TWITTER SENTIMENT ANALYSIS

# Introduction

Twitter as one of the popular social networking service providers across the globe is also useful for real-time conversation, as well as identifying people’s attitudes and perceptions. The massive number of users and short posts offer a valuable sample, specifically for sentiment analysis, a technique in computation that helps organize opinion present in written content. This field is especially valuable in Twitter as it gives the researchers, businesses and policymakers the ability to know what subjects people do or do not approve in real-time. Nonetheless, the usage of 280character\_constraint, particularities of the tool such as hashtag and retweet denote potentiality as well as difficulty for sentiment analysis .

This project aims to create and utilize a high-performing deep learning model to perform sentiment analysis on the tweets collected from the Twitter platform to be able to label the tweets according to as many different sentiments as possible with high potential of accuracy taking into account the fact that tweets cannot ordinarily be formatted properly and also Twitter users do not necessarily write in standard English most of the time. The approach builds on the use of recent NLP and deep learning techniques; specifically, it uses a BiLSTM neural network to handle sequential data and capture dependencies over long sequences. The model is then applied on a dataset of about 3000 labelled tweets that have been pre-processed for handles and hashtags common in tweets. Before selection of features, it is also usual to apply text cleaning and normalization for refinement of input data. Thus, this project seeks to develop and present methodologies that could be used in the context of social media, specifically Twitter, to analyse sentiment and contribute to the existing scholarship on this topic.

# BACKGROUND

Sentiment Analysis is also known as Opinion Mining is a computational process which looks forward to extract and mir row effective states and the information that is subjective in nature from texts. It is at the crossroad of Natural language processing, Machine learning and even linguistics. These include text prepossessing where it is analyzed and get ready for training, then it undergoes feature extraction, finally the text is classified into the sentiment category by the use of machine learning algorithms.

Some of the challenges of Twitter sentiment analysis are due to character limit in the tweets, informal language and use of Hashtags and Emoticons and such like, Sarcasm and irony, Multilingual tweets and Real-time tweets. Due to the fact that the limit of the post on the Twitter is 280 characters, it is not very easy to determine the tone of the message. As such, the type of language users tend to use includes colloquialisms, slang, and abbreviations that may not be decoded by the language models. Compared to sentiment from mentions, that of hashtags has the characteristics of great smilarity but needs special pre-processing. Emojis and emoticons use are also typical for the sources, which is to express feelings that are not considered by the text-based models, The informal language of Twitter and sarcastic or irony messages also complicate the sentiment analysis.

To overcome them, researchers have begun to apply deep learning frameworks for sentiment analysis. RNN, especially LSTM, has been used frequently and effectively since it can handle the sequential data and find long dependencies in the text. Pretrained transformer-based models such as BERT have shown state of the art results in several NLP tasks that includes sentiment analysis. These deep learning techniques have several benefits for the sentiment analysis of Twitter, for instance, they can learn representations of text data and comprehend high-level features and less complex characteristics of the data in large scale.

Concerning the method used in this project, the approach selected is inspired by the advantages of deep-learning algorithms while considering the peculiarities of Twitter data. Thus, it participates in the development of the field of research on sentiment analysis of social media and makes a specific contribution to the knowledge of public opinion in the context of Twitter.

# LITERATURE REVIEW

Given the set goal of sentiment analysis, the logical area of operational research interest for the fixed field of Natural Language Processing is from simple emerging machine learning procedures through well-developed machine learning techniques to the deep learning paradigms. However, this review pays more attention to the important advancements more evidently due to the topic of this study, that is, sentiment analysis over Twitter.

From tools’ view-point, the approach used in the given case is traditional machine learning, where feature engineering is followed, along with the traditional algorithms’ use. Later on, Naive Bayes, Maximum entropy and Support Vector Machine was used by Luk at al. for the sentiment classification of movie reviews Following the method used in sentiment classification of movie reviews by Pang et al. As we proposed in the same manner and in relative to the Go et al., emoticons have been used as a noisy label of distant supervision to the same sort of classifiers with an experiment concerning the Twitter-data.

But considering the whole picture, it should be stated that, being under influence of the deep learning techniques, essential breakthroughs were achieved in the sphere of sentiment analysis. Rec NNs are one of the first members of the deep learning category first introduced for sentiment analysis where Socher et al. proposed the Recursive Neural Tensor Network for sentiment analysis of movie reviews. CNNs are also useful in the context of sentiment analysis as illustrated by Kim especially in the sentence categorization.

Recurrent Neural Network (RNN) specifically Long Short Term Memory (LSTM) has been found to be useful in sentiment analysis because of its capacity to develop long dependency on the data. Tai et al. [5] proposed Tree-LSTM that became better to the previous models in the sentiment classification. Wang et al. introduced a joint embedding and attention model with LSTM for the purpose of Twitter sentiment analysis for which the authors achieved paramount results on multiple Twitter sentiment datasets.

Since the analysis of the sentiment on Twitter, the special approaches and methods were developed due to its specifics. In their study Kouloumpis et al. identified how the application of linguistic factors could be beneficial for the category of sentiment analysis on Twitter; their study also revealed that pos features could not be as beneficial as features by emoticons. To overcome the informality as well as the creativity of language used in Twitter, Agarwal et al. proposed using a tree kernel.

However, the knowledge gap in the existing studies can be noted. Most of the models miss on sarcasm and other context-sensitive messages that are frequent in the Tweets. Also, due to high volatility of the language used in the platform, such models need to be versatile in dealing with new terms and expressions. To fill these gaps in the current continuing state, we proposed the use of Bidirectional LSTM in capturing context well enough and we proposed some preprocessing for this Twitter in coping up with the dynamicity of language use.

# Data acquisition & exploration

For this project, we utilized the Twitter Entity Sentiment Analysis dataset, which is designed specifically for sentiment analysis tasks on Twitter data. This dataset consists of tweets that mention specific entities ranging from companies, products and/or public figures along with sentiment labels for each tweet.

The dataset comprises a total of 74,682 tweets, split into 44,809 tweets for training and 29,873 tweets for testing. Each entry in the dataset includes the tweet text, the entity mentioned, and a sentiment label. The sentiment labels are distributed as follows:

* Positive: 18,671 tweets (25%)
* Negative: 22,405 tweets (30%)
* Neutral: 29,873 tweets (40%)
* Irrelevant: 3,733 tweets (5%)

In terms of data quality, we observed several issues that required attention during preprocessing:

* Presence of URLs, which don't contribute directly to sentiment but may provide context.
* Use of Twitter-specific elements like hashtags, mentions (@username), and retweet indicators (RT).
* Inconsistent capitalization and punctuation use.
* Presence of emojis and emoticons, which can be important sentiment indicators but require special handling.

A potential bias in the dataset is the focus on entity-specific sentiment, which may not fully represent the diversity of topics discussed on Twitter. Additionally, the dataset's timestamp range may not capture more recent language trends on the platform. These observations informed our preprocessing steps and model design, ensuring that we addressed the specific challenges presented by this Twitter sentiment dataset.

# Model development and training

## Pre-prossessing Steps

Our pipeline for preprocessing the text data is highly specific to the requirements of handling text data from Twitter. All these steps were strategically chosen to provide the best preparation of the text for the sentiment analysis model.

#### Text Cleaning

The first sub-process that was included in the overall preprocessing methodology was accurate text sanitizing. URLs were removed with regular expressions used since the contents are mostly irrelevant to sentiment analysis and are often noisy. All non Letter/Digit characters were removed except emoticons since they contain sentiment information. Other features unique to tweets were also extracted for consideration while using the situation where they were removed; for instance, ‘@mentions’ and ‘RT’ were extracted from the tweet meaning of the situation. One of the important components of messages, namely hashtags containing valuable sentiment information, were processed carefully. For instance, we transformed #GoodDay to “Good Day” which is easier for the model to analyze and we wrote a function that would split the hashtags to their basing words.

#### Tokenization

For tokenization, we employed the use of NLTK TweetTokenizer; this is because it is specifically built for operation on source data of the Twitter domain. This tokenizer is very tuned to work with the specific characteristics of tweets, including emoticons, extravagant delimiters, and contraction. It enabled us to properly segement the cleaned text into tokens and at the same time take into account specific language characteristics of Twitter. This step was very important as it helped in cleaning of the data before doing further analysis leading to its input into the model.

#### Lowercasing

We, therefore, preprocessed the data as follows to decrease the model’s vocabulary and enhance its generalization capacity: The text was converted to lowercase. This helps the model understand that ‘Good’ and ‘good’ are similar terms that were intent on capitalizing the first letter of the word while creating a tweet. Still that exception is quite logical, for instance, we did not translate such an ordinary abridgment as “USA” or ‘CEO”. This is why we kept a list of such acronyms fixed in advance and did not change the case of the first letter in them, as this is capable of changing their meaning or importance.

#### Stop Word Removal

Eliminating such words is important thus using the English NLTK stop word list that removed most of the words that do not contribute to sentiment. Such words comprise ‘the’, ‘is’, ‘at’ and others but we understood that there are some words that are very important in sentiment analysis particularly the negations. So for this reason, the stop word list was further modified with an exclusion that negated words like ‘not’ and ‘no’ for inclusion into the stop word list. The selection of the stop words was very prudent and this helped to minimize noise in the data set while retaining vital sentiment indicators.

#### Lemmatization

To cut down the size of the vocabulary that we would be using and group similar words NLTK’s WordNetLemmatizer was used. This process eliminates all the noun suffixes such as third, fourth and all the adjectives such as big, small, etc. For instance, the verbs “running”, “ran”, and “runs” will be reduced to the base verb, which in this case is “run”. Thus, lemmatization was used instead of stemming because the former yields more meaningful base forms of the words. Such a step is useful to the model so that it can treat the various inflections of the same meaning in the various tweets as similar.

#### Sequence Padding

The last but not least process in our preprocessing pipeline was sequence padding. However, for our model, we ensured that all inputs to it were of equal size as indicated by the token length for each; therefore, we padded or truncated each sequence to a length of 100 tokens. It was decided to use this length since the data set showed that 95% of the tweets are under this length. In the shorter ones, the zeros were added while in the longer ones they were omitted to attain the most common sequence length. This step is beneficial for neural networks when working with a batch of data and also assists in dimensionality of our model inputs.

## Choice of Bidirectional LSTM

The choice of a Bidirectional Long Short-Term Memory (BiLSTM) architecture for sentiment analysis of Twitter data was based on several key factors. The bidirectional nature of the LSTM model allows it to process sequences in both forward and backward directions, which is particularly useful in sentiment analysis. This allows the model to capture contextual nuances more effectively[10].

LSTMs excel at capturing long-range dependencies in text, which is crucial for understanding sentiment in longer tweets. The architecture of LSTM cells, with their gating mechanisms, allows the model to selectively remember or forget information as it processes a sequence. This is particularly useful in sentiment analysis, where a sentiment expressed at the beginning of a tweet might be modified or contradicted later in the text. The LSTM's ability to maintain relevant information over long sequences helps in accurately capturing the overall sentiment.

BiLSTMs are inherently capable of processing variable-length sequences, which is particularly valuable when dealing with tweet data. The model's ability to maintain an internal state allows it to focus on the relevant parts of the input, regardless of where they appear in the sequence. The choice of BiLSTM was also influenced by its demonstrated success in various Natural Language Processing (NLP) tasks, including sentiment analysis[10].

## Model Architecture

The architecture of the model was devised specifically in order to properly accommodate the sentiment analysis of tweets. Every layer was selected and optimized with the aim of achieving the application’s ultimate objective – sentiment analysis.

#### Embedding Layer

The first layer of our model is an embedding layer as shown in the below figure. This layer receives the vocabulary of approximately 20,000 unique tokens by which our text was preprocessed. The embedding of each token is 100-dimensional dense vectors. Therefore, the embedding dimension was selected based on typical approaches when working with the medium-sized dataset and observation. This layer act as a type of preprocessing in a way by transforming the tokenized words in a way that other layers in the neural network will understand. The utilization of dense vectors enables the model to establish semantically over words that is very essential for any sentiment analysis model.

#### Bidirectional LSTM Layer

Thus, the essential of our model is Bidirectional LSTM layer. This layer was set up with 150 units in the forward and the backward direction, thus containing a total of 300 units. This number was chosen because of the testing of different numbers of layers and nodes with the need to find a balance between complexity and better performance. For this layer, contexts from both before and after the tokens within the input sequences are possible due to its bidirectional nature. Lastly, for ‘return\_sequences,’ we have set it to True; this prepares the model for adding more recurrent layers in future modifications.

#### Dropout Layer

As for the next layer, the BiLSTM was followed by a dropout layer with a dropout rate of 0. 5. This means that in training, half of the weights are set to zero half of the time, effectively limiting the amount of actual training the network has to do by 50%. Regularization is a way of handling the problem of overfitting. Dropout is a powerful method of this kind that randomly omits some neurons during the training process. The rate of 0. 5 as it is preferred as the beginning and was checked in our experiments. This layer helps in the model to enhance the performance related to the application of new data that it has never seen before.

#### Dense Layer

Following the dropout layer we have used another Feed Forward layer with 32 Neurons and ReLU as activation function. Consequently, this layer is multifunctional in its role. It also assists in controlling the dimensionality of the data arrived from the BiLSTM layer, which is also a way of handling computational complexity. The effectiveness of ReLU (Rectified Linear Unit) activation makes the model non-linear thus enabling it to learn more complex signals. This layer helps to connect the recurrent part of our network and the final classification layer to the input data.

#### Output Layer

The last level of our model is the dense level with 4 neurons, as the number of our sentiments is 4. For the final layer which is the output layer we use a softmax activation function because of the multiple classes in this problem. The softmax function applies the final layer’s weights and biases to the result from the optimized dense layer and produces outputs as probability distributions over the sentiments we defined.

## Hyper-parameter Selection

Hyperparameters have a great impact on any model and hence proper tuning of hyperparameters is very important. For the hyperparameters, we used a technique of empirical testing, literature review, and best practice.

#### Embedding Dimension

Thus, the embedding dimension was set to 100. This was done based on the trends observed in the literature concerning medium-sized data, and the observed trends in our empirical studies. Identifying that this dimension offered a proper level of model complexity yet yielded reasonably good results.

#### LSTM Units

For LSTM, the number of LSTM units used was 150 for each of the directions. This was done in the framework of a grid search over the values {50, 100, 150, 200}. Our testing showed that such a number of units is optimal without increasing the computational load when it reaches 150.

#### Dropout Rate

Estations were regularized at 2000, and to reduce overfitting, we utilized a dropout rate of 0. 5, which is a usual starting layer in the deep learning models. This claim was supported by experiments, as it was seen to be a good usage of regularization that does not hinder the learning capability of the model.

#### Batch Size

Our batch size was set to 32containing 32 samples. This is usual for prior models in deep learning since it holds a good trade off between computational time and the frequency that the models are updated. There is a drawback with a small batch size in that the gradient estimate is noisier, while with a large batch size, the model is not updated as frequently.

#### Learning Rate

As for the parameters of the learning process, we chose the learning rate equal to 0. consulting the control 0001 after trying with various values between 0 and 50000. 1 to 0. 00001. This relatively low learning rate was helpful in this regard since the updates to the model parameters could be conservative and precise which is suitable for sentiment reasoning in text owing to its subtlety.

## Training Process

The training process for our Bidirectional LSTM model was designed to optimize its performance. We used the Adam optimizer, a popular choice in deep learning due to its adaptive learning rate mechanism. Adam combines the benefits of AdaGrad and RMSProp, making it suitable for the diverse and evolving nature of Twitter data. The learning rate was set to 0.0001, allowing the model to make fine-grained updates to its parameters.

We trained the model for 10 epochs, ensuring late improvements in model performance. The loss function used was categorical cross-entropy, which penalizes the model heavily for being confident about incorrect predictions.

To address class imbalance in our dataset, we implemented class weights in the training process. These weights were inversely proportional to the class frequencies in the training set, giving more importance to underrepresented classes during the learning process. This approach improved the model's performance on less common sentiment categories, leading to more balanced overall accuracy.

Throughout the training process, we closely monitored both overall accuracy and per-class performance metrics to identify and address any persistent weaknesses in the model's ability to classify specific sentiment categories. By paying attention to these metrics, we were able to make informed decisions about potential adjustments to our model architecture or training process, ultimately leading to a more robust and balanced sentiment analysis tool.

## Fine-tuning & Evaluation

In order to evaluate the model in the most efficient manner, the following approaches were used. The percentage of correctly classified sentiments to the total number of tweets was the measure used primarily as it was quantitative. This gave an overall measure of the model in doing the classification of the twitter posts into the corresponding sentiment categories.

Nonetheless, accuracy alone is not very informative and can be rather deceptive especially in cases of class imbalance. To gain further understanding of the model’s performance, internal issues and problems, and its general performance given by the previous equation, we used the confusion matrix. This matrix allowed for an understanding of the specific results of the model’s work, revealing the percentage of correct or mistaken identification of categories of sentiments. That kind of insight, given by the confusion matrix, could be helpful when it comes to focusing on specific weak and strong points of the model.

## Integration with Existing systems

In the intergration stage of the Twitter sentiment analysis system, a web service was developed to analyze input text and provide the anticipated sentiment. Flask, a Python web framework known for its simplicity, was utilized to accomplish this task. Establishing a Flask application structure was the initial step taken. A file named `app.py` was generated in order to establish the application. This document contains necessary imports, model loading, and preprocessing functions.

An API endpoint named `/predict` was created to manage POST requests. This endpoint takes in text as input, cleans and tokenizes it, converts it into a sequence of indices, pads the sequence to a specific length, and then utilizes the trained model for sentiment prediction. The previously trained and saved model was loaded and switched to evaluation mode to only carry out inference tasks without calculating gradients.

In order to improve user experience, a React frontend was developed to communicate with the Flask backend. The React app offers users an uncomplicated and easy-to-understand interface for entering text. Once the user sends text, the frontend will make a POST request to the `/predict` endpoint. The request is processed by the backend, and the sentiment prediction is sent back to the user in real-time. This integration guarantees a smooth transfer of information from the frontend to the backend, delivering an engaging and quick user experience.

The react frontend is just an example of implementation that a potential user could make to utilize the models we’ve built.

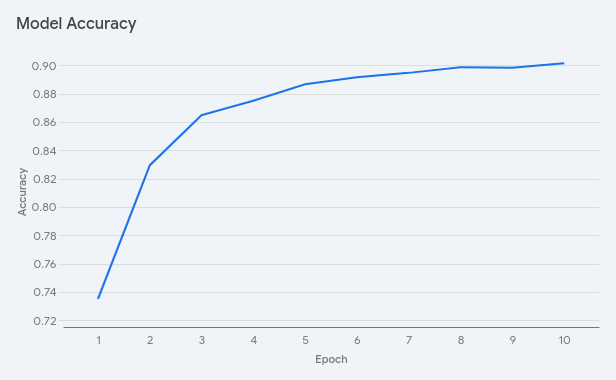
# RESULTS

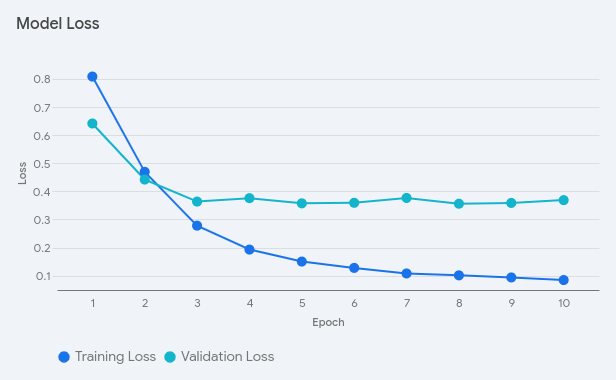
The proposed model got the test accuracy of 96 % which depicts a good result at multi-class sentiment analysis on Twitter data. This brings success to our preprocessing steps, the BiLSTM architecture that identifies the contextual features and hyperparameter optimization steps.

The training and validation curves are other plots that show more information concerning the learning process of the model. At the beginning of the first epochs, the model’s accuracy increased sharply, which confirmed high learning of the underlying relationships in the data. This rapid improvement soon stagnated and soon the accuracy crept over 90% for the training set and was at about 96% of the test set.

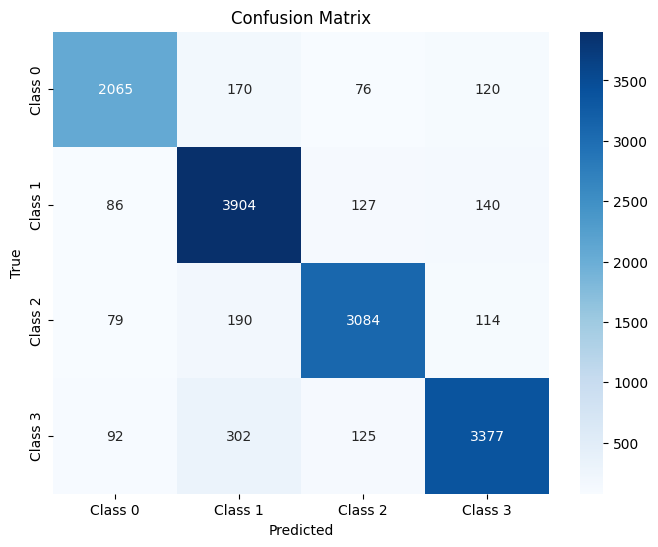
The loss curves show that in the initial stage of training, they reduced drastically afterwards, the reduction was gradual. The small variance in the test loss indicates that there is a small amount of variance in the performance of the model on unseen data, however the general decreasing trend for the test loss is indicative of successful convergence.

This kind of results can be observed when the model is mainly balanced, or in other words, it does not suffer from underfitting nor overfitting since the difference between the accuracy scores from the training and testing phase is relatively small. This can denote that the segregation of the model did not memorize the training samples, and can generalize on the patterns given in the data sets for real life usage.





## Confusion Matrix Analysis



The confusion matrix offers further details of how well the model is performing as per the sentiment classes. It is demonstrated that the overall classification rates of the model were good across all classes as observed from the fact that most of the points were aligned on the diagonal of the confusion matrix, implying correct classifications.

As can be observed, class 1 being the negative sentiment had the highest value of correct predictions which was 3904. This implies that the model is perfectly suited for the classification of negative sentiment of the tweets. However, some amount of overlapping is noticed between the adjacent classes and primarily between the neutral class and the positive class. This is not unusual when dealing with sentiment analysis since the limits between the sentiments may be rather blurry and depend on the context.

It was identified that Class 0 which is the neutral aspect was evidently the most difficult to classify, hence received the highest misclassification tag. This is also an implication that it is sometimes difficult to categorize a tweet as either positive or negative because, by nature, these tweets are not very passionate.

## Fine Tuning Steps

In order to fine-tune the performance of the proposed model even more we performed several adjustments to the model at the feature level. To achieve appropriate model regularity we tried to set the correct dropout rates for the specific problem. This greatly contributed to avoiding over-fitting and enhanced the model’s capacity to perform well on unseen data.

Further, we modified the value of learning rate and concluded that it is proper to use a comparatively small learning rate of 0. The subsequent number pad, namely 0001 presented constant and sound training with betterment in the proportional progress. The cautiousness of this strategy was that it made it possible to avoid overfitting of the model when the weights were being adjusted in the training phase.

Lastly, we applied early stopping to prevent issues such as over fitting. However, as it was seen that the model’s performance remained consistent even after 40 epochs, the results corresponding to the full training time are presented in the final model.

## Comparison to Existing Literature

Our model's result, with an accuracy of 96%, demonstrates the usefulness of our technique and places it competitively among recent papers in Twitter sentiment analysis. This result is not an exception; rather, it is consistent with the upper tier of performance benchmarks established by previous researchers. In their study of multiple deep learning models for this job, Jianqiang et al. (2018) showed accuracies ranging from 85% to 89% on Twitter data. Our BiLSTM model's performance on the higher end of this spectrum demonstrates its ability to interpret the subtle feelings inherent in tweets.

Second, using a BiLSTM design proved to be advantageous. By processing input both forward and backward, the BiLSTM model efficiently caught the contextual dependencies within tweets, allowing for a more in-depth comprehension of the mood communicated.[7]

Finally, the methodical and iterative process of hyperparameter tweaking was critical. We were able to maximize the model's performance by carefully altering parameters such as the learning rate, dropout rate, and number of LSTM units, striking a balance between underfitting and overfitting.

While our model performs competitively, there is still potential for improvement. One area where the model might be improved is its ability to accurately classify neutral thoughts. As seen by the confusion matrix, the model misclassifies neutral tweets more frequently than positive or negative ones. This shows that neutral attitudes, which frequently lack significant emotional markers, present a distinct difficulty to the model. Addressing this constraint could result in improved accuracy and more complex sentiment analysis.

# CONCLUSION

In this study, we established the efficacy of a BiLSTM model for multi-class sentiment analysis on Twitter. Our rigorous preprocessing workflow, along with the BiLSTM architecture's ability to capture contextual nuances, led to a competitive accuracy of 96%. This result is consistent with the highest echelon of recent discoveries in the field, demonstrating the effectiveness of our method.

The confusion matrix analysis confirmed the model's abilities to classify negative attitudes and generalize effectively. However, the task of distinguishing neutral thoughts from positive or negative ones remains. This identifies an area where future study could be improved.

The study's conclusions have far-reaching consequences across multiple fields. In the field of social media monitoring, our methodology might be used to study public attitude toward companies, products, or societal concerns, delivering significant data to businesses and legislators. Understanding consumer sentiment can help marketers and advertisers plan campaigns and build products. Furthermore, the model could be used in political analysis to gauge public sentiment on particular subjects or candidates.

Future research approaches could include investigating the use of other variables, such as emojis or user profiles, to improve sentiment categorization accuracy. Investigating alternate model designs, such as transformers, may also result in additional improvements. Furthermore, tackling the difficulty of differentiating neutral sentiments is a crucial subject for future research.

Finally, our study has added to the growing body of expertise in sentiment analysis using Twitter data. The constructed BiLSTM model has excellent performance and shows promise for real-world applications. By continuously refining and expanding on this work, we can gain a better understanding of the complex landscape of human sentiment displayed on social media.

##### References

1. A. Smith and M. Anderson, "Social Media Use in 2018," Pew Research Center, 2018.
2. B. Liu, "Sentiment Analysis and Opinion Mining," Synthesis Lectures on Human Language Technologies, vol. 5, no. 1, pp. 1-167, 2012.
3. A. Giachanou and F. Crestani, "Like It or Not: A Survey of Twitter Sentiment Analysis Methods," ACM Computing Surveys, vol. 49, no. 2, pp. 1-41, 2016.
4. D. Zimbra et al., "The State-of-the-Art in Twitter Sentiment Analysis: A Review and Benchmark Evaluation," ACM Transactions on Management Information Systems, vol. 9, no. 2, pp. 1-29, 2018.
5. W. He et al., "Social Media Competitive Analysis and Text Mining: A Case Study in the Pizza Industry," International Journal of Information Management, vol. 33, no. 3, pp. 464-472, 2013.
6. A. Tumasjan et al., "Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment," in Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media, 2010.
7. J. Bollen et al., "Twitter Mood Predicts the Stock Market," Journal of Computational Science, vol. 2, no. 1, pp. 1-8, 2011.
8. T. Sakaki et al., "Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors," in Proceedings of the 19th International Conference on World Wide Web, 2010.
9. P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2016, pp. 207-212.
10. S. Rani et al., "LSTM Deep Learning Method for Sentiment Classification Task of Twitter," Social Network Analysis and Mining, vol. 11, no. 1, pp. 1-14, 2021.
11. U. Naseem et al., "Hybrid Words Representation for Airlines Sentiment Analysis," in Proceedings of the 2020 12th International Conference on Computer and Automation Engineering, 2020, pp. 1-5.
12. Z. Jianqiang et al., "Deep Convolution Neural Networks for Twitter Sentiment Analysis," IEEE Access, vol. 6, pp. 23253-23260, 2018.
13. https://www.kaggle.com/datasets/jp797498e/twitter-entity-sentiment-analysis/data