Sentiments of Financial Markets: Leveraging Large Language Models for Market Timing Through Sentiment Analysis

Aleksei Makogon, Jose Rafael Campos Torrealba , Sara Jabrane

WorldQuant University

E-mails:
makogon422833@gmail.com
joser.compost@gmail.com
sara.jabrane@gmail.com

Abstract

Our project aims to enhance stock price forecasting by introducing a sentiment indicator derived from the textual components of financial reports. This report outlines the goals and objectives of our project, reviews the application of sentiment analysis in finance, and examines competing models. Additionally, we discuss the development of a trading strategy based on large language models (LLMs). In this project we have fine-tuned Mistral on annual and quarterly financial reports (form 10-K and 10-Q, obtained from SEC EDGAR), using the returns relative to benchmark over the following quarter as the target variable. We demonstrate that a long-short strategy based on the analysis of financial reports using LLMs can outperform traditional momentum strategy.

Keywords: *Large Language Models (LLMs), Sentiment Analysis, Market timing, Machine Learning, Fine Tuning.*

1. Introduction

In today's rapidly evolving financial world, it's crucial for investors to be able to assess and predict market movements accurately if they want to make the most of their portfolios. Traditional analysis methods often rely heavily on numerical data from financial reports, overlooking the wealth of insights that can be gleaned from textual content (section MD&A and risk factors in reports 10-K and 10-Q). This project aims to bridge this gap by using advanced natural language processing (NLP) technologies to analyze the sentiment conveyed in the textual parts of financial reports obtained from EDGAR. The current reality is that many investors and financial analysts may not fully exploit the potential of textual analysis due to a lack of effective models or methodologies. This project seeks to address this by fine-tuning a large language model to extract and analyze textual data from financial reports and correlate it with stock price movements. The gap exists in the underutilization of textual analysis for market timing and investment strategy optimization.

Goals and Objectives

Our project aims to develop and integrate a sentiment-based indicator for market timing and investment strategy, guided by two primary goals.

The primary goal of our project is to develop a market timing indicator based on sentiment analysis. This involves a series of structured objectives, starting with gathering and preprocessing of financial reports from EDGAR, where the focus is on distinguishing between textual and numerical data. Then, fine-tuning of a large language model, aimed specifically at analyzing the sentiment of the textual data. This process is crucial to ensure that the nuances of financial language are captured and interpreted accurately. Following this, a methodology is going to be established to convert the outcomes of the sentiment analysis into a measurable indicator that can reliably signal upcoming

market movements. The final objective involves a thorough validation of the sentiment-based indicator's effectiveness by comparing it with historical market data. This step is important for refining and enhancing the indicator's predictive accuracy, ensuring it serves as a robust tool for market timing.

The second goal of our project is to integrate the sentiment-based indicator within a broader market timing strategy. This integration aims to enhance decision-making processes by combining the insights from sentiment analysis with established numerical analysis techniques, thereby creating a more robust model for market assessment. To achieve this, we are focusing on constructing a detailed framework that merges the sentiment indicator with traditional financial metrics. This composite strategy will then undergo rigorous testing in market environments, allowing us to evaluate its effectiveness across a spectrum of market conditions. Through this trial phase, we aim to pinpoint the strengths and weaknesses of our approach, leading to the development of a comprehensive market timing strategy. This approach will cover critical aspects such as risk management and portfolio adjustment, ensuring that our sentiment-based market timing strategy is both practical and reliable for investors looking to navigate the complexities of financial markets with more precision.

2. Literature Review

Financial Sentiment Analysis

According to Cambridge dictionary (SENTIMENT ANALYSIS | English Meaning - Cambridge Dictionary, n.d.), sentiment analysis is defined as "the process of using computer software to find out people's opinions or feelings about something from things that have been written". The recent development of machine learning algorithms allows to transform opinions and attitudes into information that can be used to make decisions.

There are multiple datasets and models for sentiment analysis (see for example (Hamborg & Donnay, 2021; Ho et al., 2019; Pang et al., 2002; Pang & Lee, 2005)), these models achieved good performance in their domains, however, each domain is unique, and models designed for one domain tend to demonstrate lower efficiency in other (Han et al. 2018).

Uniqueness of Financial Texts

Finance related texts are characterized by unique vocabulary. In the work (Loughran et al., 2011) authors analysed a large sample of 10-K fillings and discovered that in most cases words are misclassified. Authors provide several examples of financial jargon, that has specific meaning, for example, "bull" in general is a neutral word, in financial world has positive connotation, while "liability", which is usually negative, is neutral in finance. In the study (Mishev et al., 2020) authors demonstrated that words annotated for non-financial domains tend to misclassify common words in finance-related texts.

Applications of Sentiment Analysis in Finance

Sentiment analysis became an important direction in finance. Studies demonstrated the application of sentiment analysis to stock prices prediction (Joshi et al., n.d.; Li et al., 2014) , global market trends (Curme et al., 2015), exchange rate movements (Crone & Koeppel, 2014) .

The introduction of deep learning models, as opposed to lexicon-based approaches, marked significant step-forward in sentiment analysis (Mishev et al., 2020) , however, these models initially suffered from low availability of good, labelled datasets and, as lexicons, struggled with domain-specific jargon. Introduction of models like GloVe (Pennington et al., n.d.) and BERT (Devlin et al., 2018) led to paradigm in sentiment analysis. These models can extract contextual information and complex features.

Transformers, especially fine-tuned for financial applications, have demonstrated superior performance in extracting sentiment from financial news and reports. This can be attributed to their ability to process large volumes of data and their understanding of the contextual meaning of words and phrases in finance-specific scenarios. As the noted in (Naseem et al., 2020), the efficiency of contextual embeddings provided by transformers shows a significant improvement over previous methods, making them suitable for real-time analysis in financial markets where the accurate and prompt extraction of sentiment from news and financial reports can inform investment decisions.

Analysis of Sentiment of Financial Reports

Financial reports (namely forms 10-K and 10-Q) contain management discussion and analysis (MD&A), Risk Factors and some other text-based sections. In the work (E. Henry, 2006) authors demonstrated that sentiment in these sections is related to the performance of companies hence, stock prices. The goal of financial reports is to accurately represent a company's economic position and activities; however, some parts of reports are based on judgements, estimates and discussions of uncertainties. Price (Price et al., 2012) analysed earning reports and found that a positive textual tone is related to anormal returns. In the work (Zhong & Ren, 2022), correlation between tone of textual sections and future performance was established.

Review of Competing Models

Considering our goal to analyse sentiment of the financial reports using LLMs to make forecasts of stock price movements to assist in decision making, we have gathered data on existing finance specific LLMs, capable of performing similar tasks.

Table was partially taken from (Lee et al., 2024) and extended by us.

Description								Open Source		
Type	**Model**	**Backbone**	**Params.B**	**Tech**	**Size**	**Task**	**Dataset**	**Model**	**PT**	**IFT**
FinPLM (Disc.)	FinBERT-19 (Araci, 2019)	BERT	0.1	Post-PT, FT	(G) 3.3B words (F) 29M words	[SA]	FPB, FiQA-SA	Y	N	
	FinBERT-20 (Y. Yang et al., 2020)	BERT	0.1	PT, FT	(F) 4.9B tokens	[SA]	FPB, FiQA-SA, AnalystTone	Y	Y	N
	FinBERT-21 (Liu et al., 2020)	BERT	0.1	PT, FT	(G) 3.3B words (F) 12B words	[SA], [QA], [SBD]	FPB, FiQA-SA, FiQA-QA, FinSBD19	N	N	N
	FLANG (Shah et al., 2022)	ELECTRA	0.1	PT, FT	(G) 3.3B words (F) 696k docs	[SA], [TC], [NER], [QA], [SBD]	FPB, FiQA-SA, Headline FIN, FiQA-QA, FinSBD21	Y	Y	N
FinLLM (Gen.)	Bloomberg GPT (Wu et al., 2023)	BLOOM	50	PT, PE	(G) 345B tokens (F) 363B tokens	[SA], [TC], [NER], [QA]	FPB, FiQA-SA, Headline FIN, ConvFinQA	N	N	N
	FinMA (Xie et al., 2023)	LLaMA	7, 30	IFT, PE	(G) 1T tokens	[SA], [TC], [NER], [QA], [SMP]	FPB, FiQA-SA, Headline FIN, FinQA, ConvFinQA, StockNet, CIKM18, BigData22	Y	Y	Y
	InvestLM (Y. Yang et al., 2023)	LLaMA	65	IFT, PE, PEFT	(G) 1.4T tokens	[SA], [TC], [QA], [Summ]	FPB, FiQA-SA, FOMC, FinQA, ECTSum	Y	N	N
	FinGPT (H. Yang et al., 2023)	6 open-source LLMs	7	IFT, PE, PEFT	(G) 2T tokens (e.g., LLaMA2)	[SA], [TC], [NER], [RE]	FPB, FiQA-SA, Headline FIN, FinRED	Y	Y	Y
	FinTral (Bhatia et al., 2024)	Mistral-7b	Est. >7	PT, IFT, RLAIF	(G) Large-scale textual and visual datasets	[SA], [TC], [NER], [QA], [SMP], [CS], [FD], [HI]	Extensive benchmark including hallucinations	Y*	Y*	Y*

	FinLlama (Konstantinidis et al., 2024)	Llama 2 7B	27	FT, LoRA	(F) 34.18k samples	[SA]	Custom financial news datasets	N	N	N

Table 1: A Summary of FinPLMs and FinLLMs. The abbreviations are following: Params. Number of model parameters in Billions; Disc. = Discriminative, Gen. = Generative; Post-PT = Post-Pre-training, PT = Pre-training, FT = Fine-Tuning, PE = Prompt Engineering, IFT = Instruction Fine-Tuning, PEFT = Parameter Efficient Fine-Tuning; (G) = General domain, (F) = Financial domain; (in Evaluation)
Tasks: [SA] Sentiment Analysis, [TC] Text Classification, [SBD] Structure Boundary Detection, [NER] Named Entity Recognition, [QA] Question Answering, [SMP] Stock Movement Prediction, [Summ] Text Summarization, [RE] Relation Extraction; (in Venue) (S) = Special Track, (D) = Datasets and Benchmarks Track, (W) = Workshop. In open source, it is marked as Y if it is publicly accessible as of Dec 2023.

Technical Insights from Analysis of Competing Models

- FinGPT uses lightweight adaptation of open-source LLMs, enabling low-cost ($100-300) customization compared to expensive models like BloombergGPT. It outperforms BloombergGPT on key financial NLP benchmarks.
- Instruct-FinGPT, fine-tuned on instruction data, shows strong financial sentiment analysis performance, surpassing FinBERT and GPT models. Instruction tuning proves effective for domain adaptation.
- PIXIU introduces the first financial instructions dataset FIT and evaluation benchmark FLARE. FinMA, the LLM trained on FIT, excels at NLP tasks but struggles with complex reasoning compared to GPT-4.
- Open questions remain around the exact model sizes, training data, and speeds for some models like PIXIU and FinLLaMA. Sharing these details could enable better comparisons.
- Opportunities exist to further improve financial LLMs through techniques like LoRA (demonstrated by xFinance), reinforcement learning on stock prices, and expanding training data with real-time information.
- LoRA (Low-Rank Adaptation) and QLoRA (Quantized Low-Rank Adaptation) are efficient fine-tuning techniques for adapting large language models (LLMs) to specific domains or tasks, such as finance in the case of FinGPT. These methods enable lightweight and cost-effective adaptation compared to training models from scratch or fine-tuning all model parameters.

2.1 Competitors Analysis

Financial Large Language Models (FinLLMs). The range of potential uses of this financial model is broad and exquisite, including robo-advisory, interpreting market sentiments, evaluating credit worthiness and credit rating, identifying irregularities for fraud mitigation, optimizing investment portfolios, strategic risk management, and algorithmic trading strategies, among various other functions (P. Henry & Krishna, n.d.) .
In our analysis we have decided to focus on 3 key players in the market that have emerged in shaping the direction and capabilities of AI-driven financial analysis and investment decision-making. In our competitor's analysis we are referring to BloombergGPT, FinMA, and FinTRAL that stand at the forefront, each offering unique strengths that cater to the diverse needs of the financial industry.

BloombergGPT utilizes Bloomberg's extensive datasets and analytical resources to provide comprehensive insights into market trends and financial news with unparalleled accuracy. According to Bloomberg, BloombergGPT demonstrates superior efficacy over analogous open-model constructs in the realm of financial natural language processing, achieving noteworthy margins of improvement. This enhancement does not come at the expense of its proficiency on established benchmarks for general large language models. (Introducing BloombergGPT, Bloomberg's 50-Billion Parameter Large Language Model, Purpose-Built from Scratch for Finance | Press | Bloomberg LP, n.d.)
FinMA is an example of a model designed to excel in the analysis and interpretation of intricate financial data, including improved accuracy in financial projections, deeper understanding of market trends, and efficient processing of extensive financial reports. These capabilities prove to be

extremely valuable across various financial services, including the development of investment strategies and risk management. However, FinMA could face several challenges such as data bias. As the model's outcomes might unintentionally mirror historical prejudices embedded within the training data. Another challenge faced would be overfitting represents, with the model potentially performing well on training data but inadequately on new unseen data type. Thus, this would be affecting its applicability in practical settings. Additionally, the fast-paced nature of financial markets necessitates continual updates to the model to ensure it remains relevant and accurate, posing an ongoing challenge. (Lee et al., n.d.; Xie et al., 2023)

FinTral, as described in the paper "FinTral: A Family of GPT-4 Level Multimodal Financial Large Language Models," represents a significant leap forward in the application of artificial intelligence within the financial sector. This family of models is engineered to leverage the advanced capabilities of Mistral, enabling it to perform real-time analysis and support decision-making across a broad spectrum of financial contexts. Its multimodal nature allows FinTral to process and interpret not just textual information but also numerical data and visual inputs, making it exceptionally adept at understanding complex financial reports, market trends, and economic indicators. By integrating these diverse data types, FinTral can provide insights and forecasts with a level of depth and accuracy previously unattainable. This capability positions FinTral to be an invaluable tool for financial analysts, investors, and institutions, facilitating more informed and timely decisions in a fast-paced financial environment. (Bhatia et al., 2024)

SWOT Analysis

Based on our observations and analysis, we have come up with a combined SWOT analysis into one cohesive examination, which involves looking at the overarching factors that impact various models in the financial AI landscape:

Strengths	**Weaknesses**
Advanced Predictive Capabilities: these models exhibit strong predictive capabilities in stock movements, leveraging advanced AI to analyze market trends. **Data-Driven Insights**: The ability to process and analyze large volumes of financial data, generating insights that can inform financial decisions and strategies. **Specialized Financial Focus**: Pushing the boundaries of natural language processing (NLP) in the financial industry, offering predictive analytics, sentiment analysis, and other advanced features.	**Complexity of Financial Terminology**: Despite their advanced training, the nuanced and evolving nature of financial terminology can pose a challenge, potentially leading to misinterpretations. **Input Length Limitations**: Models based on architectures like BERT and derivatives have limitations on the length of input they can process, making them less suitable for analyzing longer financial documents. **Integration and Adaptation and Dependency on Public Sentiment Data**: The heavy reliance on social media and public sentiment data might introduce volatility and unpredictability in their predictions. **Unknown timeframes and non-numerical outputs:** Reviewed models predict price movement (up or down) based on input text, but they are not capable of estimation of returns over specified periods of time.
Opportunities	**Threats**

Expansion into New Markets: The growing demand for AI in financial services provides opportunities for these models to be adopted across various financial applications and industries worldwide. **Expansion into Long-Form Analysis**: Developing capabilities to process and analyze longer texts and sophisticated data, could open up new avenues for comprehensive financial analysis, such as detailed report assessments. **Partnerships and Collaborations**: By partnering with financial institutions, tech companies, and academia, these models can continue to evolve and stay at the forefront of financial technology. **Product Development and Financial Engineering Intergration**: Merging sophisticated financial engineering principals with AI in product creation, like advisory services and compliance tools, promises to refine these technologies' accuracy and relevance in complex finance areas. This approach combines AI innovation with solid financial strategies, leading to more dependable and efficient industry solutions.	**Competitive Market:** The field of AI in finance is highly competitive, with continuous innovations that could render existing models obsolete if they do not keep pace. **Data Privacy and Security Concerns**: The use of public sentiment data and social media inputs raises concerns about data privacy and the security of information, potentially limiting the scope of data sources. **Regulatory Risks and Challenges:** Financial AI models must navigate a complex web of global financial regulations that could impact their functionality and deployment. **Technological Shifts**: The rapid pace of innovation in AI and financial technologies poses a threat, as newer models could quickly surpass the capabilities of these three.

Overall, BloombergGPT, FinMA, and FinTral are key players within this segment, and each has their advantages and face their own challenges. Yet, they collectively exemplify the forefront of AI in the financial industry. From our conducted analysis, we have observed that the field which we are looking into is full of potential benefits, but it also has complex challenges that require constant new ideas and careful examination.

Also, it is worth noting that most of the models do not directly predict prices and need some human intervention.

2.2 Conclusions on Literature Review

In conclusion of the Competitor Analysis section, it is pertinent to highlight that our direct competitors, namely BloombergGPT, FinMA, and FinTral, are distinguished by their capability to predict stock movements, albeit within an unknown timeframe. This prediction, universally oriented towards directionality (upward or downward), often leverages Tweets or short texts as their primary data inputs. This preference for brief textual inputs is attributed to constraints observed in models like BERT, which are limited by a maximum input sequence of 512 tokens, rendering them less effective for processing longer documents such as SEC filings. Consequently, models relying on BERT's architecture face challenges in accommodating lengthy inputs, a limitation that underscores our decision against its use.

Furthermore, while the methodology employed by these competitors demonstrates robustness in applying natural language processing (NLP) techniques, there appears to be a lack of rigorous application of financial engineering principles. This observation suggests an area where further development and refinement could enhance the predictive accuracy and utility of financial AI models, thereby presenting an opportunity for innovation in integrating NLP with comprehensive financial analysis and direct methodology to predict not only stock movements, but also anticipate their prices.

Overall, although the described competitors can be effectively utilized to facilitate in investment decision-making, there are no LLM-based approaches currently available that can directly create sentiment-based indicators from financial reports and combine them with other indicators.

3. Methodology

To achieve our goals, we are following the following steps (Figure 1):

1. Creation of dataset based on reports, obtained from EDGAR platform and historical stock data, obtained from Yahoo Finance. The periodicity of the reports is every 3 months, and the historical financial data is also presented in 3-month intervals. For regression, this includes cumulative returns starting from the date each report is published. For classification, we are creating multi-class targets by comparing the returns for a given 3-month period with the performance of a benchmark over the same period. The dataset covers all S&P500 companies, the length of the dataset spans over a period of 5 years (in some cases less, when reports for 5 years are not available)

2. Fine-tuning the model to utilize pre-trained LLM for regression/classification problem. The approaches, that we are currently using, include Low Rank Adaptation, Prefix-Tuning and Soft-Prompting (which employs modification of input token embeddings)

3. Model validation depends on the type of model: for regression, we are using Mean Squared Error (MSE), and for classification, we are calculating accuracy. The dataset is split into training and validation sets, with an additional subset reserved for backtesting the strategy.

4. The strategy that we are developing is a basic long-short strategy, the decision whether to take long or short position depends on model prediction. The exact decision-making rules are yet to be specified. The strategy will undergo a back test based on historical information.

Figure 1. Project Framework

3.1 Data Retrieval

Fetching Data from EDGAR

The links to colab and github repo are available in the supplementary information (SI).

To fetch reports from EDGAR database we are using “requests” library. Our code returns links to 10-K and 10-Q reports, for example (for code see SI):
ttps://www.sec.gov/ixviewer/ix.html?doc=/Archives/edgar/data/101778/000010177824000023/mro-20231231.html

The input of the function is ticker (in the example link ‘MRO’) and the number of reports we want to retrieve and type of report.

The output is the list of links to the reports. We treat 10-K (annual reports) and 10-Q (quarterly reports) separately, because they have slightly different fields and information is located in different sections

Processing the reports

To process the retrieved 10-K and 10-Q reports from EDGAR, we use a combination of Python libraries (Selenium, BeautifulSoup), and custom functions (for details see SI-2). The steps are:

1. Use a loop to iterate over a list of tickers (e.g. S&P 500 constituents) and a range of years to retrieve multiple reports per company over time. This allows building a comprehensive dataset for analysis.
2. For each ticker and year combination, use Selenium to load the respective 10-K or 10-Q report URL in a Chrome browser session. This enables handling any required login or navigation steps. Wait for the report page to fully load by checking for the presence of a unique element.
3. Save the loaded report HTML to a local file using Selenium's page_source attribute. This allows offline processing of the report.

Read the saved HTML file and parse it using BeautifulSoup. This facilitates easy traversal and manipulation of the HTML structure.
Separate the parsed HTML into two main components:

- Textual content: Remove unwanted tags and elements that are not relevant for text analysis, such as <script>, <style>, <header>, <footer> etc. Strip out all HTML attributes from the remaining tags. Extract the cleaned text content using BeautifulSoup's get_text() method.
- Tabular content: Identify and extract <table> elements from the parsed HTML. These tables often contain valuable structured financial data that can serve as a baseline for feature engineering and numerical analysis.

4. Save the extracted textual content to a local .txt file and the tabular content to a structured format like .csv for further analysis and processing.
5. Return the file paths of the saved text and tabular data for use in subsequent steps of the pipeline, such as sentiment analysis, feature engineering, and integration with the language model.

This process is encapsulated in functions that allow automating the retrieval, separation, cleaning and extraction of textual and tabular content from 10-K and 10-Q reports at scale. By running it in a loop over multiple tickers and years, we can build a rich dataset that combines both unstructured text and structured financial data. This hybrid dataset enables a more comprehensive analysis, where the sentiment insights from the textual content can be augmented with relevant financial metrics and indicators engineered from the tabular data. The integration of these two data types will provide a strong foundation for developing a robust sentiment-based market timing indicator.

3.2 Transformation of textual parts of the reports

The textual parts of reports are too long to be used in LLMs directly. To handle this issue and keep only the most relevant information, we have utilized the following approach:

1. Split reports into chunks of approximately 800 symbols (approximately because each chunk goes to the end of sentence), chunks have overlap of 100 symbols
2. Define 10 search queries (See supplementary information (SI-5))
3. Encode chunks and queries into vector representations using sentence transformer
4. Find cosine similarity of chunks and queries, return chunks, that are the most similar to search queries
5. Combine all relevant chunks in each report

This approach is new (to our knowledge) and allows us to efficiently extract useful information from the reports. However, the exact phrasing of chunks and the size play a crucial role and need to be explored further.

3.3 Creation of targets

We utilized a ternary methodology to forecast three distinct scenarios, aiming to set precise investment targets based on the performance of cumulative returns over a specified period.

Criteria for Setting Ternary Targets:

1. Long Position (Target = 1): If the cumulative returns over the next 84 days exceed both the S&P 500 benchmark and zero, we establish a long position. This indicates that the investment is expected to perform better than the benchmark and is profitable.
2. Short Position (Target = -1): If the cumulative returns are less than both the S&P 500 benchmark and zero during the same period, we take a short position. This suggests that the investment is expected to underperform and incur losses.
3. Hold (Target = 0): If the cumulative returns neither clearly outperform nor underperform relative to the S&P 500 and zero, no action is taken, and we maintain a hold position.

This approach ensures that our investment decisions are strategically aligned with the ternary targets, which are based on a systematic analysis of expected performance over an 84-day interval between consecutive reports.

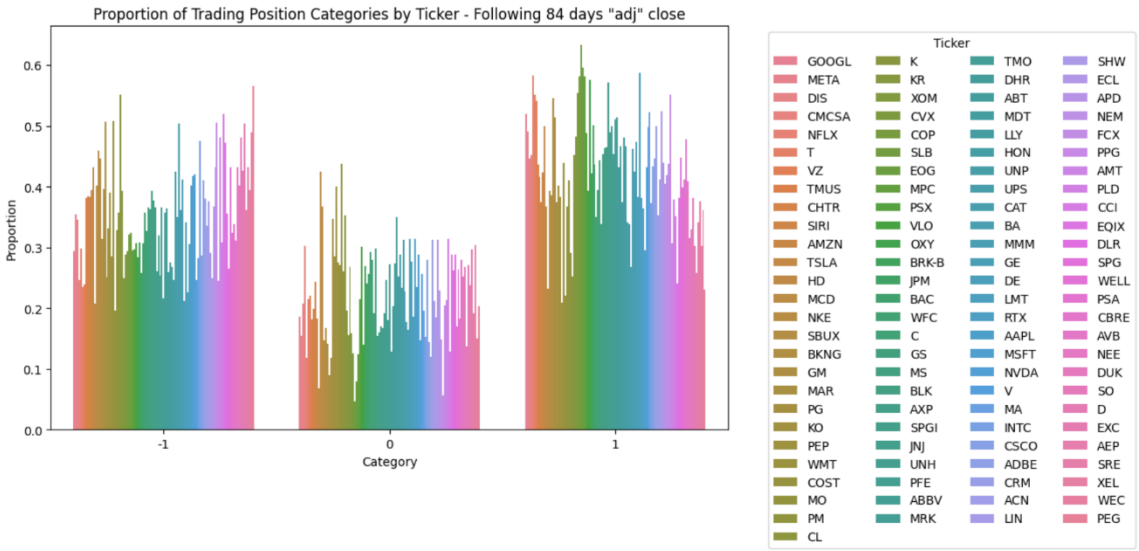

Figure 2. Proportion of Trading Position Categories by Ticker – Following 84 days

3.4 Model Training and Validation

The base model in our work is Mistral instruct v0.2 [1]. The reasoning behind choosing Mistral is its ability to process long context, which is crucial, since textual parts of reports usually exceed context length of other models (like BERT), another reason is that this model has relatively small number of parameters, which allows us to conduct multiple experiments with different fine-tuning techniques.

The model training steps are following (partial implementation is available in SI-3):

1. Download base model using Huggingface Transformers
2. Modify model head to perform regression/classification tasks (add additional output layer)
3. Fine tune the model. The approaches, that we are using to fine-tune the model are following (definitions in SI-4):
    a. Low-Rank Adaptation
    b. Prefix Tuning
    c. Soft prompting with modification of input embeddings

[1] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

The choice of best performing model is made based on accuracy/MSE on validation set.

The model was trained using Low-Rank Adaptation.
The dataset was split into train, validation and testing sets in proportion 70:10:20
The train-test spl

3.5 Strategy development

The strategy under development is an equity long-short strategy, which is a portfolio construction approach which capitalizes both increases and decreases in equity prices.
The strategy includes a combination of momentum and sentiment, extracted from reports.
The performance of the strategy is compared with performance of momentum-only strategy.
The strategy is evaluated based on historical information

4. Results

4.1 Model performance evaluation

The accuracy of the trained model is 47 (baseline (prediction of the dominant class is 39%))

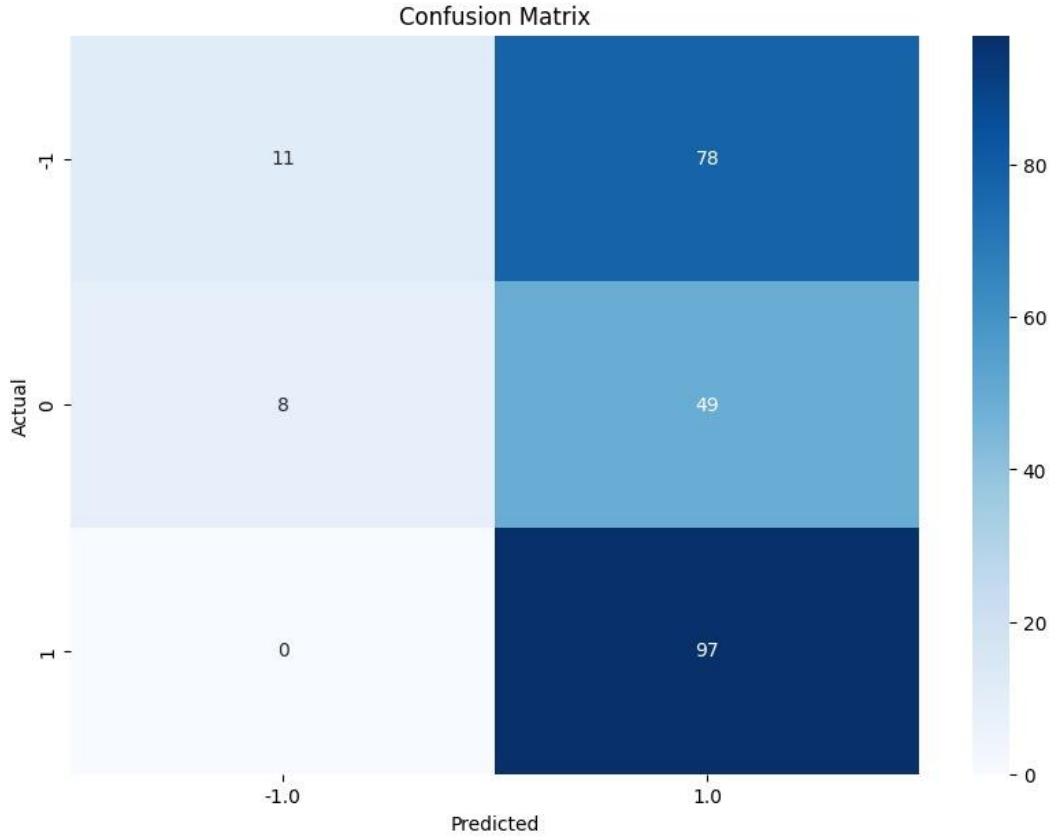

Figure 3. Confusion matrix of the model on test set. The model never predicts class "Hold", so corresponding column was removed from the

We have 3 classes (as described in section 3.3); however, the current model always predicts either short or long and never "hold" class.
The accuracy of the model is slightly higher, compared to baseline (baseline is constant prediction of dominant class).

Accuracy: 0.58 Precision: 0.57 Recall: 0.65 F1 Score: 0.60 [2]

Possible improvements

1. **Data Leakage Concerns**: There is a significant risk of data leakage due to the overlap in the time range of the data used for training the LLMs and the validation set. This overlap can lead to the model inadvertently memorizing rather than learning from the data, which may compromise the validity of the model's performance metrics.
2. **Dataset Preparation Variability**: To enhance model robustness, consider varying the dataset preparation techniques. This includes experimenting with different chunk sizes, utilizing diverse queries, and exploring a broader range of general parameters. Such variations can help in identifying optimal data processing methods that improve model performance.
3. **Advanced Training Techniques**: Implement more sophisticated training approaches, such as hyperparameter tuning. This involves systematically searching through multiple combinations of hyperparameter values to find the most effective settings that boost the model's accuracy and efficiency.
4. **Expanding Historical Data**: To improve the model's understanding and predictive accuracy, incorporate a complete set of data from all S&P 500 companies, extending the historical data to include reports from before 2011. This broader historical perspective can provide deeper insights and enhance the model's ability to generalize across different market conditions.
5. **Bias Consideration**: Address potential biases in the training data. Currently, the model primarily uses data from top-cap stocks and companies that have survived in the market (survivorship bias). Expanding the dataset to include a wider variety of companies, including those that did not survive, could help in developing a more balanced and representative model.

These improvements to be included in the next submission as we are still enhancing some parts of the project.

4.2. Strategy development

The following strategies have been conceptualized:

1. Momentum strategy (baseline strategy): 3 months of previous returns serve as an indicator for the following 3 months period (e.g. 3 months moving average).
2. LLM predictions-based strategy: predicted position (“Long”/ “Short” / “Hold”) serves as an indicator.
3. Combination of momentum and LLM predictions.

4.3 Strategies evaluation

The strategies were developed in the following way:
1. SP500 companies were divided by industry using GICS classification. Using stratified random sampling, 66 companies were selected.
2. 10-K and 10-Q fillings were downloaded from EDGAR, spanning from 2011 to 2024
3. 80% of reports were used for fine-tuning and validation of LLM, 20% were used for testing. The model was fine-tuned with the use of ternary targets, described above.
4. For both momentum and LLM-based strategy rebalancing was done on the next trading day from the publication of reports. For example, if there are n companies in the portfolio with long position, the weight of each is 1/n, if on the given date there is a long signal for another company, the updated weights of each stocks are 1/(n+1)
5. For both strategies the position spans over 84 calendar days (minimal distance between 2 consecutive reports). If given stock is in the portfolio, and the new signal for the same portfolio is “long”, the stock stays in the portfolio, if the signal is “short”, the stock is short-sold.

The performance of momentum and LLM-based strategy is presented on the figure 4.

[2] All scores were calculated for a modified testing set. The "hold" class was temporarily removed, and to simplify interpretation, the "Long" and "Short" data/classes were rebalanced.

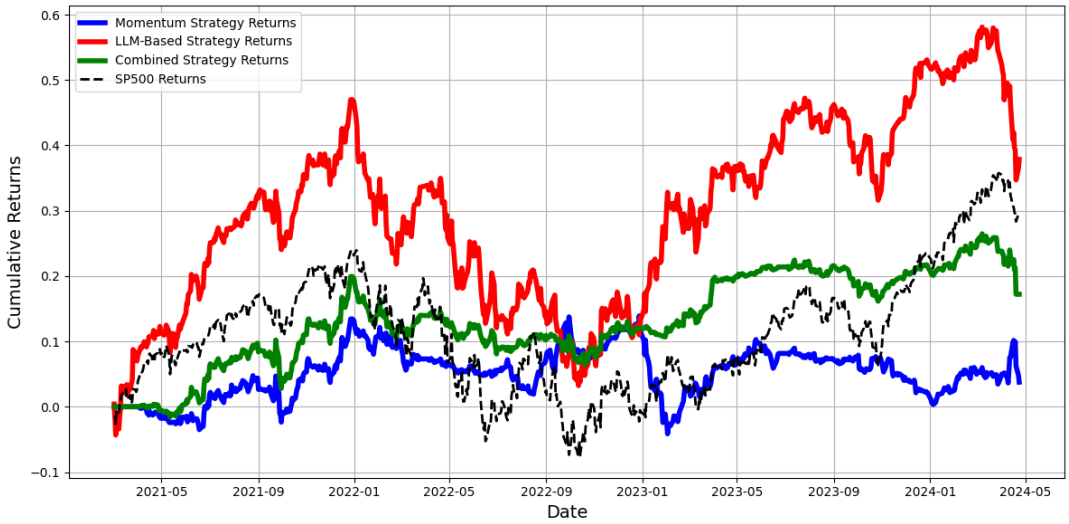

Figure 4. Cumulative returns of momentum and LLM-based strategies over back testing period. Combined strategy is defined as follows: if both momentum and LLM predict long, the long position is taken, if both predict short, then short position is taken, otherwise no action is taken

Table 2 summarizes performance metrics of strategies

Metric	*Momentum*	*LLM*	*Combined*	*SP500*
Annualized Return	0,02	0,12	0,05	0,10
Annualized Volatility	0,10	0,17	0,08	0,17
Sharpe Ratio	0,17	0,69	0,65	0,57
Sortino Ratio	0,22	1,01	0,88	0,82
Max Drawdown	-0,16	-0,30	-0,11	-0,25
Calmar Ratio	0,11	0,39	0,48	0,38
Win Rate	0,51	0,54	0,49	0,51
Profit Factor	1,03	1,12	1,13	1,10
Skewness	-0,28	0,04	-0,12	-0,13
Kurtosis	4,12	2,11	6,40	1,81
Value at Risk (VaR) 95%	-0,01	-0,02	-0,01	-0,02
Conditional Value at Risk (CVaR) 95%	-0,02	-0,02	-0,01	-0,02
Cumulative Return	0,04	0,38	0,17	0,29

4.4 Discussion

1. The developed LLM-based strategy outperformed both the momentum strategy and the benchmark.
2. The movements of the LLM-based strategy are similar to those of the benchmark. This similarity is primarily because most signals generated by the LLM are "Buy," as illustrated by the confusion matrix.
3. During fine-tuning, we observed an interesting pattern: additional fine-tuning led to decreased model performance. This decline may be due to the model forgetting the original data used to create the foundational checkpoint and focusing more on the content of the financial reports. The

original data includes news, which can significantly enhance the model's performance. This point crucial and needs to be analyzed further.

5. Conclusions

In recent months we have observed significant growth in the popularity of LLMs both in the academic world and financial industry. The main trend in the application of LLMs is using these models as assistants in the decision-making process, due to their ability to quickly extract and summarize relevant information. However, based on the review of competing models we have identified, that models, that have stock movement prediction ability are not suitable for application in the financial models directly, because they were trained on diverse dataset and designed to be multipurposed.

Our solution fills this gap by creating SotA model, intended to solve a single problem. According to our review, models designed to work in specific domain tend to outperform more general models, even if they are much lower in size.

The benefit of our project is that the framework for model construction that we are developing can be adapted to other scenarios, for example, default prediction, possibility of meeting covenants on bonds etc. However, we have observed that increased training leads to deteriorating performance as the model begins to "forget" the original data, such as news, and focuses more on the reports.

References

Araci, D. T. (2019). FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. https://arxiv.org/abs/1908.10063v1

Bhatia, G., Moatez, E., Nagoudi, B., Cavusoglu, H., & Abdul-Mageed, M. (2024). FinTral: A Family of GPT-4 Level Multimodal Financial Large Language Models. https://arxiv.org/abs/2402.10986v1

Crone, S. F., & Koeppel, C. (2014). Predicting exchange rates with sentiment indicators: An empirical evaluation using text mining and multilayer perceptrons. IEEE/IAFE Conference on Computational Intelligence for Financial Engineering, Proceedings (CIFEr), 114–121. https://doi.org/10.1109/CIFER.2014.6924062

Curme, C., Stanley, H. E., & Vodenska, I. (2015). COUPLED NETWORK APPROACH TO PREDICTABILITY OF FINANCIAL MARKET RETURNS AND NEWS SENTIMENTS. International Journal of Theoretical and Applied Finance, 18(7). https://doi.org/10.1142/S0219024915500430

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 1, 4171–4186. https://arxiv.org/abs/1810.04805v2

Hamborg, F., & Donnay, K. (2021). NewsMTSC: A Dataset for (Multi-)Target-dependent Sentiment Classification in Political News Articles. EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, 1663–1675. https://doi.org/10.18653/V1/2021.EACL-MAIN.142

Henry, E. (2006). Market Reaction to Verbal Components of Earnings Press Releases: Event Study Using a Predictive Algorithm. Journal of Emerging Technologies in Accounting, 3(1), 1–19. https://doi.org/10.2308/JETA.2006.3.1.1

Henry, P., & Krishna, D. (n.d.). Making the investment decision process more naturally intelligent How AI technologies are improving man-machine communication with natural language processing.

Ho, V. A., Nguyen, D. H. C., Nguyen, D. H., Pham, L. T. Van, Nguyen, D. V., Nguyen, K. Van, & Nguyen, N. L. T. (2019). Emotion Recognition for Vietnamese Social Media Text. Communications in Computer and Information Science, 1215 CCIS, 319–333. https://doi.org/10.1007/978-981-15-6168-9_27

Introducing BloombergGPT, Bloomberg's 50-billion parameter large language model, purpose-built from scratch for finance | Press | Bloomberg LP. (n.d.). Retrieved March 26, 2024, from https://www.bloomberg.com/company/press/bloomberggpt-50-billion-parameter-llm-tuned-finance/

Joshi, K., Bharathi, H. N., & Rao, J. (n.d.). STOCK TREND PREDICTION USING NEWS SENTIMENT ANALYSIS.

Konstantinidis, T., Iacovides, G., Xu, M., Constantinides, T. G., & Mandic, D. (2024). FinLlama: Financial Sentiment Classification for Algorithmic Trading Applications. https://arxiv.org/abs/2403.12285v1

Lee, J., Stevens, N., Han, S. C., & Song, M. (n.d.). A Survey of Large Language Models in Finance (FinLLMs). Retrieved March 26, 2024, from https://github.com/yya518/FinBERT

Lee, J., Stevens, N., Han, S. C., & Song, M. (2024). A Survey of Large Language Models in Finance (FinLLMs). https://arxiv.org/abs/2402.02315v1

Li, X., Xie, H., Chen, L., Wang, J., & Deng, X. (2014). News impact on stock price return via sentiment analysis. Knowledge-Based Systems, 69(1), 14–23. https://doi.org/10.1016/J.KNOSYS.2014.04.022

Liu, Z., Huang, D., Huang, K., Li, Z., & Zhao, J. (2020). FinBERT: A Pre-trained Financial Language Representation Model for Financial Text Mining. http://commoncrawl.org/

Loughran, T., Mcdonald, B., Battalio, R., Easton, P., Fuehrmeyer, J., Gao, P., Harvey, C., Hirschey, N., Marietta-Westberg, J., & Schultz, P. (2011). When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. The Journal of Finance, 66(1), 35–65. https://doi.org/10.1111/J.1540-6261.2010.01625.X

Mishev, K., Gjorgjevikj, A., Vodenska, I., Chitkushev, L. T., & Trajanov, D. (2020). Evaluation of Sentiment Analysis in Finance: From Lexicons to Transformers. IEEE Access, 8, 131662–131682. https://doi.org/10.1109/ACCESS.2020.3009626

Naseem, U., Razzak, I., Musial, K., & Imran, M. (2020). Transformer based Deep Intelligent Contextual Embedding for Twitter sentiment analysis. Future Generation Computer Systems, 113, 58–69. https://doi.org/10.1016/J.FUTURE.2020.06.050

Pang, B., & Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. ACL-05 - 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 115–124. https://arxiv.org/abs/cs/0506075v1

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, EMNLP 2002, 79–86. https://arxiv.org/abs/cs/0205070v1

Pennington, J., Socher, R., & Manning, C. D. (n.d.). GloVe: Global Vectors for Word Representation. Retrieved March 25, 2024, from http://nlp.

Price, S. M. K., Doran, J. S., Peterson, D. R., & Bliss, B. A. (2012). Earnings conference calls and stock returns: The incremental informativeness of textual tone. Journal of Banking & Finance, 36(4), 992–1011. https://doi.org/10.1016/J.JBANKFIN.2011.10.013

SENTIMENT ANALYSIS | English meaning - Cambridge Dictionary. (n.d.). Retrieved March 25, 2024, from https://dictionary.cambridge.org/dictionary/english/sentiment-analysis

Shah, R. S., Chawla, K., Eidnani, D., Shah, A., Du, W., Chava, S., Raman, N., Smiley, C., Chen, J., & Yang, D. (2022). WHEN FLUE MEETS FLANG: Benchmarks and Large Pre-trained Language Model for Financial Domain. Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, 2322–2335. https://doi.org/10.18653/v1/2022.emnlp-main.148

Wu, S., Irsoy, O., Lu, S., Dabravolski, V., Dredze, M., Gehrmann, S., Kambadur, P., Rosenberg, D., & Mann, G. (2023). BloombergGPT: A Large Language Model for Finance. ArXiv.Org. https://doi.org/10.48550/ARXIV.2303.17564

Xie, Q., Han, W., Zhang, X., Lai, Y., Peng, M., Lopez-Lira, A., & Huang, J. (2023). PIXIU: A Large Language Model, Instruction Data and Evaluation Benchmark for Finance. https://arxiv.org/abs/2306.05443v1

Yang, H., Liu, X.-Y., & Wang, C. D. (2023). FinGPT: Open-Source Financial Large Language Models. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.4489826

Yang, Y., Christopher, M., Uy, S., & Huang, A. (2020). FinBERT: A Pretrained Language Model for Financial Communications. https://arxiv.org/abs/2006.08097v2

Yang, Y., Tang, Y., & Tam, K. Y. (2023). InvestLM: A Large Language Model for Investment using Financial Domain Instruction Tuning. https://arxiv.org/abs/2309.13064v1

Zhong, N., & Ren, J. B. (2022). Using sentiment analysis to study the relationship between subjective expression in financial reports and company performance. Frontiers in Psychology, 13, 949881. https://doi.org/10.3389/FPSYG.2022.949881/BIBTEX

Supplementary information

The GitHub Repository
https://github.com/ralfcam/wqu_capstone/

The repository includes codes, notebooks, data, requirements and readme.

Additionally, we provide notebooks to quickly review key components of the project

SI-1 – Download Financial Reports from EDGAR
To fetch data from SEC EDGAR platform we have utilized requests package. The code in the provided colab notebook returns links to the requested reports (10-K and 10-Q) on EDGAR platform for a given company and a filing date.
https://colab.research.google.com/drive/1OPefb95aklnAmw5lmwKYv-MaK2y_fZNo?usp=sharing
Note: this code requires specific network configuration, in case it does not work, contact authors for the details

SI-2 – Processing the reports.
To process the reports, we have utilized Selenium package (to mimic behavior of the browser) and BeautifulSoup to parse the documents. The inputs to this code are links to the financial reports, the code returns separately text corpus of the reports and tables
https://colab.research.google.com/drive/1KiuVtfCDBK0MJTD8y-WsrWYad9PWRpAP?usp=sharing

SI-3 – Modeling
The in the provided notebook is the example of creation of the model. Here we are downloading Mistral-7B-Instruct, modify head of the model (in this code we are adding regression head, the code allows to add additional layers to the head to further customize architecture) and transforms dataset for training into appropriate format, using tokenizer from Mistral.
https://colab.research.google.com/drive/10Dlv4KtgTFYyThN0UXXvoWw_QKVeNT9M?usp=sharing
Note 1: we are running all the codes, related to modeling locally, because colab does not provide sufficient resources for model fine tuning (it is possible to fine tune Mistral with quantized low rank adoptation (QLoRA) inside colab, however, given available computational resources, we prefer to run locally, because of computational time and possibility to use standard LoRA and complete finetuning).
Note 2: This code does not include finetuning, because we are still experimenting with it.

SI-4 – Parameter-efficient fine-tuning

Parameter-efficient fine-tuning methods for large language models (LLMs) have become increasingly important as the size of these models continues to grow. Traditional fine-tuning, which involves updating all parameters of a pre-trained model for a specific downstream task, becomes impractical due to the high computational cost and memory requirements. This has led to the development of various parameter-efficient techniques, including Low-Rank Adaptation (LoRA), Prefix-Tuning, and Soft-Prompting. These methods aim to adapt LLMs to new tasks while training only a small fraction of the model's parameters, thus significantly reducing the resources needed for fine-tuning.

Low-Rank Adaptation (LoRA)
Low-Rank Adaptation (LoRA) proposes a novel approach to fine-tuning by freezing the pre-trained model weights and injecting trainable rank decomposition matrices into each layer of the Transformer architecture (Hu et al., "LoRA: Low-Rank Adaptation of Large Language Models"). This method significantly reduces the number of trainable parameters for downstream tasks. LoRA operates under the hypothesis that the change in weights during model adaptation has a low "intrinsic rank," allowing for efficient and effective fine-tuning with minimal trainable parameters. By optimizing the rank decomposition matrices instead of the original dense layers, LoRA achieves competitive or even superior performance compared to full fine-tuning, with the added benefits of higher training throughput and no additional inference latency.

Prefix-Tuning
Prefix-Tuning is another parameter-efficient fine-tuning method that freezes the pre-trained model parameters and learns a set of task-specific continuous vectors, or "prefixes," that are prepended to the input sequence (Li and Liang, "Prefix-Tuning: Optimizing Continuous Prompts for Generation"). These prefixes serve as soft prompts that condition the model's behavior for the specific task. Unlike traditional prompt engineering, which relies on discrete text prompts, Prefix-Tuning allows for end-to-end learning of the prompts through backpropagation. This method has shown strong results on generative tasks and offers a flexible way to adapt LLMs to new tasks without modifying the model architecture or training a large number of parameters.

Soft-Prompting
Soft-Prompting, also known as prompt tuning, is a technique that involves learning "soft prompts" to condition frozen language models on specific downstream tasks (Lester et al., "The Power of Scale for Parameter-Efficient Prompt Tuning"). Unlike discrete text prompts used in few-shot learning scenarios, soft prompts are learned through backpropagation and can be tuned to incorporate signals from labeled examples. This method significantly outperforms traditional few-shot learning approaches and closes the quality gap with model tuning as the size of the language model increases. Soft-Prompting retains the efficient serving benefits of frozen models while enabling robustness to domain transfer and efficient prompt ensembling.

SI-5. Search queries for similarity search

- "The company has incurred significant financial losses due to",
- "Capital investments for the fiscal year include",
- "A breakdown of revenue by geographic segment shows",
- "The company’s strategic initiatives for the upcoming year are",
- "Key risk factors impacting the business operations include",
- "Summary of cash flow activities for the current reporting period",
- "Major changes in shareholder equity occurred because",
- "The company's most profitable product lines are",
- "Recent acquisitions or divestitures have been made to",
- "Corporate governance practices adopted by the company include"

To convert chunks and queries to embeddings all-mpnet-base-v2

References

- Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." *arXiv preprint arXiv:2106.09685* (2021).
- Li, Xiang Lisa, and Percy Liang. "Prefix-tuning: Optimizing continuous prompts for generation." *arXiv preprint arXiv:2101.00190* (2021).
- Lester, Brian, Rami Al-Rfou, and Noah Constant. "The power of scale for parameter-efficient prompt tuning." *arXiv preprint arXiv:2104.08691* (2021).