MAFS 5440 Project5: Large Language Model - FinGPT

Congxin He 21024425: Forecaster & Portfolio Manager
Yuwei Chen 21106455: Peer Comparison
Zilin Wu 21022271: Sentiment Analysis
Jiping Wang 21105827: Analyst Rating

Abstract

This report explores the transformative impact of artificial intelligence and natural language processing in the financial industry through the development of four specialized large language models based on the Llama-2-7b-chat-hf model. These models include the Forecaster, Portfolio Manager, Sentiment Analysis model, and Peer Company Comparison model, each implemented via API calls and lightweight fine-tuning.

The Forecaster model predicts market trends and asset prices using historical data and economic indicators, enhancing decision-making for investors. The Portfolio Manager model optimizes investment portfolios by analyzing asset performance and risk factors, helping investors achieve their financial goals. The Sentiment Analysis model assesses market sentiment by processing news and social media, providing insights into public and investor emotions. Lastly, the Peer Company Comparison model evaluates competing firms, aiding stakeholders in strategic benchmarking.

Collectively, these models illustrate the significant potential of Llama-2-7b-chat-hf in financial analysis. The report discusses their construction, application scenarios, and impacts, highlighting their contributions to intelligent financial decision-making and the overall advancement of the industry.

Demo Link: https://youtu.be/SMN3aq0a5cw?si=jdwHQ9Sy58sYIxCB

1 Overall Architecture

The whole project consists of 4 parts1:

The first part is data sources, where we collect historical and streaming data from the Internet.

Next, we push the data to the Data Engineering where we clean, tokenize, and prompt the data.

The data is then pushed to the Large Language Model (LLMs). Here, we can use LLMs in different ways. Not only can we use the collected data to train our own lightweight fine-tuning models, but we can also use this data and trained models or LLM apis to support our applications.

The last section will be the application section where we can use data and LLMs to make many interesting applications.

The following is the process of obtaining training data.2.

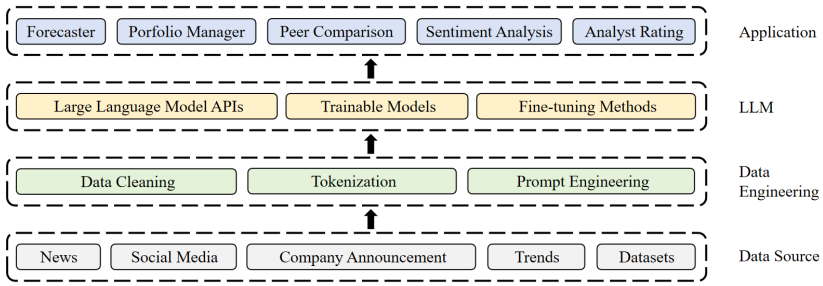

Figure 1: Architecture:Four-layer design of our framework. Data Source layer orchestrates the acquisition of extensive financial data from various online sources, including price data, news, and financial basic data. Data Curation layer focuses on the real-time processing of the text data to filter noise. LLM layer encompasses various LLMs and fine-tuning methodologies, with a priority on lightweight adaptation, to keep the model updated and pertinent. Application layer is designed to demonstrate the practical applicability of our model.

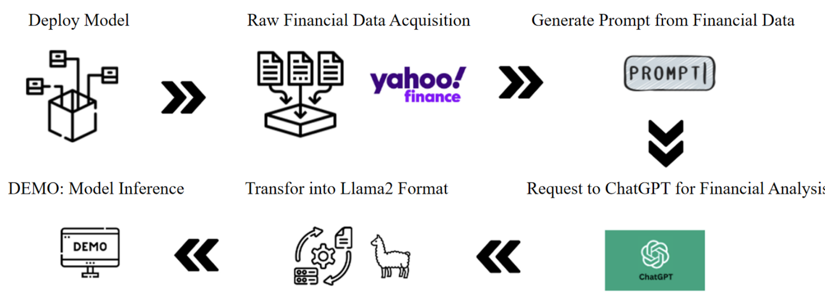

Figure 2: Process of obtaining training data

2 Model

2.1 Forecaster

2.1.1 Introduction

Forecaster can extract effective information from complex information such as price data, news, and financial basic data, and forecast the possible return of a single stock in the next week. In this model, we used weekly data to implement Forecaster's output of his Analysis of a single stock, which includes three sectors: Positive Development, Potential Concerns, Prediction and Analysis.

2.1.2 Data

Raw Data: We downloaded raw data from Yahoo for 30 stocks from November 20, 2023 to November 20, 2024, including weekly opening and closing prices, news, and optional underlying financial basic data, and built the prompt based on the raw data.

Prompt: Firstly, we briefly introduced the companies of each stock. Secondly, due to the tendency of financial data, we added time factor to the data used, that is, for each stock, we gave a parameter `n_weeks`, that is, we used the data of several weeks before the date to build prompt. As well as the optional `with_basics` option, if this parameter is true, we add the company financial data for the week in the prompt.

Answer: When using GPT to build an answer for a dataset, we need not only a prompt that we build ourselves, but also a SYSTEM_PROMPT that informs GPT of its identity, the purpose of the data generation, the format of the generated answer, etc. Our format here is as below8.

We present an example of our training set data in the appendix. In generating the data set, we used the parameters shown in the following table1.

Parameter	n_weeks	with_basics
Value	random int from 1 to 4	True

Table 1: Parameter of creating data for Forecaster

2.1.3 Training details

The loss function we use here is divided into three types: Binary Accuracy, MSE and rouge score.

Binary Accuracy: Since we forecast the return of a single stock in the coming week, we use Binary Accuracy to calculate the accuracy of the forecast. Binary Accuracy is an index used to evaluate the performance of binary classification models. It represents the percentage of the total sample that the model correctly predicted.The calculation formula is: Binary Accuracy $= \frac{\text{\# samples correctly predicted}}{\text{\# samples}}$

Mse: Mean square error is a common index used to evaluate the performance of regression models. It measures the difference between the predicted value and the actual value.The calculation formula is: MSE $= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

Among them, n is the number of samples, y_i is the actual value, $\hat{y}_i$ is the predicted value. The smaller the value of MSE, the more accurate the prediction of the model.

Rouge Score: The ROUGE score is a set of metrics used to assess the quality of automatic text summarization and machine translation. It mainly measures quality by comparing the overlap between the generated text and the reference text.

Rouge score includes the following indicators:

ROUGE-N: Calculates the overlap of n-grams between the generated text and the reference text. Common examples are ROUGE-1 (word overlap) and ROUGE-2 (double word overlap).

ROUGE-L: An indicator calculated based on the longest common subsequence (LCS), taking into account the order in which the text is generated and the reference text.

ROUGE scores, which typically include Recall, Precision, and F1 values, focus on recall and reflect how much of the generated text matches the reference text. The higher the ROUGE score, the more similar the generated text is to the reference text.

In forecaster, we used ROUGE-1, ROUGE-2, ROUGE-L.

The table shows the parameter in training the Forecaster2.

Parameter	base_model	max_length	learning_rate	weight_decay	gradient_accumulation_steps	warm_up_ratio
Value	llama2	2048	1e-4	0.01	8	0.05

Table 2: Parameter of training Forecaster

2.1.4 Performance

Here we show the change of the loss function of the model in the training process. We can see that the rouge score shows a steady increase, while the MSE and binary accuracy have very drastic changes, and the changes are not monotonous3.

Figure 3: MSE & Binary Accuracy of Forcaster

At the same time, we observe ROUGE_1 > ROUGE_L> ROUGE_2, we can draw the following conclusion4:

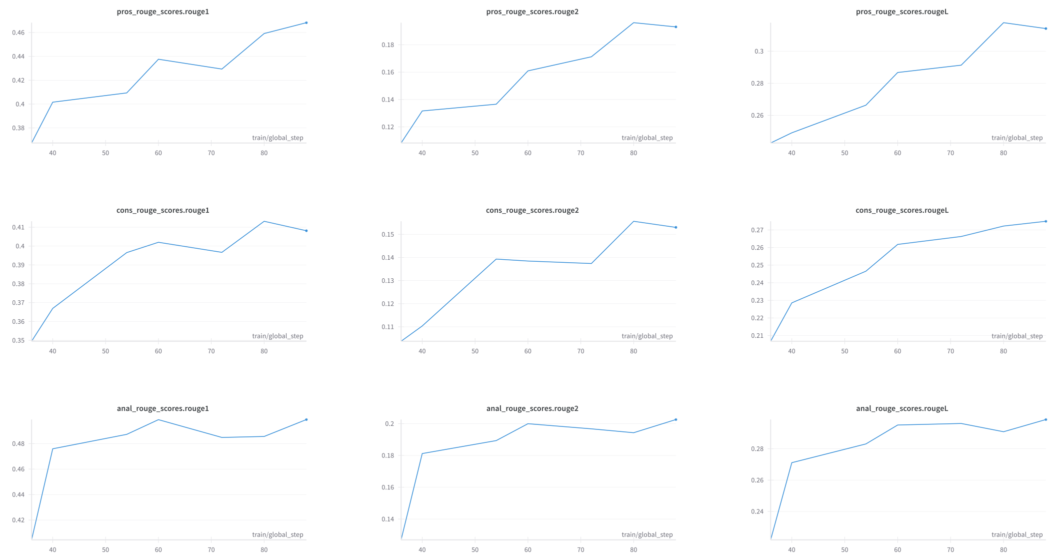

Figure 4: ROUGE SCORE of Forecaster

Good coverage: A high score of ROUGE-1 indicates good word overlap between the generated text and the reference text. This means that the model does a good job of capturing primary information and keywords, effectively covering what is important in the reference text.

Reasonable order: The score of ROUGE-L is higher than ROUGE-2, indicating that the generated text performs well in maintaining word order. Although the score of ROUGE-2 is low, a high score of ROUGE-L indicates that the generated text is structurally better able to reflect the logic and order of the reference text. Lack of phrase matching ability:

Bad abilities of generating phrases: The low ROUGE_2 score means that while the model is able to correctly generate individual words, it can have problems generating more complex phrases, resulting in less overlap with the double-word reference text.

Finally, we are trying to reach a point where five indicators trade off, so we choose the model with global-step being 81, which is the triangle point in the graph.

2.2 Portfolio Manager

2.2.1 Introduction

Portfolio Manager can compare multiple stocks from the price data, news reports and other complicated information, and output a Portfolio. In this model, we can output and explain the weights in the portfolio, and at the same time, the model can give certain risk management strategies.

2.2.2 Data

Raw Data: We use the same raw data from Forecaster.

Prompt: In this model, we first give m_stocks, and for each stock we get the profile of the stock company, n_weeks of price data, news data and financial data.

Answer: Again, here we show SYSTEM_PROMPT. It tells GPT to output content in a format that includes weights, explanations, and risk management strategies9.

We present an example of our training set data in the appendix. In generating the data set, we used the parameters shown in the following table3.

Parameter	n_weeks	m_stocks	with_basics
Value	random int from 1 to 4	random int from 2 to 5	True

Table 3: Parameter of creating data for Portfolio Manager

2.2.3 Training Details

In this model, we did not involve the prediction of stock returns, so we did not set binary accuracy and mse loss function. So we only set ROUGE_1, ROUGE_2 and ROUGE_L scores for portfolio weights, explanation and risk management respectively. For details, see 3.1.34.

Parameter	base_model	max_length	learning_rate	weight_decay	gradient_accumulation_steps	warm_up_ratio
Value	llama2	3072	1e-4	0.01	8	0.05

Table 4: Parameter of training Portfolio Manager

2.2.4 Performance

All three ROUGE scores are gradually increasing, although the three scores are not identical. Similarly, we observed the same results as Forecaster in the comparison of the three forecaster models, which shows that our model has good content coverage and good logical order5.

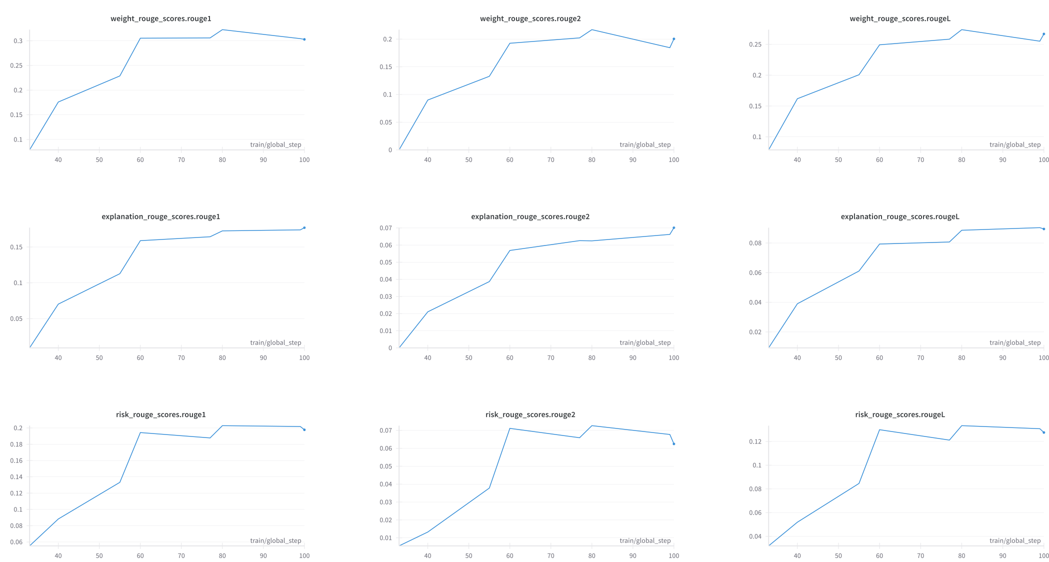

Figure 5: ROUGE Score of Portfolio Manager

At the same time, the score in our model is lower than the score in forecaster, which we guess is due to 1. More stock data, more responsible information; 2. Due to computer performance limitations, we can use fewer data sets. If we can use more and larger data sets, and show a gradual upward trend according to each score, our model may achieve better results.

2.3 Peer Company Comparison

2.3.1 Introduction

Peer Company Comparison can list three peer companies of your target company, analyze each company, and give out a comparison. In this model, we can give out the best company, explain the reasons, and output the analysis of each company. This is really helpful for someone who wants to explore more about the target company and get an insight about the other companies in the same industry.

2.3.2 Data

Raw Data: Dow 30 stocks' information and news from Yahoo Finance.

Prompt: In this model, we give out the information of the target company and its three peer companies. The information includes a Company Introduction, Relative News, and Basic Financials for the recent week.

Answer: Here we show SYSTEM_PROMPT. It tells GPT to output content in a format that includes comparison results (best company), reasons, and peer companies analysis10.

2.3.3 Training Details

For this model, we set ROUGE_1, ROUGE_2, and ROUGE_L scores for comparison results and reasons. For details, see 3.1.35.

Parameter	base_model	max_length	learning_rate	weight_decay	gradient_accumulation_steps	warm_up_ratio
Value	llama2	3600	1e-4	0.01	8	0.05

Table 5: Parameter of training Peer Company Comparison

2.3.4 Performance

We can see that all the rouge scores are rising, except for the last checkpoint. So we choose the second to last checkpoint model to utilize our app demo. Compared with the basic forecaster model and the Portfolio Manager model, the comparison result has an extremely high rouge score. We think this is because the comparison result is just a company symbol, which is simpler for the model to generate than long texts. For the Reason part, three rouge scores are all a little lower than those in the basic forecast model. This indicates some improvement in our training process, for instance, increasing the training set and trying more model parameters, which we didn't do the best due to the GPU memory6.

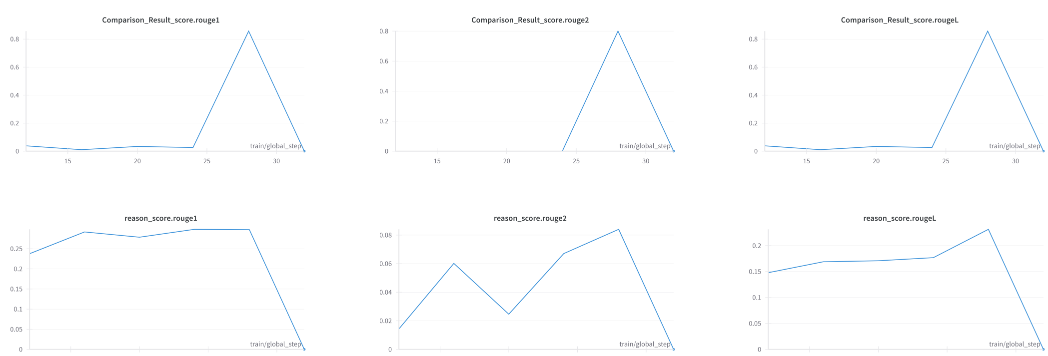

Figure 6: ROUGE Score of Peer Company Comparison

2.4 Sentiment Analysis

2.4.1 Introduction

Sentiment analysis can extract effective emotional tendencies from complex news information, helping us understand the market's attitude toward a specific stock and predict its future trends. By analyzing media reports on the stock, we can track changes in market sentiment in real-time, ensuring that investors always have access to the latest market dynamics. In our model, we focus on performing sentiment analysis on the most recent 20 news articles within the given time period related to a single stock, categorizing each article as "positive," "neutral," or "negative." This analysis not only helps identify the current market perception of the stock but also uncovers potential investment opportunities or risks.

2.4.2 Data

Raw Data:

We utilized four datasets for our sentiment analysis task, each providing unique insights into financial news and social media sentiment:

- **Financial Phrasebank (FPB)**: This dataset serves as a benchmark for financial news sentiment analysis, with labels categorized as "positive," "negative," and "neutral."
- **FiQA SA**: Comprising 17k sentences from microblog headlines and financial news, this dataset was relabeled to "positive," "negative," and "neutral" based on BloombergGPT's paper.
- **Twitter Financial News Sentiment (TFNS)**: This dataset contains 11,932 annotated finance-related tweets, classified into "Bearish" (negative), "Bullish" (positive), and "Neutral" categories. It provides valuable insights into real-time market sentiment from social media.
- **News With GPT Instruction (NWGI)**: This dataset includes 16.2k training samples and 4.05k test samples, with labels ranging from "strong negative" to "strong positive" across seven categories. Additionally, it provides detailed reasons for each classification, which is beneficial for instruction fine-tuning.

Each dataset has been divided into a training set and a test set, ensuring that the task is approached with a full-shot learning strategy. The training samples have been augmented by including duplicated entries in the case of three out of four datasets, thereby increasing the total number of training samples for each dataset.

The following table summarizes the distribution of the training and test samples across the four datasets, along with the percentage contribution of each dataset's training samples to the total training set6:

Table 6: Distribution of Training and Test Samples Across Datasets

Dataset	Training samples	Duplication	Total Training samples	Part%	Test samples
FPB	3634	6	21804	28.4	1212
FiQA-SA	938	21	19698	25.7	275
TFNS	9543	2	19086	24.9	2388
NWGI	16184	1	16184	21.0	4047
Total	-	-	76772	100	-

Prompt: In this case, we input the headline and summary of each recent news article for each stock and the model will then perform sentiment analysis.

Answer: For each stock, we provide the model with the most recent 20 news articles. For every news article, the model analyzes the sentiment and selects one of three categories: positive, neutral, or negative. We then tally the counts of each sentiment category and present them in a bar chart for a clear and intuitive visualization. This graphical representation helps stakeholders quickly understand the overall sentiment trends associated with the stock11.

2.4.3 Training Details

The loss function used in this model is the **Cross-Entropy Loss**, which is implemented internally by the AutoModelForCausalLM model by default. It measures the difference between the predicted probability distribution and the true distribution (one-hot encoded labels). The calculation formula for Cross-Entropy Loss is:

$$\text{Cross-Entropy Loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{V} y_{ij}\log(p_{ij})$$

where: - N is the batch size. - V is the vocabulary size. - y_{ij} is the true label (one-hot encoded). - p_{ij} is the predicted probability distribution by the model.

LoRA (Low-Rank Adaptation) is an efficient fine-tuning method that adapts large language models by introducing low-rank matrices into specific modules. Table 7 summarizes the hyperparameter settings used for configuring the LoRA model during sentiment analysis training.

Table 7: Hyperparameter Settings for LoRA Configuration

Parameter	task_type	inference_mode	r	lora_alpha	lora_dropout	target_modules
Value	Causal_LM	False	8	32	0.1	`{query_key_value}`

The following table summarizes the hyperparameters used for training the sentiment analysis model8:

Table 8: Hyperparameter Settings for Sentiment Analysis Training

Parameter	base_model	num_train_epochs	batch_size	gradient_accumulation_steps	learning_rate	weight_decay	warmup_steps
Value	llama2	2	4	8	1×10^{-4}	0.01	1000

2.5 Analyst Rating

2.5.1 Introduction

Analyst Rating can return the latest analyst recommendation trends for a company based on the input stock code. In this model, we can output recommendations, which allows investors to better analyze the reactions of other market participants to the company and make more informed investment decisions.

2.5.2 Parameters

We use stock codes and corresponding information obtained from Finnhub. In generating the Recommendations and Histograms, we used the parameters shown in the following table9.

Parameter	ticker symbol	n_months
Value	str	int from 1 to 4

Table 9: Parameter of Recommendations System

2.5.3 Performance

When using GPT for other inquiries, it will provide evaluations of the company's stock based on the input stock code. These evaluations include five categories: strong buy, buy, hold, sell, and strong sell, and are visually presented through a histogram. For example, the recommendations regarding NVIDIA are shown as follows7.

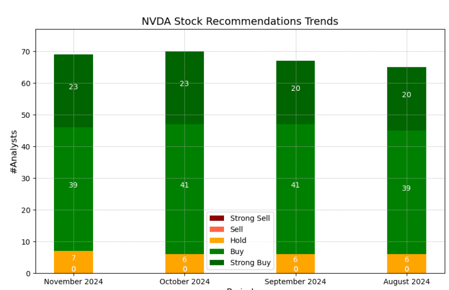

Figure 7: NVDA_rec_trends

Appendix

A System Prompt

```
You are a seasoned stock market analyst.
 Your task is to list the positive developments and potential concerns for companies based on relevant news and basic financials from the past weeks,
 then provide an analysis and prediction for the companies' stock price movement for the upcoming week. Your answer format should be as follows:

[Positive Developments]:
1. ...

[Potential Concerns]:
1. ...

[Prediction & Analysis]
Prediction: ...
Analysis: ...
```

Figure 8: Forecaster System Prompt Format

```
You are a seasoned stock market analyst.
 Your task is to construct a portfolio including weights of these stocks based  on relevant news and basic financials from the past weeks,
 then provide an explanation of why did you choose these weights. Your answer format should be as follows:

[Portfolio Weights]:
1. ... (stock name) – ...% (weight)

[Explanation]
1. ...

[Risk Management]
1. ...
```

Figure 9: Portfolio Manager System Prompt Format

```
You are a professional stock financial analyst. Your task is to analyze a target company and its industry peers based on relevant news
and basic financials for the last week, then provide company analysis, compare companies and determine the best-performing company.

Your answer format should be as follows:

[Comparison Result (Best Company)]:...

[Reasons]:1. ...

[Peer Companies Analysis]:1. ...
```

Figure 10: Peer Comparison System Prompt Format

```
Instruction: What is the sentiment of this news? Please choose an answer from {negative/neutral/positive}.
Input: L&T has also made a commitment to redeem the remaining shares by the end of 2011 .
Answer:
```

Figure 11: Sentiment Analysis Prompt Format

B Demo

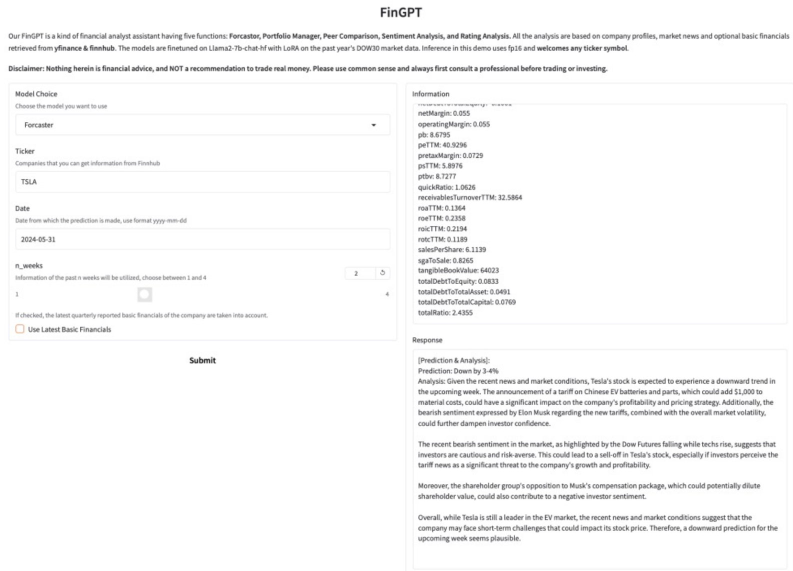

Figure 12: Demo of Forecaster

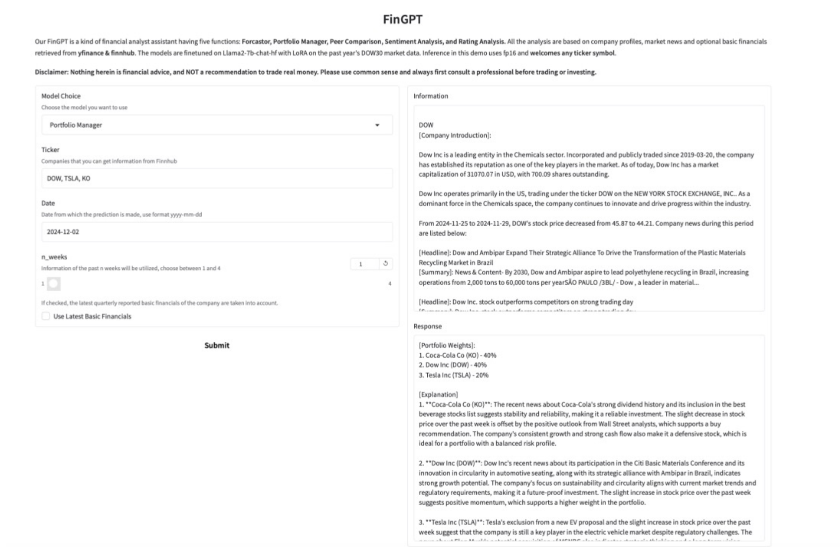

Figure 13: Demo of Portfolio Manager

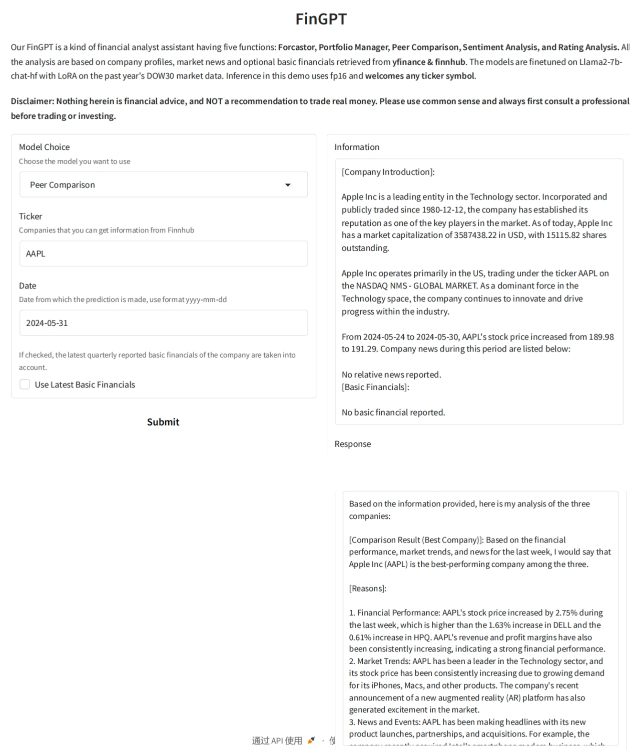

Figure 14: Demo of Peer Company Comparison

Figure 15: Demo of Sentiment Analysis

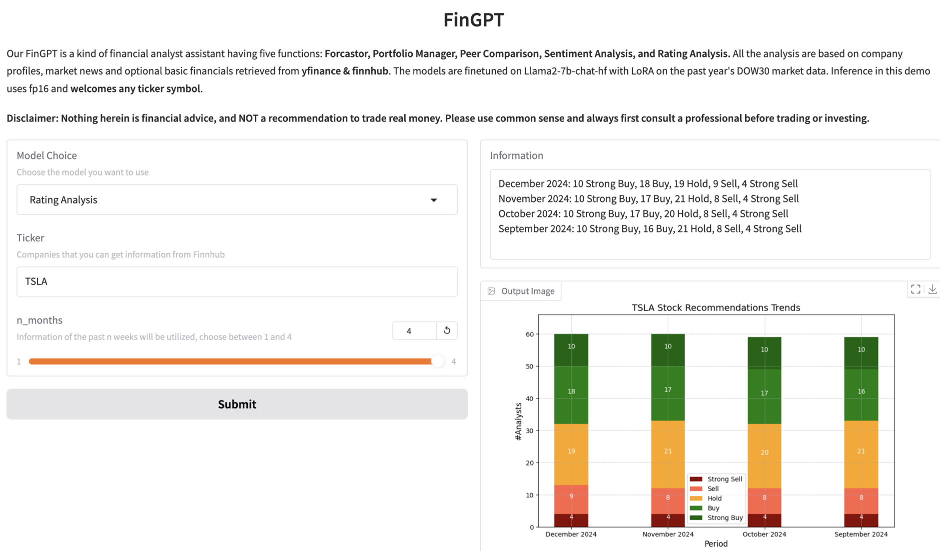

Figure 16: Demo of Analyst Rating