GPT-Based Sentiment Analysis for Predicting Dow Jones Trends

Contributors: Mateo Velarde, Hubery Jiarui Hu, Himal Pandey

1. Introduction

This documentation provides a detailed overview of our deep learning project that leverages GPT-based sentiment analysis to predict trends in the Dow Jones Industrial Average (DJIA). By using historical news headlines and stock prices in the training process, we aim to establish a correlation between public sentiment and stock market movements.

2. Previous Solutions

Bollen, Mao, and Zeng (2011) investigated the correlation between public mood states, as derived from large-scale Twitter feeds, and the Dow Jones Industrial Average (DJIA). By utilizing mood tracking tools like OpinionFinder and GPOMS, they demonstrated that specific mood dimensions, particularly “Calm,” significantly improved the accuracy of predicting DJIA movements, achieving a prediction accuracy of 86.7%. This study underscores the potential of using collective sentiment analysis to forecast stock market trends. Similarly, Li et al. (2014) examined the impact of financial news sentiment on stock price returns. By leveraging sentiment dictionaries to quantify news sentiment, they found that models incorporating sentiment analysis outperformed traditional bag-of-words models across various market levels. These foundational studies highlight the effectiveness of sentiment analysis in financial predictions, laying the groundwork for our project, which integrates GPT-based sentiment analysis to predict DJIA trends.

3. Dataset

The project utilizes two primary data sources (sourced from Kaggle - Stock Market Predictions[1]):

- News Data: Sourced from historical headlines crawled from the Reddit WorldNews Channel (/r/worldnews), covering the period from June 8, 2008, to July 1, 2016. The dataset includes the top 25 headlines for each date, ranked by Reddit users' votes.
- Stock Data: The Dow Jones Industrial Average metrics, sourced from Yahoo Finance for the same dates as the news data. For each day, our dataset contains: Date, Open, High, Low, Close, Volume, Adj Close

4. Proposed Method

Data Preprocessing

This code loads and processes DJIA and news headline datasets, merging them by date into a single string and further labeling data for longer-term trend predictions based on stock price changes. The labeling is derived from numerical data for the Dow Jones in addition to the original headline dataset. The label_trend function initializes labels to 0 and calculates short-term and long-term changes in stock prices. If both changes are positive, it labels an upward trend (1); if both are negative, a downward trend (0); otherwise, no clear trend (2). The processed DataFrame is saved to a new CSV file.

[1] https://www.kaggle.com/datasets/tanishqdublish/stock-market-predictions

The project defines a NewsDataset class for handling news headlines, labels, and optional stock price data. It uses a tokenizer to encode headlines and price information into input tensors suitable for a PyTorch model. The class includes methods to get the dataset length and to retrieve and preprocess individual items, ensuring proper tokenization and padding.

Model Implementation[2]

First Model:

We leverage the Hugging Face Transformers library to implement a GPT-2 model for sequence classification. The model is fine-tuned on the dataset to predict stock market trends based on the sentiment of news headlines.

Second Model:

We leverage the Hugging Face Transformers library to implement a bert model for sequence classification. Additionally, we integrate DJIA price information directly into the training process, incorporating both long-term and short-term trends to enhance predictive accuracy.

Training and Evaluation

- **Model Training:** The model is trained on a split dataset using the AdamW optimizer and a learning rate scheduler.
- **Evaluation Metrics:** Accuracy and classification report metrics are used to evaluate model performance. We also generate confusion matrices and ROC curves for a detailed performance analysis.

5. Evaluation Method

Metrics

- **Accuracy:** Measures the overall correctness of the model. It is calculated as the ratio of correctly predicted instances to the total number of instances. This metric provides a general overview of the model's performance but does not account for class imbalances.
- **Precision, Recall, and F1-Score:**
    - Precision: We look at the ratio of true positive predictions to the sum of true positive and false positive predictions. Precision here indicates how many of the predicted positive instances are actually positive.
    - Recall: We look at the ratio of true positive predictions to the sum of true positive and false negative predictions. Recall indicates how many of the actual positive instances were correctly identified by the model.
    - F1-Score: We then look at the harmonic mean of precision and recall, providing a single metric that balances both. This will be especially useful for us if there is an uneven class distribution.
- **Confusion Matrix:** We generate a table that visualizes the performance of the model by showing the true versus predicted classifications. It provides detailed insight into the types of errors the model makes and helps identify which classes are being misclassified. The confusion matrix is then visualized using a seaborn heatmap to provide a graphical representation of the model's performance across different classes.
- **ROC Curve and AUC:**

[2] Provided by OpenAI's Transformers

- ROC Curve: We plot the true positive rate against the false positive rate at various threshold settings. This provides a visual representation of the model's performance across different thresholds.
- AUC (Area Under the Curve): We measure the area underneath the ROC curve. A higher AUC indicates a better-performing model.

Misclassification Analysis

We conduct a detailed analysis of misclassified examples to understand the model's limitations and identify areas for improvement. This involves examining the instances where the model's predictions were incorrect, allowing us to know why and how the model might have failed. It sets the model to evaluation mode, disables gradient calculation, and iterates through a data loader. For each batch, it computes predictions and compares them to the actual labels. If a prediction does not match the label, the text, predicted label, and actual label are saved. The function returns a list of these misclassified examples.

6. Results and Discussion

Previous Model

The initial implementation of our sentiment analysis model utilized a GPT-2 architecture and was trained with a standard training loop incorporating early stopping to prevent overfitting. During training, the model's accuracy improved from 55.37% to 99.18% by the 10th epoch, indicating effective learning from the training data. However, the validation accuracy peaked at around 50%, suggesting significant overfitting and poor generalization to unseen data. The balanced accuracy score on the validation set was 50.83%, reflecting the model's difficulty in equally classifying positive and negative sentiments. Additionally, the confusion matrix and classification report revealed frequent misclassifications of stock price increases as decreases, likely due to the model's failure to account for the general upward trend in stock indices during the training period.

Further Implementation Model

Training Accuracy:

- The model achieved a training accuracy of 80.34% by the 10th epoch. This indicates that the model was able to effectively learn and generalize from the training data over the course of the training process.

Validation Accuracy:

- The validation accuracy improved significantly, reaching 81.25% by the 10th epoch. This suggests that the model performs well not only on the training data but also on unseen data, indicating good generalization capabilities.

Confusion Matrix:

The confusion matrix provides a detailed insight into the performance of the model by comparing the true versus predicted classifications. The matrix reveals that the model is particularly proficient at predicting 'Downward' trends, reflecting a strong negative sentiment captured in the news headlines. Specifically, out of 842 actual 'Downward' instances, the model correctly predicted 841 and misclassified only 1 as 'Upward'. However, the model struggled more with the 'Upward' and 'No Clear Trend' categories:

- For ‘Upward’ trends, out of 472 actual instances, the model correctly predicted 169 but misclassified 59 as ‘Downward’ and 244 as ‘No Clear Trend’.
- For ‘No Clear Trend’, out of 675 actual instances, the model correctly predicted 665, misclassified 10 as ‘Upward’, and none as ‘Downward’.

This analysis indicates that while the model excels in identifying ‘Downward’ trends, it faces challenges in distinguishing ‘Upward’ trends, often confusing them with ‘No Clear Trend’.

Our model demonstrated high precision in predicting downward market trends (with high precision and recall, as reflected in the confusion matrix and the ROC curve with an AUC of 1.00), which suggests that negative sentiment in news headlines is strongly correlated with downward movements in the DJIA. This could be due to the fact that negative news tends to have a more immediate and pronounced impact on market sentiment and investor behavior.

The model’s lower precision and recall for upward trends (as evidenced by the lower AUC of 0.92 and confusion matrix analysis) suggest that positive sentiment is less clearly defined in the news data. Positive news might be more ambiguous or less frequent, making it harder for the model to identify and predict upward trends accurately. Additionally, upward trends in the stock market may be influenced by a wider range of factors beyond just news sentiment, such as economic indicators, market trends, and investor expectations.

Despite the challenges with specific classes, the overall accuracy and macro-averaged metrics indicate a balanced performance across all classes. This suggests that while there are areas for improvement, the model is reasonably robust and can provide useful predictions across different market conditions.

7. Future Work

We can use explore the use of additional data and hyperparameter adjustments to optimize model performance. This could include experimenting with different model architectures, tuning the learning rate, adjusting the batch size, and incorporating more advanced techniques like transfer learning or ensemble methods.

We can also include other financial indicators and alternative data sources, such as social media sentiment, economic reports, and financial news articles, which could further enrich the model’s input. By incorporating a wider range of features, we can capture a more comprehensive view of the factors influencing stock market trends and improve the model’s predictive accuracy.

This project showcases the potential of GPT-based sentiment analysis in financial prediction, offering a novel approach to understanding and forecasting market dynamics. Continued advancements in natural language processing and machine learning will further enhance the accuracy and reliability of such models, paving the way for more sophisticated financial analysis tools.

8. References

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. Journal of Computational Science.

Li, X., et al. (2014). News impact on stock price return via sentiment analysis. Knowledge-Based Systems.