News to Numbers: NLP Stock Return Predictions

Stanford CS224N Custom Project

Shree Reddy
Department of Computer Science
Stanford University
shreered@stanford.edu

Henrique B. N. Monteiro
Institute for Computational and Mathematical Engineering
Stanford University
hbnm@stanford.edu

Lucas Werneck
Department of Computer Science
Stanford University
lucasrw@stanford.edu

Abstract

In the highly competitive financial markets, accurately predicting the impact of news on stock prices is crucial for traders, investors, market makers, and other participants seeking to optimize their market exposure. In this study, we explore the power of natural language processing in predicting the normalized returns of securities in the 10-minute window following the publication of news articles. Our investigation encompasses 9 encoders, from published models finetuned on financial data like FinBERT to the latest OpenAI encoders like the t3-large embedding model. Our primary objective was to measure the ability of NLP in explaining stock returns and to present a framework completely free of look-ahead bias, a pervasive issue in the existing literature. We present proper data normalization, feature generation, and training routines to eliminate look-ahead bias. Our results validate NLP's application to return prediction while highlighting how the impact of news has been grossly overstated in previous studies due to data contamination. We rigorously analyze model performance both across time, security, and encoder and benchmark it against both conventional buy and hold and traditional news sentiment analysis as measured by an LLM. Our best model achieves a mean directional accuracy of 54.4% and a Sharpe Ratio of 7.6% per prediction, outperforming the Buy and Hold respective baselines of 50.7% and 1.6% and the LLM baselines of 51.2% and 3.5% . Our findings underscore the benefits of NLP in financial prediction while advocating for a robust and unbiased methodology.

1 Key Information to Include

Our TA mentor is Kaylee Burns, and we have no external collaborators, mentors, or share the project.

Team contributions: See Appendix

2 Introduction

Predicting stock prices has long been a complex problem, defying precise modeling despite significant efforts over the past 60+ years. Financial markets are influenced by numerous factors, including macroeconomic variables, corporate earnings, geopolitical events, and public sentiment, not to mention the natural randomness of order placement and trade flow. It is a problem with inherently low signal-to-noise ratio, and news are only one driving factor. The market efficiency and the abundance of highly sophisticated arbitrageurs keep the price process hard to predict and close to a martingale.

Our main question is to what extent security price movements can be predicted by novel news information. While much of the variance in security returns is believed to be random and difficult to predict, some market movements are driven by factors such as microstructure, macroeconomic conditions, order flow, and new public information. This study leverages NLP models to predict stock returns based on news articles, carefully examining their effectiveness and addressing common pitfalls like look-ahead bias. We aim to provide a realistic assessment of the power of NLP techniques in financial prediction and enhance understanding of how public news influences market behavior.

3 Related Work

The prediction of stock market movements using news and other text data has been an active area of research, combining natural language processing and financial analysis. Various studies have explored different approaches to integrate textual data with financial time series to enhance the predictive power of models. Li et al. (2022) proposed a transformer-based attention network for stock movement prediction by incorporating historical text and stock prices [1]. Their model captures the temporal dependence of financial data and effectively analyzes key information to achieve accurate predictions. This approach offers one way to tackle the challenge of integrating text and stock prices.

Several other studies have focused on developing encoders finetuned on financial data. The FinBERT model [2], introduced by Huang et al. (2022) was specifically designed for extracting information from financial text and showed promising results in financial sentiment analysis, which is a crucial part of predicting market movements. Another model with the same name of FinBERT [3] introduced by Dogu Araci (2019), also demonstrated the effectiveness of finetuned models tailored to the financial domain. Araci's implementation of FinBERT achieved state-of-the-art results in financial sentiment analysis tasks, significantly outperforming previous models such as LSTM with ELMo embeddings and ULMFit. The study reported an accuracy of 97% on the Financial PhraseBank dataset with full annotator agreement and set new benchmarks in the FiQA sentiment dataset by achieving lower mean squared error and higher R-squared values compared to other models.

4 Approach

4.1 Problem statement

Given some news story about a publicly traded company, predict from the text content the return of the company's stock following X minutes from the publication time. Each news consists of a headline, a timestamp, and possibly a text body, and we tackle the problem at a timeframe of 10 minutes but 3, 5, 15, 30, and 60 were also investigated with worse results. More formally, we do not forecast raw returns but rather the related normalized Jensen's alphas, as defined next.

4.2 Target definition

One then uses the time series of these factors to create a "base" model, often a linear regression, to explain the part of the returns controlled by the known factors. We regress the returns at time t against the factors at time t (as opposed to some $t' < t$) so that the factor model is purely explanatory with no attempt at forecasting. One then obtains the residuals of the factor model - the unexplained part of the returns - and fits a second model - the alpha model - to forecast the residuals: the "excess" returns over the known factors. See Avellaneda 2010 [4] or Gatev et. al 2006 [5] for examples.

For this project, our factor model is a simple Capital Asset Pricing Model (CAPM) [6] in which the only factor is a market factor β^M given by the S&P 500 index, and our alpha model is an NLP model to forecast the excess returns on the index, also known as the Jensen's alpha α_S in the context of CAPM for stock S. We then define our downstream model target as the volatility-normalized Jensen's alpha $\hat{\alpha_S}$. Formally, these quantities are defined for each ticker S as

$$\alpha_S = r_S - \beta_S^M r_M \quad \text{and} \quad \hat{\alpha_S} = \frac{\alpha_S}{\sigma_{\alpha_S}} \tag{1}$$

in which r_S is the stock return, r_M is the market return, and β_S^M is the stock exposure factor to the market (S&P 500), and σ_S is a forecast of the diffusive volatility of α_S for the time interval of the

alpha. Here we use a Gaussian GARCH(1,1) for σ_{α_S} since it explicitly models heteroscedasticity, reproduces volatility clustering, and reacts fast to volatility shocks without look-ahead bias.

We emphasize that return series undergo severe distributional shifts, most visible in the high instability of its second-order moments, so dynamic normalization is crucial. It cannot be stressed enough how a naive z-scoring of the whole data would be an acute methodological flaw as (1) it preserves the distributional instability and (2) introduces look-ahead bias when all data is used for normalization. Our dynamic, temporal, causal normalization mitigates the inherent distributional term structure without look-ahead bias. Henceforth, every mention to forecasting "returns" refers to $\hat{\alpha_S}$.

4.3 Model architecture

4.3.1 General framework

The full model pipeline starts with a news headline passing through an encoder to be transformed into a d-dimensional embedding vector. The embedding is concatenated to a vector of backward-looking financial features computed from price and volume data on the stock referenced in the news. The concatenated vector is then fed into a downstream network to predict the future return. See Figure 5.

4.3.2 Encoders

We tested 9 different encoders with different dimensions d. They are defined in Tables 1 and 2 below.

Name	dim	HuggingFace identifier
finbert	768	yiyanghkust/finbert-tone
finbert2	768	ProsusAI/finbert
distilroberta	768	mrm8488/distilroberta-finetuned-financial-news-sentiment-analysis
distilroberta2	768	NLP-FEUP/FT-mrm8488-distilroberta-finetuned-financial-news-sentiment-analysis
deberta	768	nickmuchi/deberta-v3-base-finetuned-finance-text-classification
roberta	768	RogerKam/roberta_fine_tuned_sentiment_financial_news
roberta2	1024	Jean-Baptiste/roberta-large-financial-news-sentiment-en

Table 1: Encoders available in HuggingFace

Name	dimensions	OpenAI identifier
OpenAI-128	128	text-embedding-3-large
OpenAI-768	768	text-embedding-3-large

Table 2: Encoders from OpenAI

Encoders `finbert` [2] and `finbert2` [3] come from publications and were finetuned on varied financial text like SEC fillings, stock analysis pieces, and some financial news. The remaining HuggingFace models were all finetuned only on financial news and are not connected to a publication. Finally, the OpenAI encoders are application-agnostic and have not been finetuned on financial text.

4.3.3 Financial features

The financial features include (1) the forecasted GARCH volatility for the current time, (2) notional volume for the current time, (3) past 3, 5, 10, 15, 30, 60 minute returns, (4) past changes in mean volatility, and (5) mean volume between the two prior 3, 5, 10, 15, 30 minute intervals.

The financial data provides further context about the market state in terms of current price, volatility, and volume levels as well as past changes in these variables, assisting in the interpretation of the encoded text data. This is particularly useful not only because market reactions are regime-dependent but especially because the timeliness of timestamps varies significantly, i.e.: some stories are published right after the actual news is made public while other pieces have a far bigger delay of many minutes if not hours. Full financial context of both current and past market states enables the model to learn to identify signs that the impact of the news already happened in the past (stale news) or will most likely happen in the near future (breaking news) to make a correct prediction. The same headline can have very different returns following its publication had it been published a couple minutes after the reported event (impact ahead) or delayed by hours (impact before timestamp), so

combining financial data is crucial for performance. We later demonstrate how a large language model (LLM) predicting news sentiment performs significantly worse in gauging the real impact of news due to lack of financial context despite its very high performance in actual sentiment analysis.

4.4 LLM annotations

For added baseline comparisons and feature engineering, we utilized the Claude-Haiku model from Anthropic to annotate our news headlines dataset. We asked it to generate its own sentiment and relevance scores for each headline. Our approach involved a pseudo chain-of-thought prompt where we provided Haiku with the news headline and the reference ticker. First, we asked Haiku to identify whether the ticker was directly mentioned by name in the headline. This step was used to engineer a feature named 'tagging', intended to help explain away much of the variance arising from ambiguous or irrelevant headlines. Next, we prompted Haiku to analyze the sentiment of the headline from a prospective investor's perspective. Haiku was instructed to provide a short explanation of the sentiment, followed by a numerical sentiment score ranging from -100 to 100. Finally, we asked Haiku to assess the relevance of the headline to the ticker, providing a short explanation and a relevance score ranging from 0 to 100. This prompt structure was designed to mimic the chain-of-thought concept, aiming to improve the resulting performance of the annotations.

4.5 Elimination of look-ahead bias

A core strength of our work compared to both previous projects and previous papers is our diligence in eliminating look-ahead bias, which is rampant in a significant part of the literature. In particular, the common formulation of predicting returns of a given day using news or social media posts from the same day is extremely flawed as the text often carries information about the day's return, e.g.: "Apple plunges after..." or "NVDA opens green with..." introduce significant look-ahead bias. Even if the target is changed to overnight returns with only news before market opening, several products like futures and foreign exchange trade 24h per day 5 or 6 days a week across markets, with cryptocurrencies and some over-the-counter products trading uninterruptedly. Since there is strong correlation of returns across markets and asset classes, any piece of financial information introduces look-ahead bias when the start of the return interval is not ahead of the text timestamp.

Therefore, the methodologically sound approach is to predict returns immediately after the timestamp as done here. A chronological train-test split and chronological training and forecasting are also used to treat the dataset as a time series. The normalization of both the target and the financial features is similarly careful by using volatility predicted by a GARCH(1,1) model fitted on past data. As previously explained, conventional z-scoring introduces look-ahead bias. Moreover, all financial features were structured to have zero temporal overlap with any target, and statistical tests were used to screen for accidental data contamination. Finally, we even decided to avoid the LLM annotations of sentiment and relevance as inputs since the model might have been trained in a large corpus of more recent news, so even if very subtle there would be no guarantee of no look-ahead bias.

4.6 One-tier vs two-tier model

The overwhelming majority of news are not financially impactful, and the aforementioned timestamp misalignment is a significant issue for all 3 news sources, so a robust architecture needs to handle these two core issues. The inclusion of rich financial data on current and past market states is one proposed solution, but we also put forward an innovative two-tier model, which to the best of our knowledge is considerably different than anything ever attempted in past literature.

Both the one-tier and two-tier models work as described under 4.3.1, with the difference that in the two-tier approach two models are used instead of one. The first model has its target altered to future changes in volatility and volume instead of returns while the second takes those predictions as additional financial features to then predict the returns. The core motivation lies in volatility and volume increases being a proxy score for news relevance/impact, so by providing them to the second model one can create an even richer representation of the financial context. Equipped with a rough estimate of a relevance score, the second model would possibly predict returns more accurately. The two-tier approach tries to disentangle the problem of concomitantly learning sentiment and impact of text as learning both together poses a significant challenge, especially given the very low signal-to-noise ratio of the data and all the aforementioned difficulties. See Figure 5 in Appendix for

a schematic representation of the two-tier approach, which can be understood as a generalization of the one-tier, which is in turn defined by simply excluding sub-model 1 and its output.

4.7 Training, Evaluation, and Baselines

The training uses a 70-30% chronological train-test split, and we tried both training a single model on all securities and training one per security, with best results on the first. We used mean squared error (MSE), L1 Loss, and Huber Loss to compare predicted and true normalized returns, with best results on the latter. All models were trained with Adam [7] with different learning rates and weight decays.

We evaluated using (1) explained variance, (2) mean directional accuracy (MDA), (3) Pearson correlation between predicted and true returns, and (4) Sharpe ratio per trade of the portfolio constructed by weighting the true normalized return by the predictions. For the case of LLM sentiment, we weight the positions analogously with the predicted sentiment.

We compared all models as well as the one-tier and two-tier approach across all 4 metrics. For the financial baselines, we compared buy and hold of each individual stock and of the S&P 500 index in terms of MDA and Sharpe since the other metrics are not defined. For the LLM baseline, we compute MDA, Pearson, and Sharpe using the sentiment score as the prediction to compare. These metrics are invariant under linear scaling of the prediction, so the comparison is methodologically sound.

5 Experiments

5.1 Data

The news and financial data are from Jan 2015 to May 2024 for AAPL, AMD, AMZN, GOOG, MSFT, NVDA, and TSLA. The news consist of 73,412 timestamped headlines aggregated from CapitalIQ, Benzinga, and Polygon for all tickers, and some contain additional data (body, tags, etc) while the financial data is the NYSE TAQ database of all trades recorded on the tickers in the period.

TAQ was processed down to minute returns with a GARCH(1,1) fitted to forecast volatility. We defined 3, 5, 10, 15, 30, 60 minute volatility-normalized returns to test targets of different timescales. We filtered out duplicate and empty headlines and matched news to returns to get input, target pairs.

An LLM (Claude-Haiku) was used to tag the news headlines pulled Polygon, Benzinga and CapitalIQ. The tag was a simple True/False that indicates whether or not the specified ticker is directly referenced/mentioned in the headline. The model was also used to provide its predictions on the sentiment and relevance of the news to the ticker, but this was used only as a baseline comparison.

5.2 Experimental details

Model Architecture: Grid search determined optimal parameters and features for single- and two-tier models.

Single Model Approach:

- **Encoder:** OpenAI-128
- **Target Columns:** $\alpha_{S_{10M}}$
- **Extra Inputs:** tag; $\alpha_{S_{-5M}}$
- **Hidden Dimension:** 8
- **Architecture:** External encoder, 3 fully connected layers (Leaky ReLU and ReLU), Dropout (later removed)

Training:

- **Batch Size:** 1024, **Epochs:** 50, **Loss Function:** HuberLoss
- **Optimizer:** Adam (Learning Rate: 0.006, Weight Decay: 0.005)

Two-Tier Model Approach:

Model 1(Volatility and Volume Processor):

- **Encoder:** OpenAI-128
- **Target Columns:** $\Delta V_{3M,5M,10M}$ and $\Delta\sigma_{3M,5M,10M}$ (V is Volume, σ is Volatility)
- **Extra Inputs:** $\{\Delta V, \Delta\sigma\}_{-[3M,5M,10M,15M,30M]}$; σ_S; $\alpha_{S_{-[3M,5M,10M,15M,30M,60M]}}$
- **Hidden Dimension:** 16

Training:

- **Batch Size:** 1024, **Epochs:** 100, **Loss Function:** HuberLoss
- **Optimizer:** Adam (LR: 0.008, WD: 0.002)

Model 2 (Downstream Processor):

- **Encoder:** OpenAI-128
- **Target Columns:** $\alpha_{S_{10M}}$
- **Extra Inputs:** tag; $\alpha_{S_{-5M}}$; $\Delta\hat{V}_{3M,5M,10M}$; $\Delta\hat{\sigma}_{3M,5M,10M}$
- **Hidden Dimension:** 12

Training:

- **Batch Size:** 1024, **Epochs:** 100, **Loss Function:** HuberLoss
- **Optimizer:** Adam (LR: 0.02, WD: 0.007)

5.3 Results

Name	Explained var(%)	MDA(%)	Pearson	Sharpe
deberta	0.170	52.302	0.049	0.048
distilroberta	0.196	51.468	0.048	0.048
distilroberta2	0.113	51.722	0.036	0.038
finbert	-0.014	51.685	0.026	0.024
finbert2	0.266	51.776	0.052	0.053
roberta	-0.005	51.015	0.027	0.030
roberta2	0.057	51.377	0.028	0.031
OpenAI-128	**0.484**	**54.422**	**0.082**	**0.076**
OpenAI-768	0.443	53.751	0.070	0.069
LLM sentiment	-	51.236	0.031	0.035
Buy & Hold	-	50.671	-	0.016

Table 3: Model performance across encoders and compared to benchmarks

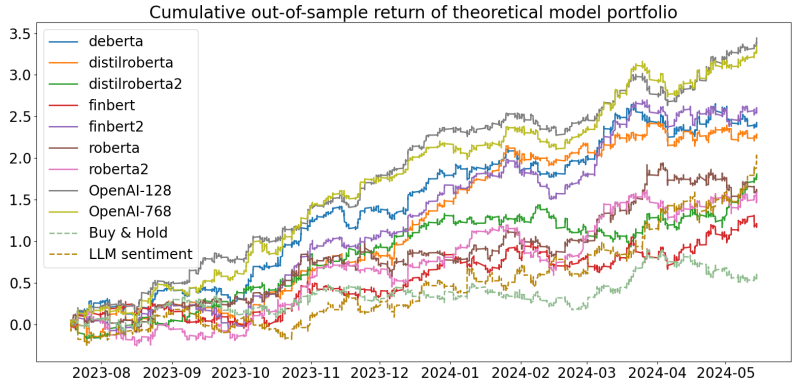

Figure 1: Cumulative returns of a theoretical portfolio constructed from model test predictions.

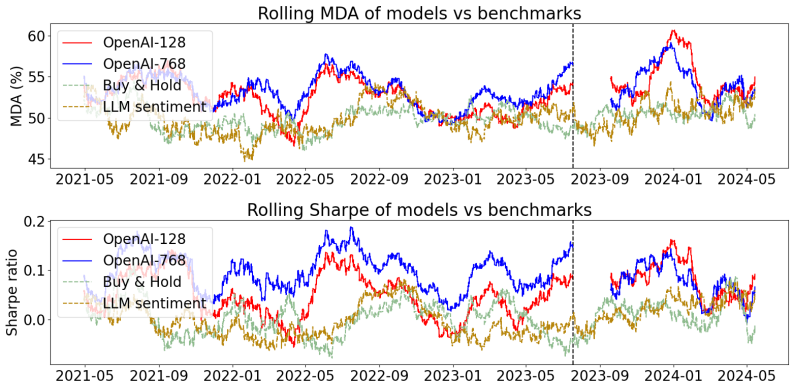

Figure 2: 1000-window rolling Sharpe and MDA of the best models.

Overall Performance The table and the cumulative return plot show that the models generally outperform the Buy & Hold strategy, indicating that the models are effective in capturing some predictive signals from the news headlines despite the low signal-to-noise ratio observed in our Exploratory Data Analysis. The OpenAI text-embedding-3-large (with both 128 and 768 dimensions) and deberta models are particularly notable for their performance, and they both largely outperform LLM sentiment, showing the power of integration of textual and financial data.

Explained Variance and Directional Accuracy

The explained variance and MDA metrics reveal that the OpenAI-128 model is the best, with an explained variance of 0.484 and an MDA of 54.422%. This indicates that the model captures a significant portion of the variability in stock returns and is the most accurate in predicting the direction of returns. The OpenAI-768 and deberta models perform slightly lower, suggesting that they are also effective in extract meaningful information from news headlines and financial data.

Risk-Adjusted Returns

The Sharpe ratio highlights the superior performance of the OpenAI-128 model (0.076), followed closely by OpenAI-768 (0.069). It indicates that these models are not only more accurate but the predictions have a more favorable risk profile. The lower Sharpe ratios with other encoders, such as distilroberta and finbert, suggest a poor representation of the embeddings for the task at hand.

Cumulative Returns

The cumulative out-of-sample return plot (Figure 1) illustrates the growth of hypothetical portfolios following the models' predictions. The OpenAI-128 and OpenAI-768 models consistently outperform other models and the Buy & Hold strategy. This consistent outperformance underscores the robustness of these models in leveraging news information for stock return prediction. The LLM sentiment model, while providing some predictive capability, significantly underperforms compared to the specialized NLP models, emphasizing the importance of targeted model architectures for this task.

Rolling Performance Metrics

The rolling MDA and Sharpe ratio plots (Figure 2) show that the OpenAI models maintain higher MDA and Sharpe ratios than the Buy & Hold strategy and LLM sentiment model, indicating consistent accuracy and a favorable risk-return profile over time. The variability in these metrics highlights the models' sensitivity to market conditions and news events, underscoring the need for continuous adaptation.Additionally, Figure 3 demonstrates that higher percentile cutoffs correlate with higher MDA, suggesting that larger predicted magnitudes indicate higher model confidence. This is crucial for financial modeling, as it implies that the model's stronger predictions (with higher absolute values) are more reliable, allowing investors to focus on the most confident forecasts for better decision-making. Further, see figure 4 which models cumulative returns in the test set.

Summary of Observations

The OpenAI text-embedding-3-large with 128 and 768 dimensions demonstrate superior performance across all metrics, effectively encoding news headlines to forecast stock returns, even without

fine-tuning on financial data. These specialized models proposed significantly outperform the LLM sentiment model and Buy & Hold strategy, underscoring the importance of tailored model architectures for financial prediction tasks. Rolling metrics indicate periods of high performance and fluctuations, reflecting the models' responsiveness to market dynamics and the need for ongoing evaluation and refinement.

6 Analysis

The challenge of predicting security prices/returns from news headlines is well-documented throughout this paper. This task is inherently noisy due to the numerous factors influencing stock prices. Evaluating the relevance of a news headline to the price of a security is particularly difficult but crucial. To understand our model's performance and provide a benchmark, we offer a selection of example headlines, their associated tickers, and the resulting normalized returns.

Table 4 lists these headlines along with the normalized returns predicted by our model, compared to the actual returns. The model's predictions are often conservative in magnitude, which does not necessarily indicate poor performance. Given the high noise in returns data, a predicted return should be seen as a central tendency or expected value rather than an exact figure. This perspective aligns with viewing normalized returns as proxies for confidence in directional predictions. A positive prediction indicates confidence in an upward movement, while a negative prediction suggests a downward trend, regardless of magnitude. When inspecting the results in Table 4, when the predicted normalized return matches the true return in direction, the model effectively captures the impact of the news headline on the security price and the data points with significant discrepancies between the predicted and true returns highlight the model's limitations. Most notably, are the ambiguous cases: Many news headlines are ambiguous or have mixed sentiment, making it difficult even for human analysts to predict their impact accurately. For instance, straight- forward headlines like "Mullen Automotive's stock at record low as it pushes ahead with plan for another reverse stock split" are easier to evaluate than more ambiguous ones like "What's going on in the oil market?" or "3 Key EPS Reports to Watch this Week."

Headline	Ticker	Actual Return	Predicted Return
Forget Nvidia: This Other "Magnificent Seven" Member Just Poured $11 Billion Into Data Centers	NVDA	-2.418821	-0.037260
Tesla, Deciphera Pharmaceuticals, Heartland Financial And Other Big Stocks Moving Higher On Monday	TSLA	1.069773	-0.074928
Amazon Spends Up To $4B On AI Startup Anthropic, Analysts Explore Deal Prospects	AMZN	0.022503	0.007440
Mullen Automotive's stock at record low as it pushes ahead with plan for another reverse stock split	TSLA	-2.467070	-0.106924
ChatGPT On Apple iPhone, Make Or Break Data For Stock Market Ahead	AAPL	0.656861	0.079368

Table 4: Headlines with Normalized Returns Predicted by Model vs Actual Returns

7 Conclusion

We have demonstrated that NLP techniques can be effectively applied to predict stock returns under a methodologically robust approach with no look-ahead bias and enhanced by the integration of financial and textual data for better context for predictions. Our statistically significant results outperform traditional benchmarks, underscoring the potential of integrating financial data with news embeddings. However, challenges remain in handling the low signal-to-noise ratio inherent in financial data since only a small part of the variance in the data can be explained by the news. Future work could explore other ways of separating the relevance and sentiment tasks beyond the two-tier model approach as well as invest more modeling effort into getting a representation of market state with respect to the expected reaction to different type of news.

8 Ethics Statement

There are two main ethical concerns. If our work is leveraged for real trading, we need to assure that the third-party news data can be used for commercial purpose without authorization from or royalties for the news creators. This can be consulted directly with the sources to prevent unauthorized use. Secondly, we need to be mindful of ESG concerns and define ethical standards with respect to controversial securities associated with undesirable societal outcomes. These include the fossil fuel industry, the arms industry, companies associated with human rights violations, etc. We might abstain from trading those or formulate strict rules to comply with most recent ESG standards.

References

[1] Yawei Li, Shuqi Lv, Xinghua Liu, and Qiuyue Zhang. Incorporating transformers and attention networks for stock movement prediction. *Complexity*, 2022(1):7739087, 2022.

[2] Allen H Huang, Hui Wang, and Yi Yang. Finbert: A large language model for extracting information from financial text. *Contemporary Accounting Research*, 40(2):806–841, 2023.

[3] Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*, 2019.

[4] Marco Avellaneda and Jeong-Hyun Lee. Statistical arbitrage in the us equities market. *Quantitative Finance*, 10(7):761–782, 2010.

[5] Evan Gatev, William N Goetzmann, and K Geert Rouwenhorst. Pairs trading: Performance of a relative-value arbitrage rule. *The Review of Financial Studies*, 19(3):797–827, 2006.

[6] William F Sharpe. Capital asset prices: A theory of market equilibrium under conditions of risk. *The journal of finance*, 19(3):425–442, 1964.

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

A Appendix (optional)

Team Contributions: Lucas pulled and processed the Polygon news, the Claude annotation, and OpenAI Embedding datasets; performed exploratory data analysis of the annotation and news datasets; finetuned the final architectures, input features, and hyperparameters via experimentation and grid search. Henrique pulled and processed the financial data and the Capital IQ news; designed the financial features and targets; experimented with different encoders; performed exploratory data analysis of the datasets; built pipelines and templates for model training, benchmarking, and visualization. Shree pulled and processed the Benzinga news, scraped the bodies of the Polygon articles for the LLM annotations; finetuned the final architecture with different loss functions and experimentation with cross-validation and addition of dropout layers.

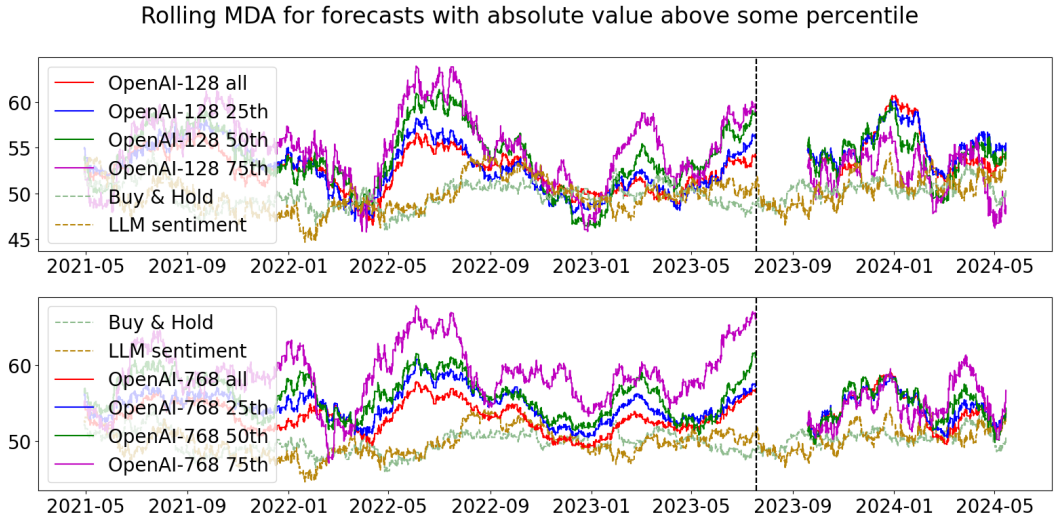

Figure 3: The magnitude of the prediction serves as a proxy for model confidence as shown by the dependence of mean directional average on the percentile magnitude of the predictions. Prediction of higher magnitude (measure here with percentile cutoffs) yield higher MDA.

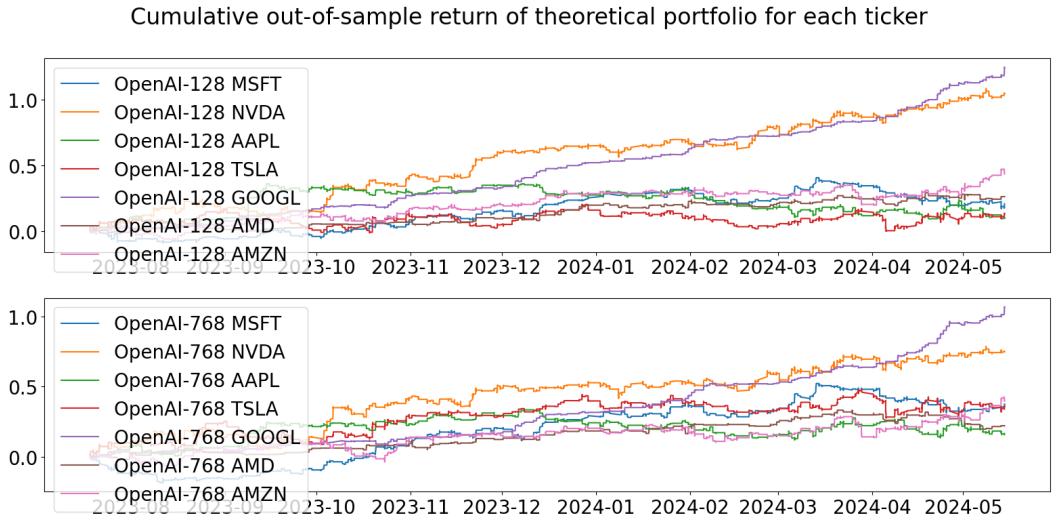

Figure 4: Cumulative returns in the test set show that most model performance comes from correctly predicting NVDA and GOOGL, revealing an unexpected structure to the difficulty of forecasting different tickers based on news.

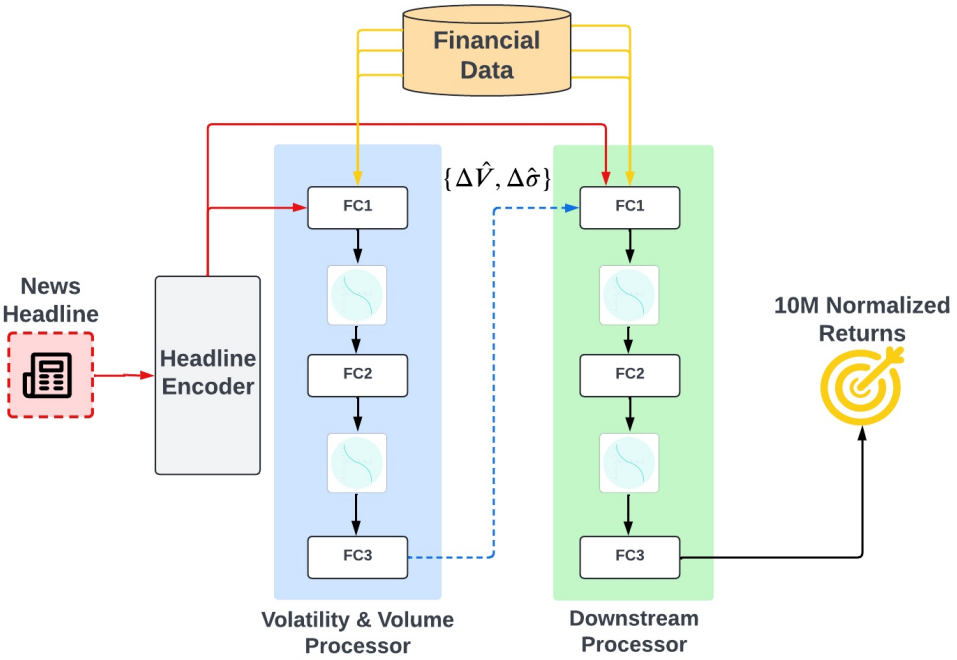

Figure 5: Architecture of the two-tier model with the one-tier as a special case