Sentiment Analysis Techniques Comparison for Stock Prediction

Abstract

This document compares four sentiment analysis techniques—ProsusAI/Finbert, FinBERT-tone, TextBlob, and VADER. applied to a financial dataset for stock prediction. After applying all models on the dataset, an aggregate function is created to infer a final label which is then used to determine the model with the best overall accuracy. The study evaluates the models based on performance metrics, computational resources, interpretability, and generalization to provide insights for choosing the most suitable technique.

Dataset Description

A dataset of financial news was collected for several stocks/companies in various sectors to evaluate the role of sentiment analysis in predicting the stock market. the stocks are: 'PG', 'META', 'AMD', 'NFLX', 'TSM', 'AAPL', 'GOOGL', 'MSFT', 'AMZN', 'TSLA',.

The dataset contains more than 20000 news entries for all the companies. A news entry can be either a news article or a news video to make sure more news is covered.

The dataset has 8 columns as follows:

**ticker**: indicates the stock symbol e.g. AAPL for Apple.

**date**: the exact date the news was published.

**title**: the title of the news article/video.

**text**: description of the news story.

**source**: what channel or news agency published the story?

**url**: news URL.

**type**: whether the news entry is an article or a video.

**sentiment**: the sentiment label of that news story i.e. Positive, Neutral, or Negative.

3. Methodology

­­The process for applying the four techniques was identical to be able to evaluate their performance against each other and pick one model for our research.

All four implementations followed this structure:

Data preprocessing:

* Data Cleaning:
  + Handle Missing Values:
    - Identify and handle any missing values in the dataset.
  + Remove Duplicates:
    - Check for and remove any duplicate records in the dataset.
* Text Cleaning:
  + Lowercasing:
    - Convert all text to lowercase to ensure consistency.
  + Remove Punctuation:
    - Eliminate unnecessary punctuation from the text data.
  + Tokenization:
    - Tokenize the text into individual words or phrases.
  + Remove Stop Words:
    - Exclude common stop words (e.g., "the," "and," "is") that may not contribute significantly to sentiment analysis.
  + Lemmatization:
    - Lemmatizing each token by returning each word/verb to its root word.
* Concatenating two columns:
  + To have more text to analyse the sentiment, we concatenated the “text” column with the “title” column to have more words, thus, getting a more accurate classification of the sentiment.

3.1 FinBERT-tone

* Sentiment Analysis Function:
  + The methodology involves the utilization of a pre-trained FinBERT model. The model is loaded, and a sentiment analysis function is introduced. This function leverages FinBERT to predict sentiment labels for each processed text.
* Applying FinBERT Sentiment Analysis to the Dataset:
  + The sentiment prediction function is applied to each pre-processed text in the dataset. This results in a new column named finbert\_label containing the predicted sentiment labels.

3.4 ProsusAI/finbert

* + similar to FinBERT-tone

3.3 TextBlob

The analysis begins by importing the TextBlob library.

* Defining Sentiment Analysis Functions:
  + get\_polarity(): Measures the positivity or negativity of the text with a score between [-1,1]
  + get\_subjectivity(): measures the subjectiveness of the text.
* Applying Sentiment Analysis to the Dataset:
  + These functions are then applied to the pre-processed text in the dataset, generating numerical values for polarity and subjectivity.
* Defining Labeling Function to simplify interpretation, a function named get\_label is introduced. This function categorizes the sentiment based on the calculated polarity:
  + Positive for values greater than 0.
  + Negative for values less than 0.
  + Neutral for values equal to 0.

3.4 VADER

* VADER Sentiment Analysis Function:
  + This function uses the SentimentIntensityAnalyzer from the NLTK library. It assesses the sentiment of each processed text, generating a compound score that indicates the positivity and negativity of the text sentiment.
* Labelling Function:
  + This function assigns sentiment labels ("Positive," "Negative," or "Neutral") based on the compound score.
* Creating Sentiment Labels:
  + The labelling function is then applied to categorize the sentiment of each text based on the computed compound scores. The resulting sentiment labels are stored in a new column named VADER\_label in the DataFrame.

3.5 Aggregate function

* After obtaining sentiment labels from multiple models—VADER, TextBlob, FinBERT, and ProsusAI/finbert—an aggregation process is introduced to consolidate these labels into a single sentiment label for each text.
* Aggregation Function Overview:
* The methodology involves defining an aggregation function named aggregate\_labels(). This function takes a row of sentiment labels generated by different models and determines the final sentiment label through a majority voting mechanism.
* The aggregate\_labels function is applied row-wise to the dataset. For each row, sentiment labels from all models are collected, and a Counter is used to count the occurrences of each label.
* The final sentiment label is determined by selecting the label with the highest count, indicating the most common sentiment label among the considered models.
* The resulting final sentiment labels are stored in a new column named final\_label in the dataset.

3.6 Performance Evaluation Methodology

* To evaluate the effectiveness of the sentiment analysis models, a consistent mechanism is applied to each model. It includes metrics such as accuracy, a confusion matrix, and a classification report.
* The final sentiment labels obtained through the aggregation process are compared with the labels generated by each model separately.
* The accuracy metric is used to measure the overall correctness of sentiment predictions.
* The confusion matrix provides a detailed breakdown of the true positive, true negative, false positive, and false negative predictions.
* The classification report presents precision, recall, F1-score, and support metrics for each sentiment category.

4. Evaluation Metrics

The evaluation metrics for comparing the models will be:

accuracy: the ratio of correctly predicted instances to the total instances.

precision: the ratio of correctly predicted positive instances to the total predicted positive.

recall: the ratio of correctly predicted positives to the total observations in the class.

F1-score: the weighted average of precision and recall.

Confusion Matrix: a table illustrating the counts of true positives, true negatives, false positives, and false negatives.

5. Model Performance

5.1 FinBERT

A screenshot of a computer screen

Description automatically generated

A comparison of a bar chart

Description automatically generated

5. Model Performance

5.2 ProsusAI/FinBERT

A screenshot of a computer screen

Description automatically generated

A comparison of a bar chart

Description automatically generated with medium confidence

5.3 TextBlob

A screenshot of a computer program

Description automatically generatedA comparison of a bar chart

Description automatically generated

5.4 VADER

A screenshot of a computer screen

Description automatically generated

A comparison of a bar chart

Description automatically generated

6. Analysis and Conclusion

Performance of each model when applied to the full dataset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1-Score |
| FinBERT | 0.64 | 0.74 | 0.64 | 0.65 |
| ProsusAI/Finbert | 0.55 | 0.72 | 0.55 | 0.53 |
| TextBlob | 0.70 | 0.69 | 0.70 | 0.69 |
| VADER | 0.78 | 0.81 | 0.78 | 0.76 |

Time taken for each model to analyze the entire dataset.

|  |  |
| --- | --- |
| Model | Time Taken |
| FinBERT | 21m 39s |
| ProsusAI/Finbert | 23m 15.4s |
| TextBlob | 3.3s |
| VADER | 3.7s |

After a comprehensive evaluation of various sentiment analysis techniques including FinBERT, ProsusAI/FinBERT, TextBlob, and VADER. we conclude the following

6.1 Model Performance

The performance metrics provide valuable insights into the effectiveness of each model. While each technique exhibited strengths and weaknesses, our decision to choose VADER is rooted in a careful consideration of several factors.

6.2 Factors Influencing the Choice of VADER

Ease of Use and Interpretability:

VADER, being a rule-based model, is known for its simplicity and ease of use. It doesn't require domain-specific training and provides a straightforward approach to sentiment analysis. This simplicity contributes to its high interpretability.

Computational Efficiency:

Our analysis included the consideration of computational resources. VADER demonstrated great efficiency, requiring only a fraction of the time taken by more complex models such as FinBERT and ProsusAI/FinBERT. This efficiency is particularly valuable when dealing with large datasets like the one we used for this comparison.

Generalization to Financial Domain:

While FinBERT and ProsusAI/FinBERT are tailored for financial sentiment analysis, VADER showcased competitive performance in our dataset. Its ability to generalize well to financial language.

Aggregation Results:

The aggregation function, which combined sentiment labels from all models, highlighted that VADER often contributed significantly to the final sentiment label. Its labels were frequently aligned with the majority votes, making it reliable for our research.

6.3 Why Other Models Were Not Selected

FinBERT and ProsusAI/FinBERT:

While FinBERT and its ProsusAI adaptation demonstrated competitive accuracy (FinBERT Accuracy: 0.64, ProsusAI/FinBERT Accuracy: 0.55), their computational demands and longer processing times limit their practicality, especially in real-time applications. Additionally, their performance did not significantly outshine VADER, considering the added complexity and resource requirements.

TextBlob:

TextBlob, although it was also computationally efficient, it showed slightly lower accuracy compared to VADER (TextBlob Accuracy: 0.70).

In conclusion, the decision to choose VADER is guided by a balance between performance, ease of use, and computational efficiency.