



SCHOOL OF BUILT ENVIRONMENT, ENGINEERING AND COMPUTING
LEEDS BECKETT UNIVERSITY

**Analyzing Sentiments at the Entity Level in Financial News Text Using NLP By:
Terdoo Michael Dugeri**

Supervisor: Dr. Gopal Jamnal

Submitted to Leeds Beckett University in partial fulfilment of the requirements for
the degree of MSc. Data Science

September 2024

Candidate's Declaration

I, Terdoo Michael Dugeri, confirm that this dissertation and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed;

Where I have quoted from the work of others the source is always given. With the exception of such quotations this dissertation is entirely my own work;

I have acknowledged all main sources of help;

I have read and understand the penalties associated with Academic Misconduct.

Signed: Terdoo Michael Dugeri

Date: 15th May 2024

Student ID No: c7320511

Acknowledgements

I give God all glory for seeing me through all the rigours of graduate school. Special thanks to Dr. Gopal Jamnal, who was always honest and supportive all throughout my dissertation. All the lessons learnt will help be better as I delve fully into the field of Machine Learning. Mum and Dad, Thank you for making sure I never stopped moving throughout this journey. I love you. Francis, Lubem, Martha and Ngutor, my guys always. To my friends and family, I appreciate you all. To my partner Ruth, Thank you for always been here.

ABSTRACT

This study investigates entity-level sentiment analysis within cryptocurrency markets by leveraging advanced natural language processing (NLP) models to analyze cryptocurrency news datasets and stock price information. The primary objective is to evaluate the effectiveness of various NLP models—DistilBERT, BERT, RoBERTa, and LSTM—in predicting market sentiment and potential price movements. The research highlights DistilBERT's superior performance, demonstrating higher accuracy, precision, and F1 scores compared to other models. Specifically, DistilBERT achieved an accuracy of 85.28% and an F1 score of 0.74818, outperforming BERT, RoBERTa, and LSTM in sentiment classification tasks. This suggests that DistilBERT's efficiency and reduced computational demands make it particularly well-suited for real-time sentiment analysis in the fast-paced cryptocurrency market. The study also highlighted areas for future research for better performance, exploring their generalizability across other financial settings in order to address the issue of bias.

TABLE OF CONTENT

Contents

Candidate's Declaration

Acknowledgements

Abstract

Abbreviations

Chapter 1: Introduction

1.1 Overview

1.2 Rationale

1.3 Aim and Objectives

1.4 Outline

Chapter 2: Literature Review

2.1 Crypto currency and News Sentiment

2.2 Sentiment Analysis in Finance

2.3 BERT

2.4 DistilBERT

2.4.2 LSTM (Long Short-Term Memory)

2.4.3 RoBERTa

2.4.4 FinBERT

Chapter 3: Methodology

3.1 Ethical Considerations

3.2 Data Collection

3.3 Data pre-processing

3.4 Crypto Named Entity Recognition (NER)

3.5 Data preparation

3.6 Classification model, Training and Evaluation

3.7 Findings, Discussions and Conclusion

3.8 Deployment

4.0 Chapter 4: Implementation

4.1 Data Collection

4.2 Data Preprocessing

4.3 Crypto Name Entity Recognition (CNER)

4.4 Model Building and Evaluation

5.0 Result and Discussions

Chapter 6: Limitations, Recommendations and Conclusion

6.1 Limitations

6.2 Recommendations

6.3 Conclusion

7.0 Project plan, timeline of execution, and feedback from supervisor

5.0 References

5.0 Appendix

Chapter 1: Introduction

A brief overview of the direction of the research is given in this first section of chapter 1 of the report. Next in the first section is a brief description of Aims and Objectives, followed by a description of how the dissertation will proceed. In the last section, the outline of the dissertation is mapped out by listing each of its chapters.

1.1 Overview

The explanation can be found in the initial section of chapter 1 of the dissertation. The following section gives a succinct overview of the aim and objectives before describing the dissertation's methodology. The final section provides a comprehensive overview of the dissertation by enumerating the chapters it comprises.

In the current dynamic financial environment, the rapid spread of unfavorable news has a substantial impact on market results. Although classic financial theory depicts investors as rational individuals, several studies have emphasized the significant influence of their irrational behavior, encompassing both negative and positive sentiment (Lee et al., 17; Baker and Wurgler,). There is a well-established correlation between these attitudes and numerous parameters such as customer ratings, stock prices, crypto prices, and other related variables. Therefore, it is clear that there is a requirement for immediate and fast responses to minimize potential losses. Given the unpredictable fluctuations in stock values, carefully analyzing news headlines to understand the underlying sentiments is a challenging and time-consuming endeavor that requires considerable amount of accuracy.

Machine learning techniques are becoming more prevalent in the financial industry, with the use of Bag of Words (BoW) techniques in the past and the recent introduction of pre-trained language models like BERT. As a result, Natural Language Processing (NLP) tasks are consistently producing positive outcomes. Natural Language Processing (NLP) plays a vital role in various domains such as insurance claims processing, loan risk assessment, fraud detection, stock market price prediction, and market trend analysis through sentiment analysis. Several experiments have been conducted utilizing the BERT model for entity-level classification in the field of finance. Utilizing machine learning

algorithms, sentiment classification allows for the examination of textual data obtained from many sources such as financial news, social media, analyst reports, and online platforms. It empowers investors and other stakeholders to make well-informed decisions by providing real-time information, which is essential for making strategic business choices (Fisher et al 2016). Although sentiment analysis may be performed at many levels, such as token, phrase, and document, token-level analysis has been seen to be specifically advantageous for analyzing Bit coin news articles that involve discussions about numerous crypto currencies (Tang Y. et al., 2023). As the token level analysis ensured that in cases where a news headline talks about more than one crypto currency, their sentiments were accurately captured in the analysis. This study helps to further understand market trends and how they influence the price fluctuations of crypto currencies. To minimize financial losses, investors are best served to be on alert at all times to the can promptly identify and respond to negative feelings reflected in news headlines due to the unpredictable nature of crypto currencies (Day and Lee, 2016; Dodevska et al., 2019). This project hopes to achieve as its objective a fine-tuned distilBERT model for crypto currency related news sentiment classification. The performance of these model will also be compared with other advanced deep learning models.

1.2 Rationale

While sentiment analysis in the financial domain using NLP Techniques isn't novel research. There is a need for improved sentiment analysis in this domain, especially for Bit-coin and other crypto-currency news, and that's why we're doing this research. The need for sophisticated natural language processing models that will deliver reliable sentiment analysis in a timely and efficient manner is evident, considering the huge influence of news sentiment on market act. Taking advantage the characteristic of DistilBERT which is a distilled version of BERT that has shorter computational time by Fine-tuning with financial news headlines curated from crypto currency should improve the accuracy of the sentiments prediction , I intend to improve the efficiency of these methods and provide more accurate entity-level sentiment predictions. By so doing

providing invaluable Insights that can lessen the financial impact of bad news by empowering stakeholders and investors to make smart choices.

1.3 Aim and Objectives

1.3.1 Aim

Fine-tuning DistilBERT Model for crypto currency news sentiment analysis.

1.3.2 Objectives:

1. Critically review relevant literature on sentiment analysis in financial domains, and understand the models and methodologies utilized in previous research.
2. To optimize the current State of the Art Models i.e DistilBERT utilizing Natural Language Processing (NLP) techniques for sentiment analysis in crypto currency news.
3. Evaluate the performance of the model in comparison to other state-of-the-art. NLP models like BERT, FINBERT, ROBERTA and LSTM.
4. Provide strategic recommendations for the future application of entity-level sentiment analysis in crypto currency markets, outlining potential areas for refinement, scalability, and integration into decision-making processes.

1.4 Outline

This rest of report is organized as follows:

Chapter 2 looks back at relevant literature in NLP as it relates to sentiment classification in finance and also contributions done in the area of crypto currencies related news

Chapter 3 defines the approach used to develop model that would be used for sentiment analysis

Chapter 4 Implementation of this process is done here

Chapter 5 discusses the results obtained and optimization of all the models considered

Chapter 6 the limitations, Recommendations, Conclusion

Finally, Chapter 7... Ethical Considerations, Project plan, timeline of execution, feedback from supervisor etc.

Chapter 2: Literature Review

Firstly, the chapter introduces the basic background knowledge about crypto-currency news and the importance of the sentiments that they hold. It is followed by an examination of the different methods applied in the field of finance for sentiments analysis. The following section reviews and critically analyses the use of data mining and deep learning techniques in sentiments analysis as it relates to the domain specific world of finance, along with a literature review to support our findings. In the last section, we discuss different deep learning classification techniques used in sentiment analysis.

2.1 Crypto Currency and News Sentiment

Crypto currency news sentiment is now considered as one of the major factors that shape the highly volatile and speculative crypto currency market (Xie et al., 2019). Those that support sentiment analysis claim that it is able to give investors and crypto traders credible information although this does not invalidate the challenges and seeming restrictions that come with it, both of which have to be handled carefully. (Park et al., 2016). Researchers have affirmed that sentiment analysis has an impact on crypto trading. Research by Karalevicius et al. (2018) and Smales (2019) proved that positive news can elevate the price level, while negative news can cause a decline. Nevertheless, the relationship between sentiment analysis and crypto currency prices is not always directly proportional. The crypto market is unrestricted and unregulated; hence, prices can be changed by opinions, mere hype around a particular currency, and fear of missing out (FOMO) instead of basic trading factors (Corbett et al., 2019). This can cause price bubbles where price increases or decreases but not according to their true value and the sentiment turns to reality (Fry & Cheah, 2016).

The quality and reliability of the information on crypto news and opinions differs, it could be from news outlets or random blogs, it could also be from social media; especially Twitter, as it has been known to be a major hub of discussing crypto currency. (Allcott & Gentzkow, 2017). Hence, there is the danger of misinformation and loss of credibility due to unverified news (Vosoughi et al., 2018). The emergency of unfounded news and the deceptive content in the crypto industry is worrisome, as this can be used to create

false impression of price variations, which could lead to investors making bad trading decisions. (Liu & Selover, 2021). The absence of a major leader to take responsibility in the crypto market is the main cause of manipulation of the market by the bad actors (Krafft et al., 2018).

Sentiment analysis keeps having challenges because of the instability and volatility of the crypto market (Xie et al., 2019). The old methods of sentiment analysis, for example, the use of machine learning algorithms to predict prices, may fail to meet up with the dynamic terrain of crypto currency. (Chen, et al., 2019). This can result into massive error since training models of machine algorithm are built on historical data (Wang & Luo, 2020).

Meanwhile, there is also the challenge of bias and subjectivity. Sentiment analysis tools are not self-predicting, rather they predict based on the training of their programmers, and this leaves room for programmer bias and sentiments. (Schumaker & Chen, 2009). Thus, a sentiment analysis tool will only predict as far as the sentiment of the programmer (Bollen et al., 2011). For example, if a sentiment analysis tool is trained on random news blog or opinionated notions of the programmer, it may not provide genuine analysis of the whole crypto market (Park et al., 2016).

The seeming dangers and restrictions of crypto currency news sentiment analysis together with the important questions they pose about its function in the investment decision and market monitoring makes it a controversial issue. Although the sentiment analysis can be a good tool to price movements and investment options, it should not be the only or the primary reason for the investment decisions (Mai et al. 2018). Mainly, investors and traders also need to take into account fundamental factors like the technology, adoption rates, and the regulation developments for evaluating the crypto currency investments (Shams, 2020). In addition to that, sentiment analysis should be applied together with other forms of market analysis such as technical analysis and on-chain metrics thus, to give a broad and in-depth view of the market conditions (Kang et al., 2019).

More so, the application of sentiment analysis on trading platforms and the decision-making based on algorithm brings up the issues of ethics and regulations. By using sentiment analysis to do high-frequency trading or to trigger the automatic rebalancing of

the portfolio, it can increase the effect of the biased or inaccurate sentiment data and thus causing the market instability (Kim et al., 2021). Using sentiment analysis in algorithmic decision-making also leads to inquiries about the transparency, accountability, and fairness, as the assumptions and value judgments that are embedded in sentiment analysis models are not always visible or subject to public examination (Tang et al., 2021).

Thus, there is a necessity for more research and development in crypto currency news sentiment. This promotes the development of more advanced and adaptable sentiment analysis tools that can deal with the changing and complicated communication of the crypto currency community (Li et al., 2020). Besides, it is also necessary to produce more diverse and representative datasets for the training of the sentiment analysis models and to apply the techniques such as transfer learning and domain adaptation to the extender of the generality of these models (Araci, 2019). Also, there is increasing awareness for greater transparency and explainability of sentiment analysis models. This would help users see the assumptions and the limitations of these tools and thus, make a wise decision about their use (Gadek et al., 2021).

2.2 Sentiment Analysis in Finance

2.2.1 Machine Learning Techniques and Sentiments analysis in finance

Sentiment analysis is a natural language processing task focused on recognizing and categorizing subjective information conveyed in texts to ascertain the related polarity .(Liu,B,2022). Its popularity stems from its wide range of applications in various fields such as business, social media monitoring, customer service, political service, political analysis and finance. The widespread popularity of sentiment analysis has prompted extensive research, notably on its applications in the field of finance. Sentiments in finance are monitored using various tools and devices that incorporate the use of artificial intelligence and Machine learning.

According to (Kaur, A. and Gupta, V., 2013) sentiment analysis approach that are applicable to finance can be grouped into the following

1. The lexicon based approach
2. N gram Modelling
3. The Machine learning Approach

The lexicon-based approach categorizes the sentiments derived from words or phrases using a list of word. It is a word list that consists of lexical feature that are labelled generally as either negative or positive based on its semantic disposition. One of such is the Bag of Words (BOW) model that uses a word list to classifier word as either negative or positive. It provides a simple an efficient way of representing text for machine learning algorithms. Commonly used lexicons are VADER ,SentiWORDnet (Sohangir, S et 2018). Datasets that are related to finance from social media platforms like stock twits were used on sentiment lexicons like VADER, SentiWordNet, Text blob and their performance evaluated alongside machine learning techniques logistic regression, naiyes bayes and SVM. With all predictions achieving accuracy of over 80%, However great disparity exist in the number of texts that were classified as neutral from the extracted data which puts into question the overall efficiency of the lexicon based approach to financial texts.This gave rise to creation finance lexicons using from financial literature with the hope that this approach improves the understanding of words in the context of finance.[Ding, Y et al. 2017].

Also, In 2019, F.Z Zing developed a set of lexicons that were inspired by cognitive processes, with the aim of improving domain adaptation. Prior to that Loughran T and McDonald B 2011) in their attempt to improve the application of sentiments lexicon to financial text produced a word list from examining words that appear 5% of SEC 10K Universe, using that to identify words that have negative colorations. The also used a term weighting system that attenuates words with the higher impact frequency and allow less frequently used words to have higher impact. This helped in the reduction of words misclassified as negative in the financial domain. The result here showed an nearly 75% improvement in correct in the prediction of negative words. Lexicon based approaches however ignore word order or context leading to loss of sematic meaning. This limitation can result in the omission of crucial information.(Turney, P.D. and Pantel, P., 2010).

2.2.2 Machine Learning Approach for sentiment analysis in finance

Machine learning methods have also been used for sentiment classification to good effect as the develop algorithms that optimize system performance based on actively learning from past data or example datasets. For sentiment classifications. Machine learning approach learn from a corpus of training data and then uses what has been learnt to classify unseen or new data.(Pang B lee et all 2002) in attempt to measure the performance of machine learning techniques used movies rating data sets and applied machine learning techniques like Naiye bayes, SVM and Maximum entropy. After learning on training data set and the test on unseen data the movie sentiments classification yielded impressive result albeit, it was observed that Naiyes bayes techniques gives better result when the training datasets has less features. Its struggles to maintain high performance when there are many features to consider, here in lies one of the strength of SVM as subjecting it to more features yield commendable results still. Machine learning approach just like the lexicon based approaches are based on the Bag of word techniques (Turney, P.D. and Pantel, P., 2010). In this model text is analyzed based on the list of words in it with no attention given to the mode in which this words are being used as such words might not necessary maintain same context when used in sentences. The word “undervalue” word normally be seen as negative by BOW whereas it be in a sentence at “undervalued crypto coin” which gives a positive tone to the text.(Khairnar, J. and Kinikar, M., 2013.)

2.2.3 Deep Learning and Sentiment Analysis

Over the last decade, deep learning has achieved significant advancements and outstanding outcomes in various fields such as computer vision, speech recognition, and more recently, natural language processing (Collobert et al., 2011; Goldberg, 2016). The resurgence of neural networks can be ascribed to numerous sources. The key factors contributing to the success of machine learning include the increased computational power resulting from hardware advancements, such as GPUs, the abundance of large training datasets, and the effectiveness and adaptability of learning intermediate representations (Bengio, Courville, & Vincent, 2013).

Deep learning- based model also became popularly adopted in sentiment analysis because of its high performance results. One of the advantages of utilizing deep

learning, also referred to as neural network, is the utilization of cascaded multi layers for intricate feature extraction and transformation [Zhang, L et al 2018]. These layers employ non-linear processing units for this operation. This approach utilizes word embedding's such as Word2vec and global vectors (Glove) (Mikolov, T et al 2013) (Artetxe, M. and Schwenk, H., 2019) to capture both semantic and syntactic relationships inside phrases. Word embedding in natural language processing (NLP) is accomplished using neural networks or matrix factorization techniques.

The approaches commonly employed in financial sentiment analysis include RNN, CNN, LSTM, and the attention mechanism. (Sohangir, s 2018) (Shijia, E. et al. 2018). While Word2Vec is effective in generating distributed representations, it does not consider word order and global context. In contrast, GloVe addresses these limitations by integrating global information into word vectors using co-occurrence matrix decomposition. Although this represents an advancement over traditional machine learning methods, it is constrained by a static, context-independent representation of each word when applied in various settings.[Salant, S. and Berant, J., 2017] ELMo (Embeddings from linguistic models) and GPT have been developed to tackle this difficulty.[Radford, A.(2018)].

2.3 Bidirectional Encoder Representation from Transformers (BERT)

Recently, there have been additional achievements in natural language processing with the advent of transformer models such as BERT. It is a sophisticated natural language processing (NLP) model that Google AI researchers developed. BERT is an acronym for Bidirectional Encoder Representations from Transformers. The publication "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" was authored by (Jacob Devlin et. al. 2018). It utilizes attention-based transformers, which, unlike RNNs, are not only bi-directional but also encode context. (Radford,A. 2018). The BERT model achieved a significant achievement by effectively performing sentiment analysis tasks. This was accomplished by utilizing pre-training on the masked language model and fine-tuning with Next sequence predictions to minimize discrepancies. Refining this approach and implementing it for sentiment analysis produces remarkable

outcomes. (J. Devlin et al., 2018). GPT-2 demonstrates the effectiveness of unsupervised pre-training using a big and diverse dataset and a deep neural network to obtain state-of-the-art results in sentiment analysis without the need for additional training. The attention mechanism in BERT enables it to choose to retain the relevant information from a text, rather than storing all of it. Hybrid models exist which aim to combine the advantages of BERT and previous models. BERT models have been applied in conjunction with deep learning techniques such as recurrent neural networks (RNN) or convolutional neural networks (CNN). These models have been modified to perform sentiment categorization at both the document level [Yang, Z; 2016] and the aspect level.[Majumder, N. et al., 2018] Wang, Y in 2016. The system has demonstrated exceptional spatial awareness and the ability to accurately predict relationships.

2.3.1 BERT Architecture

The BERT Model architecture is based on the original transformer available in the tensor2tensor library. BERT is constructed using the Transformer architecture, which consists of numerous layers of Transformer encoders and Decoders. This architecture replaces the traditional encoder-decoder structure found in competitive neural sequence models with stacked self-attention and fully connected layers for both the encoder and the decoder.

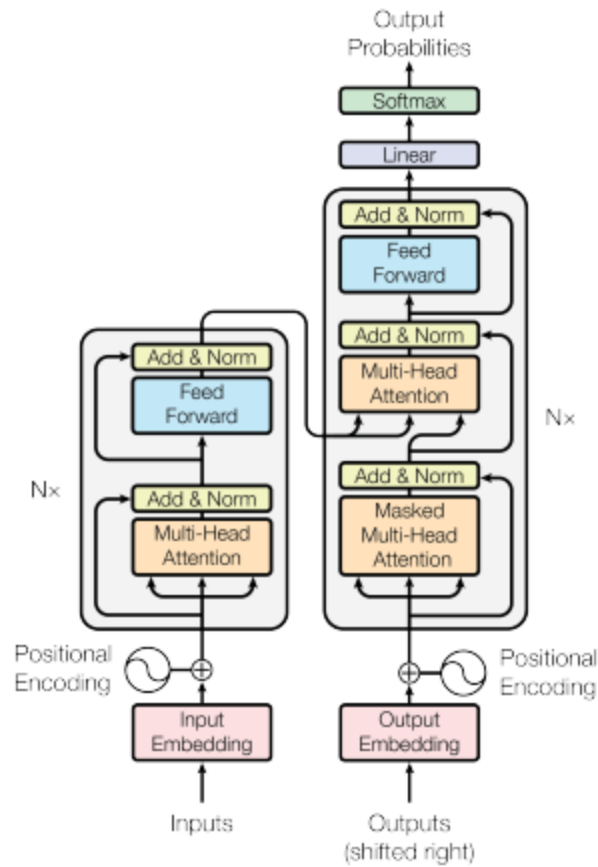


Fig 2.1 Transformer Architecture (Viswani et al, 2017)

The BERT models are in two forms , BERTBASE or BERTLARGE . They both comprising of Layers (L), Hidden sizes (H), Attention Heads (A) and other parameters.

BERTBASE which has the same model size as that of OPENAI GPT has (L= 12, H= 768, A= 12, Total Parameters= 110M) and that of BERTLARGE Attention (L= 24, H= 1024,

A= 16, Total Parameters= 340M). BERT differs strongly from GPT transformers in that its attention is Bi-directional as compared GPT Transformers being unidirectional in their operation.

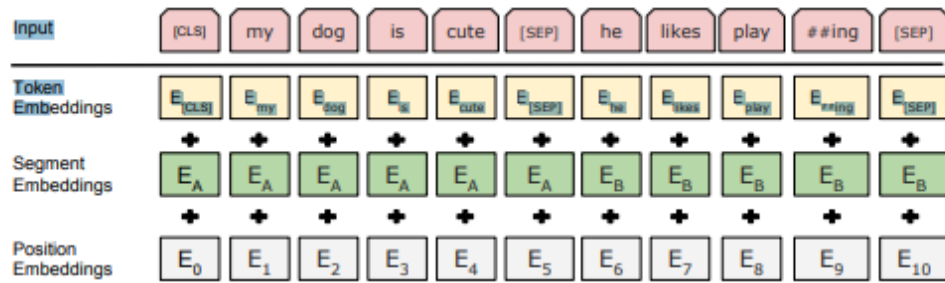


Figure: BERT Input Representation (Jacob Devlin et. al. 2018)

Input Embedding, Positional encoder, Multi-Head attention are some of the key parts of the BERT architecture. Here we look at the role the play in the setup

- I. **Input Representation:** BERT is provided with tokenized input text, This inputs representation are specially designed take either single or multiple pairs of sentences. Where each token corresponds to a word or subword." The tokens are converted into condensed vectors, known as embeddings, using an embedding layer. BERT employs WordPiece embeddings, enabling it to proficiently handle out-of-vocabulary terms by decomposing them into smaller sub words. BERT's vocabulary consists of 30,000 tokens derived from the WordPiece tokenizer. BERT uses three types of embeddings, token to split words in to sub words, Segment embedding and position embeddings.
- II. **Positional Encoding:** In order to overcome the absence of built-in encoding for token order in the Transformer design, positional encoding is incorporated into the input embeddings. This encoding provides detailed information regarding the exact position of each token in the sequence. This allows BERT to preserve the sequential order of token^s and effectively capture positional information. Given it a proper sense of order. The input representation for each token is constructed by summing the corresponding token embedding E_i segment embedding S_i and position embedding P_i

$$Input_i = E_i + S_i + P_i$$

III. **Position-wise Feed-Forward Networks:** There is also the feed-forward network that is connected to each layer in the encoder and decoder. This consists of two linear transformations with a ReLU activation in between. While the linear transformations are the same across different positions, they use different parameters from layer to layer.

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (\text{Bahdanau, D., 2014})$$

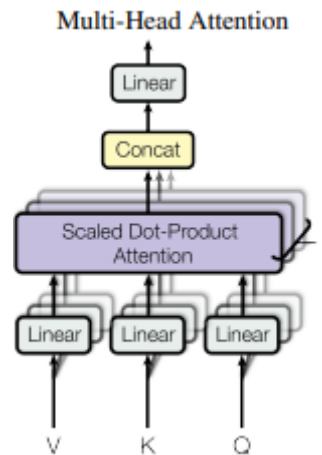
IV. **Embeddings and Softmax:** Learned embedding are used to convert input and output tokens to dimensions. Here, the input and output embeddings share the same weight and scales them by $\sqrt{d_{model}}$

V. **Attention Mechanism:** Using this function set of key-value pair and a query are mapped as outputs, here Query (Q), keys (K) and values (V) and the output itself as known as vectors. The output gotten here is a weighted sum of the values, where the assigned weight to the values are added with the use of a compatibility function of the query and it's key. Unlike when single attention function is used with the d_{model} dimension. The multi-head attention layer was found to be useful to linearly project queries, keys and values with different, learned linear projections d_Q , d_k and d_v dimensions, respectively. The attention function is calculated simultaneously of sets of queries that are stored into a matrix Q. while the matrix K and V hold the keys and values.

$$\text{matrix of output as: } Attention(Q, K, V) = softmax(QKT \sqrt{d_K}) \quad (\text{Ba, J.L., 2016.})$$

VI. **Multi-Head Self-Attention:** The Multi-Head Self-Attention Mechanism allows BERT to evaluate the significance of each token in the input sequence compared to all other tokens. This technique obtains contextual information by selectively attending to different portions of the input streams. Point-wise Feed-forward Networks are utilized subsequent to the self-attention process. These networks apply linear adjustments to the input representations at each location in a consistent and uniform manner. BERT achieves bidirectional context modeling by incorporating information from both the preceding and succeeding contexts concurrently throughout the training phase. This capability boosts

the effectiveness of BERT in comprehending complex semantic subtleties and contextual interdependencies inside a sentence.



Multi-Head Attention consisting of several layers in parallel ()

Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions. With a single attention head, averaging inhibits this

$$MultiHead(Q, K, V) = concat(head_1 \dots head_h)W^O$$

$$where head_i = Attention(QW_i^Q, KW_i^K, VW_i^Q)$$

Projection are parameter matrices $W_i^Q \in R^{d_{model} \times d_k}$, $W_i^K \in R^{d_{model} \times d_k}$, $W_i^V \in R^{d_{model} \times d_v}$, and $W_i^O \in R^{hd_v \times d_{model}}$

Where $h = 8$ parallel attention layers, or heads. For each of these $d_k = d_v = \frac{d_{model}}{h} = 64$.

Due to the reduced dimension of each head, the total computational cost is similar to that of single-head attention with full dimensionality.

2.3.2 Pre-training Tasks: BERT is subjected to pre-training by exposure to an extensive corpus of literature and the completion of two unsupervised tasks:

- i. The Masked Language Model (MLM) technique entails the random masking

of specific input tokens. BERT attempts to anticipate the masked tokens by utilizing contextual information from the neighboring token, thus thereby obtaining contextualized word representations.

- 2.3.3** ii. The Next Sentence Prediction (NSP) tasks BERT with determining if the second sentence is a logical continuation of the first sentence in the original text. This challenge boosts BERT's comprehension of sentence relationships and promotes its effectiveness in tasks such as question responding and text entailment.

Fine-tuning refers to the process of adding specialized layers to the pre-trained BERT model following the first pre-training phase. The task-specific layers of BERT are trained using labeled data for the target task, allowing BERT to adapt its representations to the uniqueness of the task at hand. The BERT model's design, which includes Transformer encoder layers, bidirectional context, positional encoding, and pre-training tasks, has greatly transformed the field of natural language processing. It has achieved top-notch performance in several NLP tasks and benchmarks.

2.4 Sentiment Analysis using BERT Models.

Due to the rising importance of online reviews,(Hu, M. and Liu, B., 2004)(Somprasertsri, G. and Lalitrojwong, P., 2010) there is a growing need for sentiment analysis in several industries. Businesses are using sentiment analysis to assess, evaluate, and improve their product or service delivery. Utilizing its remarkable efficiency, BERT has been optimized for diverse sentiment analysis tasks in natural language processing (NLP). Nguyen et al. (2020) combined BERT and RCNN to achieve superior performance compared to other models in analyzing sentiment in a Vietnamese dataset. Similarly, BERT has been effectively utilized in analyzing Chinese stock reviews, detecting hate speech in Hindi-English, and classifying emotion in Twitter data. In addition, BERT has been utilized for assessing online reviews on Google play, aiding in the enhancement of applications. (Goe.H. 2024). Similarly, these recent breakthrough in natural language

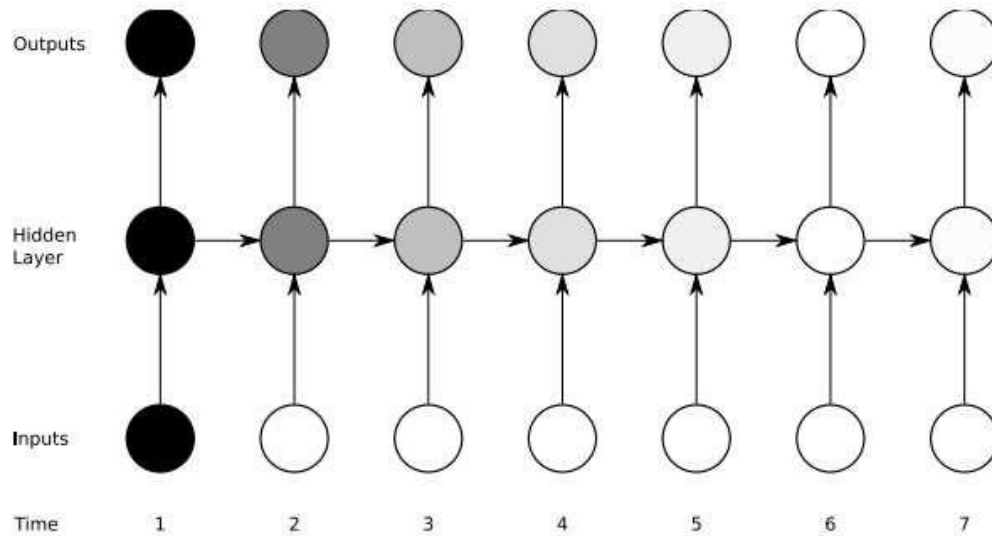
processing (NLP) and deep learning have greatly enhanced the ability to extract emotion from news and texts in the field of finance.

2.4 DistilBERT

DistilBERT is a condensed form of BERT, DistilBERT was trained on 8 16GB V100 GPUs for approximately 90 hours. [Sanh, V., 2019] it is also trained on the same original corpus as BERT [Zhu et al., 2015] created by Hugging Face, the token-type embeddings and the pooler as found in BERT are removed with the number of layers reduced by a factor of 2. The linear layer and layer normalization is are the most used parts in a transformer architecture as they are highly optimized. Variation in the hidden size dimension do not significantly affect the computational efficiency compared to the reduction in the number of layers. DistilBERT is specifically designed to be efficient for both training and deployment. It attains an efficiency rate of over 95% of BERT Performance on GLUE-Task. While operating at a speed that is 60% faster than BERT. By including knowledge distillation and a triple loss mechanism in the pre-training phase, DistilBERT effectively captures the inductive biases acquired by larger models, resulting in enhanced performance on subsequent challenges.

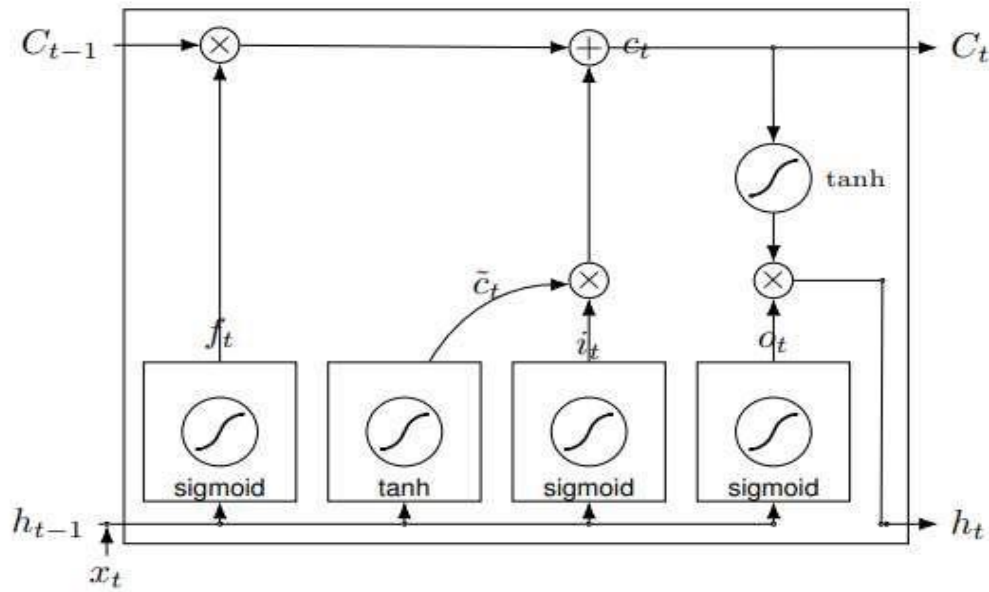
2.4.2 LSTM (Long Short-Term Memory)

LSTM is unique type of Recurrent Neural network (RNN) architecture and algorithm specially design to surmount some of the issues associated with rational RNNs especially in handling long term dependencies and extenuating some of the challenges such as exploding and exploding gradient. This gradients uses sequences of data to process and make predictions based on them. [Hochreiter and Schmidhuber (1997)] introduced LSTM in 1997 to solve the vanishing gradient problem.



An RNN with vanishing gradients (Graves and Pedrycz, 2009)

LSTM introduces a memory cell with three gate mechanism the input gate, forget gate, and output gate. They make sure information network is able to remember or forget information when it is needed. The gates in LSTM are implement via a sigmoid function that is also followed by element-wise multiplication. The inputs are broken into values of 0 and 1 by the sigmoid function, this determines the amount of information that passes through. (Staudemeyer, R.C. and Morris, E.R., 2019)The forget gate decides the portion of the cell state that should either be brought forward or discarded. The Input gate on other hand regulates the amount of new information from the current input and the previous input that should be forgotten about. The Output gate determines what the output of the hidden state would be and also becomes the hidden for the next step. LSTM are used to process inputs sequences in both forward and backward directions, giving deeper context as a result. It is also used to learn complex systems by stacking multiple on each other.



Cell architecture of LSTM Alhagry, Aly and A. 2017

It is formulated as: $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$

$$h_t = o_t * \tanh(C_t)$$

Where, σ is sigmoid activation function W_o is the weight matrices h_{t-1} is past hidden state x_t is input vector (current input) b_o is bias vector \tanh is hyperbolic tangent activation function C_t is cell state

2.4.3 RoBERTa

RoBERTa which translates to robustly optimized BERT approach is an enhanced version of BERT designed with the sole aim of improving on BERT's performance via several modifications to the original architecture of BERT and its process of training. RoBERTa is training of multiple corpora, the size totaling 160GB. (Liu, Y., 2019) The larger size of dataset used in training improves the understanding of context compared to unidirectional models. RoBERTa employs a type of masking strategy that is dynamic in the sense that the patterns of the masking changes as at every epoch. This is different to BERT where

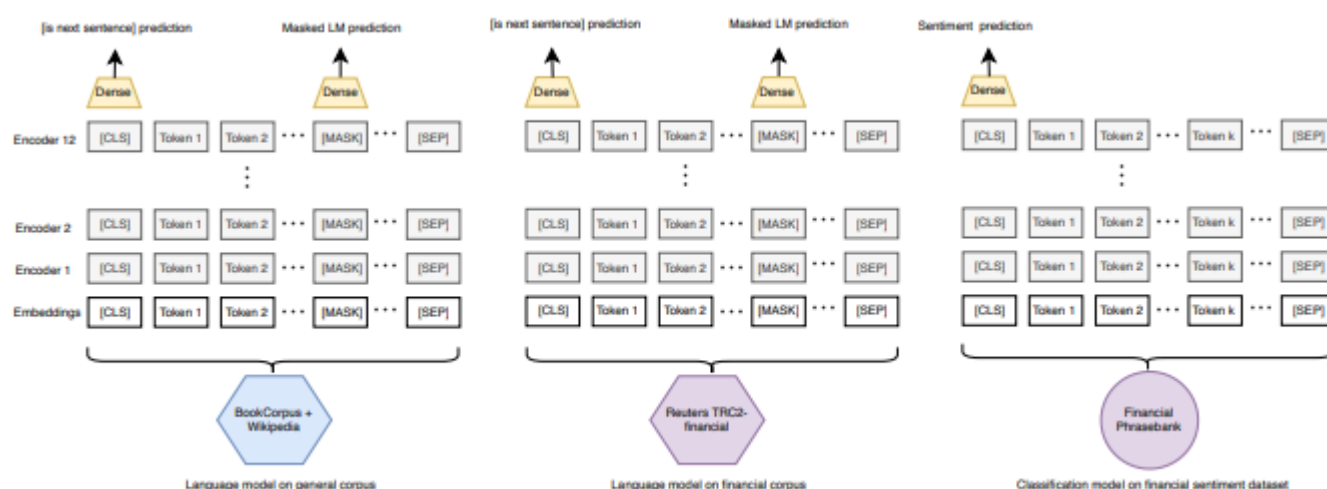
the masked tokens remain unchanged throughout. This improves its generalization ability. RoBERTa comes in different sizes you have the RoBERTa-base: 12 layers, 768 hidden units, 12 attention heads and the RoBERTa-large: 24 layers, 1024 hidden units, 16 attention heads.): RoBERTa focuses exclusively on the MLM objective, where a percentage of the input tokens are masked, and the model is trained to predict these masked tokens. The dynamic masking strategy and exclusion of NSP are key differences that contribute to RoBERTa's performance improvements over BERT.

(Zhao, L et al 2021) Fine-tuned RoBERTa for Key entities identification from online financial texts and for subsequent prediction of the sentiments that they carry. Implementing a unified approach that combines RoBERTa and Ensemble learning method with Machine Reading comprehension, to identify the key financial entities whose sentiments are negative in online text. The model performing better when compared to simply using the pre-trained language model without fine-tuning. While the model is able to identify key financial entities its focus on negative sentiments which understandable helps investors makes swift market decisions. Considerations could have been given to the advantages of positive and neutral sentiments Identification to investors. The model could also over fit from its being bias towards identifying negative sentiments. There also the worry that ambiguity might arise on the list that make up key entities as this would across different stakeholders with the possibility of the model training been impacted greatly.

2.4.4 FinBERT

FinBERT a model of BERT which is a pertained language model has been further trained on large corpus of financial communication data called TCR2-financial made up of 1.8Million financial articles published between 2008 – 2010 by reuters (ARaci 2019) . a dense layer is added just after the last hidden state of the (CLS) token for sentiment classification. The labelled sentiments datasets are used to train the classifier network. FinBERT like BERT is bidirectional as it takes note of left and right context of words during its training. It also uses the self-attention mechanism ensure that the relationship between words in sentences retains the role the play regardless of their position. FinBERT uses

the same tokenizer as BERT, however to ensure that the meaning of the financial jargons involved Vocabulary and embeddings are adapted during the process of fine-tuning.



Which was compared with other state-of-the-art methods likes LSTM, Glove embeddings, EIMo and ULMfit with FinBERT performing better, giving higher accuracy and f1 score in sentiment classification task. Gradual freezing, slanted triangular learning rates and discriminative technique were employed to minimize catastrophic forgetting The research, however, uses only datasets obtained from the financial PhraseBank with 100% annotators in agreement strengthening the generalizing ability of the model, perhaps additional datasets that reflect real-world variability and ambiguity with financials texts could improve its performance. Comparing finBERT with other transformer models would give a clearer view on its standing current in the sentiment analysis landscape.

Due to FinBERT's superiority in financial domain sentiment classification task it has been used in combination with other existing models with the aim of improving the accuracy of sentiments analysis. Zhang, Y. and Zhang, H., 2023. Using Machine reading comprehension (MRC) to evaluate the performances of BERT and FinBERT on sentiment analysis tasks. FinBERT performed slightly better, taking advantage of the semantic information expressed in financial text because it was able to easily identify multiple entities that BERT would miss as it isn't domain-specific like FinBERT. There is a question mark on how good the models generalization is beyond the datasets used here.

FinBERT was also used in combination with LSTM model by(Jiang, T. and Zeng, A., 2023.) to show that correlation exists between stock related news and stock prices which performance better than when ARIMA or LSTM alone is employed. Similarly, Halder, S., 2022. In trying to show the relationship between news sentiment and stock prices, trained BERT model on NASDAQ-100 stock data and New York Times news article which delivered sentiments and stock price predictions with greater accuracy for NASDAQ Stock. While the generalization of the findings from the latter might be limited due to the use of one data source, the former should perform better as the datasets were gotten from multiple sources.

Being able to easily identify financial entities as an important prerequisite for improved accuracy when carrying out sentiment analysis in finance Tang et Al, 2022 constructed a novel dataset consisting of financial entities with a focus on crypto currency-related news headline. Their model was designed to identify financial entities and sentiments attached to them as contained in financial text. This customized datasets was used to train modified FinBERT and BERT with CRF Layer added to this models to aid entity recognition task in its sentiment classification. Both Models performance trumped state of the art models, that included BERT, FINBERT, CHAT GPT, LSTM , ELMO as it was able to also predict multiple sentiments where text contain more than one entity.

While this performance was impressive, the news text used in this constructing this datasets were gotten from one source which is Reuters, perhaps using additional

sources to fetch this news headlines would increase the robustness of the datasets. While the approaches above have yield impressive results in finance there are still areas to improve upon, one of which my intend research attempts to address with Crypto currency gaining continuous

From the knowledge gathered so far, there has been no adoption of DistilBERT in finance, specifically predicting the sentiments present in crypto currency related news headline. I intend to construct a crypto currency news dataset from scratch and then fine tune DistilBERT for sentiment analysis of crypto currency news headline. One of the key reasons of adopting DistilBERT for this purpose is the fact that it is 40% faster than BERT while also being a 60% less in size compared to BERT.

Chapter :3.0 Methodology

The main aim of this chapter is to outline the method used to accomplish our research aims and objectives. Machine learning approaches are based on quantifying qualitative data with machine learning algorithms. As discussed in section 1, the purpose of our study is to be able to predict the sentiment of crypto currency news text using data mining techniques in order to develop a machine learning algorithm for this task. The purpose of this chapter is to present the proposed system for the implementation and evaluation of this project. In the next section, we discuss ethical considerations that were taken during the data collection process. In our study we are using python as a coding language and Vscode as the IDE and the project folder uploaded on github after completion and testing of the product.

3.1 Ethical Considerations

. This study was conducted in compliance with the ethical guidelines of Leeds Beckett University, and the necessary approvals were obtained, including the completion and supervisor approval of the university's ethics form. The data used in this research was ethically sourced from websites that explicitly permit data scraping, adhering strictly to the scraping rules set by each website. My ethical framework is rooted in responsible and transparent data usage, with a strong emphasis on privacy protection, bias mitigation, and fairness. Given the sensitivity of crypto currency news datasets and stock price information involved in our entity-level sentiment analysis within crypto currency markets, I prioritize safeguarding the privacy of data subjects. This includes implementing robust security measures and ensuring the anonymization of data to uphold ethical standards.

Bias mitigation is a critical concern and that's why we were committed to identifying and addressing any algorithmic biases that may arise during the analysis. Transparency remains a cornerstone of my approach; I will rigorously document all models, methodologies, and limitations to facilitate reproducibility and peer review. In light of the unique characteristics of crypto currency markets, I am particularly mindful of the ethical

implications of the work being carried out. I am committed to responsible AI practices, ensuring that my research does not unknowingly contribute to market manipulation. All findings will be communicated transparently, adhering to relevant laws and regulations in the crypto currency space.

Overall, my ethical considerations underscore the importance of responsible and ethical data use in generating valuable insights into the sentiment dynamics of crypto currency markets.

The proposed methodology for this study has the following process: Data Collection, Data Pre-processing, Crypto Named Entity Recognition, Data Preparation, Classification model training and Evaluation, Findings, Discussions and Deployment. The proposed methodology flowchart is as depicted in the figure below

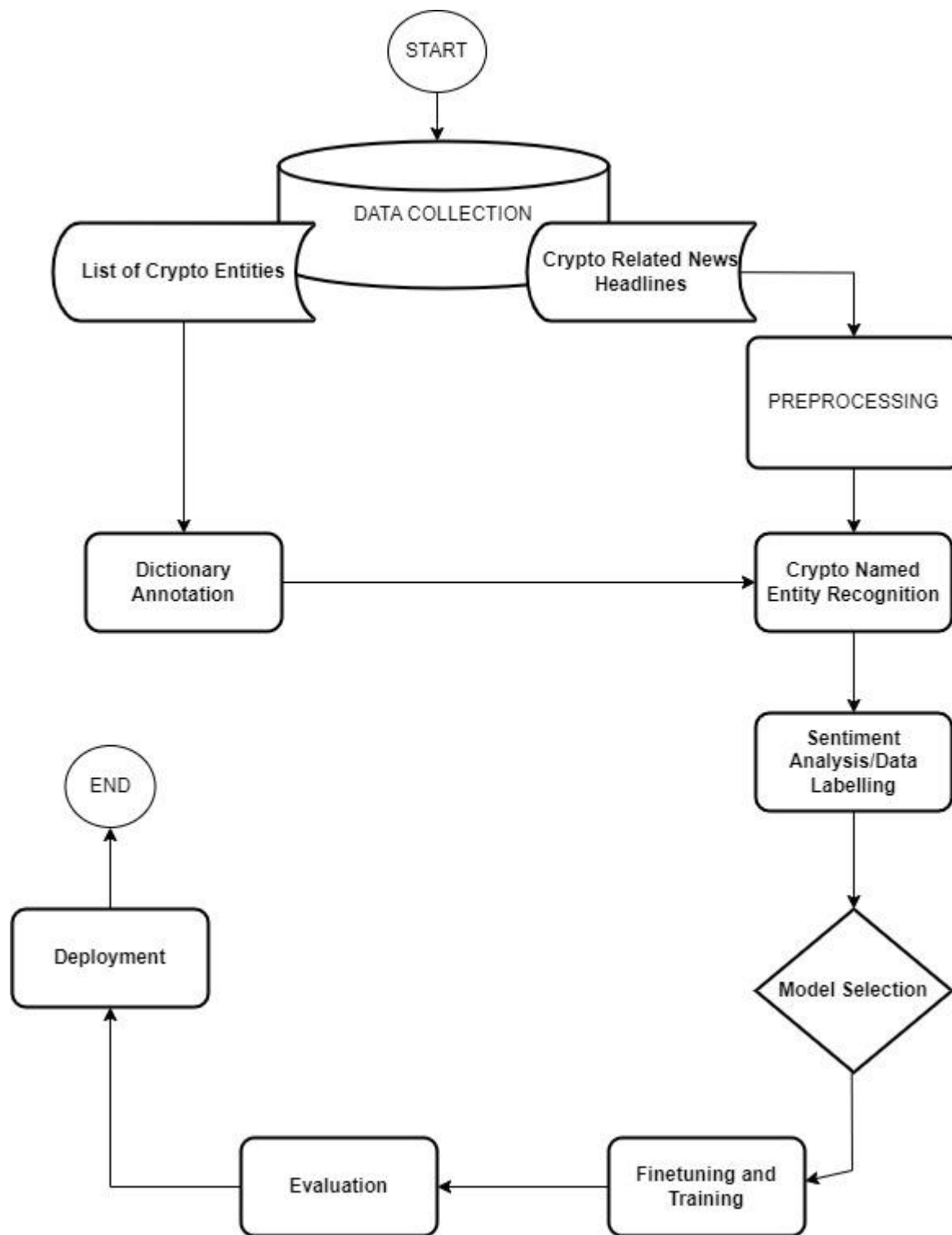


Fig 3.1 Proposed methodology for the study

3.2 Data Collection

In this section, we discuss the considerations for selecting data sources and the process followed to gather the necessary data. A crypto currency news dataset was required to fine-tune a pre-trained model for sentiment analysis specific to crypto currency-related news. The criteria for selecting websites for scraping headlines involved identifying top crypto currency news platforms that allow web scraping. This step was crucial because the model being developed will be used by individuals who regularly visit these websites for crypto currency information, and their investment decisions will likely be influenced by the insights they gain from these updates. While other sources, such as Twitter, were considered to improve the data quality, recent restrictions on data access from the platform eliminated that option.

3.3 Data Preprocessing

The data preprocessing steps included data cleaning: i.e. The extracted data contain noises, like punctuation, numbers, stop words and special characters that don't contribute to the analysis to be done. Preprocessing is done to ensure relevant data are preserved for analysis. Tokenization of the text so that it is broken down into structures that can easily processed by algorithms. The crypto currency data was then tokenized using a pre-trained tokenizer by breaking down and assigning identification numbers (id's) to each token or entity and converting them to tensors to allow the models to learn each word or entity and to be able to determine their appropriate sentiment and how they contribute to the overall sentiment of the entire text or sentence.

3.4 Crypto Named Entity Recognition (NER)

To enhance sentiment classification in the dataset used for model fine-tuning, it was essential to accurately identify crypto currency entities in news headlines. The initial approach involved using the SpaCy library for Named Entity Recognition (NER), but it proved inadequate in recognizing crypto currency-specific entities. Crypto currencies and crypto exchanges are relatively new, and traditional NLP tools might not be updated

to include them. Developing a custom approach ensures that all relevant entities in the crypto currency space are recognized, which is vital for tasks like market sentiment analysis or financial forecasting.

As a result, a semi-automated method was developed, utilizing a Python dictionary tailored to identify crypto currencies, crypto exchange companies, and a few key countries. This custom dictionary was applied to each headline, enabling the identification of relevant entities. Afterward, the TextBlob library was used to label the sentiment associated with these entities, ensuring more precise sentiment classification tailored to the crypto currency domain.

3.5 Data Preparation

The dataset was divided into a training set and a validation set using a random split from PyTorch's DataLoader class, with 80% allocated for training and 20% for validation. Random splitting ensures that each class in the dataset is proportionally represented across the training and validation sets, helping to minimize potential bias during both training and evaluation phases.

Also, the textual data was preprocessed to be compatible with BERT' models input requirements. Pre-trained models, such as BERT, expect input in a specific format, which includes converting the text into a machine-readable format. This involves tokenizing the text into subword units, assigning token embeddings, and translating the written language into binary sequences through Natural Language Processing (NLP). This prepared data is then passed into the model for training, ensuring that the input is in a format that can be effectively understood by the machine learning algorithms.

3.6 Classification model, Training and Evaluation

In this section, we discuss the performance of the model using confusion matrix and its derivatives such as precision, accuracy, recall and F-score. And also, we compare the BERT Models and also the LSTM models by hyper parameter tuning its parameters.

Classification matrix is a $n \times n$ matrix, where n is the number that describes the performance of a classifier (Salmon et al., 2015). Figure 3.4 summarizes all the confusion matrix of all the models

		Actual values	
		Positive	Negative
Predicted values	Positive	TP	FP
	Negative	FN	TN

Figure confusion matrix (Salmon et al., 2015)

True Positive (TP): The true positive rate is the total instance where a model is performing well by accurately predicting the actual positive class as positive.

False Positive (FP): The false positive rate is the total instance where a model is not performing too well by incorrectly predicting the actual negative class as positive.

True Negative (TN): The true negative rate is the total instance where a model is performing so well by accurately predicting the actual negative class as negative.

False Negative (FN): The false negative rate is the total instance where a model is not performing too well by incorrectly predicting the actual negative class as positive.

The TP, FP, TN, FN are used to mathematically determine the precision, recall, f1 score, and accuracy of a machine or deep learning model.

Precision: Precision is a calculated metric used to ascertain how well the models are performing in determining the correct sentiments labeled in the data. It is calculated by dividing the true positive rate by the total sum of positively predicted sentiments (Margherita et al., 2020).

$$\text{Precision} = TP/TP+FP$$

Recall: This is a measure that helps models to determine all the positive classes in the data. The recall is calculated through the division of the true positive classes by the total sum of the correctly predicted positive class and the falsely predicted positive class (Margherita et al., 2020).

$$\text{Recall} = TP/TP+FN$$

F1 Score: This is a measure of the harmonic average between precision and recall. The F1 score value computed is usually between 0 and 1 and is used to ascertain a good balance between the model precision and recall (Margherita et al., 2020). It is calculated by the formula:

$$\text{F1 Score} = 2*(TP/2TP+FN+FP)$$

3.7 Findings, Discussions and Conclusion

A discussion of different metrics that affected the accuracy of the models is also discussed, along with a short discussion regarding how the study may be held for further investigation with the provided suggestions. In this last section, we conclude our results by listing the best performing model and also discuss the limitations of this study with regards to the realistic settings.

3.8 Deployment

The final model was then deployed using streamlit app to create a user friendly interface for product demo test.

Chapter 4: Implementation

4.1 Data Collection

In this stage, we discuss how data was collected .The dataset comprises of **2374** news headlines related to crypto currencies: using the beautiful soup library, 2060 news were scraped from cryptopotato.com, 60 news were scraped from Coinjournal.com, 29 news were scraped from cryptotimes.io, 73 news were scraped from newsbtc.com, 152 news were scraped from cnbc.com.

4.2 Data Preprocessing

Earlier in section 3, this is the most important stage in the study since it determines how well the model performs. The raw text is made up of unstructured data mixed with special characters, symbols, etc. To get meaningful insights, these redundant data must be processed.

4.2.1 Tokenize

In natural language text processing, word tokenization is used which splits a piece of text into individual words based on a certain delimiter which is also referred as “Lexical analysis” (Bakshi et al., 2016). The most popular deep learning architectures used for NLP, such as RNN, CNN, LSTM and BERT models, also make use of tokenization to process the raw texts. For example: “I live in London” is tokenized as “I,” live,” in,” London”. This will further assist with the pre-processing of words in the subsequent steps that are aimed at manipulating individual words in pre-processing of words in the following steps which works on individual word manipulation. For this study, every news

headline was tokenized using Python libraries and functions such as word tokenize and NLTK, which supports tokenization through natural language processing. This was the followed by joining the tokenized words and storing a list in preparation for the next step in the methodology.

```
def preprocess_text(text):
    # Lowercasing
    text = text.lower()
    # Remove punctuation
    text = text.translate(str.maketrans('', '', string.punctuation))
    # Remove symbols (except alphanumeric)
    text = re.sub(r'[^a-zA-Z0-9]', ' ', text)
    # Remove numbers
    text = ''.join([i for i in text if not i.isdigit()])
    ## remove single characters eg 'k'
    text = re.sub(r"\b\w\b", "", text)
    # Remove extra whitespace
    text = ' '.join(text.split())
    # Remove special characters
    text = re.sub(r'\W', ' ', text)
    # Tokenization
    tokens = nltk.word_tokenize(text)
    # Remove stopwords
    stop_words = set(stopwords.words('english'))
    tokens = [word for word in tokens if word not in stop_words]

    return tokens

def preprocess_dataframe(df, text_column, new_column):
    df[new_column] = df[text_column].apply(preprocess_text)
    return df
```

Fig: 4.1 Preprocessing steps taken

4.3 Crypto Name Entity Recognition (CNER)

The dictionary was populated with data scraped from a crypto currency website-coinranking.com, including the top 200 crypto currencies by market capitalization and their abbreviations, labeled as 'CryptoCurrency.' Exchange companies were labeled as 'Company.' The resulting key-value pairs were saved in a JSON file and used for entity annotation. The annotated entities were then assigned sentiments (positive, negative, or neutral) using the TextBlob library.

```

entity_dict = annotated_dict

def ner_on_text(text, entity_dict):
    entities = []
    words = text.split() # Split text into words

    for word in words:
        word_lower = word.lower() # Convert word to lowercase for case-insensitive matching
        if word_lower in entity_dict:
            start = text.lower().find(word_lower)
            end = start + len(word)
            entities.append({
                "text": text[start:end],
                "label": entity_dict[word_lower],
                "start_char": start,
                "end_char": end,
            })

    return entities

def ner_on_dataframe(df, text_column, entity_dict):

    df["entities"] = df[text_column].apply(lambda text: ner_on_text(text, entity_dict))
    return df

```

Fig: 4.2 Entity Identification from Text

4.3.2 CNER and Sentiment Analysis

CNER was performed using the custom dictionary approach, and the resulting entities were assigned sentiments. The final dataset contains 1235 headlines, with a total of 1773 entities identified and classified as follows: 413 Positive, 423 Negative, and 937 Neutral. After assigning sentiments to each entity, here is what the dataset looks like.

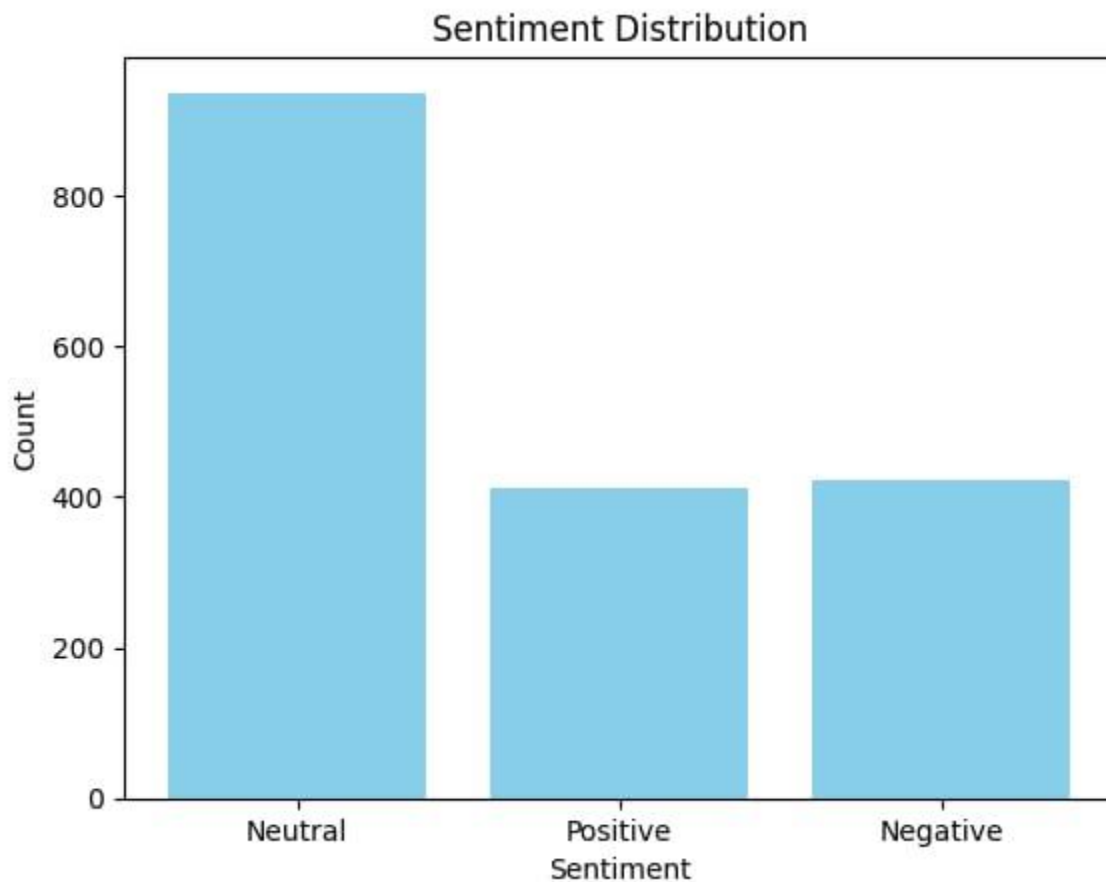


Fig: 4.3 Datasets Sentiments Distribution

Bitcoin miner Marathon Digital is mining Kasp...	[bitcoin, miner, marathon, digital, mining, ka...	bitcoin miner marathon digital mining kaspas kas	{('text': 'bitcoin', 'label': 'CRYPTOCURRENCY'...	{('start_char': 0, 'end_char': 7, 'entity': 'b...
Cryptocurrencies waver as focus shifts to BitB...	[cryptocurrencies, waver, focus, shifts, bitbo...	cryptocurrencies waver focus shifts bitbot tok...	{('text': 'cryptocurrencies', 'label': 'CRYPTO...	{('start_char': 0, 'end_char': 16, 'entity': '...
US government transfers 3,940 BTC worth \$241 m...	[us, government, transfers, btc, worth, millio...	us government transfers btc worth million coin...	{('text': 'us', 'label': 'COUNTRY', 'start_cha...	{('start_char': 0, 'end_char': 2, 'entity': 'u...
German's BKA transfers more Bitcoin to exchang...	[german, bka, transfers, bitcoin, exchanges, i...	german bka transfers bitcoin exchanges includi...	{('text': 'german', 'label': 'COUNTRY', 'start...	{('start_char': 0, 'end_char': 6, 'entity': 'g...
Crypto Fear and Greed Index hits 30, lowest le...	[crypto, fear, greed, index, hits, lowest, lev...	crypto fear greed index hits lowest level months	{('text': 'crypto', 'label': 'CRYPTOCURRENCY', ...	{('start_char': 0, 'end_char': 6, 'entity': 'c...

Fig: 4.4 CNER with Sentiments

For each headline, the entities are recognized and sentiments are assigned to them.

4.4 Model Building and Evaluation

We tested various BERT-based pre-trained language models, including BERT, RoBERTa, DistilBERT, and FinBERT, along with an RNN model (LSTM). The models were trained on 80% of the dataset and tested on the remaining 20%. The DistilBERT model demonstrated superior performance, achieving the highest accuracy and F1 score among the models tested. As the task is primarily a classification task, DistilBERT was utilized with a classification head. The training parameters were carefully selected and optimized through multiple iterations and testing.

4.4.2 Fine-Tuning: The transformer models, including DistilBERT, were fine-tuned on the custom dataset of cryptocurrency-related headlines. The fine-tuning process involved adjusting the model's parameters on a specific portion of the dataset (80% for training and 20% for testing) to optimize its performance on the classification task. This fine-tuning step was crucial to adapt the pre-trained models to the specific vocabulary and context of cryptocurrency news, enhancing the accuracy and relevance of the predictions.

4.4.3 Hyper-parameter tuning: There were huge losses and poor classification report generated by most of the models, and the necessitated the need to tune the hyper-parameters of the PLMs. The hyper-parameters tuned for better performances of the PLMs are the learning rate for the PLMs, the batch sizes, the number of epochs, the warm up steps, the logging steps, the weight decay and the value of epsilon. After trying out lots of experiments, the work finally settled with the following hyper-parameters for the proposed model to get a better performance for each of the PLMs.

```
4.  
5.         num_train_epochs=7,  
6.         per_device_train_batch_size=16,  
7.         per_device_eval_batch_size=16,  
8.         warmup_steps=500,  
9.         weight_decay=0.01,  
10.        logging_dir='./logs',  
11.        logging_steps=10,
```

```
12.         eval_strategy="epoch",
13.         report_to="none",
14.         learning_rate=1e-5
15.     )
```

Fig: 4.5 Training arguments

4.4.4 Training arguments explanation:

1. `num_train_epochs`: This specifies the number of times the model will iterate over the entire training dataset. More epochs generally improve the model's learning, but too many can lead to overfitting.
2. `per_device_train_batch_size`: The number of training samples processed together in one forward/backward pass on each device (e.g., GPU). Larger batch sizes can speed up training but require more memory.
3. `per_device_eval_batch_size`: Similar to the training batch size, but used during evaluation (validation). It's typically set equal to or smaller than the training batch size.
4. `warmup_steps`: The number of steps during which the learning rate gradually increases from zero to the specified value. This helps stabilize training early on by avoiding sudden large updates.
5. `weight_decay`: A regularization technique that reduces the size of model weights during training to prevent overfitting. It's applied to the learning updates to encourage smaller weight magnitudes.
6. `logging_steps`: The frequency (in steps) at which training logs (e.g., loss, accuracy) are reported. This helps monitor the training process and ensure the model is learning as expected.
7. `evaluation_strategy`: Specifies when and how often to evaluate the model during training, such as after each epoch or at specified intervals. It helps in tracking validation performance and adjusting the training process.
8. `report_to`: Determines where to report training metrics (e.g., "tensorboard", "wandb"). It integrates with various tools for visualizing and tracking training progress.
9. `learning_rate`: The step size at which the model's parameters are updated during training. It's a crucial hyperparameter that controls the speed and convergence of the training process.

Chapter 5: Results and Discussion

In this section, we discuss the performance of the model using a confusion matrix and its derivatives such as precision, accuracy, recall and F-score. And also, we compare the LSTM models by hyper parameter tuning its parameters. Classification matrix is a $n \times n$ matrix, where n is the number that describes the performance of a classifier.

The equations to calculate the confusion matrix configurations are as follows:

Accuracy: The accuracy is calculated as the ratio of the true prediction values to all other predictions The equation is as follows:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

Precision: Precision is calculated as the ratio of true positive predictions to all other positive predictions

$$\text{Precision} = TP / (TP + FP)$$

Recall: Recall is calculated as the ratio of true positive predictions to all the actual positive

$$\text{Recall} = TP / (TP + FN)$$

F1 Score: F1-scores are calculated by taking the harmonic mean of both the precision and recall of a classifier to come up with a single metric

$$\text{F1 Score} = TP / TP + (1/2)(FP + FN)$$

The results section includes the evaluation metrics for each model, summarized in the table below:

Model	Accuracy	F1 Score	Precision	Recall
DistilBERT	0.8528	0.74818	0.7738	0.7324
BERT	0.8358	0.72310	0.7321	0.7158
RoBERTa	0.7292	0.4766	0.5108	0.4707
FinBERT	0.6908	0.4821	0.4826	0.4826
LSTM	0.7441	0.4394	0.7398	0.4336

Table 5.1: Performance scores of the models used

5.1 Accuracy

The table shows that DistilBERT with an accuracy score of 0.8528 has the best highest accuracy as it outperforms all other model in the table. BERT which is larger in size than DistilBERT posted an accuracy score of 0.8358 which is really close to that of DistilBERT . RoBERTa, FinBERT, and LSTM followed in terms of accuracy score ranking with LSTM being the least with a score of 0.7441, the results on this side of the table gives distilBERT edge over the others.

5.2 F1 Score: DistilBERT after training and testing the finetuned models , evaluating their performance on same metrics posted a F1 score of 0.74818 ranking as the model with the highest score when compared to other models whose performance was also evaluated. F1 score which is a balance between precision and recall tells a better story of a models performance. BERT posted an F1 score of 0.7231 , RoBERTa 0.4766, FinBERT 0.4821 , and LSTM 0.4394. LSTM having the least score here implies that there is less balance between its precision and recall making it least reliable overall.

5.3 Precision: DistilBERT (0.7738) also outperforms all the other models in precision, meaning it is better at minimizing false positives (incorrectly identifying something as positive when it is not).BERT (0.7321) is again the closest, but RoBERTa (0.5108), FinBERT (0.4826), and LSTM (0.7398) struggle much more with precision. Interestingly,

LSTM's precision (0.7398) is surprisingly high compared to its F1 score and recall, suggesting that while LSTM can identify positive cases well, it fails to capture a broader set of correct predictions, which hurts its F1 score.

5.4 Recall: DistilBERT (0.7324) once more shows its balanced strength with a high recall value, meaning it is good at identifying true positives (correctly identifying positive cases). BERT (0.7158) is similarly close but not better. RoBERTa (0.4707), FinBERT (0.4826), and LSTM (0.4336) have considerably lower recall values, meaning they miss many true positives, which is a significant weakness in these models.

5.5.1 DistilBERT vs. BERT: comparing distilBERT to BERT. it is seen that DistilBERT has a slightly better performance than BERT, with its Accuracy and F1 score, precision and recall higher than that of BERT. It is worth remembering that DistilBERT is a lighter, more computationally time efficient version of BERT. Despite being less resource-intensive, these results are impressive as it offers improved performance while being less resource-intensive. It shows that model compression doesn't always lead to a drop in performance, and in this case, DistilBERT outperforms BERT in all metrics.

5.5.2 DistilBERT vs RoBERTa: DistilBERT significantly outperforms in all metrics, which is surprising given that RoBERTa being significantly larger in size having been pre-trained on a corpus larger than that of BERT or DistilBERT usually performs strongly in most NLP tasks. The poor F1 score amongst other reasons could be that it isn't well suited for the task at hand or that it would require more effective fine tuning to produce better results. The result also shows it is missing many true positives as it posted a very low recall of 0.4707.

5.5.3 DistilBERT vs. FinBERT: FinBERT is a model specifically fine-tuned for financial texts, yet DistilBERT shows clear superiority. FinBERT's weak performance across all metrics, especially the lower F1 score and accuracy, suggests it may not generalize well to broader tasks beyond its financial domain.

This shows that task-specific fine-tuning can sometimes limit the versatility of a model.

5.5.4 DistilBERT vs. LSTM:

- **DistilBERT** vastly outperforms the LSTM model in terms of both accuracy and F1 score. While LSTM has a surprisingly high precision (0.7398), its recall (0.4336) is much lower, indicating that it is missing many true positive cases, which leads to a poor F1 score. The dominance of transformer-based models like DistilBERT and BERT over traditional models like LSTM highlights the advantages of transformer architectures in modern NLP tasks.

The model comparison (Accuracy and F1 score) in relation to the DistilBERT model is shown graphically in the bar charts below:

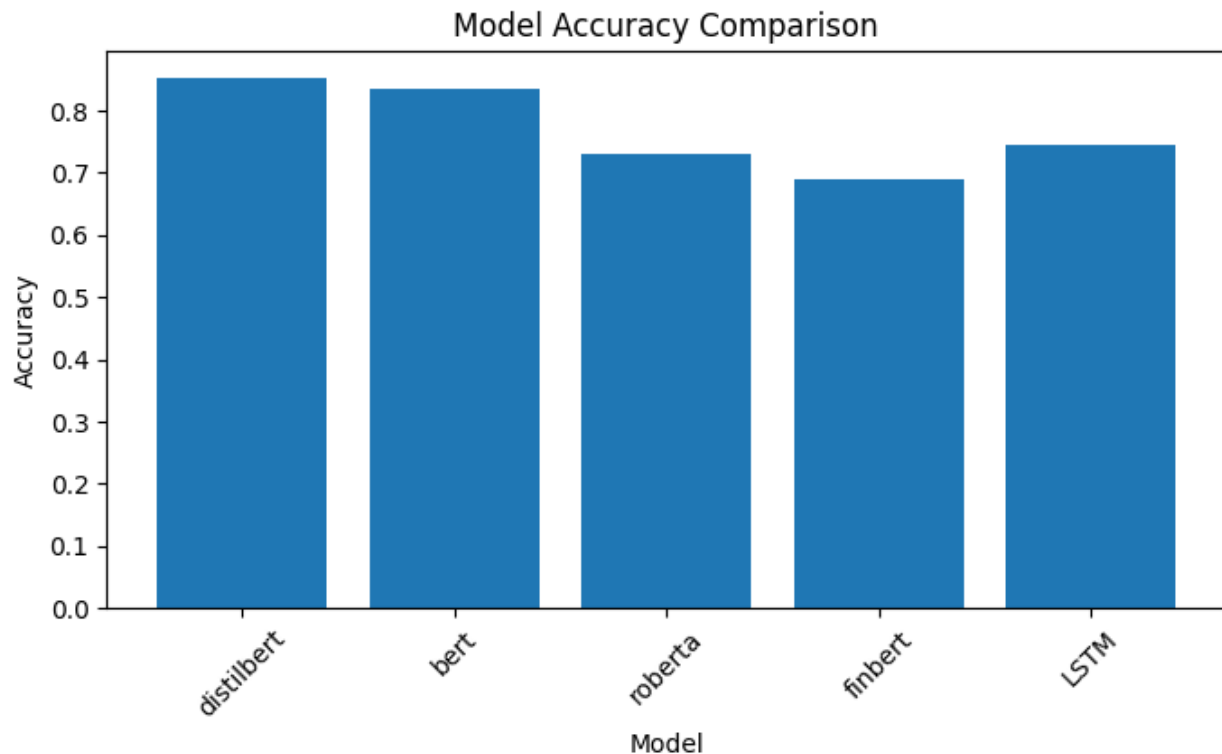


Fig: 5.1 Accuracy Score of the Models

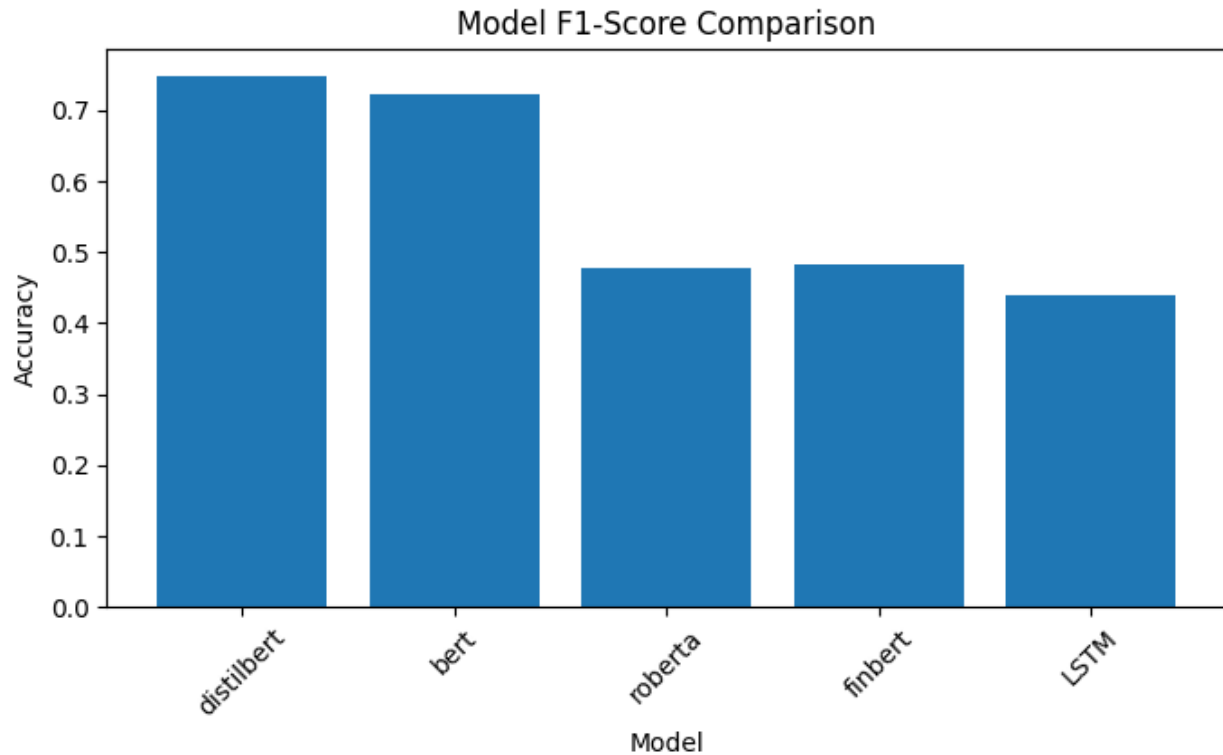


Fig:

5.2 F1 Score of the models

5.6.1 DistilBERT

DistilBERT is a smaller, faster, and lighter version of BERT, created through a process called knowledge distillation. This process involves training a smaller model (DistilBERT) to replicate the behavior of a larger model (BERT) while maintaining most of its accuracy. DistilBERT retains about 97% of BERT's language understanding while being 60% faster and using 40% fewer parameters. It's widely used in situations where computational resources are limited, such as in mobile applications and real-time inference tasks. DistilBERT being the choice model for this study, its performance will then be compared with other models used in this research. . [Sanh, V., 2019]

Analysis of the loss curves

The loss curve gives a visual representation of the model's performance and usually allows one to easily spot overfitting or any bias in the model during training or validation. To be able to further examine the performances of the models in these experiments, the loss curves of the models were plotted during training. The losses were calculated using the cross-entropy loss function and the loss at each epoch training was added to a list and plotted after the whole epoch training.

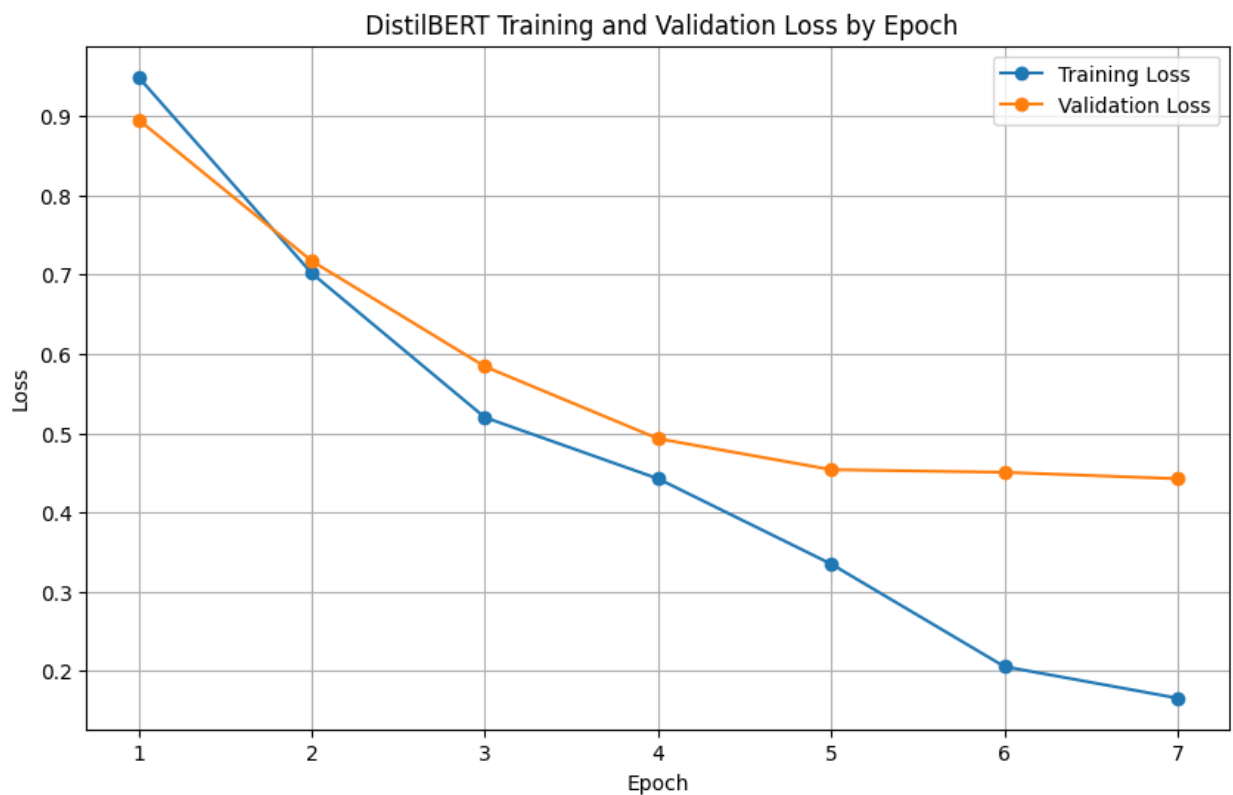


Fig: 5.6.1 Training vs Validation Loss for DistilBert

5.6.2 BERT (Bidirectional Encoder Representations from Transformers)

BERT is a transformer-based model designed by Google that revolutionized NLP by introducing a bidirectional training approach. Unlike traditional models that read text sequentially, BERT reads the text in both directions (left-to-right and right-to-left) simultaneously. This allows it to capture context more effectively. BERT is pre-trained on large text corpora using two unsupervised tasks: masked language modeling (MLM) and

next sentence prediction (NSP), making it highly effective for various downstream NLP tasks like text classification, question answering, and sentiment analysis.

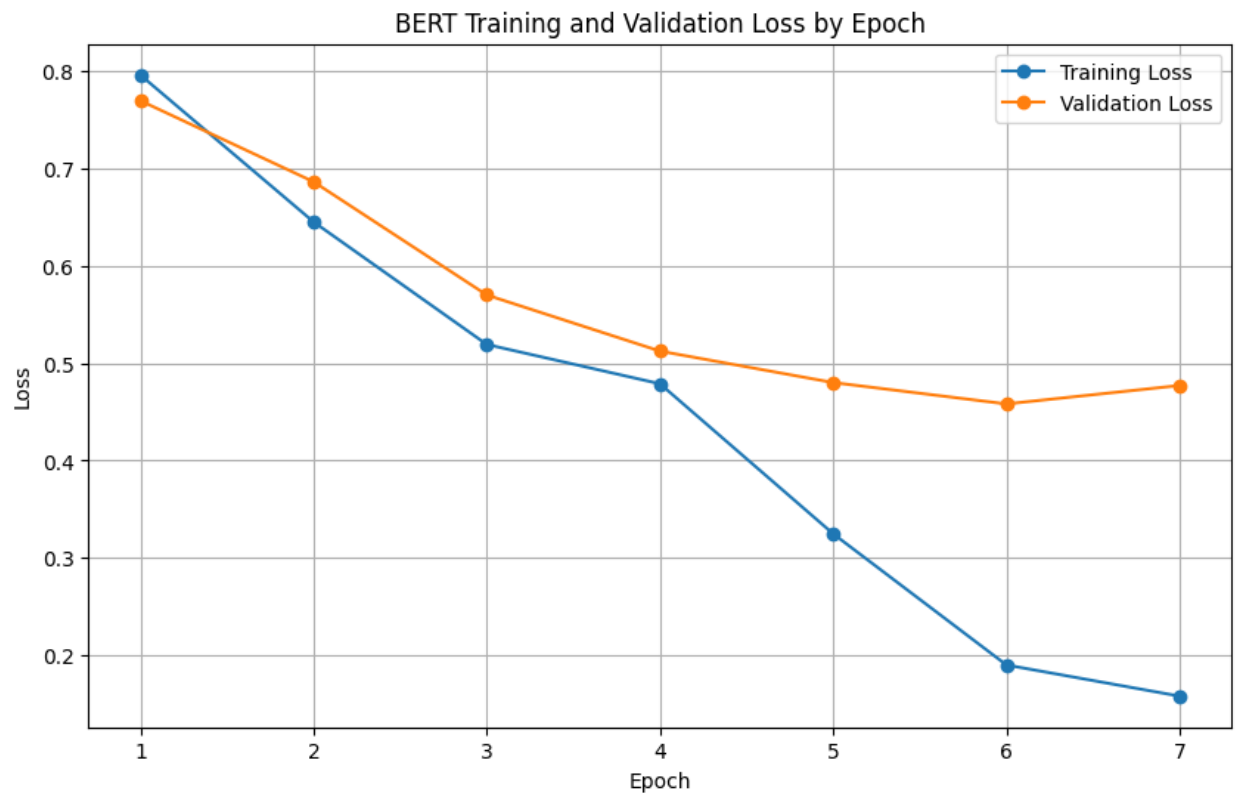


Fig: 5.6.2 Training vs Validation Loss for BERT

5.6.3 RoBERTa (Robustly Optimized BERT Pretraining Approach)

RoBERTa is a variant of BERT developed by Facebook AI that improves upon the original BERT model by optimizing its pre-training procedure. RoBERTa increases the amount of training data, removes the NSP task, and uses dynamic masking rather than static masking during pre-training. These changes allow RoBERTa to achieve better performance on various NLP benchmarks, making it one of the most powerful models for tasks like text classification, summarization, and language translation. (Liu, Y., 2019)

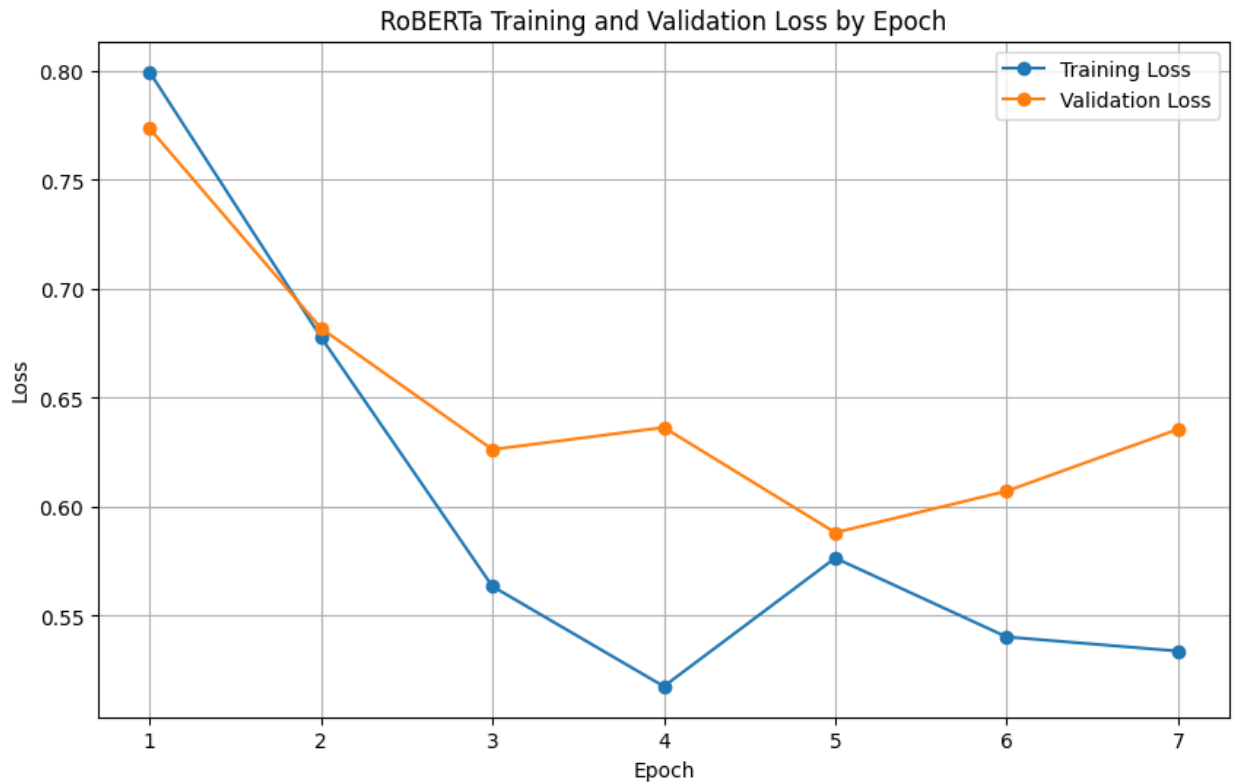
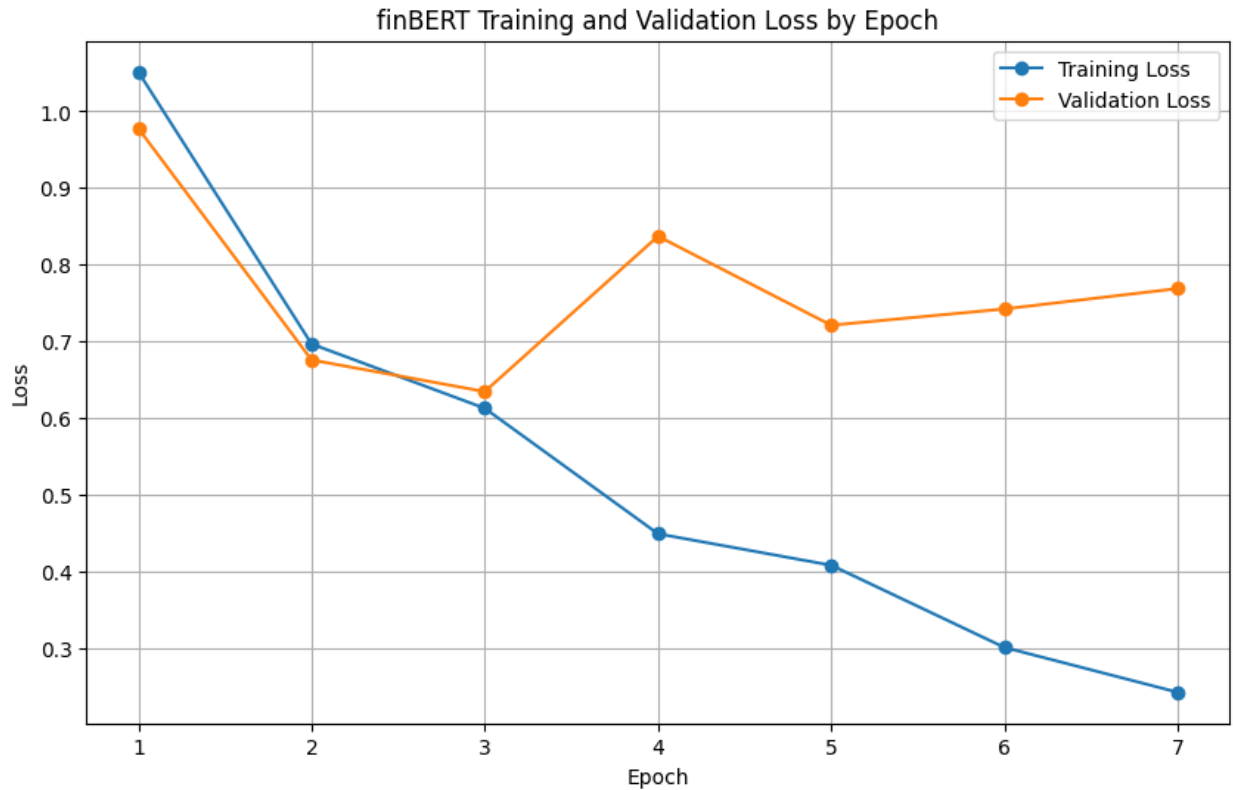


Fig: 5.6.3 Training vs Validation Loss for RoBERTA

5.6.4 FinBERT

FinBERT is a domain-specific variant of BERT, pre-trained on financial texts such as news articles, earnings calls, and analyst reports. Developed by researchers to cater to the financial sector, FinBERT excels at understanding financial terminology and nuances, making it highly effective for tasks like sentiment analysis, risk assessment, and financial forecasting. By focusing on the financial domain, FinBERT delivers more accurate results compared to general-purpose models in financial NLP tasks. (ARaci 2019)



+

Fig: 5.6.4 Training vs Validation Loss for FinBERT

5.6.5 LSTM (Long Short-Term Memory):

LSTM is a type of recurrent neural network (RNN) architecture designed to overcome the vanishing gradient problem commonly encountered in traditional RNNs. LSTMs are capable of learning long-term dependencies by using gates to control the flow of information, making them particularly effective for sequential data tasks such as time series prediction, language modeling, and machine translation. They are designed to remember information for long periods, which makes them powerful for tasks involving temporal dependencies. [Hochreiter and Schmidhuber (1997)]

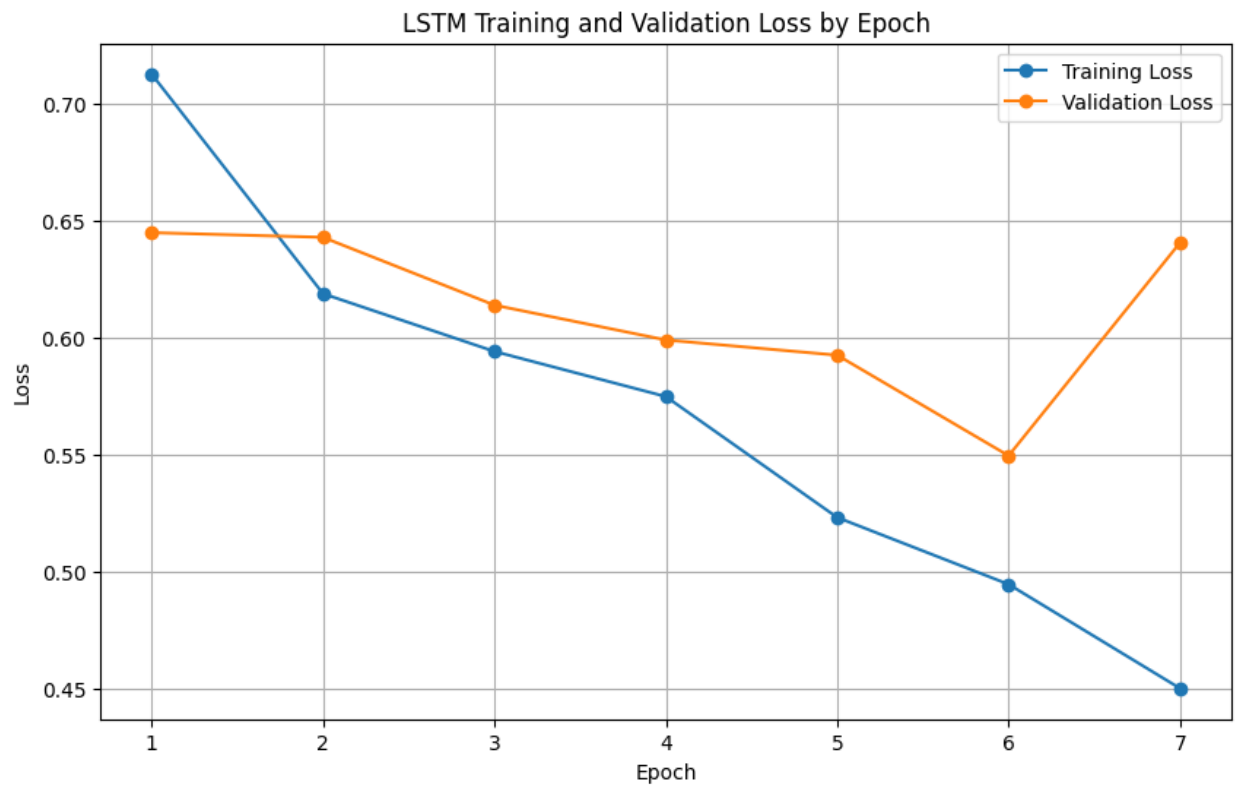


Fig: 5.6.5 Training vs Validation Loss for LSTM

Chapter 6: Limitations, Recommendations and Conclusion

6.1 Limitations

1 transformer vs Non Transformers: the results here further illustrates the gap between traditional recurrent architectures and transformer based model. LSTM performance is worse despite posting good precision score. However, transformer compared to LSTM have high cost of computation.

2. Imbalanced Metrics: Some models like LSTM show high precision but low recall. This imbalance suggests that certain models might be skewed toward avoiding false positives at the cost of missing true positives, which may not be ideal depending on the task.

3. Computation and Resource Intensity. The fact that transformer based models require significant computational power and memory, in areas where resources are constrained, easy deployment on real time applications or mobile phone could be challenging.

6.2 Recommendations for Future Research:

1. Better fine-tuning of models like FinBERT and RoBERTa for specific task.
2. More research into combining the strengths of transformer models and non-transformer models like a hybrid model may help get a better f1 score by giving balance and recall.
3. Handling Imbalanced Data The precision-recall imbalance seen in models like LSTM suggests a need for further exploration into techniques that optimize these metrics in a balanced way. Future work should focus on improving models' ability to handle imbalanced datasets (e.g., using techniques like Focal Loss, re-sampling, or cost-sensitive learning).

6.3 Conclusion

For sentiment analysis of crypto news, **DistilBERT** is the best model based on its performance metrics. If computational efficiency or capturing sequential dependencies is a priority, **LSTM** is a strong alternative. **FinBERT** is a good choice when the analysis is strictly financial, though it doesn't outperform DistilBERT in this scenario. Fine-tuning and further experimentation with BERT and RoBERTa could potentially improve their performance, as the lower f1 score shows the difficulty in achieving good balance between precision and recall, which might require more careful parameter adjustments or training on a larger dataset as they currently lag behind the other models in this specific task.

6.4 Deployment: Streamlit Interface

To facilitate the deployment and user interaction with the model, a Streamlit interface was developed. This interface allows users to input new crypto currency-related headlines and receive entity based recognition and sentiment analysis in real-time. The interface is designed for ease of use, making the sophisticated model accessible to non-technical users.

Entity-based Sentiment Analysis for News Headlines

Enter a news headline to predict its sentiment.

News Headline

Stablecoin Adoption Grows as Major Companies Integrate Crypto Payments

Predict Sentiment

Headline: **Stablecoin Adoption Grows as Major Companies Integrate Crypto Payments**

Entity: **crypto**, Label: **CRYPTOCURRENCY**, Sentiment: **positive**

Fig: 6.1 Streamlit App User interface for crypto currency news text sentiment analysis

Chapter 7: Project plan, timeline of execution, feedback from supervisor

7.1 Project Management

The first read was on NLP in finance. Discussing research area with my supervisor. In the beginning, it was more of ensuring I understood the field of research I was embarking on. Prior to the next meeting we had, my supervisor advised me to go do some readings on BERT and recent research in the area of BERT for Finance. The more I read the more I learnt of the different ways BERT is used for NLP task. Subsequently narrowing down a research area was challenging as datasets which is crucial for the dissertation was not easy to come by. This led to several tweaking of the research area, until a suitable dataset was found. Next thing was to realistically state the project delivery time. As any research has to factor in implementation time. The Project plan and proposal was then submitted. My Supervisor after seeing the time allotted for literature review, quickly advised that more time be given for review of literature and realistically speaking two weeks wasn't going to be enough to exhaust literature in this dissertation. The next hurdle was the ethical application. Approval. The first request didn't explain how the model performance would be assessed. After this was updated the ethical application was then approved. The research didn't progress as the time indicated in the initial project plan as I wasn't meeting the milestones set for each phase at as at when due. My supervisor was always patient with me and kept pointing me in the right direction and constantly advised me on how to be better organised with my research in-order to make the best use of time.

Feedback was given after the scores was released and I immediately set about addressing the issues raised. Most importantly was ensuring that data used for this project is gotten from news sites related to crypto currency. I spent the next 2 weeks webscrapping data required for this work. The next 6 weeks was spent building a working model, while the next two weeks after that wa used to re-write parts of the report. The product then tested and the link to the project folder on github.

REFERENCES

- Alhagry, S., Aly, A. and A., R. (2017). Emotion Recognition based on EEG using LSTM Recurrent Neural Network. *International Journal of Advanced Computer Science and Applications*, 8(10). doi:10.14569/ijacsa.2017.081046.
- Araci, D., 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.
- Atzeni, M., Dridi, A. and Reforgiato Recupero, D., 2017. Fine-grained sentiment analysis on financial microblogs and news headlines. In *Semantic Web Challenges: 4th SemWebEval Challenge at ESWC 2017, Portoroz, Slovenia, May 28-June 1, 2017, Revised Selected Papers*, pp. 124-128. Springer International Publishing.
- Ba, J.L., 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.
- Bahdanau, D., 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Bakshi, R.K., Kaur, N., Kaur, R. and Kaur, G., 2016. Opinion mining and sentiment analysis. [online] *IEEE Xplore*. Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7724305> [Accessed 16 Aug. 2022].
- Chen, J., Liang, J., Liu, B. and Zhang, Q., 2019. Deep learning for sentiment analysis of cryptocurrency news. In *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 1380-1385. IEEE. <https://doi.org/10.1109/ICDM.2019.00158>.
- Corbet, S., Larkin, C., Lucey, B., Meegan, A., and Yarovaya, L., 2019. Cryptocurrency reaction to FOMC announcements: Evidence of heterogeneity based on blockchain stack position. *Journal of Financial Stability*, 46, 100706. <https://doi.org/10.1016/j.jfs.2019.100706>.

Devlin, J., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fisher, I.E., Garnsey, M.R. and Hughes, M.E., 2016. Natural language processing in accounting, auditing and finance: A synthesis of the literature with a roadmap for future research. *Intelligent Systems in Accounting, Finance and Management*, 23(3), pp.157-214.

Fry, J. and Cheah, E.T., 2016. Negative bubbles and shocks in cryptocurrency markets. *International Review of Financial Analysis*, 47, pp. 343-352.
<https://doi.org/10.1016/j.irfa.2016.02.008>.

Gadek, G., Hosen, M., Budiharto, W. and Purnomo, M.H., 2021. Cryptocurrency price prediction using deep learning with long short-term memory and GRU with sentiment and emotion analysis. *Journal of King Saud University-Computer and Information Sciences*. <https://doi.org/10.1016/j.jksuci.2021.09.009>.

Graves, D. and Pedrycz, W., 2009. Fuzzy prediction architecture using recurrent neural networks. *Neurocomputing*, 72(7-9), pp. 1668-1678.
<https://doi.org/10.1016/j.neucom.2008.07.009>.

Halder, S., 2022. Finbert-lstm: Deep learning based stock price prediction using news sentiment analysis. *arXiv preprint arXiv:2211.07392*.

Hochreiter, S. and Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation*, 9(8), pp.1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>.

Jiang, T. and Zeng, A., 2023. Financial sentiment analysis using FinBERT with application in predicting stock movement. *arXiv preprint arXiv:2306.02136*.

Kang, S.H., McIver, R.P. and Hernandez, J.A., 2019. Co-movements between Bitcoin and gold: A wavelet coherence analysis. *Physica A: Statistical Mechanics and its Applications*, 536, 120888. <https://doi.org/10.1016/j.physa.2019.04.124>.

Karalevicius, V., Degrande, N. and De Weerd, J., 2018. Using sentiment analysis to predict interday Bitcoin price movements. *The Journal of Risk Finance*, 19(1), pp. 56-75. <https://doi.org/10.1108/JRF-06-2017-0092>.

Kaur, A. and Gupta, V., 2013. A survey on sentiment analysis and opinion mining techniques. *Journal of Emerging Technologies in Web Intelligence*, 5(4), pp. 367-371.

Khairnar, J. and Kinikar, M., 2013. Machine learning algorithms for opinion mining and sentiment classification. *International Journal of Scientific and Research Publications*, 3(6), pp. 1-6.

Kim, S., Ku, S. and Kim, H.Y., 2021. Predicting cryptocurrency price using news data and social media factors based on machine learning. *Applied Sciences*, 11(10), 4506. <https://doi.org/10.3390/app11104506>.

Lee, D. K. C., Guo, L., & Wang, Y. (2017). Cryptocurrency: A new investment opportunity?. Available at SSRN 2994097.

Li, X., Wu, P. and Wang, W., 2020. Incorporating stock prices and news sentiments for stock market prediction: A case of Hong Kong. *Information Processing & Management*, 57(5), 102212. <https://doi.org/10.1016/j.ipm.2020.102212>.

Liu, Y., 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Liu, Y. and Selover, D., 2021. An exploration of the nexus between cryptocurrency appreciation and rumors. *Journal of Financial Crime*. <https://doi.org/10.1108/JFC-01-2021-0014>.

Loughran, T. and McDonald, B., 2011. When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of Finance*, 66(1), pp. 35-65.

Mai, F., Shan, Z., Bai, Q., Wang, X. and Chiang, R.H., 2018. How does social media impact Bitcoin value? A test of the silent majority hypothesis. *Journal of Management Information Systems*, 35(1), pp. 19-52. <https://doi.org/10.1080/07421222.2018.1440774>.

Margherita, G., Enrico, B. and Giorgio, V., 2020. Metrics for multi-class classification: An overview. <https://arxiv.org/abs/2008.05756>.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S. and Dean, J., 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26.

Nguyen, Q.T., Nguyen, T.L., Luong, N.H. and Ngo, Q.H., 2020, November. Fine-tuning bert for sentiment analysis of vietnamese reviews. In *2020 7th NAFOSTED conference on information and computer science (NICS)* (pp. 302-307). IEEE.

Pang, B., Lee, L. and Vaithyanathan, S., 2002. Thumbs up? Sentiment classification using machine learning techniques. *arXiv preprint cs/0205070*.

Park, A., Sabourian, H. and Sgroi, D., 2016. Learning, belief elicitation and trading in asset markets. Working Paper, University of Toronto.

Salant, S. and Berant, J., 2017. Contextualized word representations for reading comprehension. *arXiv preprint arXiv:1712.03609*.

Sanh, V., 2019. DistilBERT, A Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *arXiv preprint arXiv:1910.01108*.

Salmon, B.P., Kleynhans, W., Schwegmann, C.P. and Olivier, J.C., 2015. Proper comparison among methods using a confusion matrix. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* (pp. 3057-3060). IEEE.

Schumaker, R.P. and Chen, H., 2009. Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Transactions on Information Systems (TOIS)*, 27(2), pp. 1-19. <https://doi.org/10.1145/1462198.1462204>.

Shams, A., 2020. The structure of cryptocurrency returns. *Journal of Alternative Investments*, 22(4), pp. 8-28. <https://doi.org/10.3905/jai.2020.1.110>.

Smales, L.A., 2019. Bitcoin as a safe haven: Is it even worth considering? *Finance Research Letters*, 30, pp. 385-393. <https://doi.org/10.1016/j.frl.2018.11.002>.

Sohangir, S., Petty, N. and Wang, D., 2018. Financial sentiment lexicon analysis. In *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, pp. 286-289. IEEE.

Tang, J., Jiang, B. and He, L., 2021. Interpretable text-driven neural network for cryptocurrency price prediction. *Expert Systems with Applications*, 173, 114632. <https://doi.org/10.1016/j.eswa.2021.114632>.

Tang, Y., Yang, Y., Huang, A.H., Tam, A. and Tang, J.Z. (2023). FinEntity: Entity-level Sentiment Classification for Financial Texts. [online] arXiv.org. doi:<https://doi.org/10.48550/arXiv.2310.12406>.

Turney, P.D. and Pantel, P., 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37, pp. 141-188.

Vaswani, A., 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Vosoughi, S., Roy, D. and Aral, S., 2018. The spread of true and false news online. *Science*, 359(6380), pp. 1146-1151. <https://doi.org/10.1126/science.aap9559>.

Wang, S. and Luo, Y., 2020. Cryptocurrency price prediction based on deep learning. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 14(4), pp. 56-68. <https://doi.org/10.4018/IJCINI.2020100105>.

Xie, P., Chen, H. and Hu, Y.J., 2019. Predicting and understanding cryptocurrency price using machine learning. In *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 1373-1379. IEEE. <https://doi.org/10.1109/ICDM.2019.00157>.

Zhang, Y. and Zhang, H., 2023. FinBERT–MRC: financial named entity recognition using BERT under the machine reading comprehension paradigm. *Neural Processing Letters*, 55(6), pp. 7393-7413.

Zhao, L., Li, L., Zheng, X. and Zhang, J., 2021. A BERT based sentiment analysis and key entity detection approach for online financial texts. In *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 1233-1238. IEEE.

Zhu, Y., 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *arXiv preprint arXiv:1506.06724*.

APPENDIX

The python code of the study can accessible in the Github link below:

https://github.com/1NTRO4/crypto_news

please read the Readme file and follow the steps listed to be able to run the streamlitt App to launch the streamlit app for sentiment analysis task.

Making Predictions with the Fine-Tuned Model

A streamlit app interface is used for sentiments prediction.

To run this model take note of the following steps

- Clone the repo

```
git clone https://github.com/1NTR04/crypto_news  
cd crypto_news
```

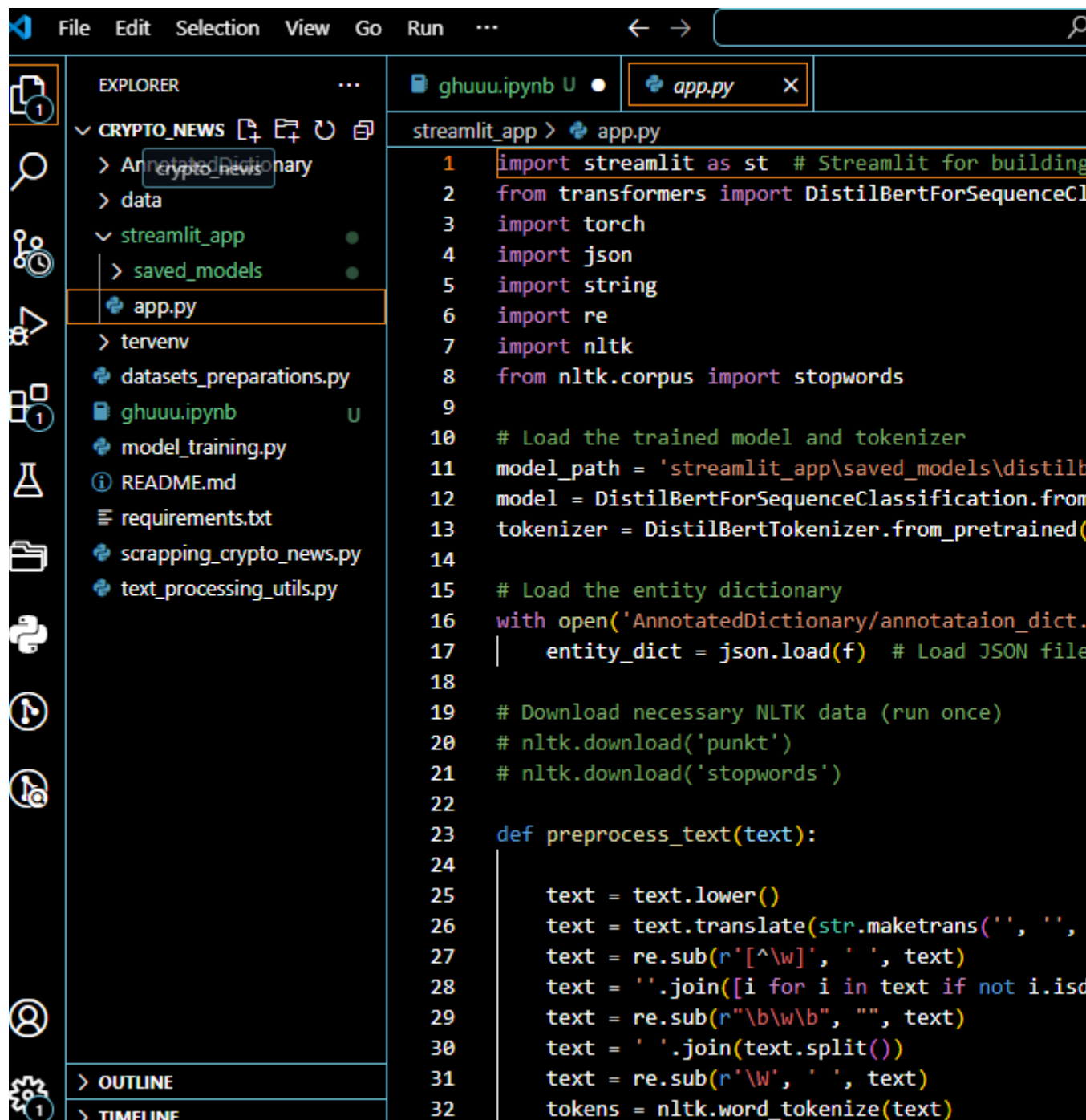
- Create a folder in `streamlit_app` folder named `saved_models`
- Download the already fine-tuned model titled `distilbert_class_head` from [google drive](#) folder
- Setup a virtual enviroment
- Install the packages

```
pip install -r requirements.txt
```

- Launch sentiment analysis streamlit app

```
streamlit run streamlit_app/app.py
```

Some part of the streamlit python script is show below



```
File Edit Selection View Go Run ... < >
EXPLORER
CRYPTO_NEWS
  > AnnotatedDictionary
  > data
  > streamlit_app
    > saved_models
      app.py
  > tervenv
  datasets_preparations.py
  ghuuu.ipynb
  model_training.py
  README.md
  requirements.txt
  scrapping_crypto_news.py
  text_processing_utils.py
OUTLINE
TIMELINE

streamlit_app > app.py
1 import streamlit as st # Streamlit for building
2 from transformers import DistilBertForSequenceCl
3 import torch
4 import json
5 import string
6 import re
7 import nltk
8 from nltk.corpus import stopwords
9
10 # Load the trained model and tokenizer
11 model_path = 'streamlit_app\saved_models\distilb
12 model = DistilBertForSequenceClassification.from
13 tokenizer = DistilBertTokenizer.from_pretrained(
14
15 # Load the entity dictionary
16 with open('AnnotatedDictionary/annotataion_dict.
17 | entity_dict = json.load(f) # Load JSON file
18
19 # Download necessary NLTK data (run once)
20 # nltk.download('punkt')
21 # nltk.download('stopwords')
22
23 def preprocess_text(text):
24
25     text = text.lower()
26     text = text.translate(str.maketrans('', '',
27     text = re.sub(r'^\w', ' ', text)
28     text = ''.join([i for i in text if not i.isc
29     text = re.sub(r"\b\w\b", "", text)
30     text = ' '.join(text.split())
31     text = re.sub(r'\W', ' ', text)
32     tokens = nltk.word_tokenize(text)
```


EXPLORER

CRYPTO_NEWS

> AnnotatedDictionary

> data

streamlit_app

> saved_models

app.py

tervenv

datasets_preparations.py

ghuuu.ipynb

model_training.py

README.md

requirements.txt

scrapping_crypto_news.py

text_processing_utils.py

ghuuu.ipynb

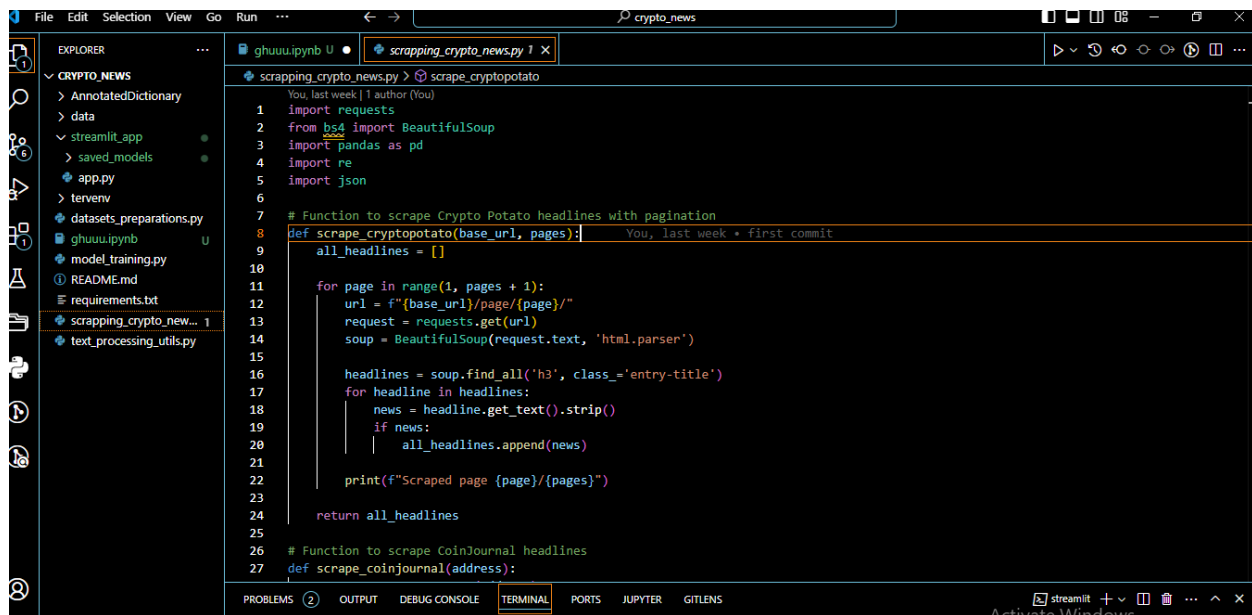
app.py

streamlit_app > app.py > predict_sentiment_and_entities

```
51 def predict_sentiment_and_entities(model, tokeniz
92     results.append(headline_results) # Append th
93     return results # Return the results
94
95 # Streamlit interface for user interaction
96
97 st.title("Entity-based Sentiment Analysis for New
98 st.write("Enter a news headline to predict its se
99
100
101 headline = st.text_input("News Headline")
102
103 # Button to trigger sentiment prediction
104 if st.button("Predict Sentiment"):
105     if headline: # Check if a headline has been
106         cleaned_headline = preprocess_text(headli
107         results = predict_sentiment_and_entities(
108         if results: # Check if there are any res
109             for result in results: # Iterate th
110                 st.write(f"Headline: **{headline}
111                 for entity_info in result['entiti
112                     st.write(f"Entity: **{entity_
113             else:
114                 st.write("No recognized entities in t
115         else:
116             st.write("Please enter a news headline.")
117
118
119
```

OUTLINE

Web scrapping function



The screenshot shows a Jupyter Notebook window with the file explorer on the left and the code editor on the right. The file explorer shows a directory named 'CRYPTO_NEWS' with several files. The code editor shows the following code:

```
1 import requests
2 from bs4 import BeautifulSoup
3 import pandas as pd
4 import re
5 import json
6
7 # Function to scrape Crypto Potato headlines with pagination
8 def scrape_cryptopotato(base_url, pages):
9     all_headlines = []
10
11     for page in range(1, pages + 1):
12         url = f"{base_url}/page/{page}/"
13         request = requests.get(url)
14         soup = BeautifulSoup(request.text, 'html.parser')
15
16         headlines = soup.find_all('h3', class_='entry-title')
17         for headline in headlines:
18             news = headline.get_text().strip()
19             if news:
20                 all_headlines.append(news)
21
22         print(f"Scraped page {page}/{pages}")
23
24     return all_headlines
25
26 # Function to scrape CoinJournal headlines
27 def scrape_coinjournal(address):
```



The screenshot shows the same Jupyter Notebook window, but with the code editor scrolled down to show the second part of the script. The code continues as follows:

```
28     request = requests.get(address)
29     soup = BeautifulSoup(request.text, 'html.parser')
30     anchor = soup.find_all('h2')
31
32     headline_news = [headline.get_text().strip() for headline in anchor if headline.get_text().strip()]
33     return headline_news
34
35 # Function to scrape Crypto Times headlines
36 def scrape_cryptotimes(address):
37     request = requests.get(address)
38     soup = BeautifulSoup(request.text, 'html.parser')
39     anchor = soup.find_all('a', {"class": 'p-flink'})
40
41     headline_news = [headline['title'] for headline in anchor if 'title' in headline.attrs]
42     return headline_news
43
44 # Function to scrape NewsBTC headlines
45 def scrape_newsbtc(address):
46     request = requests.get(address)
47     soup = BeautifulSoup(request.text, 'html.parser')
48     anchor = soup.find_all("h4", {"class": 'block-article_title'})
49
50     headline_news = [headline.get_text().strip() for headline in anchor if headline.get_text().strip()]
51     return headline_news
```