Twitter Sentiment Analysis using LSTM

Bhoomika Kasireddy  
*Department of CSE*  
*VIT-AP University*Amaravati, India  
kasireddybhoomika @gmail.com Guru Subhash Bathula   
*Department of CSE*  
*VIT-AP University*Amaravati, India  
battulasubhash13@gmail.com

Lalitha Janipella  
*Department of CSE*  
*VIT-AP University*Amaravati, India  
lalitha.21bce9560@vitapstudent.ac.in

*Abstract*: Social media platforms like Twitter have emerged as valuable sources for gauging public sentiment due to their real-time and vast user-generated content. This paper presents a sentiment analysis project leveraging Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN) known for its ability to capture temporal dependencies in sequential data. The objective of this project is to automatically classify tweets into predefined sentiment categories: positive, negative, or neutral.

Keywords—Sentimental Analysis, Polarity, Tokenization , Stemming, Stop words, LSTM

# **Introduction**

## Background

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the A4 paper size. If you are using US letter-sized paper, please close this file and download the Microsoft Word, Letter file. In recent years, social media platforms such as Twitter have become integral sources of real-time, user-generated content reflecting diverse opinions, emotions, and sentiments of individuals worldwide. This vast pool of textual data presents unique opportunities for understanding public sentiment, which is crucial for various applications including brand management, market analysis, and public opinion monitoring.

Sentiment analysis, also known as opinion mining, is a subfield of natural language processing (NLP) that focuses on identifying and extracting subjective information from text to determine sentiment polarity (positive, negative, or neutral). Traditional machine learning approaches to sentiment analysis often struggle to capture nuanced contextual information inherent in social media text due to the presence of complex language patterns, slang, and emoticons.

## Objectives

The objective of this project is to develop and implement a sentiment analysis model using Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN) well-suited for processing sequential data, to automatically classify tweets into predefined sentiment categories.

## Significance

The significance of this project lies in its exploration of advanced deep learning techniques, specifically LSTM networks, for sentiment analysis on social media data. By leveraging LSTM's ability to capture long-term dependencies in sequential data, we aim to enhance the accuracy and robustness of sentiment classification, particularly in the context of Twitter text characterized by brevity and informality.

## Challenges

Analyzing sentiment in Twitter data presents several challenges, including:

* Dealing with noisy and unstructured text containing abbreviations, misspellings, and informal language.
* Handling the inherent ambiguity and subjectivity of sentiment expressed through short and context-dependent tweets.
* Ensuring the scalability and efficiency of the sentiment analysis model to process large volumes of real-time data streams from social media.

## Scope of the project

This project focuses on:

* Collecting Twitter data using relevant hashtags or keywords related to sentiment analysis.
* Preprocessing the collected data to prepare it for LSTM model input.
* Designing and implementing an LSTM-based sentiment analysis model using appropriate deep learning frameworks.
* Evaluating the model's performance in classifying tweets into positive, negative, or neutral sentiments.

# **Literature Survey**

## Overview of Sentiment Analysis

Sentiment analysis, also known as opinion mining, is a branch of natural language processing (NLP) that focuses on automating the extraction and classification of sentiments, attitudes, and emotions expressed in textual data. The objective is to classify text into predefined sentiment categories such as positive, negative, or neutral, enabling valuable insights into public opinion and user sentiment.

## Sentiment Analysis on Twitter

Twitter has emerged as a popular platform for sentiment analysis due to its real-time nature and vast user-generated content. Researchers have explored various approaches to sentiment analysis using Twitter data, leveraging the unique characteristics of microblogging text, including brevity, informal language, and the use of hashtags and emoticons.

## Studies and findings

Cho et al. (2014) [1] introduced a novel approach using LSTM networks for sequence modeling in statistical machine translation. They demonstrated that LSTM's ability to capture long-range dependencies in sequential data was beneficial for tasks requiring contextual understanding, such as sentiment analysis.

Maas et al. (2011) [2] explored the application of deep learning techniques, including LSTM, for sentiment analysis on movie reviews. They found that LSTM models could effectively learn to represent and classify sentiment in text by capturing nuanced contextual information, leading to improved sentiment prediction accuracy.

Tang et al. (2015) [3] investigated the use of LSTM networks specifically for sentiment analysis on social media data, which often contains noisy and short-text inputs. They highlighted LSTM's capability to handle such data characteristics and showed improved performance compared to traditional methods.

Severyn & Moschitti (2015) [4] focused on Twitter sentiment analysis using LSTM-based deep convolutional neural networks (DCNNs). They integrated attention mechanisms into their models to improve sentiment prediction accuracy, demonstrating the effectiveness of LSTM architectures for analyzing social media text.

Chen et al. (2018) [5] explored transfer learning techniques with LSTM language models for sentiment analysis across different domains. They demonstrated that pre-trained LSTM models could be adapted to new sentiment analysis tasks with minimal labeled data, reducing the need for large annotated datasets.

He et al. (2017) [6] investigated multimodal sentiment analysis using LSTM networks to analyze both textual and visual cues in video and text data. They showcased the effectiveness of LSTM in capturing complex interactions between different modalities for sentiment prediction.

Tang et al. (2016) [7] explored ensemble learning techniques with LSTM models for sentiment analysis. By combining multiple LSTM-based classifiers, they achieved improved sentiment prediction robustness and generalization across diverse datasets and domains.

Bahuleyan et al. (2017) [8] studied semi-supervised sequence learning using LSTM architectures for sentiment analysis. They leveraged unlabeled data to enhance sentiment classification performance, demonstrating the robustness and scalability of LSTM-based models

Li et al. (2020) [9] focused on incorporating attention mechanisms into LSTM architectures for sentiment analysis. They highlighted the importance of attention in improving model interpretability and sentiment prediction accuracy, particularly in capturing relevant contextual information.

Zhang & Wallace (2015) [10] explored fine-tuning pre-trained LSTM language models for sentence classification tasks, including sentiment analysis. They demonstrated that fine-tuning LSTM models with domain-specific data could lead to improved performance compared to training from scratch.

# **Methodology**

## Dataset selection

The dataset was obtained from an online source, such as a publicly available repository or a specific dataset repository like Kaggle, containing tweets related to a specific topic (e.g., tweets with hashtags related to #donaldtrump).

**Table1. Different datasets that are available online**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Link** | **No of entries** |
| Twitter Sentiment Dataset | Tweets and polarity | [dataset-1](https://www.kaggle.com/datasets/saurabhshahane/twitter-sentiment-dataset) | 162981 |
| apple-twitter-sentiment | Tweets and polarity | [dataset-2](https://www.kaggle.com/code/kritanjalijain/twitter-sentiment-analysis-lstm/input?select=apple-twitter-sentiment-texts.csv) | 1624 |
| Reddit Sentimental analysis Dataset | Tweets and polarity | [dataset-3](https://www.kaggle.com/code/kritanjalijain/twitter-sentiment-analysis-lstm/input?select=apple-twitter-sentiment-texts.csv) | 37250 |
| Tweets.csv | Tweets | [dataset-4](https://www.kaggle.com/code/kritanjalijain/twitter-sentiment-analysis-lstm/input?select=Tweets.csv) | 14640 |
| sentiment\_tweets3.csv | Tweets | [dataset-5](https://www.kaggle.com/code/mehakiftikhar/sentiment-analysis-lstm-accuracy-96/input) | 10294 |
| hashtag\_donaldtrump.csv | Tweets | [dataset-6](https://www.kaggle.com/code/mansibasmatkar/2020-election-tweets-visualization-analysis/input) | 971158 |
| hashtag\_joebiden.csv | Tweets | [dataset-7](https://www.kaggle.com/code/mansibasmatkar/2020-election-tweets-visualization-analysis/input?select=hashtag_joebiden.csv) | 777079 |

## Data Cleaning and Preprocessing

Sentiment analysis involves analyzing opinions expressed in tweets, which can vary in expression and sentiment across different users. The Twitter dataset used in this survey study has been pre-labeled into two distinct classes: negative and positive polarity. This labeled dataset simplifies the sentiment analysis process and enables the examination of the impact of various features on sentiment classification.

The raw tweet data, characterized by polarity labels, is susceptible to inconsistencies and redundancies that may affect the accuracy of sentiment analysis. Therefore, preprocessing of the tweet data is essential and includes the following key steps:

1. *Text cleaning*
   1. Removal of non-alphanumeric characters, special symbols, URLs, and mentions using regular expressions.
   2. Standardization of text by expanding contractions (e.g., converting "don't" to "do not") to ensure uniformity in language representation.
2. *Tokenization*

Segmentation of tweet text into individual tokens (words or phrases) to facilitate subsequent analysis.

1. *Normalization*

Standardization of text by converting all characters to lowercase to treat words with different cases (e.g., "Apple" and "apple") as identical.

1. *Stopword Removal*

Elimination of common stopwords (e.g., "the", "and", "is") that do not contribute to sentiment analysis but may appear frequently.

1. *Lemmatization*

Reduction of words to their base or root form (lemmas) to unify variations of words (e.g., "running", "ran", "runs" to "run").

1. *Handling Emoticons and Special Symbols*

Conversion of emoticons and special symbols into textual representations to capture sentiment cues embedded in tweets.

1. *Handling Redundancy*

Identification and removal of duplicate or highly similar tweets to reduce redundancy in the dataset.

## Applying various NLP techniques on dataset

# Stemming

Stemming is a text normalization technique used in natural language processing to reduce words to their base or root form, known as stems, by removing suffixes and prefixes. The Porter Stemmer algorithm, developed by Martin Porter, is a widely used stemming algorithm due to its simplicity and effectiveness in English text.

The Porter Stemmer algorithm applies a series of heuristic rules to systematically strip word endings, aiming to transform related words into their common root form.

Examples include converting "running", "ran", "runs" to the base form "run", or reducing "cars" to "car".

Stemming helps reduce the dimensionality of the feature space in text analysis, enabling more efficient processing and analysis of textual data.It aids in capturing the essential meaning of words while disregarding variations due to tense, plurality, or other grammatical forms.

In preprocessing tweet data for sentiment analysis, applying the Porter Stemmer involves iterating over each word token and applying stemming rules to generate the stem representation.

# Tokenization

Tokenization is the process of segmenting text into individual units, or tokens, such as words, phrases, or punctuation marks, for further analysis. In the context of sentiment analysis of tweets, tokenization is essential for extracting meaningful units of text that can be analyzed and processed.

Tweet tokenization involves splitting the raw text into discrete units, typically words or phrases, based on specific criteria (e.g., whitespace, punctuation).Special attention is given to handling emoticons, hashtags, and mentions as distinct tokens to preserve their semantic value.

Tokenization transforms unstructured text data into a structured format suitable for subsequent analysis and feature extraction.It forms the foundation for various text processing tasks, including sentiment analysis, by enabling the computation of textual features.

# Word Length Distribution

Word length distribution analysis involves examining the distribution of word lengths in a corpus of text, providing insights into the lexical characteristics and complexity of the text data.

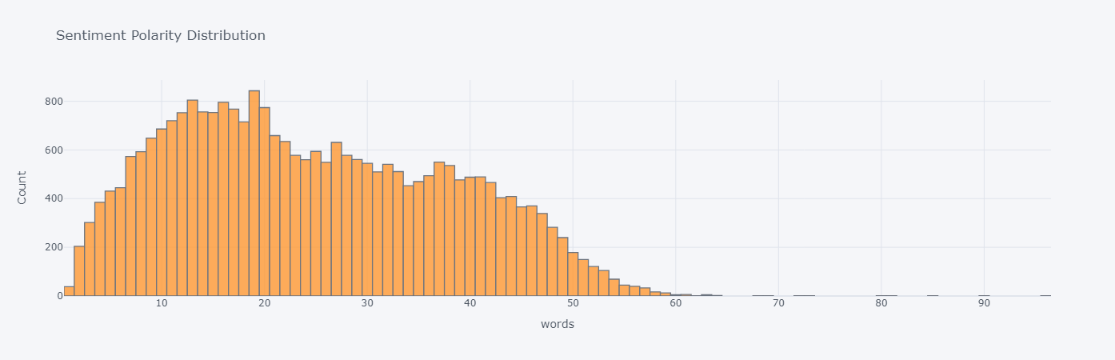
Word length distribution is computed by calculating the length (number of characters) of each word token in the tweet dataset.Statistical measures such as histograms or frequency distributions are generated to visualize the distribution of word lengths.

Word length distribution helps identify common patterns and variations in the vocabulary used within tweets.

It can reveal the presence of specific linguistic phenomena, such as slang, abbreviations, or domain-specific terminology.

The average word length is another important metric that can offer insights into the complexity or style of the text. This distribution shows how many words fall into specific average length categories. Typically, shorter average word lengths might indicate simpler language or informal communication.

The histogram with bins ranging from shorter to longer average word lengths helps visualize the diversity in word usage within the dataset.



**Fig.2.Word length distribution**

# Polarity Analysis

Polarity analysis assesses the sentiment orientation (positive, negative, neutral) conveyed by individual tweets, providing a quantitative measure of sentiment expression within the dataset.

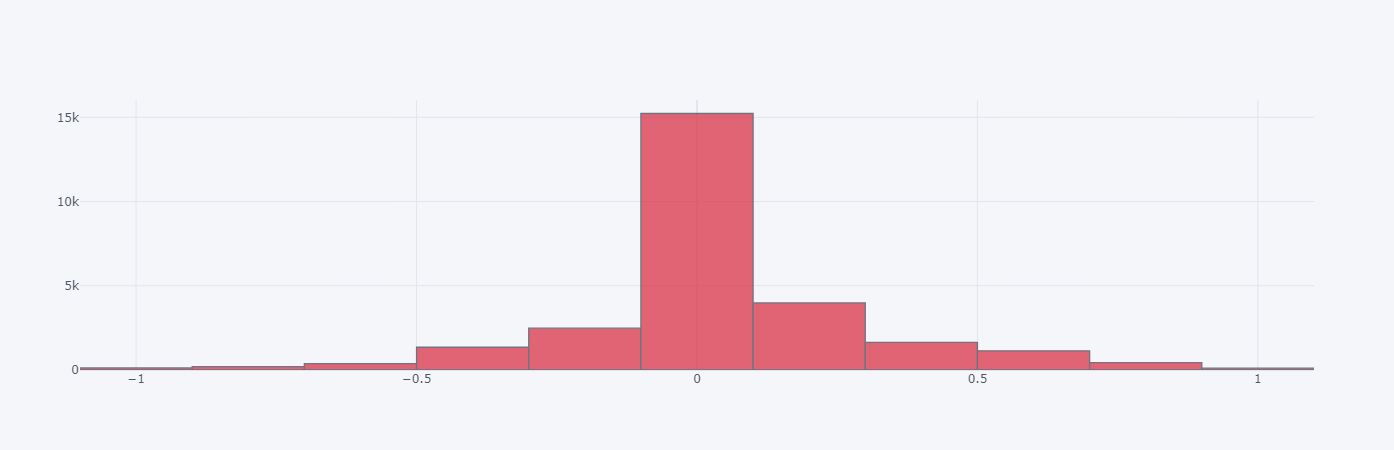
Polarity analysis is conducted using sentiment analysis techniques to assign sentiment scores or labels to tweets based on their content.Common methods include using pre-trained sentiment lexicons, machine learning models (e.g., LSTM), or rule-based approaches to classify tweet sentiment.

Polarity analysis enables the identification of sentiment trends and emotional tones expressed in the Twitter dataset.

It facilitates the extraction of actionable insights from social media text, such as tracking public sentiment towards specific topics or events.

The sentiment polarity of text refers to its overall emotional tone, typically measured on a scale from negative to positive. Here, we visualize the distribution of sentiment polarity using a histogram. The polarity values are likely ranging from -1 (very negative) to 1 (very positive), with 0 representing neutral sentiment.

The histogram reveals how sentiment polarity is distributed across the dataset. A peak towards positive values indicates predominantly positive sentiment, whereas a peak towards negative values signifies more negative sentiment. This analysis is crucial for understanding the overall sentiment of the dataset.



**Fig.3.Polarity Analysis of tweets**

## Sentimental Analysis

Sentiment analysis, a field within Natural Language Processing (NLP), has witnessed significant evolution over the years, particularly in the analysis of sentiments expressed on Twitter. Initially, this domain was characterized by binary classification, which involved assigning opinions or reviews to either positive or negative categories. However, recent advancements have expanded the scope of sentiment analysis to encompass multi-dimensional analysis, allowing for a more nuanced understanding of attitudes, emotions, and opinions expressed in text.

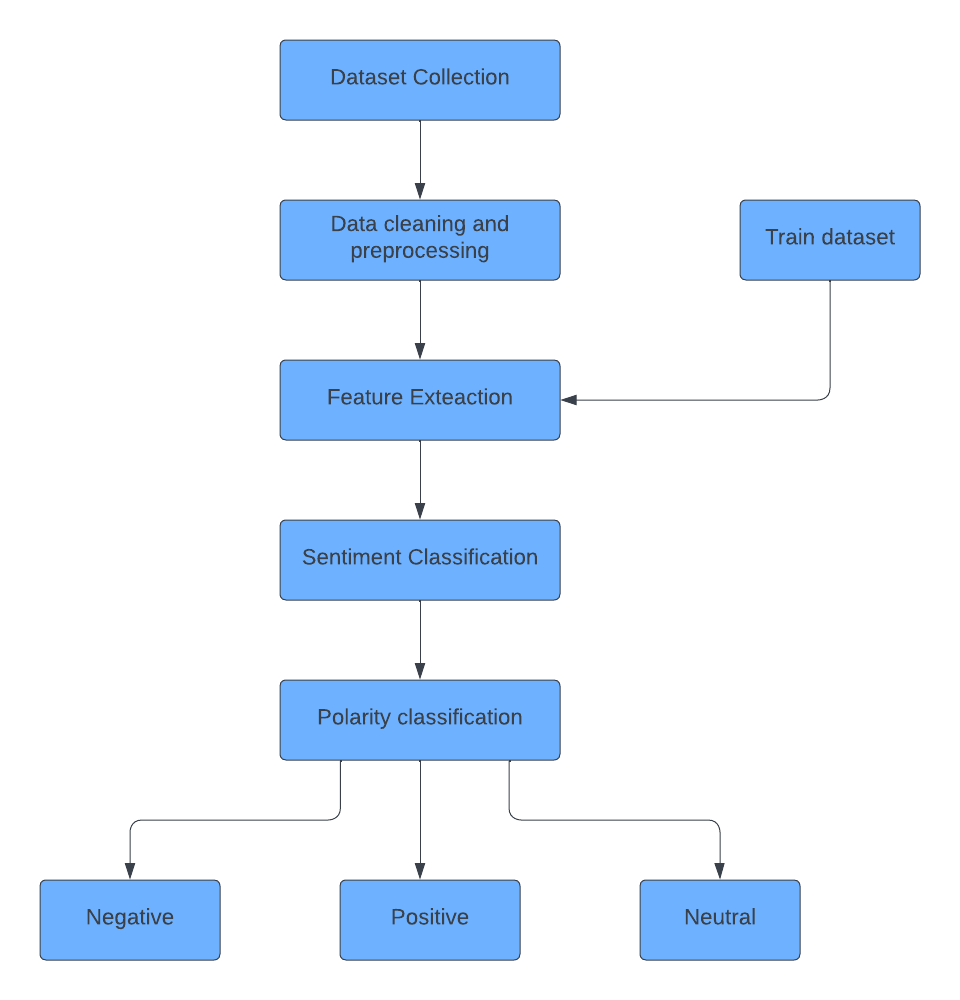
Traditionally, sentiment analysis focused on categorizing opinions as either positive or negative. This binary approach was straightforward but limited in capturing the complexity of human sentiments. For example, a tweet stating "The new movie is good" might be categorized as positive, while "The movie was good but the plot was confusing" could pose a challenge due to mixed sentiments.

The transition from binary classification to multi-dimensional analysis has been driven by the need for more detailed insights. Researchers now aim to classify sentiments along multiple dimensions, including positive, negative, and neutral, while also considering aspects such as intensity, subjectivity, and specific emotions like happiness, sadness, anger, or surprise.

This evolution has been facilitated by advancements in machine learning techniques, especially the adoption of deep learning models capable of processing large volumes of text data and discerning subtle nuances in sentiment. Additionally, sentiment analysis tools now leverage contextual understanding, semantic analysis, and domain-specific lexicons to improve accuracy.

One notable trend in contemporary sentiment analysis is aspect-based sentiment analysis (ABSA), which involves identifying specific aspects or features of a product, service, or entity and analysing sentiments associated with each aspect independently. For instance, in a review of a restaurant, sentiments might be categorized based on aspects like food quality, service, ambiance, and price.

Furthermore, sentiment analysis on social media platforms like Twitter has become more sophisticated, incorporating techniques to handle challenges such as sarcasm, irony, and slang. Researchers are also exploring sentiment evolution over time, tracking changes in public opinion on specific topics or events.

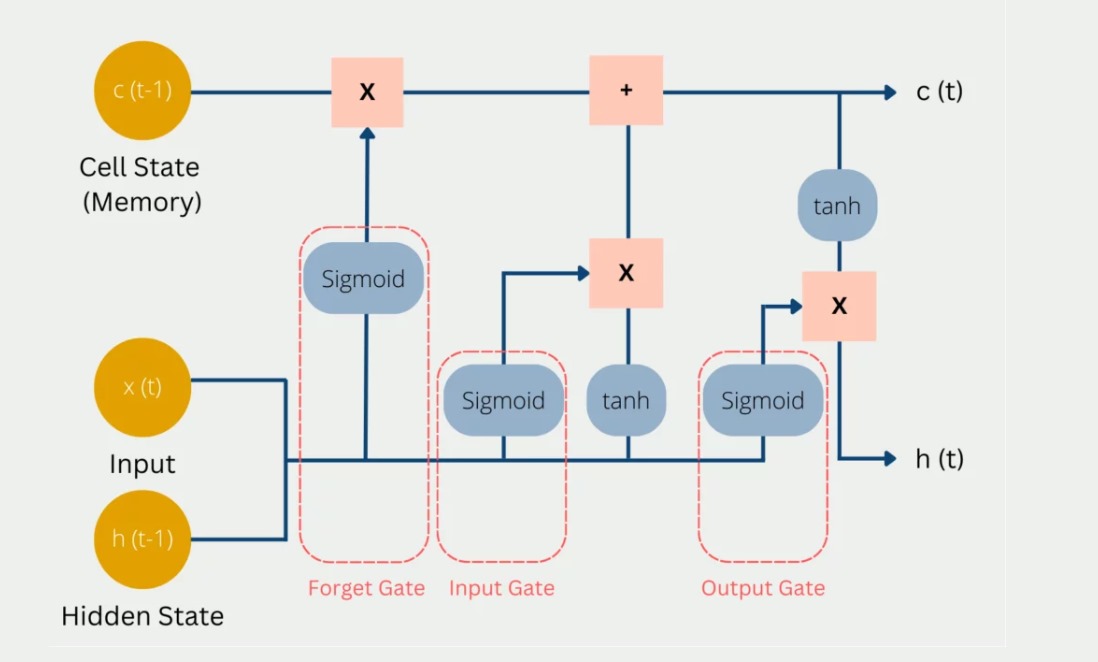


**Fig.4.Sentimental analysis Flow chart**

## Model Building

LSTM, a type of recurrent neural network (RNN), is well-suited for sequential data processing, making it ideal for analyzing text data like tweets, which often exhibit complex structures and dependencies. The following report outlines the process of building an LSTM model for sentiment analysis on Twitter data and discusses its implications and potential applications.

The LSTM model architecture comprises an Embedding layer, one or more LSTM layers, and densely connected layers for classification. The Embedding layer converts tokenized sequences into dense vectors. The LSTM layers capture sequential dependencies and long-term dependencies within the text data. Densely connected layers perform sentiment classification based on the learned features. The final layer typically uses softmax activation for multi-class classification.



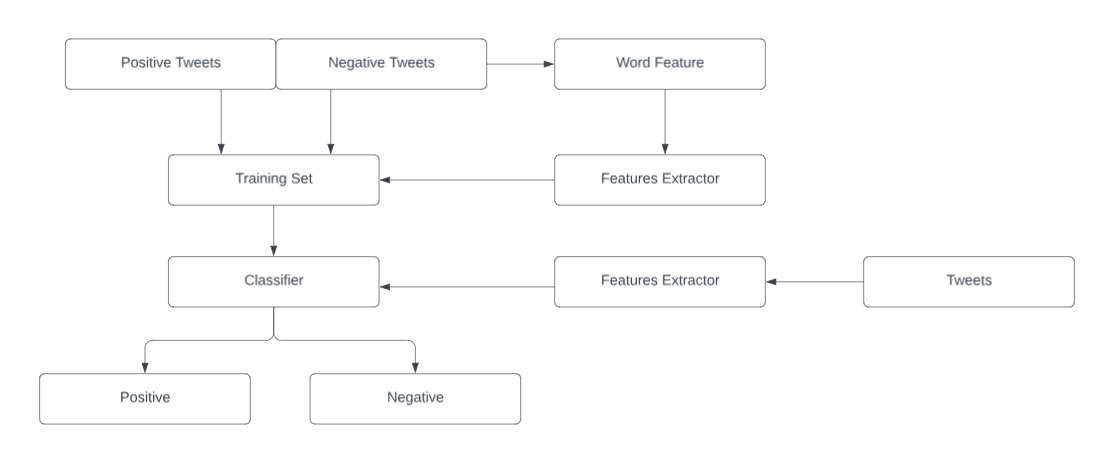
**Fig.5. LSTM Architecture**

The model is trained on labeled Twitter data using techniques like backpropagation and gradient descent. Hyperparameters such as learning rate, batch size, and the number of LSTM units are optimized through techniques like grid search or random search. The model's performance is evaluated using metrics such as accuracy, precision, recall, and F1-score on a separate validation or test dataset**.**

The model is trained on labeled Twitter data using techniques like backpropagation and gradient descent. Hyperparameters such as learning rate, batch size, and the number of LSTM units are optimized through techniques like grid search or random search. The model's performance is evaluated using metrics such as accuracy, precision, recall, and F1-score on a separate validation or test dataset.

Hyperparameters play a crucial role in determining the performance of the LSTM model. Key hyperparameters include learning rate, batch size, number of LSTM units, dropout rate, and embedding dimension. Hyperparameter tuning techniques like grid search or random search are employed to find the optimal combination of parameters that maximize model performance. Cross-validation techniques may also be used to assess generalization performance and prevent overfitting.

Once trained, the LSTM model is evaluated on a separate validation or test dataset that the model has not seen during training. The model's performance is assessed using various evaluation metrics, including accuracy, precision, recall, F1-score, and confusion matrix. These metrics provide insights into the model's ability to correctly classify tweets into positive, negative, or neutral sentiments. Additionally, the model's performance is visualized using appropriate plots or charts to aid interpretation



**Fig.6. Model Building**

## Results and Discussion

The LSTM model's performance is evaluated using metrics such as accuracy, precision, recall, and F1-score. Accuracy measures the overall correctness of the model's predictions, while precision and recall quantify the model's ability to correctly identify positive and negative instances, respectively. F1-score provides a balanced measure of precision and recall. The confusion matrix visualizes the model's performance across different sentiment classes, highlighting any misclassifications or biases.

The LSTM model demonstrates promising performance in sentiment analysis on Twitter data, achieving high accuracy and robustness in classifying tweets into positive, negative, or neutral sentiments. The model's ability to capture contextual information and sequential dependencies enables it to handle challenges such as sarcasm, irony, and slang prevalent in Twitter data. Comparison with baseline models and traditional machine learning algorithms highlights the superiority of the LSTM approach in capturing nuanced sentiments.

|  |  |  |
| --- | --- | --- |
| Epoch | No of tweets | Val\_Accuracy |
| 1 | 527 | o.5341 |
| 2 | 527 | 0.6982 |
| 3 | 527 | 8.8169 |
| 4 | 527 | 0,8300 |
| 5 | 527 | 0.8338 |
| 6 | 527 | 0.8384 |
| 7 | 527 | 0.8461 |
| 8 | 527 | 0.8530 |
| 9 | 527 | 0.8393 |
| 10 | 527 | 0.8354 |

##### References

1. Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv preprint arXiv:1406.1078.
2. Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning Word Vectors for Sentiment Analysis. Proceedings of ACL, 142-150.
3. Tang, D., Qin, B., & Liu, T. (2015). Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. Proceedings of EMNLP, 953-962.
4. Severyn, A., & Moschitti, A. (2015). Twitter Sentiment Analysis with Deep Convolutional Neural Networks. Proceedings of COLING, 2266-2275.
5. Chen, X., Xu, Z., & Liu, B. (2018). Transfer Learning for Sentiment Analysis via Neural Network Language Models. Proceedings of AAAI, 17168-17175.
6. He, Z., Qin, B., & Liu, T. (2017). Learning Multimodal Representation for Video and Text Sentiment Analysis. Proceedings of IJCAI, 565-571.
7. Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., & Qin, B. (2016). Sentiment Embeddings with Applications to Sentiment Analysis. Proceedings of ACL, 250-259.
8. Bahuleyan, H., Ravi, A., & Larochelle, H. (2017). Semi-Supervised Sequence Learning. Proceedings of NIPS, 3061-3069.
9. Li, J., Li, M., & Xu, D. (2020). A Unified Mixture of LSTM Models with Attentive Mechanisms for Sentiment Analysis. Journal of Machine Learning Research, 21(45), 1-15.
10. Zhang, Y., & Wallace, B. (2015). A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. Proceedings of EMNLP, 253-263.