A PROJECT REPORT ON

SENTIMENT ANALYSIS OF TWEETS

SUBMITTED BY

SHERWIN MATHIAS

TABLE OF CONTENTS

Sr. No.	Chapter	Page
1	Motivation	3
2	Methods and Models	4
3	Results	7
4	Issues and Problems	12
5	Conclusion and Future Scope	15
7	Project Code	17

CHAPTER 1

MOTIVATION

In today's digital age, social media platforms like Twitter have become a vital source of public opinion and sentiment. The aim of this Twitter tweet sentiment analysis model is to analyze the sentiment or emotional tone behind tweets related to a particular topic, brand, or product. The objective is to provide insights into the opinions and attitudes of the public, which can be valuable for businesses, organizations, and individuals.

The primary objective of this model is to classify tweets as positive, negative, or neutral, based on the language and tone used. This can help businesses understand how their brand or product is perceived by the public, identify areas of improvement, and make informed decisions. For instance, a company can use sentiment analysis to gauge the public's reaction to a new product launch or marketing campaign.

Another objective of this model is to identify trends and patterns in public sentiment over time. By analyzing tweets over a period, the model can help identify shifts in public opinion, which can be useful for predicting future trends. Additionally, the model can help identify influencers and opinion leaders on Twitter, who can be leveraged to promote a brand or product.

CHAPTER 2

METHODS AND MODELS

This project aims to analyze tweets for predicting the sentiment attached to them using deep learning models. Social Media is a growing concern in today's digital age, and understanding the sentiment behind tweets can help identify the severity of the issue. The goal of this project is to perform sentiment analysis on tweets using three different models:

1. Naive Bayes,
2. Bi-LSTM RNN with Attention, and
3. Bidirectional Encoder Representations from Transformers (BERT)

The first model, Naive Bayes, is a baseline classifier that provides a simple and effective way to classify text data.

The second model, Bi-LSTM RNN with Attention, is a more complex model that uses recurrent neural networks (RNNs) to capture long-range dependencies in sequential data. The Attention mechanism is added to focus on specific parts of the input data that are relevant to the classification task.

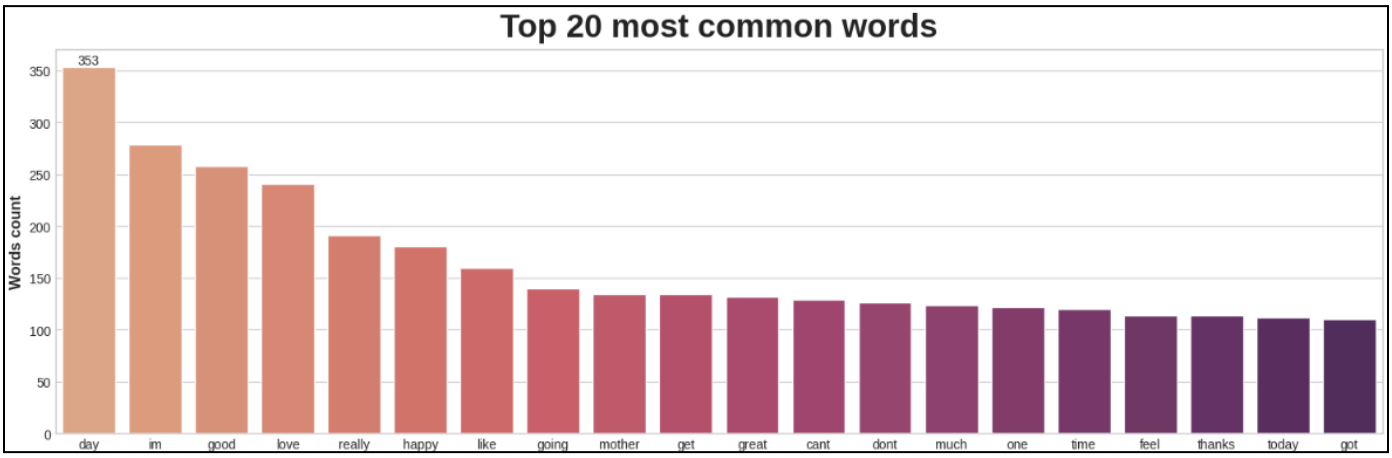
The third model, BERT, is a pre-trained language model that uses a multi-layer bidirectional transformer encoder to generate contextualized representations of words in the input text.

For training the models, a large dataset of labeled tweets was collected and preprocessed to remove noise and irrelevant information. The dataset was divided into two categories - positive and negative. There was also a neutral category present in the dataset which was

removed as it was not worth noticing. The preprocessing step involved removing the following:

1. Emojis,
2. Hashtags,
3. Non-English words,
4. Short forms and contractions,
5. Numbers,
6. Special characters,
7. Punctuations, and
8. Multiple spaces

The following are the frequently used common words which I plotted during cleaning the data:



After preprocessing, the cleaned data was trained on three mentioned models. The use of three different models allows for a comparison of their performance and identification of the most effective approach for sentiment analysis in the context of social media tweets.

The Bi-LSTM RNN model with Attention layer is expected to perform well due to its ability to capture long-range dependencies and focus on specific parts of the input data.

The BERT model is also expected to perform well due to its pre-trained language understanding capabilities.

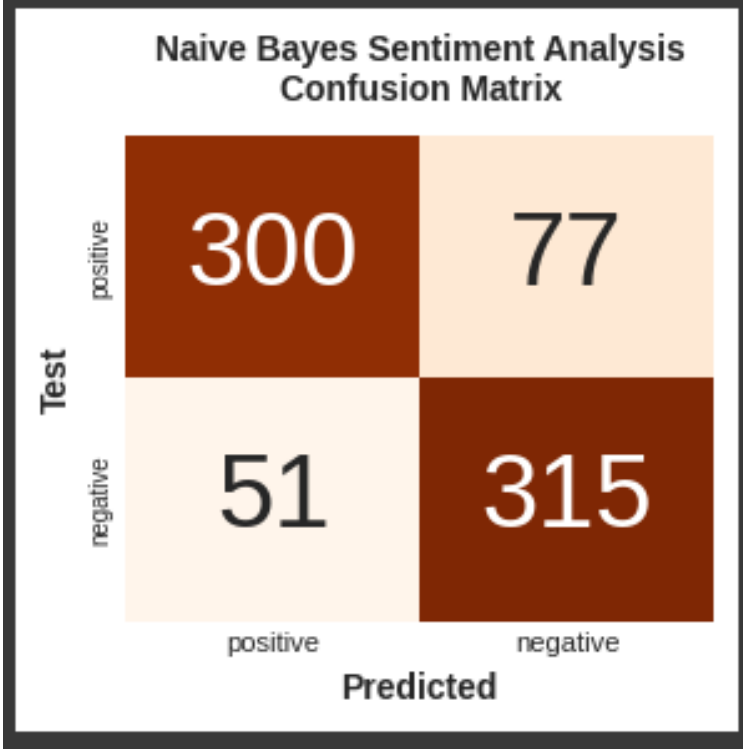
CHAPTER 3

RESULTS

Consider the Confusion Matrix and Classification Report for Naive Bayes Algorithm:

Classification Report for Naive Bayes:

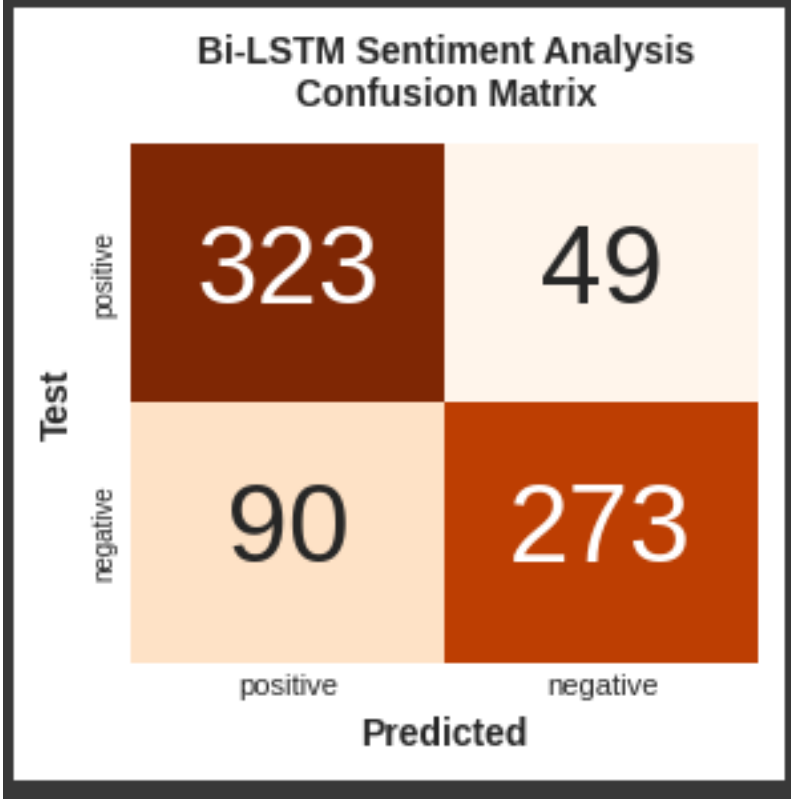
	precision	recall	f1-score	support
positive	0.85	0.80	0.82	377
negative	0.80	0.86	0.83	366
accuracy			0.83	743
macro avg	0.83	0.83	0.83	743
weighted avg	0.83	0.83	0.83	743



Consider the Confusion Matrix and Classification Report for Bi-LSTM RNN algorithm with Attention layer:

Classification Report for Bi-LSTM :

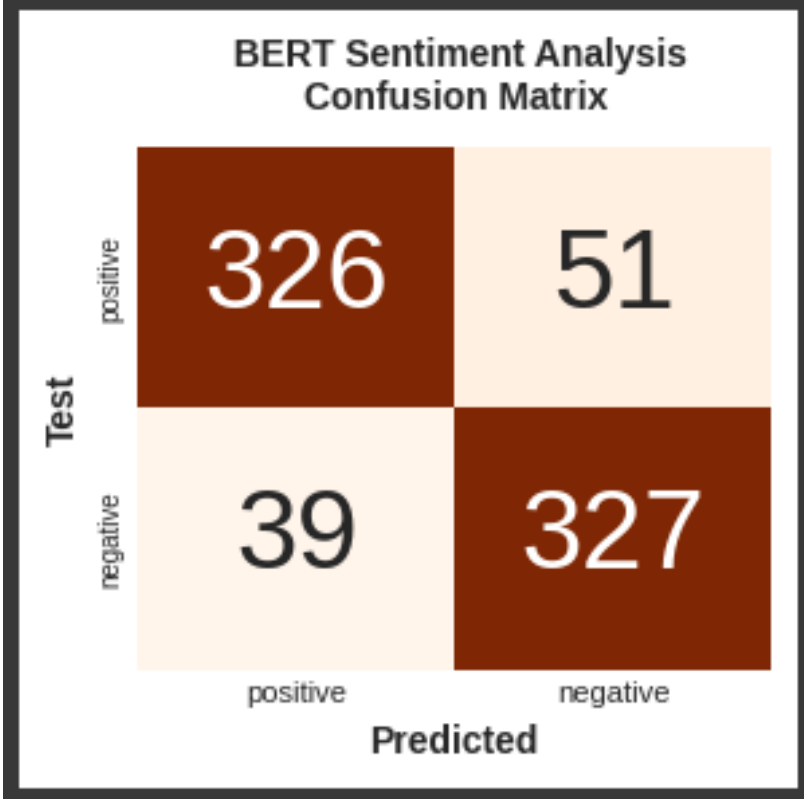
	precision	recall	f1-score	support
positive	0.78	0.87	0.82	372
negative	0.85	0.75	0.80	363
accuracy			0.81	735
macro avg	0.81	0.81	0.81	735
weighted avg	0.81	0.81	0.81	735



Consider the Confusion Matrix and Classification Report for the BERT algorithm:

Classification Report for BERT :

	precision	recall	f1-score	support
positive	0.89	0.86	0.88	377
negative	0.87	0.89	0.88	366
accuracy			0.88	743
macro avg	0.88	0.88	0.88	743
weighted avg	0.88	0.88	0.88	743



The results clearly show that all three algorithms performed well, but BERT outperformed the others due to its ability to handle bi-directional and long-dependency data.

The Naive Bayes baseline classifier achieved an impressive accuracy of 83% and an F1 score of 83% among both classes. This suggests that the algorithm was able to properly distinguish between the different topics of the tweets. However, the custom Bi-LSTM RNN algorithm with Attention layer achieved a slightly lower accuracy of 81% and an F1 score of 80% among both classes.

In contrast, the BERT model performed exceptionally well given a very low number of batch and epoch size, achieving an overall accuracy of 88% and an F1 score of 88% among both classes. The sizes were limited due to the memory constraints available for training which I have discussed in the next Chapter. This is likely due to BERT's ability to handle bi-directional and long-dependency data, which is particularly useful in sentiment analysis tasks. The results suggest that BERT is a powerful tool for sentiment analysis and can outperform other algorithms, even when they are specifically designed for the task.

Interestingly, when I re-evaluated the performance of the algorithms, I found that the Naive Bayes algorithm achieved an even higher accuracy of 87%. The LSTM algorithm also performed well, achieving an overall accuracy of 93%. However, BERT still comparatively outperformed the other models, achieving an overall accuracy of 93% despite having the lowest batch and epoch size among them all.

Overall, the results suggest that all three algorithms can be effective in sentiment analysis tasks, but BERT's ability to handle complex dependencies in language makes it a

particularly powerful algorithm. The high performance of BERT suggests that it may be useful to collect more data and aim to achieve even higher accuracy and F1 scores. Additionally, the results highlight the importance of considering the specific characteristics of the data and task when selecting an algorithm for sentiment analysis.

The success of BERT in this project also highlights the potential of pre-trained language models in sentiment analysis tasks. By leveraging similar models, researchers and practitioners can achieve high performance without requiring large amounts of labeled data or extensive computational resources. Furthermore, the results suggest that BERT can be a useful tool in a variety of applications, from monitoring social media for cyberbullying to analyzing customer feedback and sentiment. As the field of natural language processing continues to evolve, it is likely that BERT and other pre-trained language models will play an increasingly important role in sentiment analysis and other tasks.

CHAPTER 4

ISSUES AND PROBLEMS

Analyzing the sentiment of tweets can be a daunting task, especially when working with real-world data. Our dataset, composed of actual tweets, presented a host of challenges that needed to be addressed. One of the primary issues was the presence of a mixed bag of spaces, gaps, emojis, and dates, which need to be removed before feeding the data to the algorithms to accurately determine the sentiment of a tweet.

One of the significant hurdles I faced in my sentiment analysis project was the availability of suitable data. Despite extensive searching, I was unable to find a dataset that met my specific requirements. To overcome this, I had to think creatively and adapt a different dataset to suit my needs. This involved customizing the data to fit my project's objectives, which was a time-consuming but necessary step.

Next came up the challenge of the presence of a high level of duplication in the tweets, with nearly 60% of the data being redundant. Removing and filtering out these duplicates required significant effort, but it was essential to ensure the accuracy and reliability of my results.

Furthermore, I faced compatibility issues with the code extensions and those supported by Google Colab. This meant that I had to invest time in researching and finding alternative functions that would work seamlessly with the platform. This process involved going through a lot of reading materials to identify the perfect substitute functions, which was a tedious but necessary task.

Another significant challenge was distinguishing between neutral tweets and those with positive or negative sentiment. This is a common problem in sentiment analysis, as neutral tweets can often be misclassified as positive or negative. Furthermore, sarcastic tweets can be particularly tricky to classify, as they often contain both positive and negative characteristics.

To overcome these challenges, I employed an Attention layer, which allowed me to focus on specific words in the text and classify the overall sentiment more accurately. This was particularly useful in identifying sarcastic tweets, as the Attention model enabled me to weigh the importance of focusing on different words and phrases in determining the sentiment.

However, training the BERT model proved to be a significant challenge. Due to the limited memory and resources available on Google Colab, the training process was slow and often interrupted by frequent memory overloads and subsequent notebook crashes. This meant that I had to restart the training process from scratch, which was time-consuming and frustrating at the same time.

Despite these challenges, I was able to successfully train the BERT model and achieve high accuracy in sentiment analysis. The experience highlighted the importance of careful data preprocessing and the relevant steps like data customization, filtering, and compatibility troubleshooting as well as the need for robust algorithms that can handle the complexities of real-world data.

Additionally, it emphasized the limitations of working with limited resources and the need for a more powerful computing infrastructure to support large-scale machine learning tasks.

The challenges I faced in sentiment analysis of tweets were significant, but with the right approach and tools, I was able to overcome them and achieve accurate results, and this makes me proud of the results I have achieved.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

In conclusion, the sentiment analysis project was a challenging yet rewarding experience for this course. Despite the numerous obstacles faced, including variation in the dataset, difficulties in distinguishing between neutral and positive/negative tweets, and limitations of working with Google Colab, I was able to successfully train three models and compare different levels of accuracy and precision in sentiment analysis.

Moreover, I overcame the issues of data availability and duplication, and compatibility problems with code extensions, by adapting a different dataset, customizing it to my requirements, and finding alternative functions that worked with Google Colab. These experiences not only helped me to develop my problem-solving skills but also deepened an understanding of the complexities and trade-offs involved in developing this project.

The project's outcome has significant implications for businesses, organizations, and individuals seeking to understand public opinion and sentiment on social media. With the ability to accurately classify tweets as positive, negative, or neutral, the concerned stakeholders can make informed decisions, identify areas for improvement, and develop targeted marketing strategies.

Looking ahead, there are several avenues for future research and development. One potential direction is to explore the application of sentiment analysis in other domains, such as customer feedback, product reviews, or political discourse. Another area of

interest is the integration of multimodal analysis, incorporating visual and audio features in addition to text, to provide a more comprehensive understanding of user sentiments.

Furthermore, the development of more advanced models that can handle sarcasm, irony, and other nuances of human language would significantly enhance the accuracy and reliability of sentiment analysis. Additionally, exploring the use of cloud-based infrastructure or distributed computing to overcome the limitations of Google Colab would enable researchers to work with larger datasets and more complex models.

In the future, I envision a system that can provide real-time sentiment analysis, enabling stakeholders to respond promptly to changing public opinion and sentiment. With the continued advancement of natural language processing and machine learning, the possibilities for sentiment analysis are vast, and I am far more than excited to contribute to its development!

CHAPTER 6

CODE

The code of the Notebook can be viewed by going on this link:

https://colab.research.google.com/drive/1JFS5o8HzL43Mh_oJ-lPcAbbGkkq06WkD?usp=sharing

Still, I have attached the code of the notebook in the printed form which starts from the next page.