FinSight AI: Virtual Assistant for Real-Time Financial Market Insights

Prithvi Elancherran

December 08, 2024

1. Objective

The objective of this project is to develop a cutting-edge, AI-powered virtual assistant (VA) for financial market insights. Leveraging advanced Retrieval-Augmented Generation (RAG) and Natural Language Processing (NLP) techniques, the system aims to deliver accurate, real-time responses to user queries. It will provide actionable insights on stock performance, market trends, and trading metrics, empowering users with data-driven decision-making in the financial domain.

2. Data Sources

2.1 Data Collection:

- Stock market data sourced using yFinance API
- S&P 500 tickers extracted from official lists

2.2 Data Processing:

- Data preprocessed to remove null values
- Cleaned and normalized stock price, trading volume, and return data
- Data stored in Pinecone for efficient retrieval

3. Model Selection

- **Sentence Transformer:** all-MiniLM-L6-v2 (for document embedding)
- **Large Language Model API:** Gemini API for advanced question answering and enhanced contextual understanding
- **Text Generation:** Google Flan-T5-Large for response generation

4. Implementation

4.1 RAG Setup:

- Used Pinecone as a vector database for embedding storage and retrieval
- Designed a Gradio-powered web interface for user interaction

4.2 Query Handling Pipeline:

1. User submits a query
2. System queries Pinecone for relevant documents
3. Retrieved data is embedded into an adaptive prompt
4. The prompt is passed to Google Flan-T5-Large for response generation

5. Evaluation Metrics

5.1 Metrics Used:

- **Cosine Similarity:** To measure response relevance
- **Accuracy:** Based on matching ground-truth responses

5.2 Results:

- **Initial Accuracy:** 66.67%
- **Initial Average Cosine Similarity:** 75.09%

6. Improvement Techniques

1. **Adaptive Prompting:** Enhanced prompts based on query type (e.g., performance, trends, prices).
2. **Ground-Truth Extraction:** Dynamically generated adaptive ground-truth responses improved evaluation accuracy.
3. **Response Reformatting:** Reformatted incomplete numerical answers into full descriptive sentences.
4. **Filtering Response:** Filtered response by applying grammar check.

7. Improved Results

- **Improved Accuracy:** 70.00%
- **Improved Average Cosine Similarity:** 78.22%

8. User Interface

- **Gradio:** Gradio is an open-source Python library that allows users to quickly create user-friendly web interfaces for machine learning models, APIs, and data workflows.
- **Dynamic Interaction:** Incorporated Gradio UI for real-time chat.

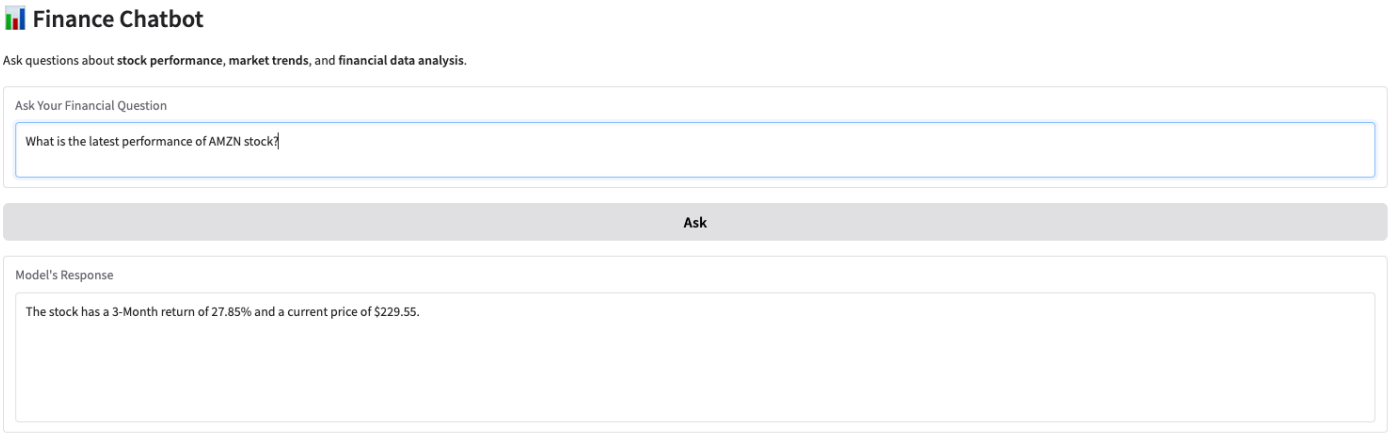

Figure 1: Chatbot UI

9. Challenges & Resolutions

9.1 Creating Embeddings for Numerical Data:

Challenge: Converting rows of numerical data into meaningful text representations for embeddings.
Resolution: Preprocessed data into descriptive strings before generating embeddings.

9.2 Choosing Lightweight Models:

Challenge: Most models either ran for too long or failed due to the large query/prompt size.
Resolution: Tested over 10 models and finalized the best-performing ones, including Flan-T5 and Gemini API.

9.3 Response Redundancy in Flan-T5:

Challenge: Flan-T5 generated multiple or redundant responses.
Resolution: Applied early stopping and refined prompt structure to limit verbosity.

9.4 Improving VA Performance:

Challenge: Several individual improvement techniques failed to yield significant gains.
Resolution: Combined three key approaches—adaptive prompting, revising model response, and filtering (grammar check)—to achieve notable performance improvements.

10. Conclusion & Future Work

This project successfully implemented a financial market VA capable of real-time insights using the RAG framework. Future enhancements include integrating additional financial APIs, fine-tuning LLMs, and expanding evaluation metrics for improved performance.