Stock Behavior Using RNN Models and Transformer-Based Text Analysis.

Diego Vallarino *Independent Researcher,* Atlanta, GA, USA. September 2024

Abstract

Predicting stock prices is essential in financial markets, where anticipating future movements based on historical data and relevant events can significantly impact trading strategies. This study presents a methodology for forecasting Apple Inc.'s (AAPL) stock price by combining Recurrent Neural Networks (RNN) and Transformer models for text analysis. The objective is to be able to incorporate a proxy of the agents' expectations regarding the share price. By leveraging both numerical and textual data, this approach enhances predictive performance and provides a comprehensive understanding of market dynamics.

Key words and phrases. Asset Pricing; Trading Volume, Financial Forecasting, Forecasting Models, Neural Networks and Deep Learning

Introduction

Stock price prediction poses challenges due to complex financial dynamics and high levels of uncertainty. While traditional models primarily use statistical techniques, they often struggle to capture intricate patterns in price movements. The advent of machine learning has enabled the use of more sophisticated models, such as Recurrent Neural Networks (RNNs) and Transformers, for financial forecasting. This paper explores using an RNN model to predict Apple Inc.'s stock price, integrating a Transformer model to evaluate how daily news impacts stock behavior. The approach combines historical stock data and textual information, aiming to improve prediction accuracy and offer deeper insights into market trends.

Methodology

The methodology involved several stages, beginning with feature generation using the Transformer model. Transformer analyzed daily news articles related to Apple Inc., extracting features such as *Sentiment Score*, *Event Type* (e.g., Product Launch), *Polarity*, and *Relevance Score*. This step aimed to capture relevant information from the text to enrich the dataset used for analysis, thereby helping to understand the impact of news on stock prices. Libraries like *torch* provided tools for creating the Transformer model, while *tibble* facilitated data manipulation and structuring. Functions such as *generate_simulated_data()* and *simulate_embeddings()* were employed to create sample embeddings for textual data processing.

Once the features were generated, they were aligned with daily financial data to ensure temporal coherence between the textual characteristics and financial information, including *Closing Price*, *Volume*, and *High/Low Prices*. To do that, we use *quantmod* library. This alignment helped form a consistent dataset where each day contained both numerical and text-derived features. Libraries like *dplyr* and functions like *merge()* were used for data alignment.

The enriched inputs were then prepared by concatenating the features extracted from the Transformer with financial data, forming the input for the RNN model. This step was intended to provide a comprehensive input for the model, combining historical price data and textual analysis to improve predictive performance. The *cbind()* function was utilized to merge these data sources.

The RNN model, built using an LSTM architecture, was trained with enriched inputs. Leveraging both historical stock data and features derived from the Transformer, the model was configured to predict future stock prices. The training process used the *keras* library[1] to define the layers of the neural network, including an LSTM layer and a dense output layer. The model was compiled with the *Adam* optimizer and

[1] *Por qué con Keras: utiliza Adam para la optimización automática de parámetros, incluye capas preconfiguradas como LSTM para simplificar el diseño de redes, y ofrece escalabilidad eficiente para implementación en servidores o en la nube.*

Mean Squared Error (MSE) as the loss function, with functions like *compile()* and *fit()* facilitating the training process.[2]

To evaluate the model, predictions were generated on a test set, measuring how well the incorporation of additional features improved accuracy. The model's performance was assessed by comparing its predictions to actual stock prices, with MSE used as the primary evaluation metric. Visualizations created with the *plotly* library helped compare the predicted and actual values, providing a clear view of the model's accuracy.

Finally, the trained model was used to make future predictions, considering both the historical market trends and the influence of daily news. The enriched features allowed the model to capture market dynamics more effectively, resulting in more accurate forecasts. Functions like *prepare_input_data()* and *predict_future_prices()* were employed to generate these predictions.

Empirical Results

The RNN model demonstrated effective predictive capabilities, with an actual closed price of $218 on July 27th 2024 where the model estimated the stock price at $217 (using a walk-forward validation approach). This result highlights the model's robustness for short-term forecasting. The inclusion of features derived from the Transformer, such as *Sentiment Score*, *Event Type*, *Polarity*, and *Relevance Score*, contributed to a noticeable improvement in prediction accuracy compared to traditional methods. This comparison demonstrated the enhanced performance in analyzing agents' expectations, highlighting the added value of incorporating sentiment and event-based features from the Transformer model.

Visualizations generated using *plotly* illustrated the model's ability to align with real stock trends while incorporating textual data, providing a more comprehensive understanding of market behavior.

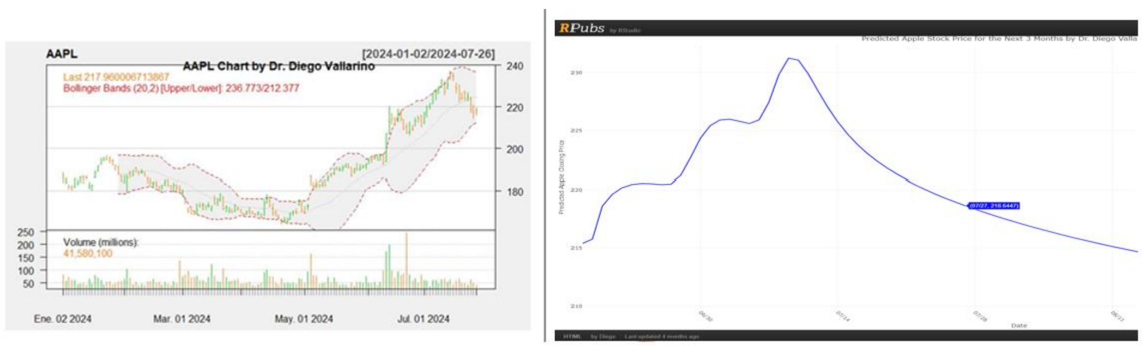

Figure 1: *AAPL Stock Price on July 27th (left chart with quantmod), with the Value Estimated 30 Days Prior (right chart with plotly).*

The model's accuracy was assessed using Mean Squared Error (MSE), with the RNN based on LSTM (price data only) achieving an MSE of 0.0021, while the RNN + Transformer (Sentiment/Expectations) model improved performance with an MSE of 0.0015. *The improvement in the MSE was approximately 28.57%, comparing the RNN model with the RNN + Transformer model*.

Conclusions

The integration of RNN and Transformer models offers a robust approach to stock price prediction. The RNN effectively utilized numerical data, including *Closing Price*, *Volume*, and *High/Low Prices*, while the Transformer incorporated valuable insights from daily news articles. This hybrid methodology not only enhances the precision of stock predictions but also enriches the understanding of factors influencing market behavior. The study underscores the importance of combining different modeling techniques to tackle complex financial forecasting tasks, providing a solid foundation for further exploration in financial analytics.

[2] *Las RNN permiten considerar relaciones no lineales, a diferencia de ARIMA o ARIMAX o GARCH*