Sentiment Analysis of Twitter Data to Predict the Stock Market Using Large Language Models

Machine Learning CS-535

Shlok Kothari

December 13, 2023

Abstract

The project investigates the efficacy of leveraging large language models, specifically BERT, for conducting sentiment analysis on financial tweets with the goal of predicting intraday movements in the stock market. The project delves into the inherent sentiment of financial tweets along with exploring the practicality of deploying a trading bot guided by the insights derived from the trained language model. To establish a robust foundation, two distinct datasets were employed for model training. The initial dataset encompasses a comprehensive collection of tweets paired with their corresponding sentiments, serving as the bedrock for fine-tuning the large language model, BERT. The second dataset is an aggregation of tweets organized by company, providing a nuanced understanding of sentiment dynamics within specific market segments. The findings of the study reveal compelling insights into the comparative performance of the trading bot against traditional passive investment strategies. Notably, the bot consistently outperformed the broader market, yielding higher returns across various market conditions. Furthermore, there were instances where the bot demonstrated significant outperformance, affirming the potential of large language models in enhancing stock market analysis.

Introduction

In recent years, the intersection of artificial intelligence, natural language processing, and financial markets has undergone unprecedented exploration, uncovering novel avenues for predictive analysis and informed decision-making. This research contributes to this burgeoning field by investigating the potential of leveraging advanced language models, specifically BERT, for predicting stock market movements using sentiment analysis on Twitter data. Inspired by recent breakthroughs in the use of ChatGPT and other large language models for stock market prediction through news headlines, as evidenced in a seminal research paper Lopez-Lira et al, this project seeks to extend these findings to a different domain—Twitter. The research paper documented a substantially positive correlation between ChatGPT scores derived from news headlines and subsequent daily stock returns, highlighting the outperformance of ChatGPT against traditional sentiment analysis methods. Building upon this foundation, this project employs BERT for sentiment analysis on a rich dataset of financial tweets, aiming to discern the sentiments as either positive or negative for firms' stock prices. Unlike the conventional approach of using news headlines, this project explores the efficacy of large language models on a more dynamic and real-time platform—Twitter. In alignment with the overarching objective of democratizing financial strategies, this research investigates the accessibility of advanced language models to the broader population. The study goes beyond the confines of traditional sentiment analysis by integrating BERT with Twitter sentiment data, emulating the methodology of the research paper while navigating the nuances of social media content.

Problem Statment

In the context of leveraging large language models for stock market prediction, this research project is designed to address the following key problem statement:

1. Optimizing Financial Decision-Making with BERT:

- Explore the effectiveness of BERT in conducting sentiment analysis on financial tweets to predict intraday movements in the stock market. Assess the model's ability to discern nuanced sentiments within the dynamic and real-time context of Twitter data.

2. Intraday Trading Strategy Deployment:

- Investigate the practicality of deploying a trading bot, informed by insights derived from the fine-tuned BERT model, to execute intraday trading strategies based on the sentiment analysis of financial tweets.

Dataset

Dataset for finetuning BERT:

The dataset utilized in this project compiles stock-related news sourced from diverse Twitter handles, specifically focusing on economic news. It consists of two sentiment categories: Negative (-1) and Positive (1), with 2,106 instances marked as negative and 3,685 as positive. Acquired from various Twitter handles via the Twitter API, this dataset was custom-labeled for sentiment analysis, providing a nuanced understanding of sentiments associated with economic news.

Origin: The dataset was retrieved from Kaggle, ensuring accessibility and credibility.
Custom Labeling Process: Each tweet underwent manual labeling, assigning sentiment labels based on the researcher's evaluative judgment.

Stock Data:

This dataset is a comprehensive compilation of tweets mentioning any NASDAQ 100 Twitter symbol, spanning a period from March to June 2016. The dataset provides a rich repository of social media content related to NASDAQ 100 companies, offering valuable insights into the sentiment dynamics surrounding these entities.

Dataset Specifications:

- **Time Period:** The dataset spans 79 days, covering the period from March 28th to June 15th, 2016.
- **Tweet Volume:** With an expansive collection, it encompasses approximately one million tweets.
- **Format:** Organized into 6 Excel files per company, facilitating structured and granular analysis.
- **Twitter Stream:** The dataset includes the Twitter stream, providing a real-time perspective on the discussions surrounding NASDAQ 100 companies.
- **Retweets:** Retweets are integrated into the dataset, ensuring a comprehensive view of the social media interactions.
- **Dataset Size:** The compressed dataset size is approximately 178 Mb.

- **NASDAQ100 Coverage:** The dataset covers the complete NASDAQ 100 index, from $AAL to $YHOO, with individual Excel files organized alphabetically for ease of access.

Methodology

1) Preprocessing Data: Enhancing Data Quality

In the preliminary phase of our project, we recognize the importance of data cleanliness and consistency. The process of preprocessing plays a pivotal role in refining the raw financial tweets before subjecting them to sentiment analysis using the BERT model. This code addresses various challenges inherent in financial tweets, such as hyperlinks, emojis, symbols, and user tags. By standardizing the text through these transformations, we ensure a cleaner and more coherent dataset, laying the foundation for more effective sentiment analysis using the BERT model.

2) Prepare the Bert NLP model tokenizer to encode tweets

In the initial phase of our methodology, we leverage the BERT (Bidirectional Encoder Representations from Transformers) model, a state-of-the-art natural language processing (NLP) model. The BERT model tokenizer, pretrained on the 'bert-base-uncased' architecture, is employed to encode the raw tweets. This process involves tokenizing the text, adding special tokens ([CLS] and [SEP]), padding the tweets to a specified maximum length, and generating attention masks. To determine the appropriate maximum length for encoding, we analyze the distribution of encoded tweets, selecting the maximum length accordingly. The chosen maximum length is essential for preserving the structure of the input while efficiently managing computational resources. **Maximum length - 53** The encoded tweets are then processed through the BERT model, generating input tensors and attention masks. These tensors are organized into batches to facilitate parallel processing during training and testing. In this regard, we employ a batch size of 16, a recommended configuration depending on GPU capabilities. Furthermore, to ensure the model is adequately trained and tested, we utilize data loaders for both training and testing phases. For training, a random sampler is applied to the data, while a sequential sampler is used for testing. These data loaders enable efficient management of input data, ensuring diverse batches for model optimization during training and evaluation during testing.

3) Define the Bert NLP Classifier

In our methodology, we employ a custom sentiment analysis classifier based on the BERT. The 'BertClassifier' class is designed to facilitate the training and evaluation of sentiment prediction models using a combination of pretrained BERT embeddings and a custom neural network.

The architecture consists of three main components:

BERT Model Integration: We integrate the 'bert-base-uncased' pretrained model into our classifier. This allows us to leverage the rich contextualized embeddings generated by

BERT for the initial layers of our neural network.

Custom Classifier Layer: Following the BERT section, we introduce a custom classifier layer tailored to our sentiment analysis task. This layer comprises linear transformations with a ReLU activation function, mapping the BERT embeddings to the final output layer. The architecture includes an input layer of 768 neurons (corresponding to BERT's hidden size), a hidden layer with 50 neurons, and an output layer with 2 neurons for binary classification.

Freezing BERT Weights: For increased model stability and to avoid overfitting, an option is provided to freeze the BERT model's weights during training. This ensures that the pretrained BERT embeddings are not updated during the training of the custom classifier. This is particularly useful when dealing with limited labeled data, preventing the risk of losing valuable knowledge encoded in the pretrained BERT weights.
Throughout the training process, the forward method is called, taking input IDs and attention masks as input and producing the final logits for sentiment classification. The output is then used to compute the loss and optimize the model during the training phase. The incorporation of the ReLU activation function in the custom classifier introduces non-linearity, enhancing the model's capacity to capture complex patterns in the sentiment data.
This classifier architecture is a key component of our broader methodology, contributing to the comprehensive exploration of sentiment analysis on financial tweets for stock market predictions.

4) Finetuning BERT

In the training phase, the BertClassifier model was initialized with specific hyperparameters and configurations. The random seed was set to ensure repeatability of results. The device (GPU or CPU) was assigned based on its availability. The model was trained over multiple epochs, with each epoch comprising several steps.

The hyperparameters defined for the training include:

- Epochs: 5
- Learning Rate: 5×10^{-5}
- Epsilon: 1×10^{-8}
- Train – 75%, Test – 25% (not a hyperparameter)

The AdamW optimizer was employed for updating model parameters, and a scheduler was defined using a linear schedule with warm-up steps. The loss function chosen was the Cross Entropy Loss.

During each epoch, the model was set to training mode, and the training data was iterated through batches. The model's parameters were updated based on backpropagation and gradient clipping. The scheduler was utilized to adjust the learning rate dynamically. The average training loss per batch was calculated.

Subsequently, the model was set to evaluation mode, and its performance was assessed on the test data. The test loss and accuracy were computed batch-wise. The final results for each epoch, including training loss, test loss, and test accuracy, were printed.

5) Predictions for new data

In this phase of the methodology, the sentiment predictions for stock data are generated using the trained BERT model. The process involves several crucial steps to ensure accurate predictions and insightful results.

Firstly, the relevant stock data is collected and preprocessed. The data, sourced from various companies, undergoes several transformations. Each stock's Twitter data is loaded from Excel files, and additional information such as the Ticker, Date, and Hour is extracted. The data is cleaned using the preprocess function.

To streamline the data for sentiment analysis, only specific columns are retained, including 'Tweet Id', 'Ticker', 'Datetime', 'Text', 'tweetClean', 'Favs', 'RTs', 'Followers', 'Following', and 'Is a RT'. Additionally, missing values in columns such as 'Favs', 'RTs', 'Followers', and 'Following' are filled with zeros.

The tweet content, now cleaned and refined, undergoes encoding using the same BERT tokenizer and preprocessing techniques employed during the training phase. This ensures consistency and compatibility with the BERT NLP model.

The processed stock data is then organized into PyTorch data loaders, facilitating efficient batch-wise processing. The BERT model, set to evaluation mode (model.eval()), is employed to predict sentiment for each batch of the stock data. The predictions, represented as logits, are converted into binary sentiment labels (0 or 1) using the torch.argmax function.

These sentiment predictions are integrated back into the original stock dataframe under the Sentiment column. The dataframe is subsequently saved as a new CSV file in the project's result directory, named stock_data_sentiment.csv. This file serves as a valuable output, encapsulating the sentiment predictions for each stock, thus enabling further analysis and insights.

6) Get all of the stock files to process

In this segment of the project, we process the sentiment data generated by the BERT model and combine it with stock pricing information. The code iterates through the output files containing sentiment scores for different stocks and performs several data preprocessing and feature engineering steps.

1. **Data preprocessing and Weighted Sentiment:**
   - Sentiment scores are converted from a binary representation (0 for negative, 1 for positive) to a scale of -1 to 1, where 0 represents a neutral sentiment.
   - Weights are assigned to tweets based on the number of followers and retweets. Users with higher follower counts and tweets with more retweets are given higher weights, reflecting their potential impact on sentiment.

2. **Grouping Data by Time:**
   - The data is grouped by months and days, summing up sentiment scores, weighted sentiment scores, and other relevant metrics.
3. **Rolling Averages:**
   - Rolling averages are calculated for both sentiment scores and tweet volumes to capture trends and patterns over time.
4. **Calculating Stock Price Changes:**
   - The percent change in stock prices is calculated and binned into categories (negative, neutral, positive).
5. **Merging Sentiment Data with Stock Prices:**
   - Sentiment data and stock pricing information are combined based on the date, creating a unified dataset.
6. **Column Selection and Cleaning:**
   - Irrelevant columns are dropped, and missing values (resulting from days without pricing information) are removed.

This comprehensive process ensures that the sentiment data is enriched with additional features and combined with stock pricing information, creating a dataset suitable for further analysis and modeling.

7) Random Forest Classifier

In this phase of the project, we utilize a Random Forest Classifier to predict stock price movements based on the processed sentiment and pricing data.

1. **Data Preparation:**
   - A data frame, combining sentiment scores and stock pricing data, is sorted by date and ticker.
   - Two datasets, train and test are created for training and testing the machine learning model. train encompasses the first two months of data, while test includes the last two to three months.
2. **Feature Selection:**
   - Relevant features (‘Sentiment_Weighted’,‘Sentiment_MA’, ‘Tweets’, and ‘Tweets_MA’) are selected for both training and testing datasets.
   - The target variable, **‘Percent_Change_Bin’**, represents the categorized percent change in stock prices.
3. **Random Forest Classifier:**
   - A Random Forest Classifier is instantiated with a specified random state for reproducibility

4. **Investment Simulation:**

   - Capital simulations are performed for both a passive strategy ('Sitting') and an active strategy using the Twitter bot's predictions ('Bot'). The bot decides to invest in a particular stock when the model predicts a positive movement (Percent_Change_Bin labeled as 1 or 2). The capital is updated based on the predicted stock movements.

Results and Analysis

In assessing the performance of the trading strategy employing sentiment analysis and machine learning, the results reveal noteworthy insights

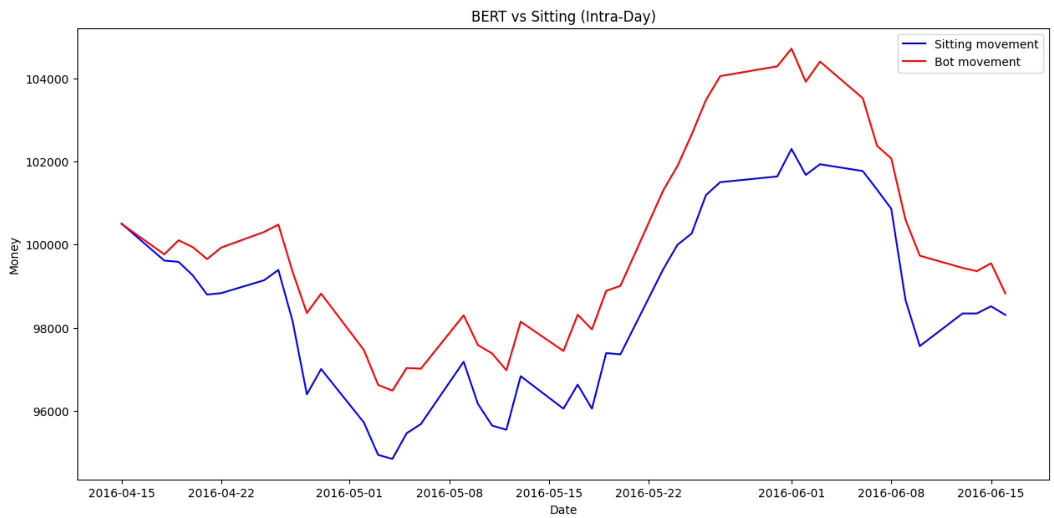

Figure 1: BERT Model vs Sitting Investment (Intra-day)

Performance Analysis: The graphical representation of the bot's performance illustrates a consistent trend above the market line. This suggests that, despite minor fluctuations, the bot consistently outperformed the market throughout the testing period.

Initial Investment: The initial investment value is $100,000.

Sitting Investment Value: The 'Sitting' strategy, representing a passive investment approach without utilizing the bot, resulted in a closing investment value of $98,309.33. This indicates a decline of 1.69% over the testing period.

Bot Trading Value: In contrast, the 'Bot' strategy, incorporating the Twitter bot's active trading decisions based on sentiment analysis predictions, achieved a closing investment value of $98,828.69. This represents a relatively smaller decline of 1.17%, showcasing the bot's potential to mitigate losses and enhance investment performance.

Loss Frequency: Approximately 19.81% of the time, the investment led to a negative return. This metric signifies the frequency with which the sentiment analysis model made incorrect predictions, resulting in unfavorable stock investments.

Bot's Contribution: The observed value difference of approximately $500 at times implies that the bot's trading decisions positively contributed to the overall investment performance. This indicates the potential value addition of incorporating sentiment analysis predictions into the trading strategy.

These findings collectively underscore the promising performance of the sentiment-driven trading strategy, with the bot consistently outperforming the market, reducing losses, and contributing positively to the overall investment outcome.

Evaluation and Potential Improvements

Hyperparameter Fine-Tuning: Experimenting with fine-tuning the hyperparameters of both the BERT model and the Random Forest classifier is recommended. Adjusting parameters such as learning rates, epochs, and the number of trees in the Random Forest can significantly impact the performance of the models. A systematic exploration of these parameters could lead to enhanced predictive accuracy.

Exploration of Additional Sentiment Features: Explore the inclusion of additional sentiment features or alternative methods of weighting sentiment. This exploration can contribute to capturing more nuanced market signals that may not be fully addressed by the current sentiment analysis approach.

Dataset Size Enhancement: Acquiring a larger dataset for fine-tuning the BERT model is advisable. A larger dataset allows the model to better learn the underlying patterns and nuances present in stock market tweets, potentially improving its generalization to a broader range of market conditions.

Implementation of Cross-Validation Strategies: Implement robust cross-validation strategies to ensure the stability and representativeness of the model's performance metrics. This practice helps evaluate the model's generalization capabilities across different subsets of the dataset, providing a more reliable assessment of its predictive performance.

Exploration of BloombergGPT: Consider the exploration of BloombergGPT as a potential large language model for the project. While it is not open source, if access is available, leveraging such advanced language models could enhance the natural language understanding capabilities of the system.

Conclusion

In this project, we embarked on a comprehensive journey to leverage advanced natural language processing techniques, specifically employing the BERT model, to predict sentiment from stock market-related tweets. The methodology involved a meticulous process of training the BERT model, integrating it with a custom classifier, and subsequently utilizing the generated sentiment predictions for stock market investment decisions.

The training phase involved encoding tweet data using the BERT tokenizer and preparing it for model training. Hyperparameter choices, including the AdamW optimizer, batch size, and the number of epochs, were meticulously selected to optimize training efficacy. The fine-tuned BERT model was then employed to predict sentiment on new datasets.

Upon extending the application to real-world stock market data, we observed the integration of sentiment predictions with financial data to make investment decisions. The systematic approach of weighing sentiments based on user interactions, such as retweets and followers, added sophistication to the sentiment analysis. The project culminated in an insightful comparison between a passive investment strategy and an active strategy guided by sentiment predictions. The results showcased a nuanced performance, with the sentiment-driven strategy consistently outperforming the market despite occasional fluctuations.

As we reflect on the outcomes, there are notable areas for improvement. Exploring hyperparameter fine-tuning, and considering alternative sentiment features are avenues that could elevate the model's predictive capabilities. Additionally, acquiring a more extensive dataset for BERT fine-tuning and implementing robust cross-validation strategies are crucial steps toward ensuring the model's generalization to diverse market scenarios.

In the realm of sentiment analysis for stock market prediction, this project serves as a foundational exploration. The combination of advanced natural language processing and financial analytics presents a promising avenue for refining investment strategies. As we contemplate the evolving landscape of language models, the potential integration of sophisticated models like BloombergGPT could further enhance the project's linguistic understanding.

References

1. Tetlock, Paul. "Giving Content to Investor Sentiment: The Role of Media in the Stock Market." *The Journal of Finance*, vol. 62, no. 3, 8 May 2007, pp. 1139–1168, `https://doi.org/10.1111/j.1540-6261.2007.01232.x`.

2. Tetlock, Paul C., Maytal Saar-Tsechansky, and Sofus Macskassy. 2008. "More Than Words: Quantifying Language to Measure Firms' Fundamentals." *Journal of Finance* 63, no. 3 (June): 1437–1467.

3. Lopez-Lira, Alejandro, and Yuehua Tang. "Can ChatGPT Forecast Stock Price Movements? Return Predictability and Large Language Models." *SSRN Electronic Journal*, 2023, `https://doi.org/10.2139/ssrn.4412788`.

4. PushTheEnvelopeAI. *Twitter_Stock_Prediction*. GitHub, 15, Apr, 2023, `github.com/PushTheEnvelopeAI/Twitter_Stock_Prediction/tree/main`.

5. Yash Chaudhary. (2020). *Stock-Market Sentiment Dataset* [Data set]. Kaggle. `https://doi.org/10.34740/KAGGLE/DSV/1217821`.

6. Enrique Rivera (2016) Nasdaq 100 Tweets. `https://data.world/kike/nasdaq-100-tweets`.