Financial Sentiment Analysis using LSTM

Dr. David Raj Micheal
Department of Mathematics, School of Advanced Sciences
Vellore Institute of Technology Chennai
Tamil Nadu – 600127
davidraj.micheal@vit.ac.in

Vellengara Adithi Sunil
Department of Mathematics, School of Advanced Sciences
Vellore Institute of Technology Chennai
Tamil Nadu – 600127
vellengara.adithi2023@vitstudent.ac.in

1 Introduction

Sentiment analysis has become a key tool in financial markets for predicting market movements, understanding investor behavior, and enhancing decision-making processes. By analyzing financial texts—such as news articles, social media posts, and corporate reports—sentiment analysis can classify the sentiment expressed in the text as positive, negative, or neutral, providing insights into how the market might react. Research has demonstrated that sentiment analysis can provide valuable predictions in stock market trends and investor behavior [1], with financial news content playing a significant role in price movements [2]. This report presents a system that uses deep learning techniques, specifically an LSTM (Long Short-Term Memory) network, to perform sentiment analysis on financial data.

2 Objective and Problem Statement

2.1 Objective

The objective of this study is to build and evaluate an LSTM-based model to classify financial texts into sentiment categories (positive, neutral, and negative). By using a deep learning approach, the model aims to accurately predict sentiment from financial text data, helping financial analysts and investors gauge market sentiment more effectively. The ultimate goal is to enhance financial decision-making and contribute to better market predictions by leveraging sentiment data.

2.2 Problem Statement

In the rapidly changing financial markets, the need for timely and accurate sentiment analysis is critical. Many existing sentiment analysis models rely on traditional machine learning techniques, which can be limited by their ability to capture the complex temporal dependencies present in text data. This limitation is particularly apparent when processing financial data, which is often jargon-heavy, context-sensitive, and subject to rapid changes in sentiment. While LSTM models have shown promise in capturing these complex patterns due to their ability to remember long-term dependencies, there remains a challenge in accurately classifying negative sentiment in financial text. The problem addressed in this report is to build an LSTM-based model that can classify financial sentiment accurately, particularly focusing on improving the classification of negative sentiment, which often gets overshadowed by more frequent neutral and positive sentiments.

3 Methodology

The methodology for this sentiment analysis system involves several key steps:

1. **Data Loading and Preprocessing**: The financial dataset is loaded from a CSV file containing text and sentiment labels. Preprocessing includes cleaning the text by removing special characters, converting it to lowercase, and trimming extra spaces. Tokenization is then performed to convert the text into numerical sequences suitable for deep learning input.

2. **Text Tokenization and Padding**: Keras' `Tokenizer` is used to tokenize the text, converting it into sequences of integers. These sequences

are padded to ensure uniformity in input length, allowing the LSTM model to handle the data efficiently.

3. **Sentiment Encoding**: Sentiment labels—such as "positive", "neutral", and "negative"—are encoded using Scikit-learn's `LabelEncoder` to convert the categorical sentiment values into numerical representations.

4. **Model Architecture**: The LSTM model is structured with:

   - An `Embedding` layer to convert tokenized sequences into dense vectors.
   - Two `LSTM` layers to capture temporal dependencies in the text data.
   - A `Dropout` layer to reduce overfitting.
   - A `Dense` layer with a softmax activation function to output one of the three sentiment classes.

5. **Training and Evaluation**: The model is trained over five epochs using the Adam optimizer and sparse categorical cross-entropy loss. The model's performance is evaluated using metrics such as accuracy, precision, recall, F1-score, and confusion matrix.

4 Results

The model achieved the following results after training for five epochs:

- **Accuracy**: 69.97% on the validation set.
- **Classification Report**:
  - **Class 0 (Negative sentiment)**: Precision: 0.35, Recall: 0.22, F1-score: 0.27
  - **Class 1 (Neutral sentiment)**: Precision: 0.73, Recall: 0.82, F1-score: 0.77
  - **Class 2 (Positive sentiment)**: Precision: 0.75, Recall: 0.73, F1-score: 0.74

  The overall weighted average F1-score is 0.69, indicating moderate performance.

- **Confusion Matrix**: A heatmap of the confusion matrix is shown below, which highlights that the model struggles with accurately classifying negative sentiment. This results in lower performance in predicting Class 0.

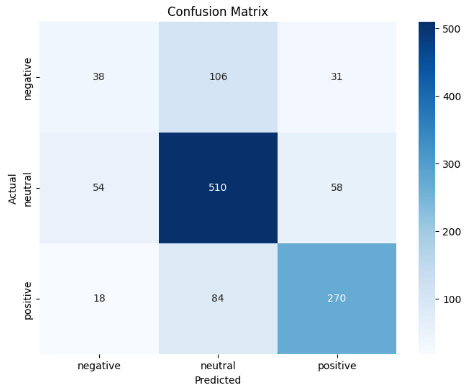

Figure 1: Confusion Matrix of the Sentiment Analysis Model

5 Model Performance Analysis

The model performed well in classifying neutral and positive sentiments, with relatively high precision and recall for these categories. However, the model faced challenges in accurately classifying negative sentiment. This could be attributed to class imbalance in the dataset, where negative sentiment examples are less frequent than neutral and positive ones. Addressing this imbalance could enhance the model's ability to detect negative sentiment more accurately.

6 Improvements and Future Work

Several strategies could be employed to improve model performance:

1. **Class Imbalance Handling**: Techniques like oversampling the minority class or adjusting class weights during training can help balance performance across all sentiment categories.

2. **Hyperparameter Tuning**: Experimenting with different hyperparameters, such as the number of LSTM units and dropout rates, may lead to better model performance.

3. **Data Augmentation**: Generating additional samples for the under-represented classes can improve model generalization.

4. **Advanced Architectures**: Exploring advanced architectures, such as Transformer-based models like BERT, could help in capturing more nuanced relationships in the text.

7 Conclusion

Financial sentiment analysis is a valuable tool for understanding market trends and assisting in financial decision-making. The LSTM-based model presented in this report demonstrated moderate success, particularly in classifying neutral and positive sentiments. However, there is room for improvement, especially in the classification of negative sentiments. Future work will focus on addressing these limitations, optimizing the model, and exploring advanced techniques to enhance the predictive accuracy and reliability of financial sentiment analysis.

8 References

References

[1] Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1), 1-8. https://doi.org/10.1016/j.jocs.2010.12.007

[2] Das, S. R., & Chen, M. Y. (2007). Yahoo! for Amazon: Sentiment extraction from small talk on the web. *Management Science*, 53(9), 1375-1388. https://doi.org/10.1287/mnsc.1060.0708

[3] Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1), 1-167. https://doi.org/10.2200/S00416ED1V01Y201204HLT016

[4] Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *Journal of Finance*, 62(3), 1139-1168. https://doi.org/10.1111/j.1540-6261.2007.01232.x