Enhancing Sentiment Analysis Models for Cryptocurrency Discourse: A Study of LSTM and Transformer Architectures

Baoying Feng
Department of Computer Science
University of Arizona
fengb1@arizona.edu

Chunlin Huang
Department of Electrical
Computer Engineering
University of Arizona
chunlinhuang@arizona.edu

Abstract

This study explores sentiment analysis on Bitcoin-related tweets using two advanced neural network models: LSTM with attention and a Transformer model. Employing distinct tokenization methods, the Enhanced LSTM model focuses on model-specific features, while the Transformer leverages a pre-trained BERT tokenizer to handle complex linguistic patterns.

1 Introduction

In the rapidly evolving field of financial technology, social media platforms such as Twitter significantly influence market behaviors through the public's real-time sentiments, especially concerning cryptocurrencies like Bitcoin. This project seeks to decipher the underlying sentiments in Twitter discussions about Bitcoin, categorizing them into positive and negative emotions, to provide insights into market trends and investor behavior. Utilizing sophisticated machine learning techniques, we employ two distinct approaches: an Enhanced Long Short-Term Memory (LSTM) network with an attention mechanism and a Transformer model using a pre-trained BERT tokenizer. This research not only advances academic understanding but also offers practical insights for financial analysts and cryptocurrency enthusiasts, comparing different neural network architectures' efficacy in processing the noisy and unstructured data typical of social media.

2 Background

Initially, our project aimed to develop an advanced program for analyzing sentiments in Twitter posts using Natural Language Processing (NLP) techniques. We planned to utilize a combination of contemporary and historical datasets—one from Kaggle and another compiled via a web crawler—to categorize sentiments into emotions like positive, negative, and neutral, while also examining temporal shifts in sentiment expression. This dual-dataset approach was intended to provide insights into how public sentiment on Twitter evolves, reflecting societal trends and the emotional impact of events over time. However, we faced significant challenges in data collection due to limitations with the free version of the Twitter API and other free web crawler libraries such as Twint, Snscrape, and NTScraper, which were blocked by Twitter. This limitation hindered our ability to gather recent tweets, forcing us to rely on available datasets focused on Bitcoin-related posts for our experiments.

The choice of LSTM and Transformer models was strategic, driven by their proven capabilities in handling complex NLP tasks. The LSTM model, with its ability to process sequences and remember long-term dependencies, was ideal for analyzing the flow and evolution of sentiments within tweets. Its inherent nature allows it to capture temporal dynamics, which is crucial for understanding how

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

sentiments change over time. We enhanced the LSTM with an attention mechanism to improve its focus on pertinent segments of text, thereby enhancing its ability to discern relevant emotional cues. Conversely, the Transformer model was selected for its superior handling of large-scale language data, facilitated by its self-attention mechanism that evaluates the weights of all parts of the input data simultaneously. This feature makes the Transformer especially effective in dealing with the varied linguistic patterns found in Twitter data, which often includes colloquial language, hashtags, and abbreviations. Both models were considered to address the unique challenges presented by the noisy, informal nature of social media text, aiming to robustly analyze and interpret the sentiments expressed in Bitcoin-related tweets.

3 Methdology

To address the project's objective of categorizing sentiments in Bitcoin-related tweets into positive and negative emotions, we employed two distinct neural network models: an Enhanced LSTM with Attention and a Transformer model, each tailored for their unique capabilities in text processing.

3.1 Data Preprocessing

Prior to feeding the data into these models, significant preprocessing was required to ensure the quality and uniformity of the input data. Our preprocessing pipeline involved a series of functions designed to clean the tweets by removing URLs, hashtags, and mentions, which are common but often irrelevant elements in the textual data from Twitter. This was achieved using regular expressions: URLs were stripped from tweets, hashtags were removed, and mentions were eliminated, followed by the removal of all non-alphabetical characters to standardize the text. Additionally, the tweets were trimmed of leading and trailing whitespace to tidy up the dataset.

Another aspect of our preprocessing involved handling the timestamps associated with each tweet. The dates were converted to a standard datetime format and any erroneous entries were discarded, ensuring that our analyses could reliably segment data based on precise time frames.

For the dataset that lacked pre-labeled sentiments, we utilized the TextBlob library to conduct a preliminary sentiment analysis. TextBlob provided a simple interface to obtain polarity scores, which we used to assign a sentiment label to each tweet. Tweets with a positive polarity were labeled as 'Positive', while those with negative polarity were labeled as 'Negative'. This step was crucial for preparing the second dataset for comparative analysis alongside the labeled dataset.

3.2 LSTM

In our sentiment analysis project, the LSTM (Long Short-Term Memory) model was a cornerstone due to its ability to process sequences and remember information over extended periods, making it ideal for text data where context and sequence matter. We started with a basic LSTM model, using the

```
torchtext.data.utils.get_tokenizer("basic_english")
```

to tokenize the input data into a manageable form. This tokenizer simply splits the text into words based on spaces and punctuation and is ideal for establishing a baseline for more complex operations.

To improve the model's ability to focus on significant textual features and enhance its prediction accuracy, we integrated an attention mechanism into our LSTM architecture. This Enhanced LSTM with Attention allowed the model to weigh different parts of the input data differently, focusing more on segments that are crucial for understanding the overall sentiment. Specifically, the attention mechanism calculated a weighted sum of all hidden states of the LSTM, assigning more importance to words that are more relevant to the sentiment being expressed. This adaptive focusing capability is particularly useful in sentiment analysis, where certain keywords or phrases can have a strong indication of positive or negative sentiment.

Initially, the basic configurations of our LSTM model showed promising yet suboptimal results, with an accuracy around 54%. This prompted a series of experiments with various hyperparameters, including the number of LSTM layers (testing 0, 2, 5, and 8 layers) and adjustments in the frequency parameter for vocabulary building from 5 to 10, aiming to concentrate the model's learning on more relevant terms. The most effective configuration turned out to be a three-layer model run across nine training epochs, which improved accuracy to approximately 68%.

Hyperparameter	Value
Batch Size	512
Learning Rate (lr)	0.0005
Epochs	30
Optimizer	Adam
Loss Function	CrossEntropyLoss
Train-Test Split Ratio	80% train, 20% test
Validation Split (from train)	25%
Embedding Dimension	300
Hidden Dimension	300
Number of LSTM Layers	5
Bidirectional	True
Dropout Rate	0.5
Vocabulary Size	Length of vocab

Further refining the model, we integrated GLoVe word embeddings to provide a richer pre-trained vector representation for words, capturing deep semantic and syntactic meanings that significantly boosted the model's linguistic understanding. This enhancement helped lift the model's validation accuracy from about 70% to 72%.

Despite these advancements, the Enhanced LSTM model's accuracy plateaued around 72%. The breakthrough came when we decided to remove neutral sentiment data from our training dataset, significantly reducing ambiguity in sentiment classification. This strategic adjustment allowed the model to focus more effectively on distinguishing between positive and negative sentiments, catapulting the accuracy to an impressive 98.2%.

3.3 Transformer

In parallel to the LSTM model, we employed the Transformer model, leveraging the architecture of the pre-trained bert-base-uncased from the transformers library. This model was chosen for its superior capabilities in handling complex linguistic patterns, which are prevalent in the informal and dynamically evolving language found on Twitter. The Transformer distinguishes itself with a self-attention mechanism that processes all words in the dataset concurrently, rather than sequentially. This feature allows it to evaluate the context of each word relative to every other word in a sentence, providing a significant advantage for understanding the nuanced and often context-dependent meanings of language used in social media.

The bert-base-uncased model is particularly adept at handling these complexities because it has been pre-trained on a vast corpus of text, enabling it to grasp subtle linguistic nuances before any project-specific fine-tuning. For tokenization, the Transformer utilized the pre-trained tokenizer associated with bert-base-uncased, which performs beyond simple word splitting. This tokenizer not only segments text but also cleans it, integrates special tokens like [CLS] for classification tasks and [SEP] for separating segments, and manages the necessary padding and truncation. This robust preprocessing supports the Transformer's complex architecture, where each input token must be precisely formatted to leverage the model's pre-trained knowledge effectively.

Training the Transformer involved fine-tuning the model on our specific dataset of Bitcoin-related tweets. Fine-tuning a pre-trained model like BERT involves re-training it on a smaller dataset with a task-specific focus, allowing the model to adapt its broad linguistic capabilities to the specifics of sentiment analysis. During this phase, we adjusted various hyperparameters, including learning rate, batch size, and the number of epochs, to optimize the model's performance for our data.

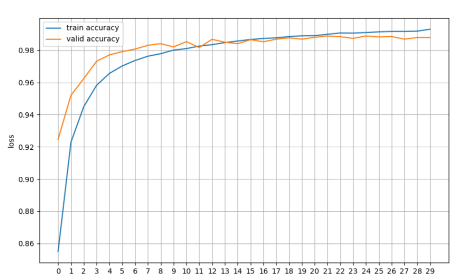
Hyperparameter	Value
Batch Size	16
Learning Rate (lr)	1e-5
Epochs	<5
Optimizer	Adam
Loss Function	CrossEntropyLoss
Train-Test Split Ratio	80% train, 20% test
Validation Split (from train)	25%
Model Freeze	False
Tokenizer	bert-base-uncased

However, the training initially presented significant challenges in terms of computational resources. When we attempted to train the model with batch sizes of 64 and 32, we encountered memory overflow errors, indicating that the required computational memory exceeded what was available. This issue is common with large models like BERT, which require substantial memory for their multiple layers of attention and token embeddings. To resolve these issues, we were compelled to reduce the batch size to 16, which allowed us to proceed with the training without further memory overflows. This adjustment, while necessary, required us to recalibrate other training parameters to maintain an efficient and effective training process.

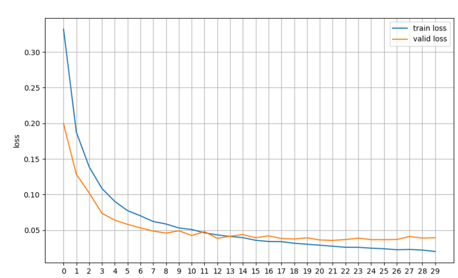
Like the LSTM, the initial accuracy of the Transformer model after extensive hyperparameter tuning and model adjustments plateaued at around 72%. However, following the removal of neutral data from the dataset, which sharpened the focus on more clearly defined positive and negative sentiments, the Transformer model's performance dramatically improved, achieving an accuracy of 98.53%. This outcome highlights the profound impact of dataset quality and composition on model performance and underscores the potential of focused training data to enhance model effectiveness significantly.

4 Results

The accompanying graph presents the accuracy and loss metrics for the LSTM model configured with three layers, demonstrating its performance dynamics over the course of training.

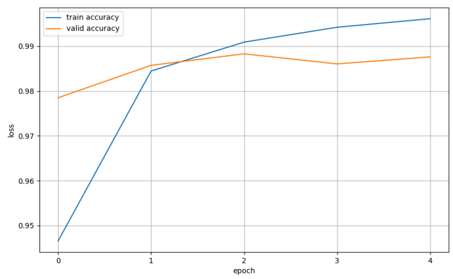


(a) LSTM 3 Layers Accuracy

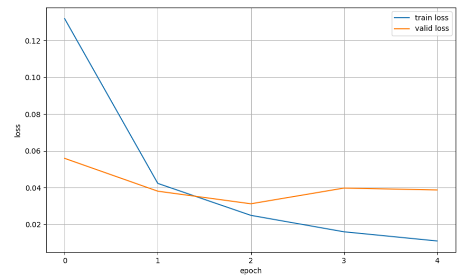


(b) LSTM 3 Layers Loss

The following graph illustrates the performance of the Transformer model with a batch size of 16, showcasing the progression of accuracy and loss throughout the training process.



(a) Transformer Accuracy



(b) Transformer Loss

The LSTM model with three layers demonstrated significant improvements in both training and validation loss after the removal of neutral sentiment data, as depicted in the updated loss graph. This adjustment led to a more effective learning curve, with loss rates sharply decreasing and stabilizing at lower levels, indicating that the model was better able to generalize from the training data without overfitting. The corresponding accuracy graph shows a rapid ascent in both training and validation accuracy, reaching a remarkable plateau at 98.2%. This high level of accuracy suggests that by focusing on more clearly defined sentiment polarities (positive and negative), the LSTM model could optimize its predictive capabilities to near perfection.

Similarly, the Transformer model, adjusted for batch size to manage computational resources effectively, showed equally promising dynamics. The training loss decreased substantially and maintained a low and stable rate, reflecting robust learning and model stability throughout the training phases. The validation loss mirrored this positive trend, albeit with slight fluctuations, likely due to the model's sensitivity to batch size and learning rate adjustments. The accuracy measurements for the Transformer model, documented in the updated graphs, indicate an impressive convergence of training and validation accuracy, culminating at a high of 98.53%. This performance underscores the model's ability to generalize well to unseen data, confirming its efficacy in handling complex linguistic patterns.

The strategic removal of neutral data from the dataset resulted in both models not only surpassing previous accuracy levels but achieving near-perfect classification performance. This breakthrough suggests that the initial plateau at around 72% accuracy was largely due to the ambiguity introduced by neutral sentiments, which complicated the models' learning algorithms. With these sentiments removed, both models were able to focus more effectively on a binary classification task, enhancing their discriminative power.

The exceptional results achieved with these adjustments illustrate the potential of both neural network approaches—LSTM and Transformer—in processing natural language data for sentiment analysis. It also highlights the importance of dataset composition and preprocessing in machine learning, where data quality and clarity can significantly influence model performance. Moving forward, these results provide a compelling case for further exploration into more sophisticated data handling and model tuning techniques to maintain and possibly exceed these levels of performance in dynamically changing domains like cryptocurrency sentiment analysis.

References

1. Gaurav Dutta. (2021). *Bitcoin tweets - 16M tweets With Sentiment Tagged, Version 1*. Retrieved April 19, 2024, from `https://www.kaggle.com/datasets/gauravduttakiit/bitcoin-tweets-16m-tweets-with-sentiment-tagged`.
2. Jeffrey Pennington. (2017). *Enhanced LSTM for Natural Language Inference*. Retrieved April 19, 2024, from `https://www.kaggle.com/datasets/gauravduttakiit/bitcoin-tweets-16m-tweets-with-sentiment-tagged`.
3. Kash. (2023). *Bitcoin Tweets, Version 1*. Retrieved April 19, 2024, from `https://www.kaggle.com/datasets/kaushiksuresh147/bitcoin-tweets`.
4. Qian Chen. (2014). *GloVe: Global Vectors for Word Representation*. Retrieved April 19, 2014, from `https://nlp.stanford.edu/projects/glove/`.