The impact of varying Dataset sizes on Sentiment Analysis Model Performance

Xin Li
Science Academy
University of Maryland College Park
College Park, Prince George's County
chrislii@hotmail.com

***Abstract*—This paper digs into sentiment analysis, a way to figure out if people are expressing positive, negative, or neutral feelings in the vast sea of words on social media. Rather than getting into complex algorithms, we focus on how the size of the dataset affects how well sentiment analysis models work. Our dataset comes from the 2019 Indian General Election chatter on Twitter and Reddit, offering a close look at sentiments toward political leaders.**

We use Apache Spark for data processing, cleaning up the text to make it more understandable. The analysis shows patterns in how sentiments are distributed, uncovering potential biases in the models. We also look at how text lengths differ, especially due to Twitter's character limit. Examining common words on Twitter and Reddit reveals the platform-specific language quirks that models need to grasp.

Visual representations Word Clouds help us see the most frequent words for positive, neutral, and negative sentiments. We use LSTM models with self-trained Word2Vec embeddings, highlighting their ability to handle different text lengths and understand meanings.

Splitting the data and using baseline and complex models, we explore how the dataset size influences model performance. Results suggest that bigger datasets generally help, but they also bring challenges like imbalances and platform-specific differences. The paper closes with thoughts on where future research could go, suggesting ways to fix dataset imbalances and make data more representative. As technology and language keep changing, this study adds to our understanding of sentiment analysis and how it fits into our digital conversations.

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

The advent of the digital era in today's world has provided people with extensive opportunities to express their opinions, ranging from product reviews to personal emotions and perspectives on major events. All of these can be disseminated through written words on the internet. Social media platforms, as one of the primary venues for information exchange, generate thousands of texts every second, encompassing diverse thoughts, viewpoints, and attitudes. Behind this massive information flow, the crucial task of classifying and understanding these posts becomes paramount.

In this context, sentiment analysis has emerged as an art of extracting emotional insights from text. Utilizing advanced natural language processing techniques, the goal of sentiment analysis is to identify the emotions conveyed in a piece of text—whether they are positive, negative, or neutral. Particularly on social media, product reviews, and customer feedback platforms, where information is abundant, sentiment analysis acts as a lighthouse guiding us through the reefs of public opinion, providing direction for businesses, researchers, and decision-makers.

The importance of sentiment analysis in today's interconnected world can be attributed to several factors. Firstly, by understanding public sentiment, we can better respond to societal needs, address public concerns, and achieve more precise communication. Secondly, businesses can safeguard their brand reputation by monitoring social media comments related to their products and promptly addressing negative sentiments. Furthermore, sentiment analysis on product reviews and market feedback aids companies in gaining a comprehensive understanding of market trends and optimizing product strategies. Most importantly, sentiment analysis not only plays a role in the business domain but also provides governments with a gauge of public opinion, assisting in policy formulation and maintaining social stability. In this era of information overload, sentiment analysis has become a crucial tool for comprehending and guiding public emotions, offering decision-makers in various fields a safer and wiser course.

This project is not intended to delve into the algorithmic discussion of sentiment analysis models but rather focuses on experimenting with a dataset specific to certain topics and examining the impact of dataset size on the results. The dataset exhibits singularity and lacks general representativeness.

II. Significance and Motivation

The project possesses several layers of significance. First, it bridges the gap between big data analytical tools, like Apache Spark, and machine learning tasks, specifically in the domain of NLP and sentiment analysis. This fusion of technologies holds notable implications in contemporary data-driven environments.

Second, sentiment analysis serves as an invaluable tool with widespread applications. It can be employed to detect hate speech, cyberbullying, and even expressions of suicidal tendencies within online content. These applications contribute substantially to the common good by enhancing online safety and well-being. Furthermore, understanding the emotional undertones in social media comments can offer profound insights into public sentiment, thereby piquing the interest of businesses, governmental organizations, and researchers.

From a personal perspective, this project presents an opportunity for us to expand our knowledge and skill set in both big data tools and sentiment analysis. This is in line with potential career development in the fields of data science and analytics.

III. Inroduction of Data Set

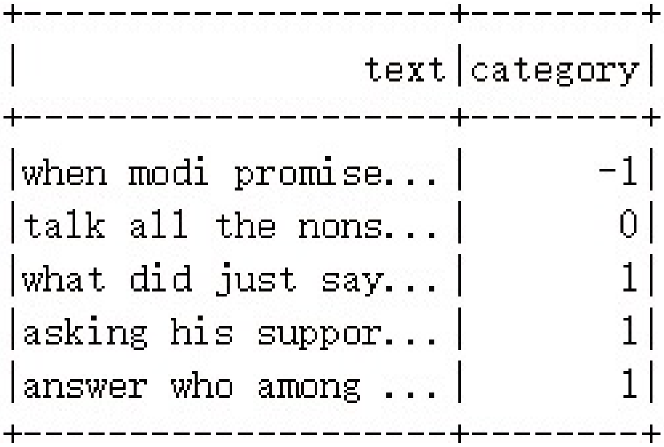
In the ever-evolving realm of contemporary political dialogue, social media platforms have emerged as influential mediums for the articulation of public sentiments and opinions. The 2019 General Election in India, marking a crucial juncture in the nation’s political narrative, experienced a notable upswing in online discussions, notably on platforms like Twitter and Reddit. This project undertakes a detailed examination of the sentiments conveyed through tweets and comments during this electoral period, with a particular emphasis on opinions aimed at pivotal leaders, including Narendra Modi, and the prevailing anticipation surrounding the designation of the next Prime Minister.

The dataset, meticulously compiled by Chaithanya Kumar A and his team from Kaggle[1], was curated using the Tweepy and PRAW APIs to extract tweets and comments. Leveraging Python’s re and NLP, the tweets and comments from Twitter and Reddit were carefully cleaned and assigned sentiment labels ranging from -1 to 1. A label of 0 indicates a neutral sentiment, 1 denotes a positive sentiment, and -1 represents a negative sentiment. The Twitter.csv dataset comprises approximately 163K tweets, while the Reddit.csv dataset includes around 37K comments. Each dataset’s structure involves two columns, with the first column containing the cleaned tweets and comments, and the second indicating their corresponding sentiment labels.

One of particular interest in this study is the exploration of how varying dataset sizes impact the performance of sentiment analysis models, specifically within the context of the 2019 Indian General Election. By analyzing subsets of different sizes, we aim to identify patterns and trends in model performance. The inclusion of both Twitter and Reddit data provides a nuanced perspective, considering the distinct characteristics and user engagement patterns inherent to each platform. The Twitter dataset, with over 163,000 individual entries, captures the succinct and real-time nature of tweet-based interactions. In contrast, the Reddit dataset, comprising approximately 37,000 unique rows, reflects the more elaborate and threaded conversations often found on the platform. Figure 1 shows some examples of this data set.

IV. Data Processing

Given the expansive scale of our dataset, totaling 200,000 rows of text, we strategically employed Apache Spark over the pandas library for its superior efficiency in handling such voluminous data. Databricks served as our chosen platform, harnessing Spark’s distributed computing capabilities to execute parallel processing across a cluster of machines. This approach facilitated seamless horizontal scaling, rendering Spark well-suited for the intricacies of large-scale data processing.



```
+--------------------+--------+
|                text|category|
+--------------------+--------+
|when modi promise...|      -1|
|talk all the nons...|       0|
|what did just say...|       1|
|asking his suppor...|       1|
|answer who among ...|       1|
+--------------------+--------+
```

Fig. 1. Example data

Within our meticulous data processing pipeline on Databricks using Apache Spark, we initiated a critical two-step transformation to refine the text data by addressing potential abbreviations and contractions. Scrutinizing each of the 200,000 rows and their constituent words, our pipeline systematically identified and converted abbreviations and contractions into their expanded forms. For instance, commonplace abbreviations like ”b4,” ”brb,” and ”lol” were replaced with their respective full forms such as ”before,” ”be right back,” and ”laugh out loud.” Similarly, contractions like ”aren’t”, ”wasn’t”, and ”how’d y” were meticulously expanded to ”are not”, ”was not”, and ”how do you,” respectively.

This rigorous preprocessing is fundamental in enhancing the readability and interpretability of the textual content by substituting prevalent abbreviations with their complete forms. The expansion of contractions contributes to linguistic explicitness, fostering a clearer comprehension of conveyed messages. These preprocessing steps hold paramount importance in natural language processing tasks, serving to standardize the text and thereby enabling more precise analysis, feature extraction, and downstream applications like sentiment analysis or topic modeling. By eradicating linguistic shortcuts, we ensure data consistency, thereby facilitating more effective text mining and information extraction from our extensive dataset.

When the initial transformation steps were finished, our focus shifted toward the meticulous cleaning of the organized dataset. We embarked on a comprehensive cleansing process, beginning with the removal of all HTTP and URL addresses, non-English alphabetic characters, extraneous spaces, and numerical values. Additionally, we addressed residual ”\n” characters that might persist after the initial cleaning, ensuring a thorough sanitization. Furthermore, all letters were uniformly converted to lowercase to maintain consistency. Following this cleaning phase, we subjected the sentences to tokenization, promptly scrutinizing the dataset for any instances of null data.

The subsequent step involved the removal of stopwords using the StopWordsRemover from pyspark.ml.feature[2], utilizing the default ”english” stop word list. Finally, the tokens were concatenated back into strings, and the cleaned data was saved as a CSV file, ready for the subsequent stages of analysis in Colab. The culmination of this data cleansing process resulted in a refined dataset, with the Reddit data ultimately comprising 37,024 entries and the Twitter data totaling 162,967 entries. This meticulous cleaning and preparation of the data lay the foundation for robust and accurate analyses in subsequent phases of our research.

V. Exploratory data analysis

A. Sentiment Category Distribution

In this section, we first delve into the distribution of label categories within the Reddit and Twitter datasets, as illustrated in Fig 2 and Fig 3, respectively. Despite marked differences in magnitude, both datasets reveal a shared underlying pattern in their sentiment categories. Notably, positive sentiments dominate, followed by neutral sentiments and a notably smaller count of negative sentiments, with the positive instances approximately twofold that of the negative category.

Examining these sentiment distribution patterns, it becomes evident that the datasets display significant imbalances, particularly in the prevalence of positive instances compared to negative instances. Such imbalances can substantially impact the performance and reliability of deep learning models trained on this data, predisposing them to favor the majority class of positive sentiments. Consequently, the model's ability to accurately predict and generalize patterns associated with the minority class (negative sentiments) may be compromised, leading to suboptimal performance and an increased likelihood of misclassification.

During the training process, the model may unintentionally prioritize optimizing for the prevalent positive class, potentially overlooking critical features and patterns crucial for discerning the minority negative class. As a result, the model's overall predictive capability may be compromised, affecting its capacity to make accurate and nuanced predictions across all sentiment categories.

B. Text Length Distribution by Category

Having analyzed the distribution of sentiment categories within both datasets, our focus now shifts to examining the distribution of text lengths across these platforms. Notably, the nature of Twitter, predominantly accessed via mobile devices, imposes a character limit of 280 characters per tweet. In stark contrast, Reddit caters to computer users and lacks such character constraints. This distinction is clearly reflected in the respective charts for each platform.

The text length distribution on Reddit's chart in Fig. 4 reveals a wide range, spanning from 0 to 6000 words. However, the majority of posts tend to cluster within the 1000-word range, showcasing the platform's flexibility for longer-form content. Conversely, the distribution of text lengths on Twitter in Fig. 5 is characterized by a more uniform pattern within the confined 280-character limit. This difference underscores the inherent constraints on tweet length imposed by the platform, in contrast to the varied and often more extensive textual expressions found on Reddit.

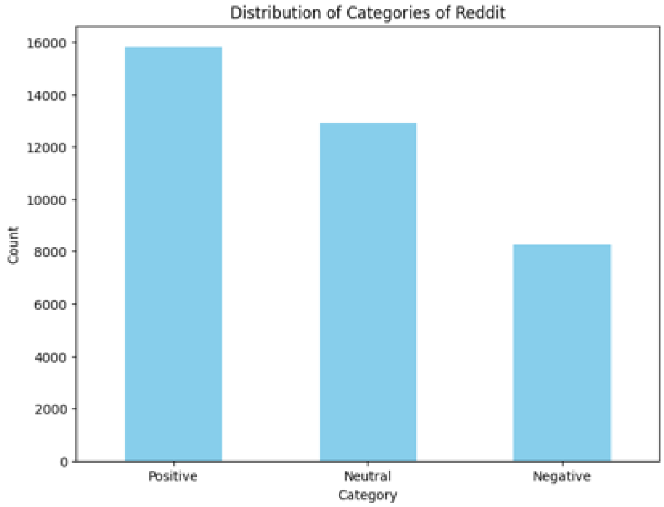


Fig. 2. Distribution of Categories of Reddit

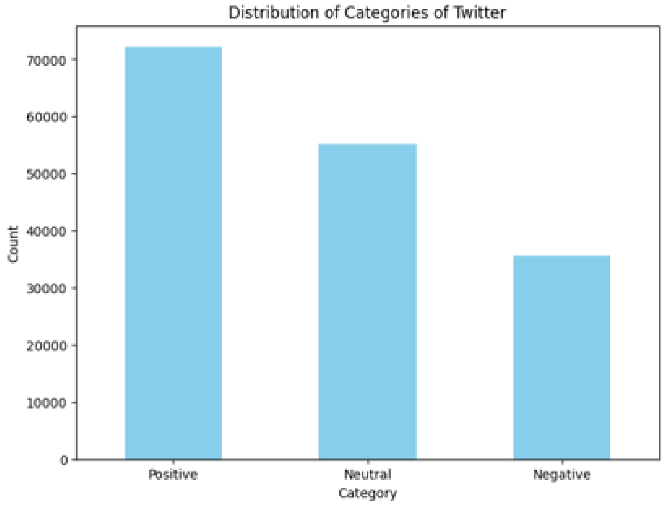


Fig. 3. Distribution of Categories of Twitter

C. Top Common Words

In this section, we gain a more intuitive understanding of the lexical disparities between Reddit data and Twitter data, along with an identification of the top 20 most frequently occurring words in each dataset. By comparing Fig. 6 and Fig. 7, we discern a pronounced contrast in the distribution patterns of the most common words between the two datasets.

The overall quantity of data in the Reddit dataset is lower than that of Twitter, yet its distribution appears more evenly spread. Generally, the term ”dollar” stands out as the most frequently occurring, with a word count from ”people” to ”modi” hovering around 5000 instances. The remaining words gradually decrease in frequency, falling within the range of several tens to around 300 instances. Notably, the quantity of the top-ranked word is not significantly disparate from that of the last-ranked word, with a ratio of approximately 3:1.

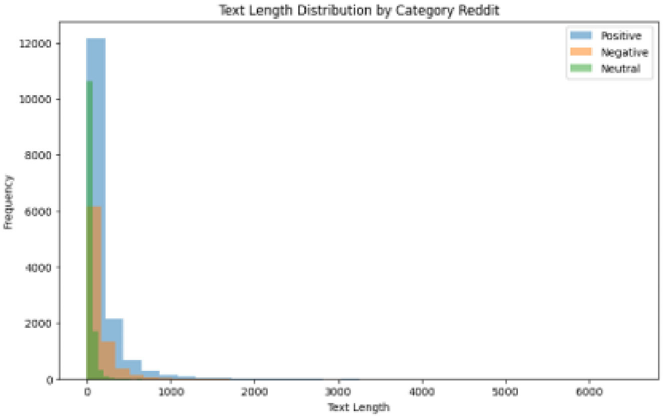


Fig. 4. Text Length Distribution by Categories Reddit

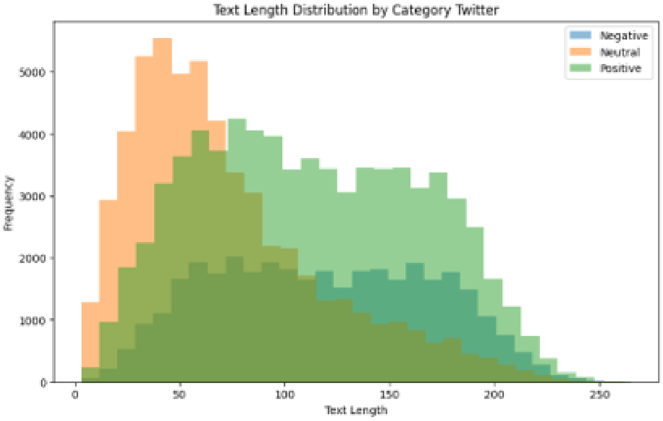


Fig. 5. Text Length Distribution by Categories Twitter

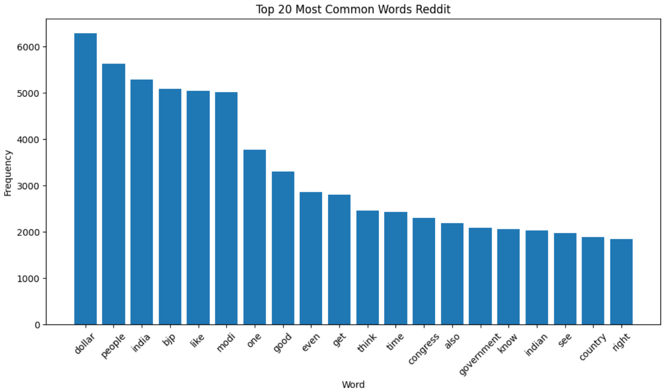


Fig. 6. Top 20 Most Common Words Reddit

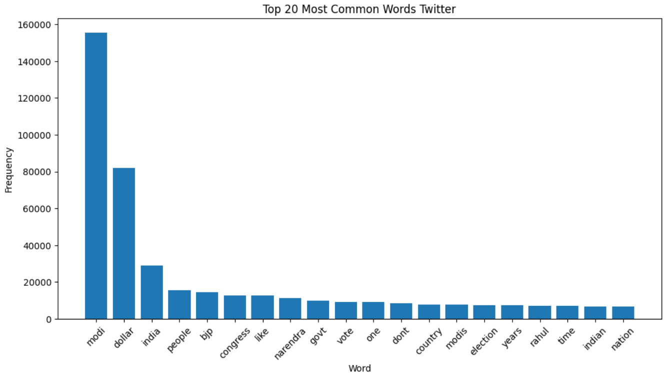


Fig. 7. Top 20 Most Common Words Twitter

Contrastingly, in the case of Twitter data (Fig. 7), the most prevalent word is ’modi,’ which approaches 160,000 instances. The second-ranked word, ’dollar,’ accounts for only around 80,000 instances. The third-ranked word, ’india,’ drops even further to around 30,000 instances. The subsequent words exhibit counts of fewer than 20,000, and the remaining words have a gradual and diminishing trend. The quantity of the top-ranked word is significantly disparate from that of the last-ranked word, with a ratio of approximately 16:1.

A distinction likely stems from the absence of character limits per post on Reddit, unlike the constrained nature of Twitter. This divergence prompts a critical question about the representativeness of the Reddit dataset, considering the distinct characteristics observed in the Twitter dataset. The disparities in word distribution underscore the impact of platform-specific constraints on content creation and suggest the need for a nuanced understanding of these variations when interpreting and generalizing findings from each dataset.

D. *Word Cloud*

Fig.8, Fig.9, and Fig.10 are the word clouds for Positive, Neutral, and Negative categories respectively from the Reddit dataset. In a word cloud, font size typically signifies the importance or frequency of words. Specifically, larger font sizes indicate higher frequency or, to some extent, greater significance of a word within a given text. In the Positive and Negative word clouds on Reddit, the terms ”india” and ”people” exhibit considerable prominence. However, in the Neutral word cloud, the term ”dollar” emerges as the most salient.

Fig.11, Fig.12, and Fig.13 are the word clouds for Positive, Neutral, and Negative categories respectively from the Twitter dataset. In Twitter data, ”modi” and ”dollar” emerge as the most prominent terms in the Positive, Neutral, and Negative word clouds.

VI. Methodology

For this project, we will use LSTM with a self-trained Word2Vec as our sentiment analysis model. LSTMs offer advantages in sentiment analysis due to their proficiency in capturing complex relationships and dependencies in text

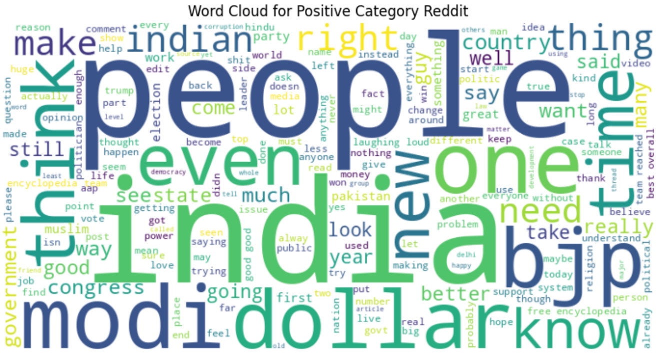


Fig. 8. Word Cloud for Positive Category for Reddit

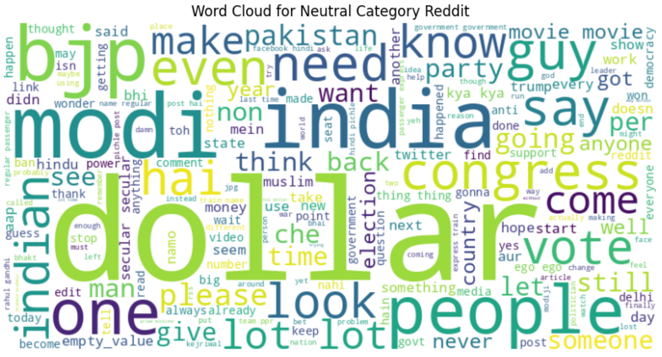


Fig. 9. Word Cloud for Neutral Category Reddit

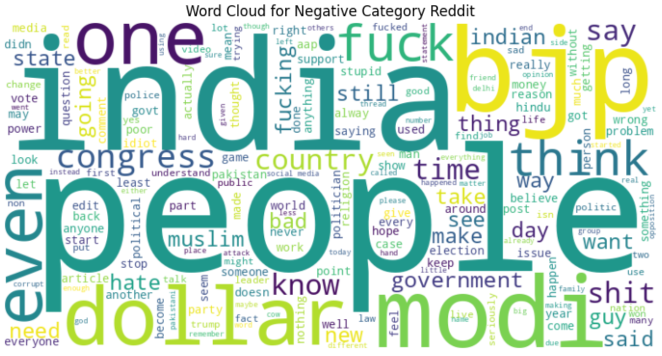


Fig. 10. Word Cloud for Negative Category Reddit

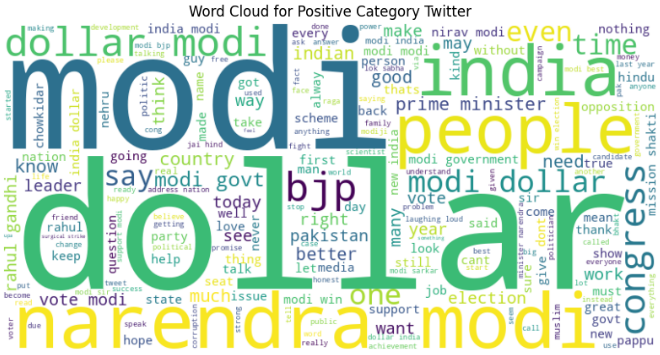


Fig. 11. Word Cloud for Positive Category Twitter

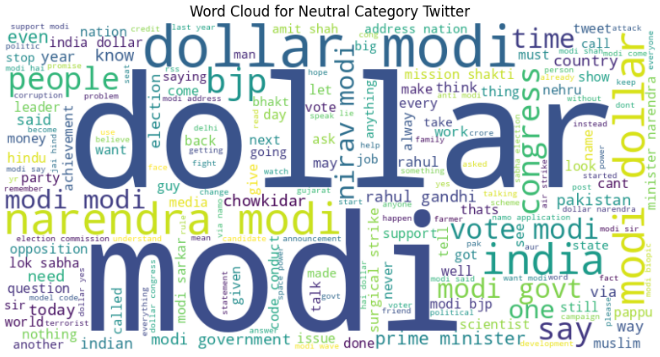


Fig. 12. Word Cloud for Neutral Category Twitter

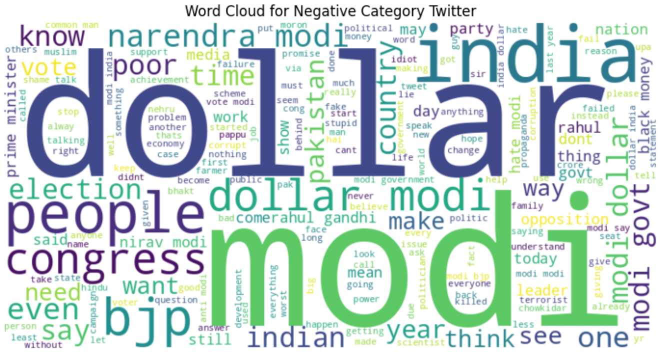


Fig. 13. Word Cloud for Negative Category Twitter

data. Understanding the nuances of language is crucial in sentiment analysis, and LSTMs excel at identifying patterns over extended sequences, enabling the model to recognize emotion-carrying information even in complex sentences. The unique architecture of LSTM, with its memory units and ability to selectively retain or discard information, provides advantages in identifying contextual emotions. Moreover, the adaptability of LSTM to different input sequence lengths is crucial when processing various textual data sources, such as short tweets or lengthy product reviews. The bidirectional nature of some LSTM implementations adds complexity by taking into account both past and future context, providing a holistic understanding of emotional expressions.

Additionally, the self-trained Word2Vec word embedding offers semantic representation, capturing key information in the data to help the model better understand the context. As previously demonstrated, some of the data in the Reddit dataset exceeds 1000 words. Using self-trained Word2Vec embedding significantly reduces the dimensionality of input data without losing semantic information, enhancing the model's generalization ability to unseen vocabulary.

A. *Model Dataset*

This project aims to investigate how the size of the dataset influences the performance of the model. We intend to leverage our dataset in three distinct ways. Initially, we will exclusively employ the Reddit dataset to evaluate the model's performance on a smaller dataset. Subsequently, the exclusive utilization of the Twitter dataset will enable an assessment of the model's performance on a larger dataset. Finally, the combination of

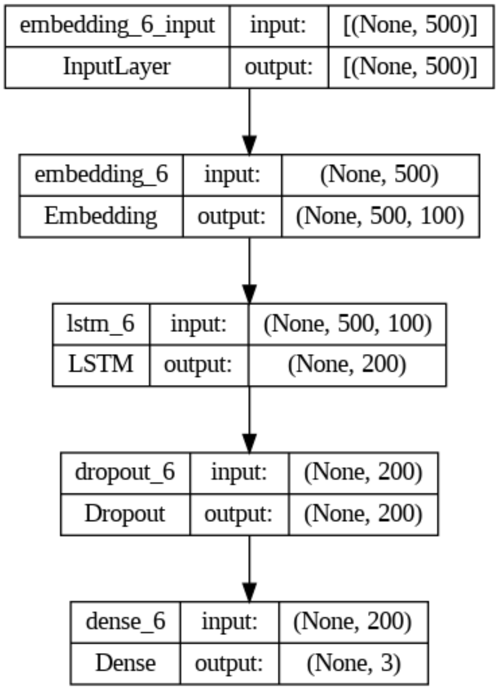


Fig. 14. Baseline model

both the Reddit and Twitter datasets will allow us to assess the model's performance on the largest dataset, encompassing the unique features of both the Reddit and Twitter platforms.

Regarding data segmentation, we will allocate 20 percent of the dataset for testing purposes. Subsequently, an additional 20 percent will be set aside from the remaining data as a validation set. The remaining portion of the data will constitute the training set.

B. Model Architecture

In order to underscore the differential impact of distinct datasets on model performance while maintaining control over experimental variables, we have devised two models. As previously mentioned, these two models will undergo sequential training and testing across the three datasets. In this experimental framework, the simple model serves as a baseline to ensure relative comparability of experimental outcomes. Concurrently, we introduce a complex model designed to simulate scenarios involving intricate algorithms, thereby facilitating a more comprehensive assessment of the model's adaptability and performance across varied datasets. This design reflects a conscientious consideration of the imperative to balance simplicity and model complexity within the experimental context.

1) Baseline Model: The baseline model shown in Fig. 14, leveraging a Long Short-Term Memory (LSTM) architecture for sequential data processing, is characterized by several additional key specifications. Commencing with an embedding layer that converts input word indices into dense vectors of 100 dimensions, this layer is initialized with pre-trained word embeddings from the Twitter dataset. Following the embedding layer, an LSTM layer boasting 200 memory units is incorporated to adeptly capture sequential dependencies within input sequences. In a bid to prevent overfitting, a dropout layer with a dropout rate of 0.5 is introduced. Concluding the architecture is a dense layer featuring a softmax activation function, equipped with three units catering to a three-class sentiment classification task.

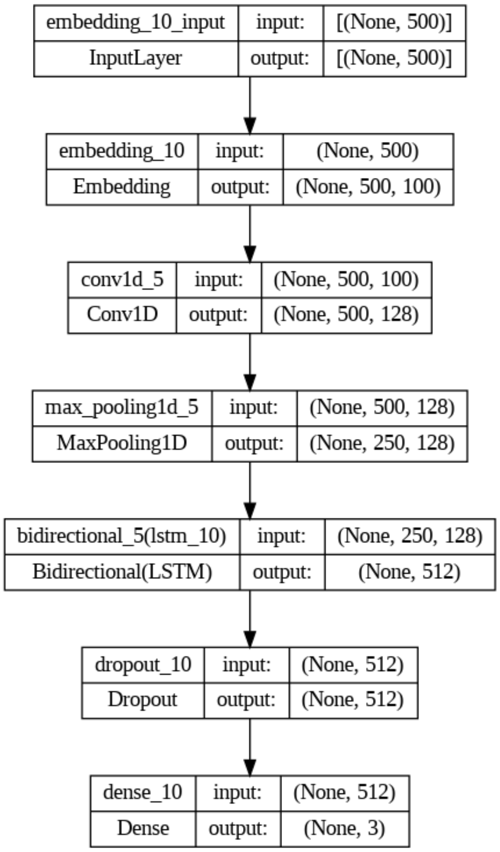


Fig. 15. Complex model

This model is configured with a batch size of 64, dictating the number of samples processed in each iteration during training. Furthermore, the training process is capped at a maximum of 30 epochs, with the integration of an early stop callback mechanism. This callback is instrumental in terminating training once the model's performance ceases to improve on a separate validation dataset, thereby optimizing computational efficiency. The model is compiled using sparse categorical cross-entropy as the loss function, employs the Adam optimizer, and utilizes accuracy as the evaluation metric. It's noteworthy that the model architecture is expressly tailored for execution on a distributed training strategy and harnesses TPU acceleration within the Google Colab environment.

2) Complex Model: The complex model shown in Fig. 15 also executes a distributed training strategy and harnesses TPU acceleration within the Google Colab environment. The model initiates with an embedding layer, utilizing pre-trained word embeddings from our dataset. Following this, a Convolutional

Neural Network (Conv1D) layer with 128 filters, a kernel size of 3, and a rectified linear unit (ReLU) activation is employed, enhancing the model's ability to discern spatial features. Subsequent spatial downsampling is achieved through a MaxPooling1D layer with a pool size of 2. The model's sequential understanding is further enriched with a Bidirectional Long Short-Term Memory (LSTM) layer containing 256 memory units. To mitigate overfitting, a Dropout layer with a dropout rate of 0.4 is introduced. The model concludes with a dense layer featuring a softmax activation function with three units, aligning with the three-class sentiment classification task.

Key configurations include a batch size of 64, dictating the number of samples processed in each training iteration. The optimization strategy employs Stochastic Gradient Descent (SGD) with a learning rate of 0.1, a momentum of 0.8, and a decay rate dynamically calculated for 30 epochs. The model is compiled using sparse categorical cross-entropy as the loss function, and accuracy serves as the evaluation metric. The model also incorporates an early stopping mechanism like the baseline model did.

VII. Result

TABLE I
LSTM Baseline mode Result

	Accuracy	F1	Precision	Recall	Runtime
Reddit Baseline	0.8579	0.8547	0.8571	0.8579	**3 mins**
Twitter Baseline	0.9185	0.9186	0.9197	0.9185	12 mins
R+T Baseline	**0.9222**	**0.9217**	**0.9222**	**0.9222**	15 mins

Table I presents the outcomes of the Baseline models applied to datasets from Reddit, Twitter, and a combined Reddit and Twitter dataset. Notably, the "R+T Baseline" model emerges as the top-performing model across all three datasets, showcasing superior performance in terms of Accuracy, F1 score, Precision, and Recall with a running time of 15 minutes.

TABLE II
LSTM complex mode Result

	Accuracy	F1	Precision	Recall	Runtime
Reddit Complex	**0.8135**	**0.7998**	**0.8171**	**0.8135**	**15 mins**
Twitter Complex	0.7628	0.7583	0.7631	0.7628	29 mins
R+T Complex	0.7805	0.7771	0.7817	0.7805	35 mins

Table II presents the outcomes of the Complex models applied to datasets from Reddit, Twitter, and a combined Reddit and Twitter dataset. Notably, the "Reddit Complex" model emerges as the top-performing model across all three datasets, showcasing superior performance in terms of Accuracy, F1 score, Precision, Recall, and Runtime with a running time of 15 minutes.

From Appendix A, an examination of the charts in Fig. 16, Fig. 17, and Fig. 18 reveals discernible signs of overfitting in the Reddit Baseline model. The Twitter Baseline and R+T Baseline models demonstrate comparable patterns, with the latter displaying superior performance, likely attributable to the increased volume of data within the R+T dataset. The observed trends in the confusion matrices across these three models are fundamentally analogous, suggesting a noteworthy impact of an imbalanced dataset on the outcomes. For instance, the prevalence of positive sentiment labels in the dataset correlates with higher predictions in the positive category, emphasizing the influence of class distribution on model predictions.

From Appendix B, On one hand, the Confusion Matrices of Complex models exhibit a pattern similar to that of Baseline models, suggesting a correlation with an imbalanced dataset. However, the trajectories of accuracy and loss graphs for Complex models differ from those observed in Baseline models. Contrary to expectations, the overall performance of Complex models, as evidenced by the results in Table II, is inversely related to the anticipated improvement with an increase in dataset size. On the other hand, the validation accuracy and loss of the Reddit Complex model do not entirely align with the trends observed in the rest of the Complex model group. Notably, the fluctuating pattern observed in the validation accuracy and loss of the Reddit Complex model may indicate a lack of representativeness in the Reddit dataset, impeding the model's ability to discern meaningful patterns. This outcome underscores the perspective that smaller datasets may yield suboptimal performance compared to larger, more diverse datasets, indirectly affirming the superiority of large datasets over smaller counterparts.

VIII. Future Work

In this project, we conducted an initial exploration into the impact of dataset size on the performance of sentiment analysis models. However, the presence of an imbalanced distribution within the dataset introduced certain distortions to our results. In future endeavors, mitigating the influence of dataset imbalance could be achieved by manually adjusting the number of labels while ensuring a consistent dataset size. Alternatively, without reducing the dataset, enhancing the representativeness of the existing data could be pursued through the incorporation of additional features, for instance, leveraging the SocialSent Subreddit[6] dictionary.

IX. Conclusion

In conclusion, we can conclude that, within the realm of simple models, larger dataset sizes are associated with stronger model performance. Larger datasets inherently offer greater diversity, a crucial factor in effective model training. Despite the considerable horizontal expansion of the Reddit dataset compared to the Twitter dataset, the performance exhibited by the former is not commensurately impressive. Admittedly, enhancing the input dimensions of the Reddit model might yield different results. However, to maintain variable consistency and ensure comparability between models, such adjustments were not implemented in this experiment. Yet, augmenting the input dimensions of a model introduces additional challenges, such as increased training time and heightened computational requirements. Based on the experimental outcomes and the aforementioned considerations, we can summarize that larger datasets contribute to improved model performance, with an emphasis on the vertical dimension of the dataset being prioritized over its horizontal dimension.

X. Appendix A

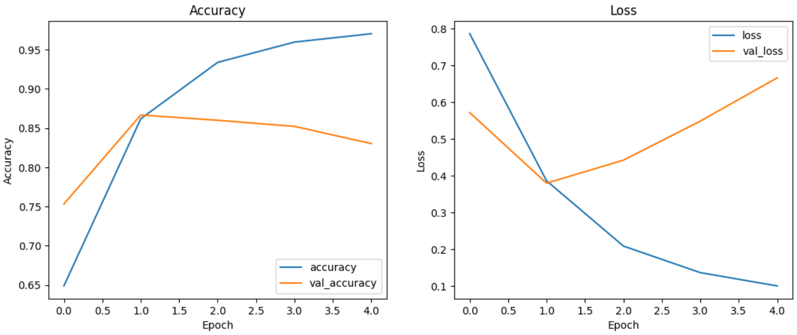


Fig. 16. Reddit Baseline graph

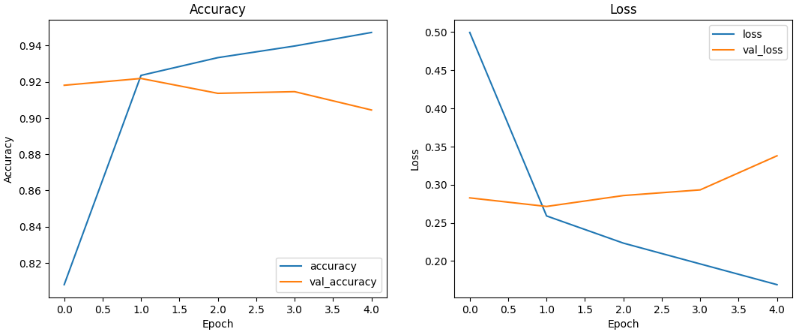


Fig. 17. Twitter Baseline graph

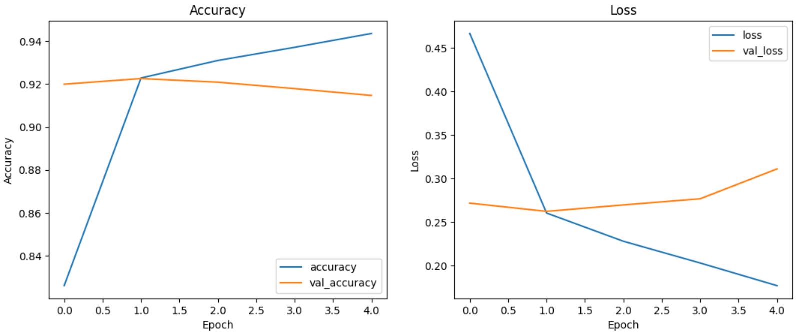


Fig. 18. R+T Baseline graph

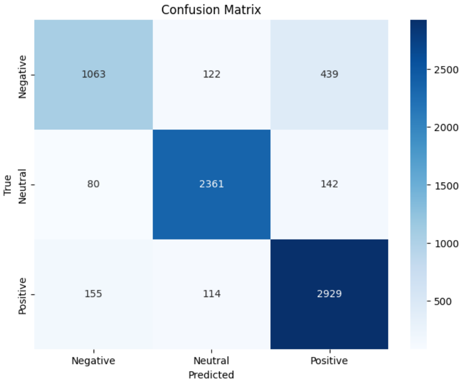


Fig. 19. Reddit Baseline Confusion Matrix

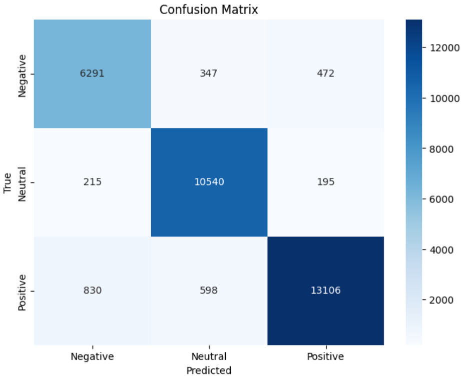


Fig. 20. Twitter Baseline Confusion Matrix

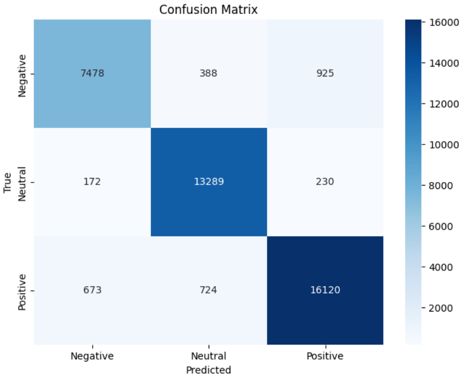


Fig. 21. R+T Baseline Confusion Matrix

XI. Appendix B

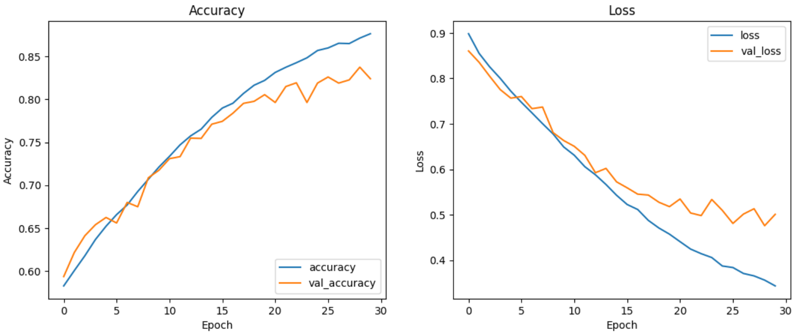


Fig. 22. Reddit Complex graph

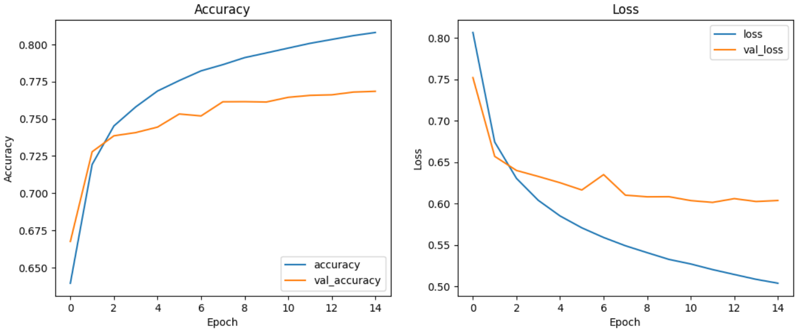


Fig. 23. Twitter Complex graph

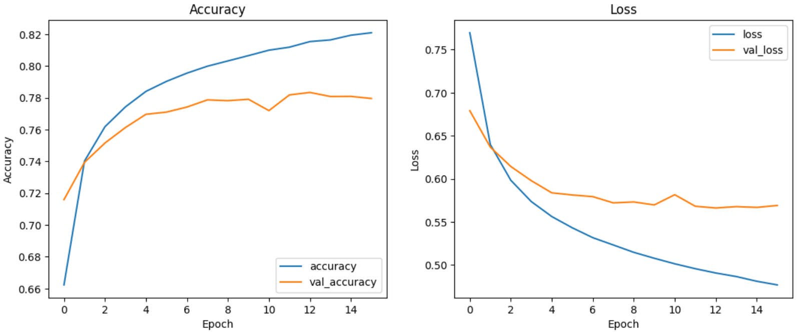


Fig. 24. R+T Complex graph

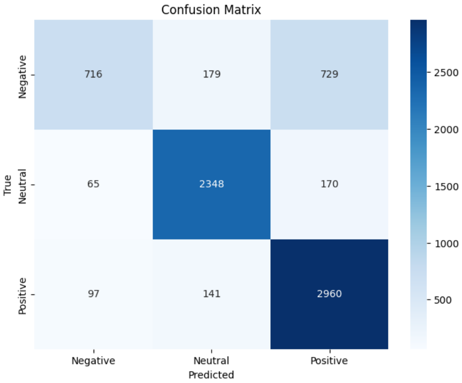


Fig. 25. Reddit Complex Confusion Matrix

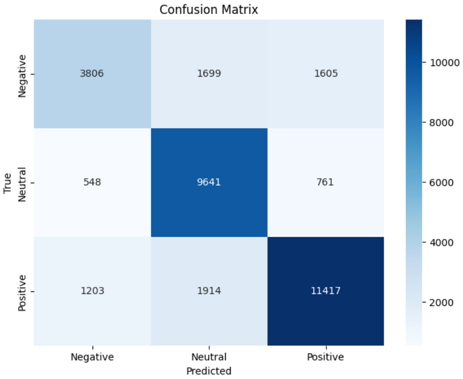


Fig. 26. Twitter Complex Confusion Matrix

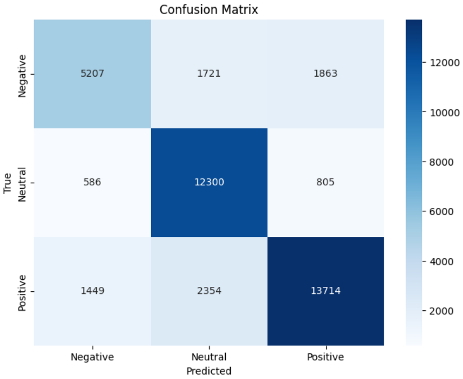


Fig. 27. R+T Complex Confusion Matrix

[1] [2] [3] [4] [5] [6]

References

[1] C. K. A. (2019). “Twitter and reddit sentimental analysis dataset.” Accessed on Dec. 15, 2023, [Online]. Available: https://www.kaggle.com/datasets/cosmos98/twitter-and-reddit-sentimental-analysis-dataset/data.

[2] (2023). “MLlib — Apache Spark.” Accessed on Dec. 15, 2023, [Online]. Available: https://spark.apache.org/mllib/.

[3] R. Kim. (2018). “Sentiment analysis with pyspark - towards data science.” Accessed on Dec. 15, 2023, [Online]. Available: https://towardsdatascience.com/sentiment-analysis-with-pyspark-bc8e83f80c35.

[4] S. Li. (2018). “Multi-class text classification with pyspark - towards data science.” Accessed on Dec. 15, 2023, [Online]. Available: https://towardsdatascience.com/multi-class-text-classification-with-pyspark-7d78d022ed35.

[5] W. L. Hamilton. (2014). “SocialSent: Domain-Specific Sentiment Lexicons.” Accessed on Dec. 15, 2023, [Online]. Available: https://nlp.stanford.edu/projects/socialsent/.

[6] williamleif. (2016). “GitHub - williamleif/socialsent: Code and data for inducing domain-specific sentiment lexicons.” Accessed on Dec. 15, 2023, [Online]. Available: https://github.com/williamleif/socialsent.