Progress Report: Enhancing Financial Domain Language Models with Auxiliary Information

Muhammad Hamza, Christian Berger

May 8, 2023

Abstract

We propose a new learning method that combines auxillary information along with language to better ground large language models. In order to prove the efficacy of this approach we share a large labelled dataset that has been built using our method. We further provide a new model that learns stock price prediction using self supervised learning. Finally, we combine it with a number of language models to show potential improvements in performance this method yields.

1 Introduction

Traditionally, numerous financial firms have relied on rapid access to pertinent news in order to gain a competitive edge in trading activities. The emergence of large language models (LLMs) and their increasing proficiency in handling natural language has become crucial in this domain, as it significantly accelerates the news processing speed. However, most LLMs have limited exposure to financial jargon, resulting in insufficient domain expertise. To address this issue, several approaches have attempted to fine-tune existing models on small subsets of financial news. While this strategy has led to modest performance improvements, language models tailored for the financial sector continue to be hindered by a lack of exposure to labeled financial domain data.

In contrast to prior approaches, our project aims to harness auxiliary information in conjunction with text to enhance LLMs, such as BERT, in comprehending complex language. Focusing on the financial domain offers two distinct benefits over other domains where similar techniques may be employed:

1. The domain is highly specialized, so even individuals fluent in English may struggle to understand the intricate nomenclature and jargon in specialized news. This presents an excellent opportunity to demonstrate how auxiliary information can provide greater context to LLMs, compensating for their lack of understanding of specific terms they may not have encountered.

2. The financial domain possesses a wealth of auxiliary information that can be easily accessed and utilized alongside textual data, such as publicly available stock prices. By narrowing down market indicators within a small time window and concentrating on the entities mentioned in the news, we can associate news with market indicators. Although this method does not eliminate all noise, it should effectively provide a robust bias grounded in real-world knowledge.

Our project comprises three distinct and progressively sophisticated approaches for utilizing market information to label data:

1. Initially, we employed simple statistical correlation to map trends in news to trends in the market. Considering the causal-reflective relationship between news and market trends, we explored this association using available statistical methods. This straightforward approach benefits from its simplicity, transparency, and speed.

2. Our second approach relied on predictive modeling, wherein we created feature vectors from market data and attempted to use these to label news independently of the text. Although this method still required less data and was relatively fast and interpretable, it suffered from not considering the textual data, similar to the statistical approach. Despite this limitation, its performance exceeded that of the statistical approach, suggesting a complex relationship between news and the market.

3. To achieve maximum accuracy, we decided to embed market information within the text and allowed the model to learn the relationship between both sources. We developed a custom transformer that learns stock price embeddings analogous to BERT for text. The embeddings from language models and the market were then combined and fed into a neural network for sentiment classification. Although we employed this model for sentiment classification, replacing the network's head can enable its adaptation to various tasks, such as stock market price prediction or finance-oriented question answering.

In summary, we sought to utilize market indicators to obtain latent vectors for financial news. We explored three different approaches to optimally leverage these market indicators, each with varying levels of complexity and performance.

2 Methodology

Our project followed a sequence of steps to validate our hypothesis, consisting of the following components:

1. Gathering the dataset
2. Statistical analysis of market trends
3. Predictive modeling using out-of-the-box methods
4. Custom Transformers

2.1 Gathering Data

To the best of our knowledge, no other approach has attempted to utilize auxiliary information to assist language models. Consequently, we needed to collect not only textual data but also corresponding market data. To create a comprehensive dataset for our project, we compiled data from various sources, including Wharton Research Data Services (WRDS), Kaggle, and personally sourced stock data.

Wharton Research Data Services (WRDS)

- We employed the Python library *wrds* to access financial datasets provided by WRDS.
- The Center for Research in Security Prices (CRSP) dataset, accessible through WRDS, was utilized in our study. More information on CRSP can be found at https://wrds-www.wharton.upenn.edu/pages/about/data-vendors/center-for-research-in-security-prices-crsp/.
- CRSP offers several datasets, covering US stocks, including those traded on the NYSE, AMEX, and NASDAQ.
- We obtained datasets with end-of-day and end-of-month stock prices to capture both short-term and long-term market trends.
- SQL queries were used to extract the relevant data from WRDS, enabling us to efficiently filter and process the information needed for our study.

Kaggle Dataset

We utilized a dataset from Kaggle called us_equities_news_dataset, which provided news headlines and their associated stock tickers. With the stock tickers, we were able to obtain the corresponding price information: daily return, price, trading volume, number of trades, and return of the S&P500 (market return).

Yahoo Finance

In order to further enrich our dataset with stock price information, we developed a script to extract stock prices from Yahoo Finance. Yahoo Finance is a popular platform that provides financial news, data, and commentary, including stock quotes, press releases, financial reports, and original content.

The script was designed to retrieve essential stock price details, such as historical prices, trading volume, opening and closing prices, and other relevant metrics for each stock ticker. By leveraging Yahoo Finance's extensive database, we were able to enhance our analysis by incorporating a comprehensive range of stock market information.

Data Processing and Challenges

While our initial approaches did not demand substantial amounts of data, our transformer model required an extensive dataset. One challenge we encountered with news headlines was their predominantly neutral nature. We suggest employing a model that summarizes news articles, such as ChatGPT, to extract more meaningful information. This manual, time-consuming process can provide valuable contributions for future research.

We utilized key performance indicator (KPI) metrics, such as open, close, high, and low prices, and normalized them for each company.

Summary

By combining data from various sources, such as WRDS, Kaggle, and Yahoo Finance, we created a comprehensive dataset for our project. This approach allowed us to adequately cover the financial domain, address potential data gaps, and facilitate the development of a more accurate and reliable model. The dataset provided a robust foundation for our analysis, enabling us to better assess the potential of auxiliary information in improving LLMs' proficiency in the financial domain.

Combining the stock price data from Yahoo Finance with the news headlines and market reactions from the Kaggle dataset allowed us to create a more robust and accurate dataset for our project. This comprehensive dataset facilitated a deeper understanding of the complex relationships between news events and stock market trends, enabling us to better assess the potential of auxiliary information in improving LLMs' proficiency in the financial domain.

In summary, the process of gathering data involved obtaining and processing financial news headlines and stock market data from various sources. By combining this information and addressing the challenges encountered, we were able to create a robust dataset that served as the basis for our study. This dataset played a crucial role in helping us evaluate the effectiveness of using auxiliary information to enhance the performance of LLMs in the financial domain.

2.2 Correlation Analysis with Market Indicators

This approach focuses on computing the correlation between market performance indicators and news sentiment. The primary goal is to investigate the importance and viability of different market indicators in the presence of noisy sentiment labels obtained from the news.

2.2.1 Labeling News Sentiment

We used a number of open-source FinBERT models to label news sentiment. Due to the imperfect nature of the models, the resulting labels were noisy. We observed an imbalance in the ratio of neutral headlines to positive or negative ones. To address this, we corrected the ratio to a more acceptable level while still preserving the characteristics of in-the-wild data.

2.2.2 Processing Stock Market Data

Processing the stock market data required several steps. First, we normalized the market trends, mapping all values to the range $[-1, 1]$. Next, we computed the daily changes in these normalized values to create a time series representing the daily variations. Our goal was to relate these variations to news headline sentiment. Two main challenges arose during this process:

1. Deciding which changes are large enough to be considered significant and which are merely noise.
2. Determining whether a change is part of the overall market trend or an outlier.

To address these challenges, we employed the following approaches:

1. **Benchmarking with Market Index:** We used the S&P 500 market index and computed a moving average over a specified period. This moving average served as a benchmark to decide whether a change was significant.

2. **Noise Thresholds:** We set various noise thresholds, chosen empirically, to maximize the performance of all metrics.

2.2.3 Correlation Computation

Finally, we used Pearson's correlation coefficient on our dataset to assess the effectiveness of this simple and transparent method in capturing the relationship between news sentiment and market performance indicators.

2.3 Predictive Modelling

Motivated by the challenges encountered in our correlation-based approach, we explored a more powerful family of predictive models from machine learning. These models can better address the complexities of the relationship between news sentiment and market trends. We used the same sentiment labels and normalized market trends as input features for the models.

2.3.1 Machine Learning Models

We experimented with a variety of machine learning models to determine their effectiveness in predicting market trends based on news sentiment. The models we employed included:

1. **Logistic Regression:** We used logistic regression with polynomial expansion to capture nonlinear relationships between the features.

2. **Random Forests:** We employed random forests, which are known for their robustness and ability to handle high-dimensional data.

3. **Support Vector Machines (SVM):** We utilized SVMs, which are effective in high-dimensional spaces and can model complex decision boundaries.

4. **Linear and Quadratic Discriminant Analysis:** We applied linear and quadratic discriminant analysis to model the differences in feature distributions across the classes.

5. **Ensemble Techniques:** We experimented with ensemble methods, such as bagging and boosting, to improve the overall prediction accuracy by combining the strengths of multiple base models.

By exploring various machine learning models, we aimed to identify a suitable approach that can effectively capture the intricate relationship between news sentiment and market performance indicators.

3 Outlook

3.1 Custom Transformer Model

Our final approach involves utilizing a custom transformer model trained on stock market data in a self-supervised manner. By leveraging bidirectional context and a masked value prediction objective, the model can learn appropriate embeddings for the market that can be used in conjunction with language embeddings. Depending on the task, different architectures can be employed, such as adding a small classifier head for sentiment analysis or a regressor head for predicting stock prices.

3.1.1 Data Preparation and Preprocessing

1. **Stock Data Collection:** We used Yahoo Finance to obtain stock prices for the last 13 years for a set of companies and four related companies for each.
2. **Data Processing:** We performed the following steps to process the dataset:
   (a) Normalize the market trends for each company.
   (b) Select a small time window, such as three days, and obtain stock prices for the related companies.
   (c) Merge the stock prices for each company using a separator token.
   (d) Merge the data for all key performance indicators (KPIs).
   (e) Randomly mask values using a probability of 0.15, inspired by the BERT model.

3.1.2 Model Training and Fine-tuning

1. **Self-Supervised Training:** Train the custom transformer to predict the masked values from the processed dataset.
2. **Language Embeddings:** Use a pre-trained model, such as BERT, to obtain language embeddings for the news headlines.
3. **Embeddings Combination:** Concatenate the embeddings from the custom transformer with the language embeddings and feed them to the classifier.
4. **Model Refinement:** Freeze the pre-trained language model and train the custom transformer using the noisy labeled dataset obtained from FinBERT.
5. **Fine-tuning:** Further fine-tune the model on a small subset of labeled financial data to compare its performance with other models.

3.1.3 Potential Challenges and Improvements

1. **Language Model Selection:** Experiment with different language models, such as RoBERTa or GPT, to determine their impact on the model's performance.
2. **Transformer Architecture:** Explore alternative transformer architectures, such as the Transformer-XL or the Longformer, to adapt to potential long-term dependencies in the data.
3. **Noise Reduction:** Develop strategies for reducing noise in the self-supervised training process, such as using denoising autoencoders or improving the masking strategy.
4. **Domain Adaptation:** Experiment with domain adaptation techniques to better transfer knowledge from the pre-trained language model to the financial domain.
5. **Hyperparameter Optimization:** Perform a comprehensive hyperparameter search to optimize model performance, including learning rates, batch sizes, and the number of layers.