Sentiment Analysis of Real-Time Tweets

A. Saras Chandrika - 21BDS003, Chandana R - 21BDS010, G. Thanmai - 21BDS033
Sentimental Analysis of Real Time Tweets
Data Science and Artificial Intelligence
Indian Institute of Information Technology Dharwad
Email: 21bds003@iiitdwd.ac.in, 21bds010@iiitdwd.ac.in, 21bds033@iiitdwd.ac.in

***Abstract*—The wide spread of World Wide Web has brought a new way of expressing the sentiments of individuals.In this project, we conducted sentiment analysis on a dataset comprising 163,000 tweets obtained from Kaggle. The dataset contains real-time tweets and their respective sentiment labels (-1 for negative, 0 for neutral, and 1 for positive).Naive Bayes and Bag of Words Vectorization-based models were utilized as traditional machine learning approaches, LSTM-based models representing a recurrent neural network (RNN) architecture, while BERT ,a pre-trained language model are used. After comparative analysis, the BERT model demonstrated the highest accuracy in sentiment prediction of tweetsAdditionally, a web interface was created using Flask, enabling users to input a tweet and view the predicted sentiment.**

I. Introduction

Sentiment Analysis, a vital technique in text analysis, plays a significant role in understanding the emotional tones of text, categorizing them into Positive, Neutral, or Negative sentiments. Its main purpose is to grasp the essence of any given data or topic.

Over the recent years, the explosion of micro-blogging platforms has mirrored the tremendous growth in online user engagement, particularly on social media platforms. This surge has empowered users to voice their opinions, share feedback, and express sentiments, shaping perceptions of individuals and organizations alike.

Twitter, boasting approximately 145 million daily active users, stands as a giant in the microblogging realm. With such a massive user base, the platform serves as a valuable resource for sentiment analysis. Its real-time nature and vast user engagement make it an invaluable resource, especially for monitoring user responses to new products or brands.

The direct and immediate feedback provided by sentiment analysis on Twitter enables businesses and organizations to gain insights into consumer sentiment, allowing them to make informed decisions and tailor their strategies accordingly. Given Twitter's reputation and widespread adoption, sentiment analysis emerges as a crucial tool across a multitude of domains, facilitating nuanced understanding and actionable insights.

Identify applicable funding agency here. If none, delete this.

II. Dataset Explanation

The dataset utilized in this model was obtained from the Kaggle website. It comprises 1,63,000 records consisting of tweets and their respective sentiment labels(-1: negative, 0: neutral, 1: positive).All these Tweets were extracted using there Respective APIs Tweepy and PRAW .Tweets from twitter are cleaned using Python and also NLP with a Sentimental Label to each ranging from -1 to 1.

III. Methodology

In this section, an in-depth overview of the approach for sentiment classification of tweet text is provided.This study utilizes machine learning and deep learning techniques, employing algorithms like TF-IDF for word embedding. The model selection includes Naive Bayes, Bag of Words vectorization and various Deep learning models like LSTM, BERT (Bidirectional Encoder Representations from Transformers). This comparative analysis aims to effectively determine the sentiment conveyed in tweet content. The Figure1 gives an overview of the approach used in this work.

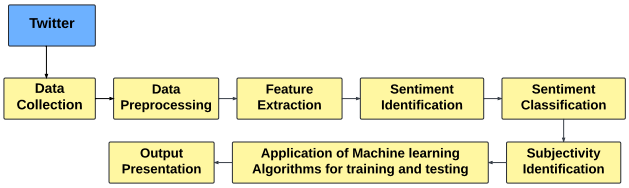

Fig. 1. Workflow

A. Data Visualization

Data visualization plays a crucial role in understanding and interpreting the sentiment analysis results. In this study, we employed various visualization techniques to gain insights into the distribution and characteristics of sentiment categories within the Twitter dataset.

1) Pie Chart of Sentiment Distribution: We utilized matplotlib to create a pie chart illustrating the percentage distribution of sentiment categories (positive, negative, and neutral) within the dataset.This visualization provides a clear overview of the relative proportions of each sentiment

category.

2) Countplot of Sentiment Distribution: Another visualization technique involved creating a countplot using seaborn, which displays the distribution of sentiment categories as individual bars. The custom color palette highlights each sentiment category, allowing for easy visual differentiation.

3)Word Clouds for Each Sentiment Category: Additionally, we generated word clouds for each sentiment category using the wordcloud library. These word clouds visually represent the most frequent words or phrases associated with each sentiment category, offering valuable insights into the prevalent themes and topics within the dataset.

B. Data Preprocessing

Before proceeding with the analysis, we have performed several preprocessing steps on the raw Twitter dataset to ensure its quality and suitability for sentiment analysis:

1) Text Cleaning: The raw text of the tweets was cleaned to remove any irrelevant or noisy elements, such as special characters, URLs, hashtags, and mentions. This step helps streamline the text data and remove any potential distractions that could affect the sentiment analysis process.

2) Tokenization: The cleaned text was tokenized, meaning it was split into individual words or tokens. This step is essential for further analysis as it breaks down the text into manageable units for processing.

3) Stopword Removal: Common stopwords, such as ”the,” ”is,” and ”and,” were removed from the tokenized text. Stopwords are frequent words that typically do not carry significant meaning and can be safely excluded from analysis to reduce noise.

4) Lowercasing: All text was converted to lowercase to ensure consistency in word representation. This prevents duplicate words with different cases from being treated as separate entities during analysis.

5) Stemming or Lemmatization: Later, stemming or lemmatization techniques were applied to further normalize the text data. Stemming reduces words to their root form, while lemmatization maps words to their base or dictionary form, ensuring consistency in word representation.

C. Machine Learning Models

We have implemented various machine learning models such as Naive Bayes classifier, Bag of Words Vectorization-based model.

1) Naive Bayes: Naive Bayes is a probabilistic classifier based on Bayes' theorem with the assumption of independence among features. In our sentiment analysis task, we employed the Gaussian Naive Bayes variant, which assumes that the features follow a Gaussian distribution. This model is particularly effective for text classification tasks like sentiment analysis due to its simplicity and efficiency.

2) Bag of Words Model: The Bag of Words (BoW) model is a simple yet effective technique for text representation, where each document is represented as a vector of word frequencies. In our implementation, we constructed a vocabulary from the entire corpus of tweets and represented each tweet as a vector indicating the presence or absence of each word in the vocabulary. This model disregards the order of words but captures the overall word frequency distribution, making it suitable for sentiment analysis tasks.

D. Deep Learning Models

In our sentiment analysis project, we're using different deep learning models designed to understand how language expresses emotions. These systems help us accurately identify whether text is positive, negative, or neutral, by looking at the words used and their meanings.

1) LSTM (Long Short-Term Memory): LSTM is a type of recurrent neural network (RNN) architecture specifically designed to capture sequential dependencies in data. It consists of memory cells and various gates that control the flow of information through the network. In our implementation, we constructed an LSTM model for sentiment analysis. The model comprises an embedding layer followed by an LSTM layer and dense layers for classification. It learns to extract features from sequential text data and make predictions based on the learned representations.

2) BERT (Bidirectional Encoder Representations from Transformers): BERT is a powerful deep learning model designed for natural language understanding tasks. It utilizes a bidirectional transformer architecture, which allows it to capture contextual information from both left and right contexts of a word in a sentence. In our sentiment analysis task, we fine-tuned a pre-trained BERT model for sequence classification. The model takes as input tokenized text sequences and outputs predictions for sentiment categories.

IV. Results and Discussion

The results of our sentiment analysis on real-time Twitter data using various machine learning models, RNN architecture and pretrained model revealed significant insights into the effectiveness of different approaches.

Initially, we evaluated the performance of Naive Bayes, Bag of Words Vectorization-based models, LSTM, and BERT in

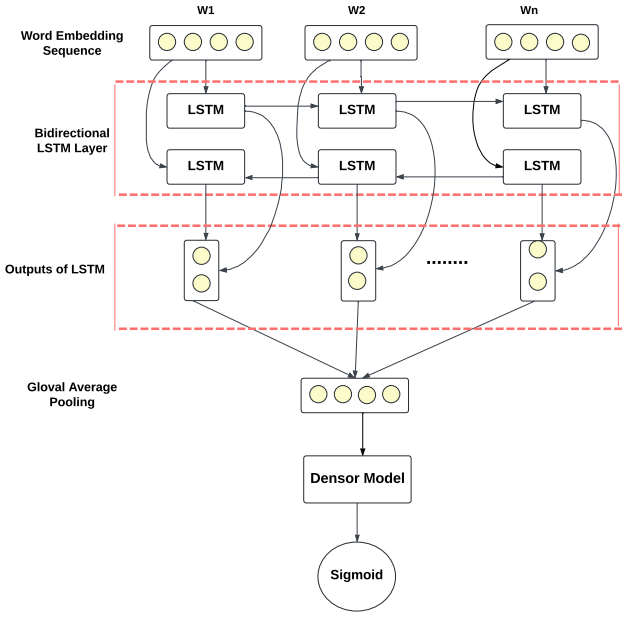

Fig. 2. Architecture of LSTM

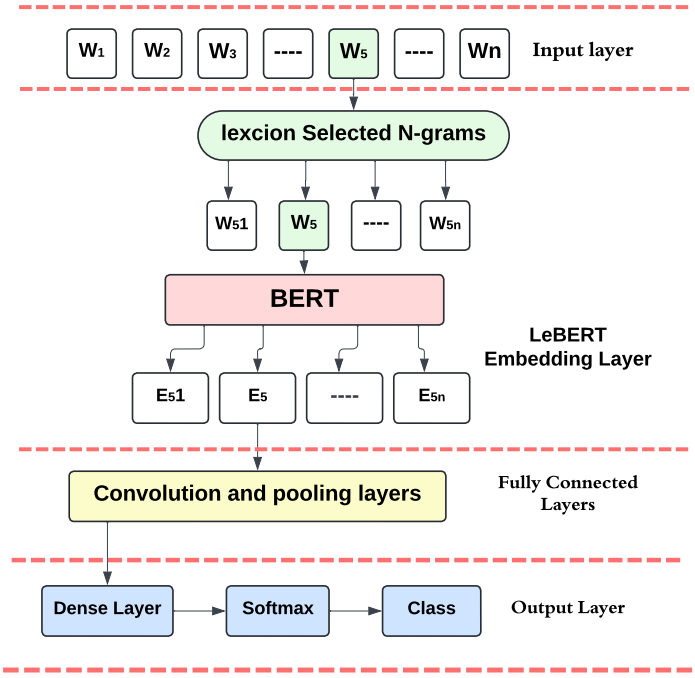

Fig. 3. Architecture of BERT

predicting sentiment. Among these models, BERT exhibited the highest accuracy of 85

BERT's bidirectional attention mechanism captures contextual information effectively, while its pre-training on extensive text data enhances understanding of word relationships. Additionally, BERT's fine-tuning capability optimizes performance for specific tasks like sentiment analysis, culminating in superior accuracy compared to traditional models.

TABLE I
COMPARATIVE ANALYSIS OF MODELS

Models	Accuracy
Long Short-Term Memory (LSTM)	73.64%
BERT	85.54%
Naive Bayes	72%
Bag of Words	69.52%

These findings highlight the efficacy of advanced deep learning models like BERT in analyzing sentiments in large-scale social media datasets.

The development of the user-friendly web interface using Flask involved several steps to ensure seamless access to sentiment predictions. Firstly, Flask provided the framework for building the interface, allowing for easy integration of various components. API calls were then integrated into the interface, enabling communication between the frontend and backend.Upon receiving input tweets from users, the web interface utilized these API calls to access the model code, which contained the sentiment analysis algorithms. This facilitated real-time sentiment prediction, as the model code processed the input tweets and generated predictions instantaneously.

The predictions were then returned to the web interface, where they were displayed to the user in a convenient and user-friendly manner. By leveraging Flask and API calls, the web interface enabled efficient and hassle-free access to sentiment predictions, enhancing the overall user experience.

V. CONCLUSION

In this project, we conducted sentiment analysis on real-time Twitter data using machine learning models. Through experimentation, we found that the BERT model demonstrated the highest accuracy in sentiment prediction, primarily due to its ability to capture contextual nuances and understand the intricate relationships between words in natural language text. In addition to leveraging machine learning models, such as BERT, for sentiment analysis, this project also utilized Flask to develop a user-friendly web interface. By integrating API calls, the web interface dynamically accesses the model code, enabling real-time sentiment prediction on input tweets.Our findings highlight the efficacy of advanced deep learning models like BERT in analyzing sentiments in large-scale social media datasets.

Future work could involve exploring ensemble methods and incorporating domain-specific knowledge to further improve sentiment analysis performance.This study contributes to the ongoing development of sentiment analysis models by showcasing the practical effectiveness of advanced deep learning architectures, such as BERT, in accurately capturing and interpreting the subtle variations of sentiment present in large-scale social media datasets.

VI. Acknowledgment

We extend our sincere thanks to Dr. Animesh Chaturvedi for his guidance and unwavering support throughout this project.

References

[1] Qi Y, Shabrina Z. Sentiment analysis using Twitter data: a comparative application of lexicon- and machine-learning-based approach. Soc Netw Anal Min. 2023;13(1):31. doi: 10.1007/s13278-023-01030-x. Epub 2023 Feb 9. PMID: 36789379; PMCID: PMC9910766.

[2] C. Kariya and P. Khodke, "Twitter Sentiment Analysis," 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-3, doi: 10.1109/INCET49848.2020.9154143.

[3] A. Sarlan, C. Nadam and S. Basri, "Twitter sentiment analysis," Proceedings of the 6th International Conference on Information Technology and Multimedia, Putrajaya, Malaysia, 2014, pp. 212-216, doi: 10.1109/ICIMU.2014.7066632.

[4] AminiMotlagh M, Shahhoseini H, Fatehi N. A reliable sentiment analysis for classification of tweets in social networks. Soc Netw Anal Min. 2023;13(1):7. doi: 10.1007/s13278-022-00998-2. Epub 2022 Dec 12. PMID: 36532862; PMCID: PMC9742011

[5] Bello, A.; Ng, S.-C.; Leung, M.-F. A BERT Framework to Sentiment Analysis of Tweets. Sensors 2023, 23, 506. https://doi.org/10.3390/s23010506

[6] N. C. Dang , M. N. Moreno-García, and F. De la Prieta, "Sentiment Analysis Based on Deep Learning: A Comparative Study", Department of Information Technology, Electronics, 2020. https://doi.org/10.3390/electronics9030483

[7] M. H. Abd El-Jawad, R. Hodhod and Y. M. K. Omar, "Sentiment Analysis of Social Media Networks Using Machine Learning," 2018 14th International Computer Engineering Conference (ICENCO), Cairo, Egypt, 2018, pp. 174-176, doi: 10.1109/ICENCO.2018.8636124.