio, EBITDA) in simple language. By tailoring depth to user needs, the system delivers actionable guidance and simultaneously fosters financial literacy.

3.2.8 Delivery via Discord Notification

As the final step, after generating all analyses and summaries by the LLMs discussed, a concise notification for news articles curated with sentiment and summary is delivered automatically via Discord, enabling a real-time and personalized notification on relevant news altogether. This is achieved by using Discord's webhook API, where our system sends

an HTTP request to the Discord Webhook server that sends the notification to the selected Discord channel. This allows delivering real-time and personalized notifications on relevant news to each user.

3.3. Feedback for Embedding Search Results

Another core feature to support better relevancy matching by the embedding search is providing users with a feedback button to receive feedback on whether the matched article was in fact relevant to the user's stock. Since the concept of relevancy can be subjective, especially when articles is indirectly related to stock name where it do not explicitly mention the company name but are still contextually related, the feedback loop has been implemented to address this limitation.

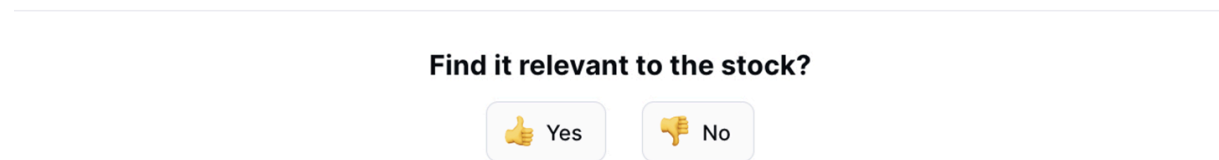


Figure 15. *Stock detail page with feedback loop feature.*

The UI/UX has been implemented in a simple way so that the user is able to click either “Yes” or “No” in response to the question “Did you find it relevant to the stock?” in order to simplify the feedback collection process for better user experience.

Initially, the team experimented with adjusting the similarity score threshold value to reflect user feedback, for example, making it higher by 0.1 when “No” button was clicked. However, this method was too general and ineffective, as it applied changes to all keywords at once even though the user might have felt that only the specific keyword matched was irrelevant. Therefore, to implement the feedback loop without affecting other relevant matches, the current implementation solves this by selectively removing only the matched keyword vector, enabling a more precise personalization.

When the user clicks “No”, the system proceeds to delete that keyword vector for the user-stock pair in the Milvus vector database. This means the user didn't find the article captured by that specific keyword was relevant to the sstock, hence removing the keyword will ensure that future incoming news will not be captured by that keyword. After the

removal, the Keyword Generator LLM will re-generate another keyword to maintain a total of 10 keyword vectors.

This process allows each individual user to maintain their own list of relevant keywords related to each stock they own, reflecting their subjectivity in viewing the relevancy, thus giving the most possible personalization to each user based on unique interpretation of what is relevant.

3.4 Engineering choices

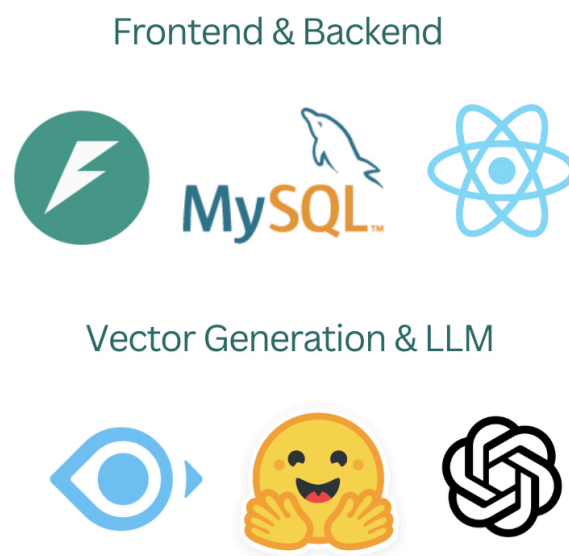


Figure 16. *Development tools used: FastAPI, MySQL, React, Milvus, HuggingFace, and OpenAI, in that order.*

The main backend system of this project has been implemented using FastAPI and Python. Since the system heavily utilizes large language models, the components related to the LLMs were built with FastAPI and Python too, as their implementation is best supported the best in a Python environment. FastAPI also provides fast and high performance, supported by the Asynchronous Server Gateway Interface (ASGI) which operates on a single-threaded model, providing high performance.

MySQL was used to store user data, stock information, news articles and their corresponding analysis and key metrics. A relational database has been utilized for efficient storage of structured data with a concrete schema.

The Frontend was implemented using React, leveraging its strong ecosystem and reusability of components to build a dynamic single-page application. For example, the dashboard UI consists of multiple reusable components, such as news card list.

On the other hand, Milvus was selected as a vector database due to its high performance and ease of use by providing abstraction. HuggingFace and OpenAI were chosen for their cost-effectiveness and State-of-the-Art Models supported by a robust ecosystem.

The server will be deployed on Amazon Web Services (AWS) to ensure high reliability and scalability.

3.5 Database Design

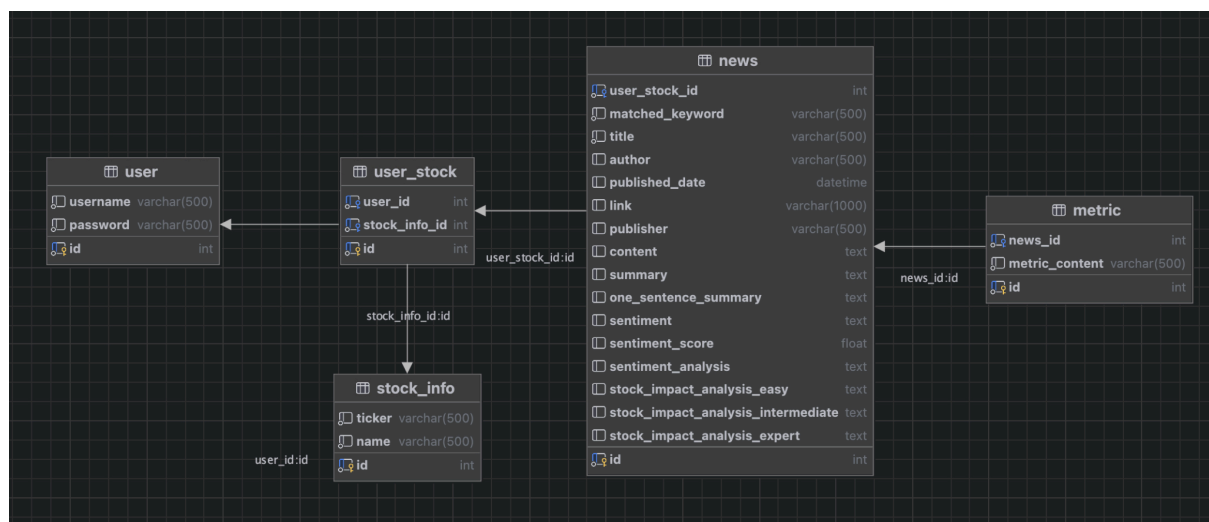


Figure 17. Entity-Relationship Diagram of the system database schema

Figure 17 above illustrates the relational database schema designed to support the core functionalities of the backend system.

The user information, including username and password is stored in the 'user' table, while stock-related information fetched from the Refinitiv API is stored in the 'stock_info' table, which includes the stock ticker and stock name. The list of stocks shown on the entering user stock page (see Figure 4) is retrieved from this table.

The intermediate 'user_stock' table represents the many-to-many relationship between users and stocks, allowing each user to subscribe to multiple stocks and vice versa. This structure

supports personalization by maintaining a unique `user_stock_id` for each user-stock pair, which is later used to associate matched news.

The `news` table is central to the system, storing all matched articles per `user-stock`. It includes fields such as title, author, publisher, link, and published_date as raw metadata, and also stores LLM-generated content like summary, sentiment analysis, stock impact analysis, etc.

Importantly, each news entry records the `matched_keyword`, which is essential for the feedback loop discussed in Section 3.3. If a user clicks “No” on the feedback, the system identifies the keyword associated with that article and deletes its corresponding vector from the Milvus database.

Lastly, since each news article can produce up to 5 metrics, the `metric` table was added to represent a one-to-many relationship between news and metrics.

This schema supports the personalized workflow of the whole system, from user-stock interest tracking to news retrieval, analysis, and saving them for efficient lookup and reuse.

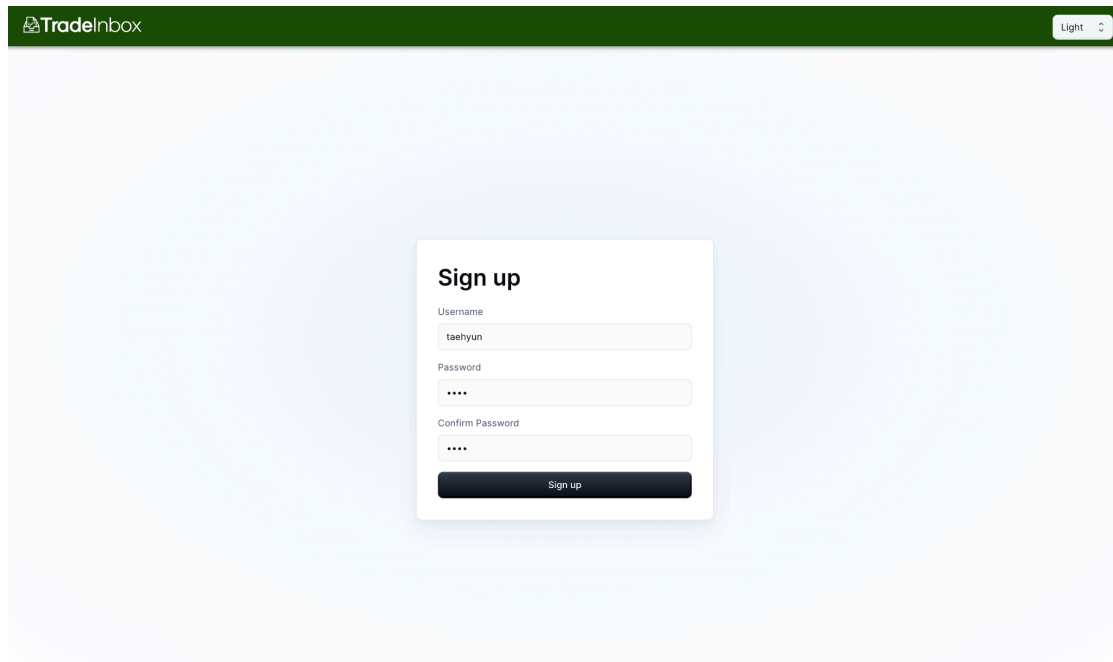
4. RESULTS & DISCUSSION

This section discusses the results of the project including the frontend implementation discussed from the user's perspective (Section 4.1), as well as the results and experiments from the LLMs discussed in the methodology sections. (Section 4.2).

4.1 Frontend Result

This section discusses the results of the frontend implementation, along with some details of the backend implementation.

4.1.1 Sign-up page

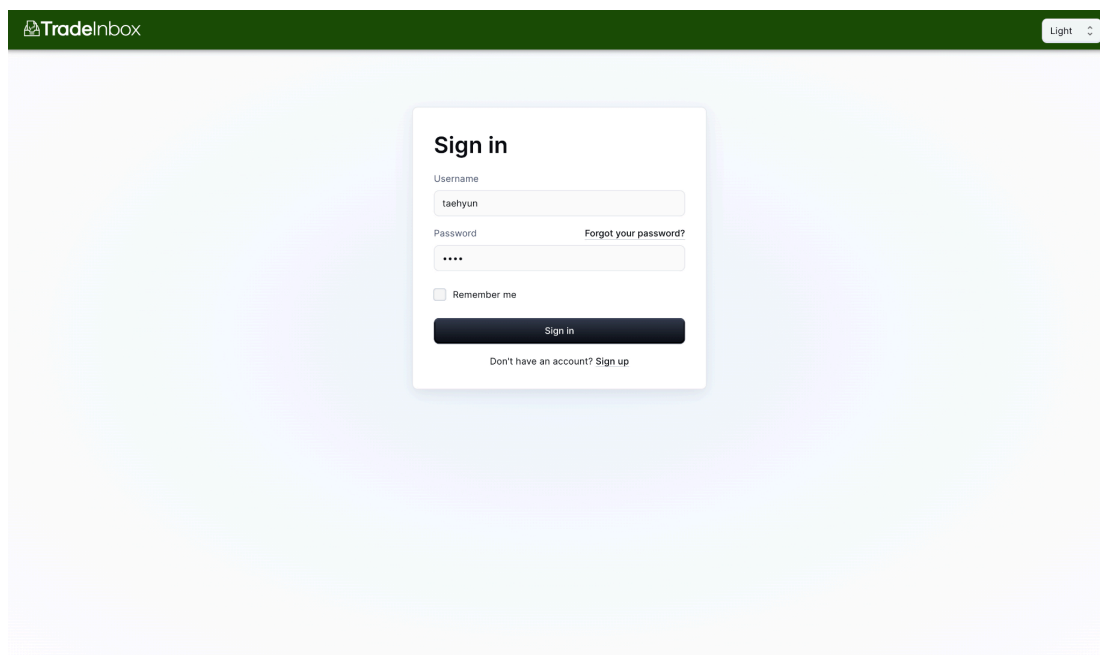


The screenshot shows the 'Sign up' page of the TradeInbox application. The page has a dark green header with the 'TradeInbox' logo on the left and a 'Light' theme toggle on the right. The main content area is light gray. In the center, there is a white card with a dark green border. The card is titled 'Sign up' in bold. Below the title, there are three input fields: 'Username' with the value 'taehyun', 'Password' with four dots, and 'Confirm Password' with four dots. At the bottom of the card is a dark green button labeled 'Sign up'.

Figure 18. *Sign-up page*

First, the sign-up page has been implemented. It allows a simple and straightforward sign-up process with a username and password.

4.1.2 Sign-in page



The screenshot shows the 'Sign in' page of the TradeInbox application. The page has a dark green header with the 'TradeInbox' logo on the left and a 'Light' theme toggle on the right. The main content area is light gray. In the center, there is a white card with a dark green border. The card is titled 'Sign in' in bold. Below the title, there are two input fields: 'Username' with the value 'taehyun' and 'Password' with four dots. To the right of the password field is a link that says 'Forgot your password?'. Below the password field is a checkbox labeled 'Remember me'. At the bottom of the card is a dark green button labeled 'Sign in'. Below the button is a link that says 'Don't have an account? Sign up'.

Figure 19. *Sign In page*

After a successful sign-up, the user is directed to sign in. Upon user sign-in, a JSON Web Token (JWT) is issued and added in the 'Authorization' Header of every subsequent HTTP request. The JWT contains the userId of the user using the system, enabling the system to always identify the user using the header. The userId is being used to identify which user is using the app across all API calls, to fetch their personalized news curation.

4.1.3 Stock Input page

Upon the first sign-in, users are directed to select the stocks they wish to track. Figure 20 shows the page where the user can choose or search from the drop-down list of all the stocks listed on the New York Stock Exchange.

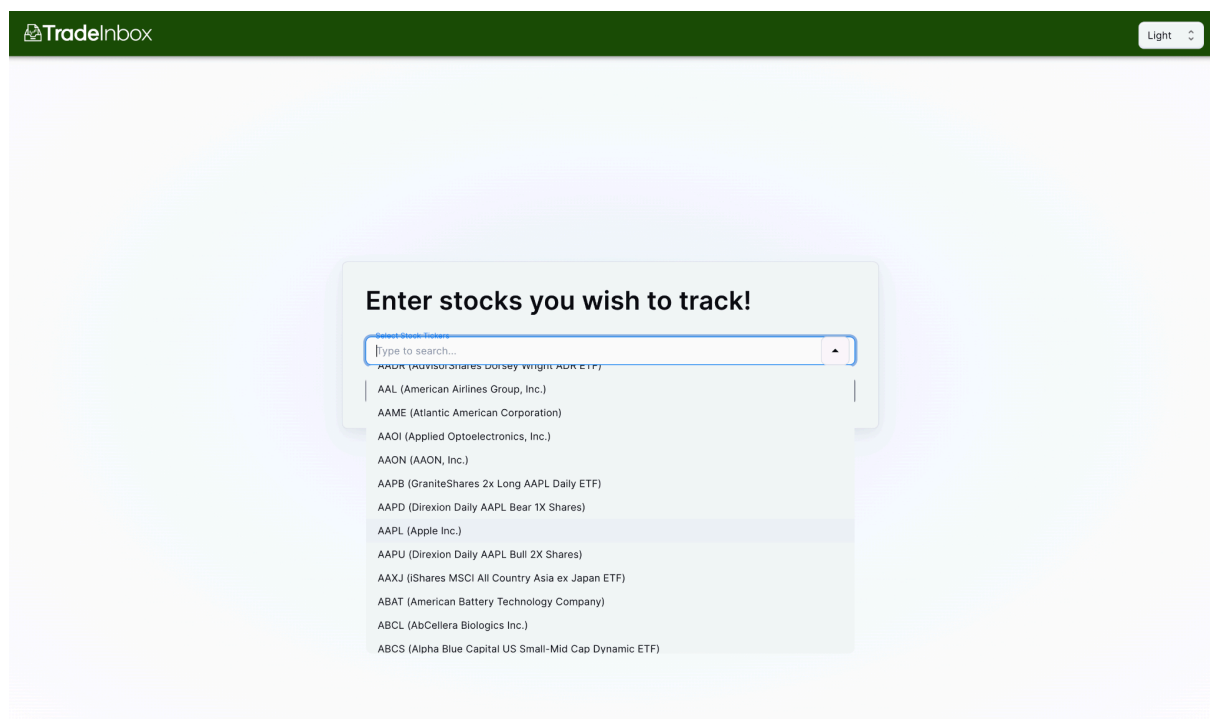


Figure 20. *User stock input page with stock lists*

Users can choose multiple stocks, allowing them to track their various portfolio holdings effectively. Figure 21 below shows the same page after the user has chosen multiple stocks.

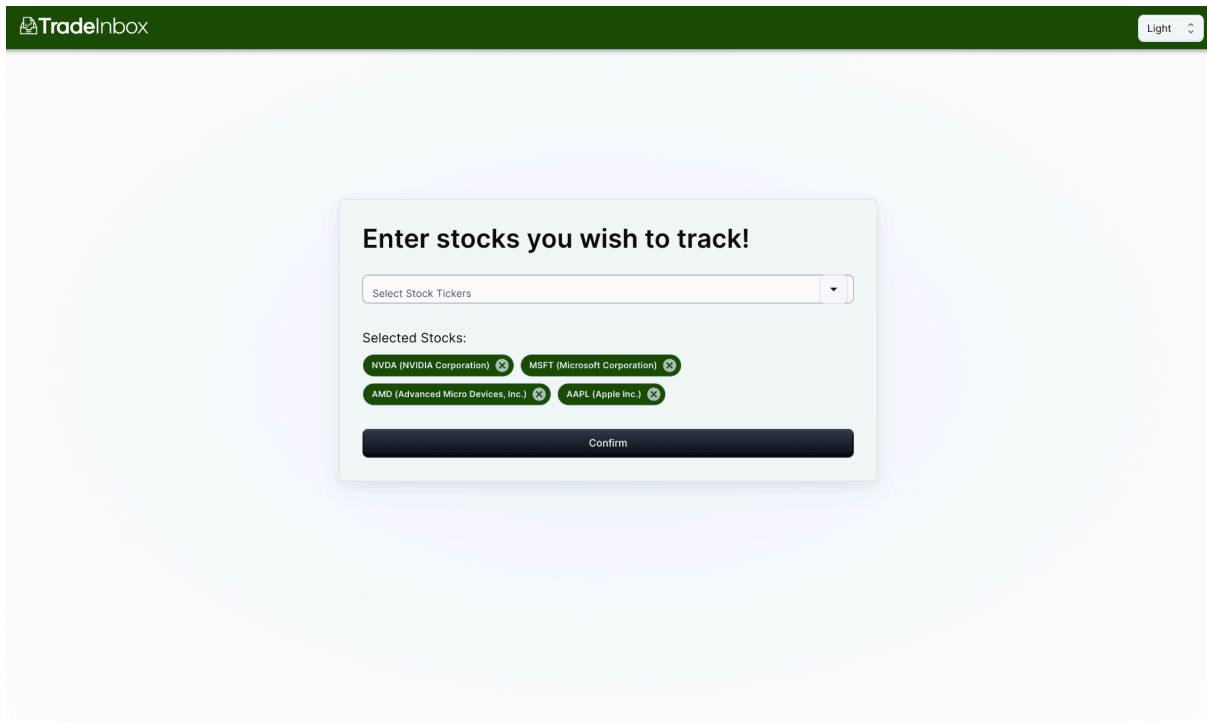


Figure 21. *User stock input page with stocks selected*

As demonstrated in Figure 21 above, users can choose multiple stocks and easily remove them. Additionally, when the user chooses a specific stock, that stock will no longer be visible in the news drop-down list for a better user experience. When the user removes the selection, it reappears in the list.

When the user clicks the “Confirm” button, the process of generating relevant keywords begins. The Keyword Generator LLM discussed in the methodology section (see Section 3.2.1) is triggered and generates 10 keywords per stock, which are stored in the Milvus vector database along with the `user_stock_id` for personalized news curation.

4.1.4 Dashboard Page

Once the user confirms their stock selections, they are redirected to the main dashboard page. Several experiments and improvements were made to the dashboard page, including a major UI/UX redesign. Based on iterative feedback, we optimized the layout to enhance readability and reduce the time required to interpret news content.

The initial version had a top-down layout in dark mode, where news articles appeared at the bottom of the page. However, this design made it difficult for users to quickly grasp relevant content (See Figure 22 below.)

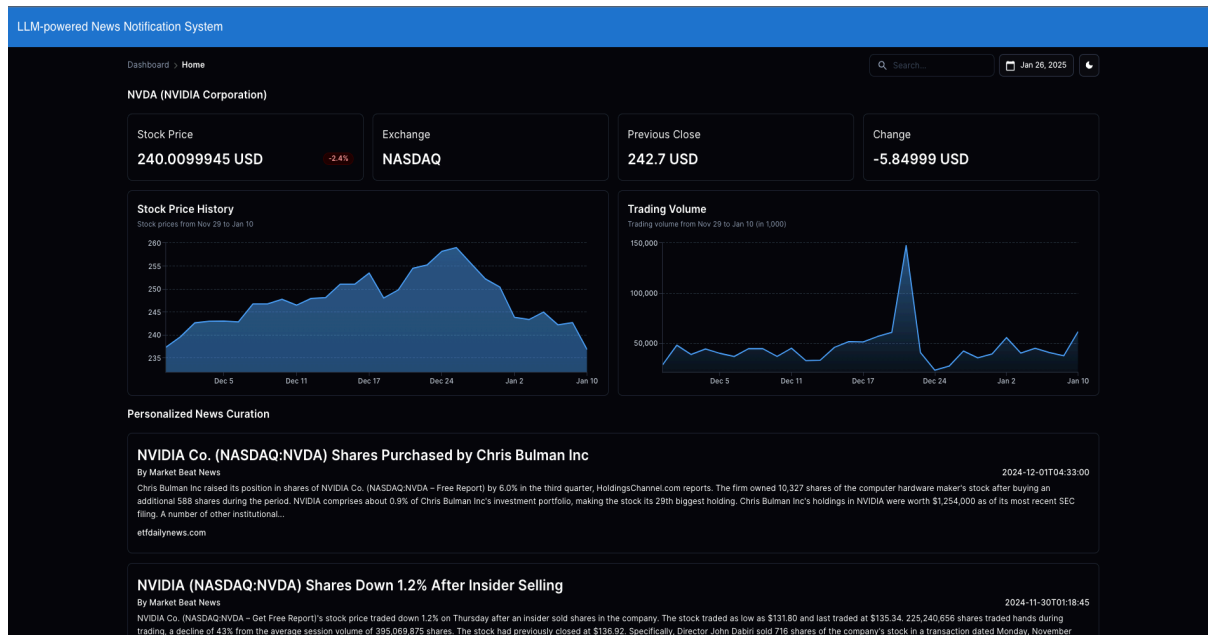


Figure 22. *Initial dashboard page design*

To enhance readability and streamline user experience, the dashboard was restructured into a left-right layout. The left section of the revised dashboard page (See Figure 23 below) displays output of the Refinitiv API integration, presenting key stock-related information including exchange, sector, stock price, and previous close. The stock price history and trading volume history were visualized through charts, providing users with an all-in-one platform where they can monitor relevant news alongside corresponding price and volume movements. Also, the “Personalized News Curation” section was moved to the right part from the bottom since the main focus of TradeBox is to provide users with personalized news curation with analysis.

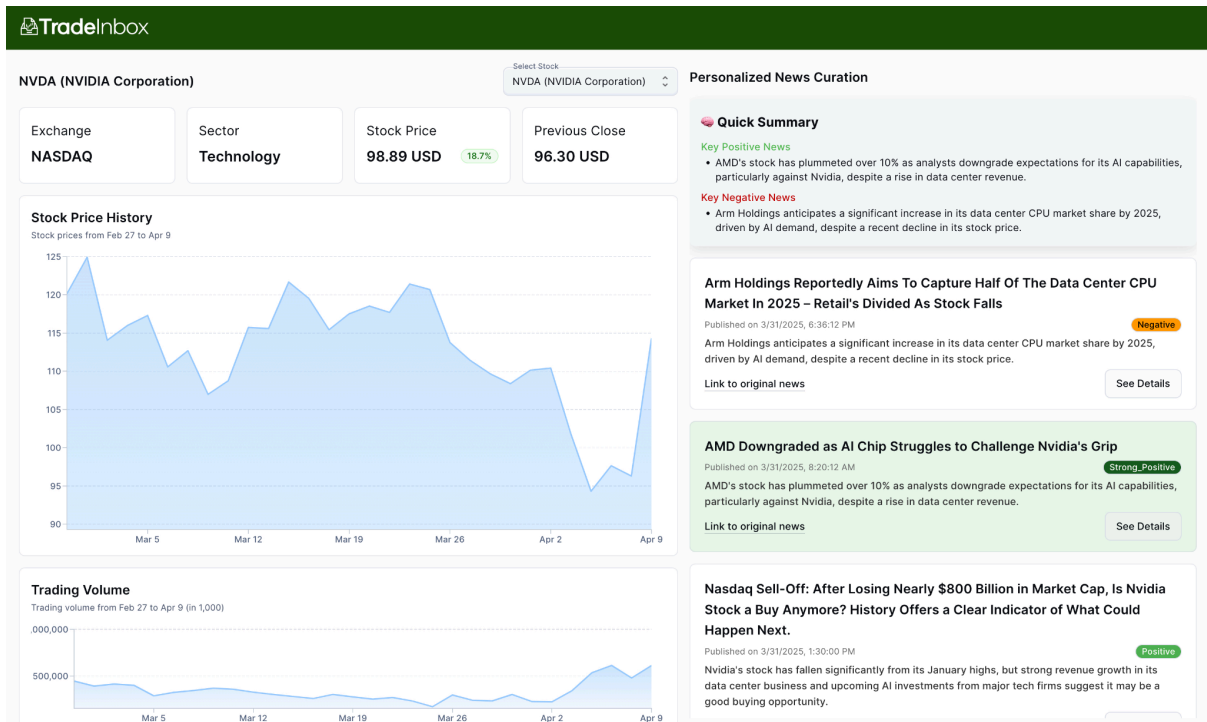


Figure 23. *Revised Dashboard page*

Meanwhile, the right section focuses on personalized news curation, including real-time articles matched through embedding search. To help users quickly identify high-impact updates, news articles classified with `strong_positive` or `strong_negative` sentiment were visually highlighted using green and red background colors, respectively.

Moreover, a “Quick Summary” section was implemented at the top-right corner of the dashboard to emphasize the key positive and negative news on the selected stock. (See Figure 23 above). Here, the ‘sentiment score’ generated by the Sentiment Analyzer LLM was utilized to automatically choose the top 2 news articles with the highest and lowest sentiment score, displaying them under “Key Positive News” and “Key Negative News”.

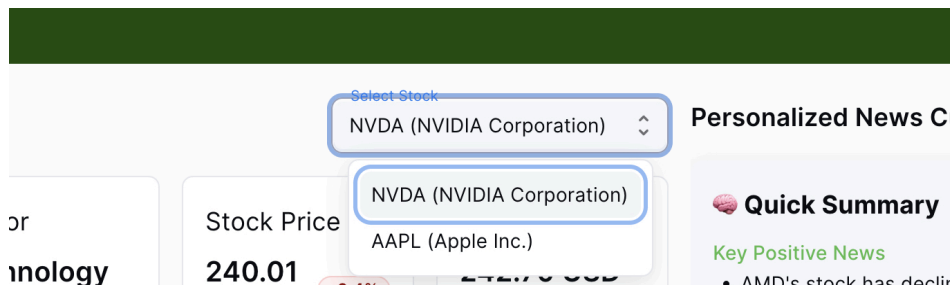


Figure 24. *Choosing a stock among multiple stocks*

Finally, multi-stock functionality has been developed. Users can seamlessly change the stock they are currently viewing, which dynamically updates the dashboard content to show only the selected stock’s information and news curation.

4.1.5 News Detail Page

Lastly, clicking the “See Detail” button on each news article opens a dedicated news detail page (See Figure 25 below).

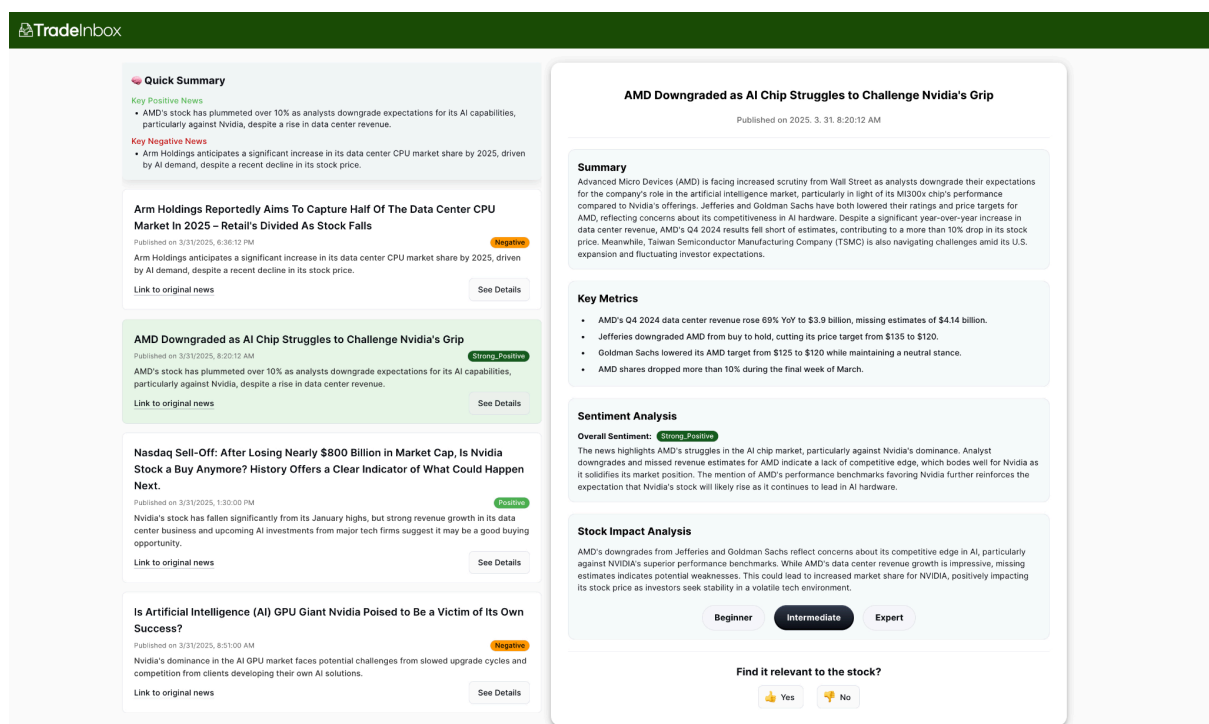


Figure 25. *News Detail Page*

This page provides users with a comprehensive, yet effectively concise information including summary, key metrics, sentiment analysis, and stock impact analysis, which are generated by Large Language Models discussed in the Methodology section (See section

3.2). The page has a side-by-side panel structure, where the left panel allows users to browse articles and the right panel displays detailed content, making it easy for users to navigate through multiple news articles.

In the “Stock Impact Analysis” section, users can select the desired level of explanation from Beginner, Intermediate, or Expert based on their financial literacy. The analysis content dynamically updates without any delay of API calls or page reload, offering a seamless user experience.

Lastly, at the bottom-right corner, a feedback component allows users to indicate whether they found the article relevant to the stock. Based on the response, background logic is triggered as described in the feedback loop mechanism (see Section 3.3), enabling the system to refine future recommendations.

4.1.6 Real-time Notification via Discord Channel

Simultaneously, every news article that appears in the dashboard news curation is sent to the user through a designated Discord channel. This enables users to receive timely updates without needing to manually revisit our website to check for fresh new articles.

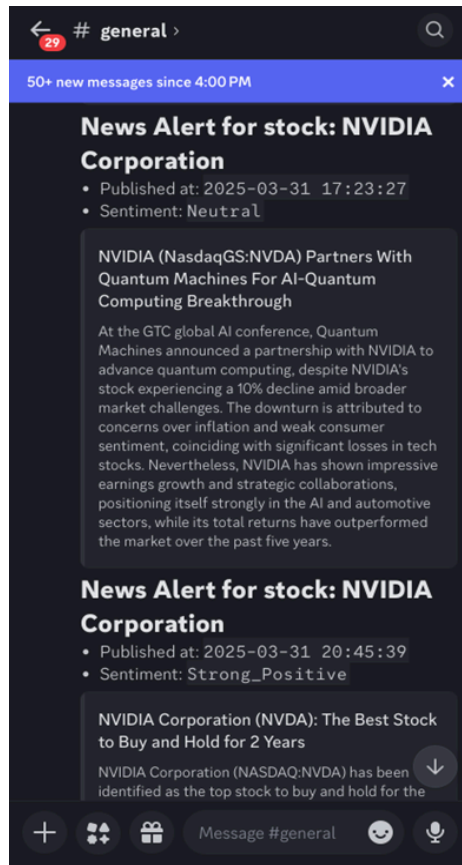


Figure 26. *Discord Notification for Nvidia stock*

Figure 26 shows an example of the notification related to Nvidia. The content of the notification includes the stock name, published date, sentiment, news title and a summary, which is selected to provide a brief overview of the news at a glance. Each message also contains a clickable link that redirects the user to the corresponding News Detail Page of the system, allowing for seamless access to full analysis if desired.

4.2 Results & Experiment on Embedding Search and Keyword Generator LLM

This section discusses the experiment and results of implementing the keyword Generator LLM and the embedding search.


```

Stock Name: NVIDIA Corporation
Example Keywords: [
    'graphics processing units',
    'AI technology trends',
    'gaming industry growth',
    'data center demand',
    'autonomous vehicle partnerships',
    'semiconductor supply chain',
    'cloud computing expansion',
    'major competitor AMD',
    'machine learning applications',
    'CEO Jensen Huang statements'
]

Stock Name: Apple Inc.
Example Keywords: [
    'iPhone sales',
    'Apple Watch market',
    'MacBook performance',
    'App Store revenue',
    'smartphone competition',
    'supply chain disruptions',
    '5G technology impact',
    'CEO Tim Cook statements',
    'consumer electronics trends',
    'global chip shortage'
]

```

Figure 27. *Results of Keywords generated from the Keyword Generator LLM*

Figure 27 above demonstrates the output of the Keyword Generator LLM implemented (See section 3.2.1 for implementation details). For the stock “Nvidia Corporation”, semantically relevant keywords such as ‘AI technology trends’, ‘graphics processing units’, and ‘data center demand’ were generated.

A qualitative analysis on the keywords generated has been conducted with validation from peers, along with the quantitative evaluation based on the similarity scores between generated keywords and news articles. These results demonstrated successful implementation of the embedding search model and Keyword Generator LLM, confirming their effectiveness (see Table 1 below).

Initially, a higher similarity score threshold (0.6) was tested, but it excluded many indirectly or contextually relevant articles that didn’t mention the stock name directly. Based on experimental results, the threshold was adjusted to 0.4 which achieved a better balance that

retrieved a wider range of indirectly relevant news articles while effectively excluding unrelated ones.

News Article Title	Matched Keyword	Similarity Score
Arm Holdings Reportedly Aims To Capture Half Of The Data Center CPU Market In 2025 – Retail's Divided As Stock Falls	AI technology trends	0.437
Nasdaq Sell-Off: After Losing Nearly \$800 Billion in Market Cap, Is Nvidia Stock a Buy Anymore? History Offers a Clear Indicator of What Could Happen Next.	NVIDIA Corporation	0.521
AI datacenters want to go nuclear. Too bad they needed it yesterday	data center demand	0.494
AMD Downgraded as AI Chip Struggles to Challenge Nvidia's Grip	major competitor AMD	0.616
‘Spurs take on the Grizzlies on 3-game losing streak’	-	0.016

Table 1. *Results of Embedding Search against keywords generated*

Table 1 above summarizes the results of the embedding search between five news articles and the keywords generated for Nvidia. The results demonstrate how the model captures not only directly relevant articles but also indirectly related ones. For instance, the first and third articles do not mention “Nvidia” explicitly but still receive a similarity score of 0.437 and 0.494, respectively.

Despite the score being lower than directly matched articles (e.g., the second and fourth articles with 0.521 and 0.616 respectively), the margin is not significant, suggesting the effectiveness of the model in identifying semantically or indirectly relevant content. In contrast, clearly irrelevant articles, such as the last one, showed a very low similarity score of

0.016. This supports the decision to use 0.4 as a threshold for identifying relevant news, possibly eliminating risks in delivering totally irrelevant articles.

Moreover, the matched keywords were accurate and contextually appropriate. For example, the news article titled “Arm Holdings Reportedly Aims To Capture Half Of The Data Center CPU Market...” was matched with “AI technology trends,” and the article “AI datacenters want to go nuclear...” was matched with “data center demand.”

These results demonstrate that the Keyword Generator LLM can produce meaningful, semantically aligned keywords that enable the system to detect both directly and indirectly relevant news articles through embedding search.

5. CONCLUSION & FUTURE WORKS

5.1 Conclusion & Findings

TradeInbox addressed the challenges retail investors face in consuming and analyzing the high volume of real-time financial news. Existing platforms often lack personalization or immediate and actionable analysis. To address these limitations, an LLM-based Real-time Personalized Financial News Notification System has been developed, which is designed to efficiently filter, summarize, and analyze relevant news tailored to individual user stock portfolios, aiming to reduce information asymmetry and enhance financial literacy by educating them through tailored educational analysis.

The findings confirm the successful implementation and effectiveness of the core components. The prompt-engineered Keyword Generator LLM, combined with embedding search using a 0.4 similarity threshold, identified both directly and indirectly relevant articles. The successful implementation and integration of multiple LLMs to generate concise summaries, key metrics, sentiment analysis, and multi-tiered stock impact analysis provided a detailed and educational news analysis function. As a result, initial user testing yielded positive results, with over 70% user satisfaction reported attributed by participants to the system's prompt delivery (via real-time scraping and embedding search), personalization, educational value (from multi-level analysis), and comprehension (aided by the intuitive

UI/UX). The functional frontend implementation, real-time Discord notifications, and the user-driven feedback mechanism demonstrate a viable end-to-end system.

In conclusion, TradeInbox demonstrates the practical application of Large Language Models in creating a personalized financial news system. The positive user feedback supports the finding that TradeInbox effectively addresses the core objectives by providing timely, relevant, and analyzed information. The system's architecture which combines multiple LLM-driven contents with Milvus vector database search and user feedback, provides a robust foundation for further development in personalized financial news processing.

5.2 Future Works

The current system successfully delivers core functionality, but a few improvements can enhance the system's performance and overall user experience.

Firstly, fine-tuning the Keyword Generator LLM with a larger, curated financial dataset, could enhance the quality of the keywords generated beyond the current prompt-engineering approach. This approach could improve the performance of embedding search, especially in capturing indirect and semantically related news articles to a specific stock.

Secondly, enhancing the feedback loop mechanism beyond simple keyword deletion to dynamically adjust relevance scoring could lead to better long-term personalization and better user experience.

Lastly, integrating the Trading Vendor API by connecting it to automatically pull users' portfolio holdings data instead of manually entering stock names could enhance the user experience and provide more meaningful data in the dashboard, such as portfolio holding changes.

References

- [1] D. E. Allen, M. McAleer, and A. K. Singh, "Daily market news sentiment and stock prices," *Applied Economics*, vol. 51, no. 30, pp. 3212-3235, Feb 2019. [Online]. Available: <https://typeset.io/pdf/daily-market-news-sentiment-and-stock-prices-4klgrey2bo.pdf>. [Accessed: April 16, 2025].
- [2] [Seeking Alpha - My Portfolio]. seekingalpha.com. Available: <https://seekingalpha.com/account/portfolio>. [Accessed: April 16, 2025].
- [3] "How do I filter news articles by industry, portfolio or interest on my Android phone or tablet?," help.seekingalpha.com. [Online]. Available: <https://help.seekingalpha.com/android-app/how-do-i-filter-news-articles-by-industry-portfolio-or-interest-on-my-android-phone-or-tablet>. [Accessed: April 16, 2025].
- [4] "Yahoo Finance," finance.yahoo.com. [Online]. Available: <https://finance.yahoo.com>. [Accessed: Oct. 16, 2024].
- [5] "Who Reads Finance News? Traffic and User Behaviour," fintext.io. [Online]. Available: <https://www.fintext.io/case-studies/benchmarking/who-reads-financial-news-web-traffic-and-user-behaviour/>. [Accessed: April 16, 2025].
- [6] "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: April 16, 2025).
- [7] 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.