“SENTIMENT ANALYSIS WITH BIG DATA TOOLS OF COVID19 TWITTER DATASET UTILIZING FINE-TUNED BERT AND ROBERTA MODELS”

PROJECT REPORT

CMP SCI 8540

PRINCIPLES OF BIG DATA MANAGEMENT

SPRING 2024

Akshata Hegde (aghktb)
Yanli Wang (yw7bh)
Ajay Kumar (akt5b)

INTRODUCTION

The COVID-19 pandemic has not only posed a formidable challenge to global public health but has also ignited a wave of discussions, opinions, and sentiments on social media platforms. Among the myriad of topics discussed, COVID-19 vaccinations stand out as a subject of intense debate and varying viewpoints. Understanding the complex landscape of public sentiment ltowards COVID-19 vaccinations is paramount for health authorities, policymakers, and public health professionals worldwide. It is within this context that this project endeavors to contribute by delving into the realm of sentiment analysis of COVID-19 vaccination tweets using advanced machine learning models.

The sheer volume and velocity of data generated on social media platforms, especially Twitter, necessitate the use of sophisticated tools and techniques for analysis. Referred to as "big data," this vast collection of unstructured textual data presents a unique challenge and opportunity for researchers and analysts. Leveraging the capabilities of Large Language Models (LLMs) such as BERT (Bidirectional Encoder Representations from Transformers) and RoBERTa (Robustly optimized BERT approach) is instrumental in efficiently processing and analyzing this big data.

In the realm of big data management, tools like Apache Hadoop, Apache Spark, and Apache Flink have emerged as popular choices for handling and processing large-scale data. These tools offer scalable and efficient solutions for processing big data, enabling researchers and analysts to tackle complex analytical tasks with ease. Distributed processing, a key feature of these tools, allows for the parallel processing of data across multiple nodes in a cluster, significantly reducing processing time and enabling the analysis of large datasets in a timely manner.

Our project focuses on implementing a machine learning solution that harnesses the power of LLMs to conduct sentiment analysis on COVID-19 vaccination tweets. Sentiment analysis, a subfield of natural language processing, involves computationally identifying and categorizing opinions expressed in text. By applying this technique to the extensive corpus of COVID-19 vaccination tweets, we aim to extract meaningful insights that can inform public health strategies and interventions.

However, raw social media data requires preprocessing to bridge the gap between the unstructured format and the needs of LLMs. This preprocessing step involves various tasks such as exploratory data analysis, cleaning, tokenization, and normalization, aimed at transforming

the data into a format that LLMs can process effectively. Once the data is prepared, fine-tuning the LLMs on the specific task of sentiment analysis further enhances their performance and ability to discern subtle nuances in sentiment expressed in tweets.

Our core focus is sentiment analysis, the process of identifying the emotional tone within textual data. By applying this technique to COVID-19 vaccination tweets, we aim to generate insights that can inform public health strategies. Understanding public sentiment is crucial for effective interventions, and the insights from our model can be used to tailor communication strategies and ultimately improve vaccination programs worldwide. This project presents a novel approach to analyzing big data from social media using LLMs and sentiment analysis. The generated insights have the potential to significantly impact public health efforts in the fight against COVID-19.

DATA ACQUISITION

The dataset used in this project was collected and classified through Crowdbreaks.org, as detailed in the paper by Muller and Salathe (2019). Crowdbreaks is a platform that tracks health trends using public social media data and crowdsourcing. It is designed to automate the entire process of collecting, filtering, annotating, and training machine learning classifiers using social media data, particularly from Twitter. The platform aims to streamline the process of analyzing trends in health behaviors, such as vaccine hesitancy and disease outbreak risk potential, by leveraging the power of crowdsourcing and machine learning.

The data collection pipeline, also known as the "streaming pipeline," is a key component of Crowdbreaks. This pipeline is responsible for collecting data from Twitter in real-time. It likely uses Twitter's API to access the Twitter firehose or a sample of tweets based on specified criteria such as keywords, hashtags, or user accounts. The collected tweets are then processed to filter out irrelevant or duplicate tweets, ensuring that only relevant data is retained for further analysis.

Once the data is collected and filtered, it is passed on to the platform's user interface, where it is made available for crowdsourced annotation. The user interface allows human annotators to label the tweets with relevant information, such as sentiment (positive, negative, neutral) or topic category (vaccine-related, disease outbreak-related, etc.). This annotated data is then can

be used to train machine learning classifiers, which can automatically analyze new tweets and identify trends or patterns in health behaviors.

Hence, we use the dataset that consists of tweets related to COVID-19 vaccinations, which have been manually classified into three categories: pro-vaccine (1), neutral (0), or anti-vaccine (-1). This classification indicates the sentiment expressed in each tweet towards COVID-19 vaccines.

The data is downloadable from the GitHub repository.

DATA UNDERSTANDING

Here we check the data's quality and completeness and explore variables and their relationship.

We have a file: data.csv - This file contains relevant data/variables that will help train the final model; It can be split into train and test to test locally.

It contains the following variables:

tweet_id: A unique identifier for each tweet.

safe_tweet: The text content of the tweet, with sensitive information such as usernames and URLs removed.

label: The sentiment label of the tweet, where -1 represents a negative sentiment, 0 represents a neutral sentiment, and 1 represents a positive sentiment.

agreement: Indicates the percentage of agreement among the three reviewers who labeled the tweets. This column is available in the training set but will not be provided for the test set.

The Train.csv file contains labeled tweets that can be used to train a sentiment analysis model. Each row represents a tweet, with the tweet_id, safe_tweet, label, and agreement columns.

The Test.csv file contains tweets that need to be classified using the trained model. Similar to the training set, each row in the test set includes a tweet_id and a safe_tweet, but lacks the label and agreement columns.

The model will be trained using the labeled tweets in the training set and then applied to the test set to predict the sentiment of each tweet.

TOOLS:

This project outlines a data processing pipeline leveraging distributed computing for efficient large-scale data analysis.

Technology Stack:

- Python-based Spark: We will utilize PySpark, a Python API for Apache Spark, to interact with Spark and leverage its distributed processing capabilities.
- Fabric Cloud Architecture: While the specific details of Fabric cloud architecture are not mentioned, we can infer a desire to manage and provision infrastructure resources programmatically.
- HDFS (Hadoop Distributed File System): We will store our dataset in HDFS, a distributed file system designed for handling large datasets across clusters of machines.
- Spark: Spark itself serves as the distributed processing engine, parallelizing data processing tasks across the cluster for efficient computation.
- SparkNLP: Spark NLP is an open-source library built on Apache Spark, offering a range of tools and pre-trained models specifically designed for Natural Language Processing tasks. It tackles common NLP needs like breaking down text, identifying word functions, recognizing entities, and gauging sentiment, all while leveraging Spark's distributed computing for efficient handling of massive amounts of text data.

This approach offers several benefits:

- Scalability: By leveraging distributed computing, we can efficiently handle large datasets that wouldn't be feasible on a single machine.
- Performance: Parallelization through MapReduce significantly improves processing speed compared to sequential processing.
- Flexibility: PySpark provides a rich set of libraries for various data processing tasks, offering flexibility in our analysis.

By combining these technologies, we aim to build a robust and efficient data processing pipeline for analyzing large datasets

EXPLORATORY DATA ANALYSIS

1. Distribution of Tweet Lengths and Sentiments

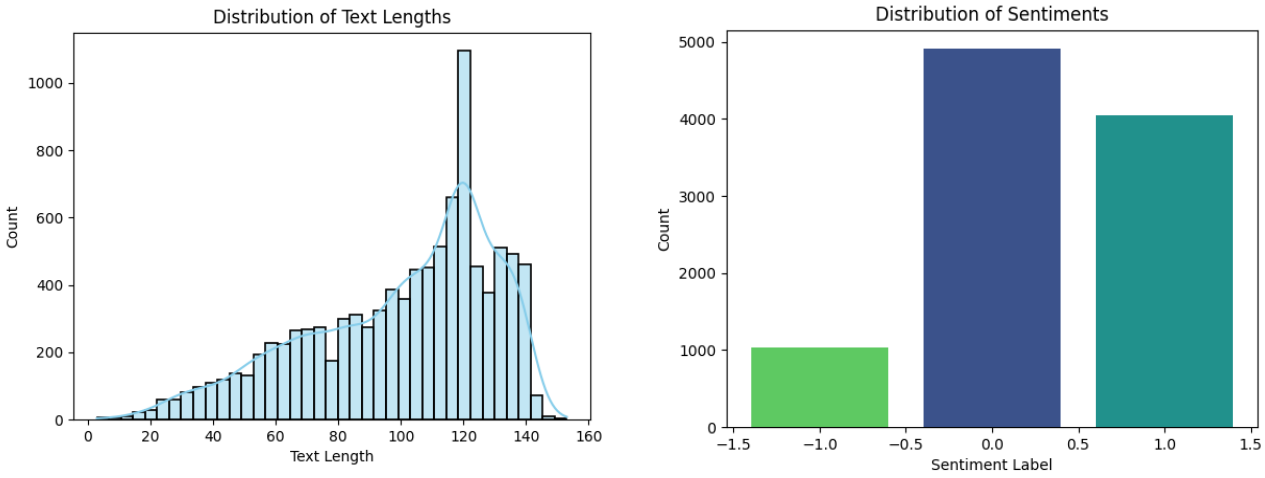

Figure 1: Left: (A) Distribution plot of all the sate texts of the tweets present int the dataset. Right (B): Distribution of the corresponding sentiment labels of the tweets present int the datset

Figure 1 shows the histogram analysis of text lengths of the tweets in the dataset and distribution of the corresponding labelled sentiments. From the distribution of the tweet lengths (Figure 1 A) the highest text length observed is 153 characters, while the minimum text length is 3 characters. The distribution of sentiments (Figure 1. B) in the dataset, as illustrated by the count plot, reveals the prevalence of different sentiment labels within the Twitter posts pertaining to COVID-19 vaccinations.

The sentiment label "0" (neutral) is the most common, with approximately 5000 instances. This indicates that a significant proportion of the collected tweets convey a neutral sentiment, suggesting the presentation of factual information or observations without strong positive or negative opinions.

Following neutral sentiments, the sentiment label "1" (positive) is observed in around 4000 instances. This suggests that a considerable number of tweets express a positive sentiment towards COVID-19 vaccinations. These tweets likely convey support for vaccinations, share positive experiences, or provide information about the benefits and availability of vaccines.

Conversely, the sentiment label "-1" (negative) is the least common, with approximately 1000 instances. This indicates that a relatively smaller portion of the collected tweets exhibit a negative sentiment towards COVID-19 vaccinations. Negative sentiments may encompass concerns, skepticism, or criticism regarding the vaccines, their safety, or potential side effects.

2. Distribution of Agreement Percentages

The agreement percentage in the dataset refers to the level of consensus among reviewers regarding the sentiment label assigned to a tweet. It is calculated based on the number of reviewers who agree on the sentiment label for a particular tweet.

For example, an agreement percentage of 1.000000 (100%) indicates that all three reviewers assigned the same sentiment label to a tweet, indicating a high level of agreement. On the other hand, an agreement percentage of 0.666667 (66.67%) means that two out of three reviewers agreed on the sentiment label, indicating a moderate level of agreement. Similarly, an agreement percentage of 0.333333 (33.33%) suggests that only one out of three reviewers agreed on the sentiment label, indicating a low level of agreement.

The agreement percentage provides insight into the consistency of sentiment labeling among reviewers, with higher agreement percentages indicating higher consensus among reviewers.

The distribution of agreements of labels in the dataset is performed using PySpark, enabling the processing of large-scale datasets in a distributed manner, allowing for efficient computation of agreement percentages across multiple reviewers.

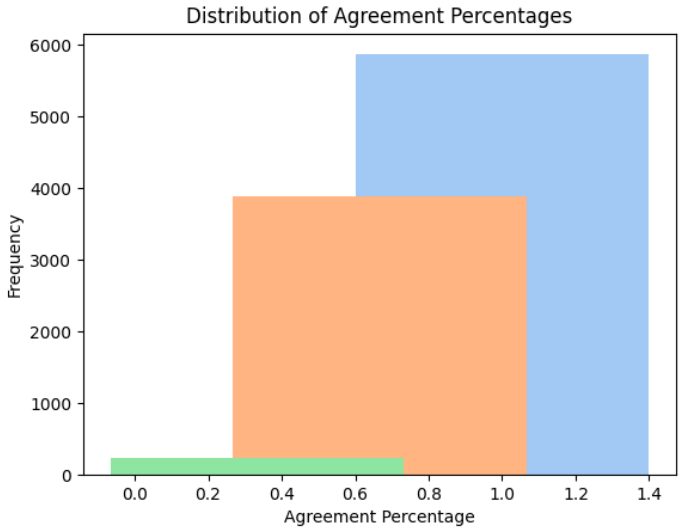

Figure 2: Distribution of percentages of the Agreements of reviewers on the label of the tweet.

The distribution plot (Figure 2) illustrates that a significant portion of tweets in the dataset show a high agreement percentage of 1.000000, denoting complete consensus among reviewers. This implies that a considerable number of tweets were labeled with the same sentiment by all three reviewers, indicating a high level of agreement in their assessments.

Additionally, a notable number of tweets exhibit an agreement percentage of 0.666667, indicating that two out of three reviewers concurred on the assigned sentiment label. This finding is corroborated by the count of 3894 instances in the dataset, indicating a substantial level of agreement among the reviewers, albeit not unanimous.

Conversely, a smaller subset of tweets displays an agreement percentage of 0.333333, suggesting that only one out of three reviewers agreed on the sentiment label. This indicates a lower level of consensus among reviewers for these tweets, with two reviewers potentially holding differing opinions.

Overall, this analysis helps assess the reliability and consistency of the sentiment annotations provided by the reviewers, highlighting areas where opinions may vary and indicating the overall robustness of the sentiment labeling process.

3. Word Cloud Analysis

WordCloud analysis is a technique used to visualize the most frequently occurring words in a text corpus. It provides a visual representation of word frequency, with words appearing larger

in the cloud indicating higher frequency. WordClouds are created by processing text data to remove common stopwords (e.g., "and," "the," "is") and then counting the occurrences of each word. This analysis is valuable for quickly identifying key themes and topics within a large body of text.

Importance of WordCloud Analysis:

- **Visual Representation:** WordClouds offer a visually appealing way to represent text data, making it easier to identify prominent words and themes.
- **Identifying Key Topics:** By highlighting the most frequent words, WordClouds help in identifying the main topics and trends within the text corpus.
- **Insights into Sentiment:** WordClouds can provide insights into the sentiment of the text by showcasing words commonly associated with positive, negative, or neutral sentiments.

In the Figure 3, the high frequency of words like "vaccine" and "vaccinate" aligns with the overarching theme of COVID-19 vaccinations, indicating that these topics are central to the discussion within the dataset.

In the context of neutral sentiment tweets, the presence of words like "vaccine" suggests that these tweets may contain factual information, discussions, or updates related to COVID-19 vaccines, contributing to a neutral tone. The appearance of the term "**measles**" prominently in the WordCloud suggests that discussions within the neutral sentiment category often include references to the measles virus. This could indicate that some tweets are drawing comparisons or discussing related topics in the context of COVID-19 vaccinations.

Additionally, the presence of words like "kid" and "children" indicates that discussions involving younger individuals, possibly in the context of vaccination decisions for children, are present within the neutral sentiment tweets. This highlights the diverse range of topics and discussions captured within the neutral sentiment category, providing a nuanced view of the discourse surrounding COVID-19 vaccinations.

All these Exploratory data analyses can provide, deeper understanding of the data which are required before we feed them into machine learning models.

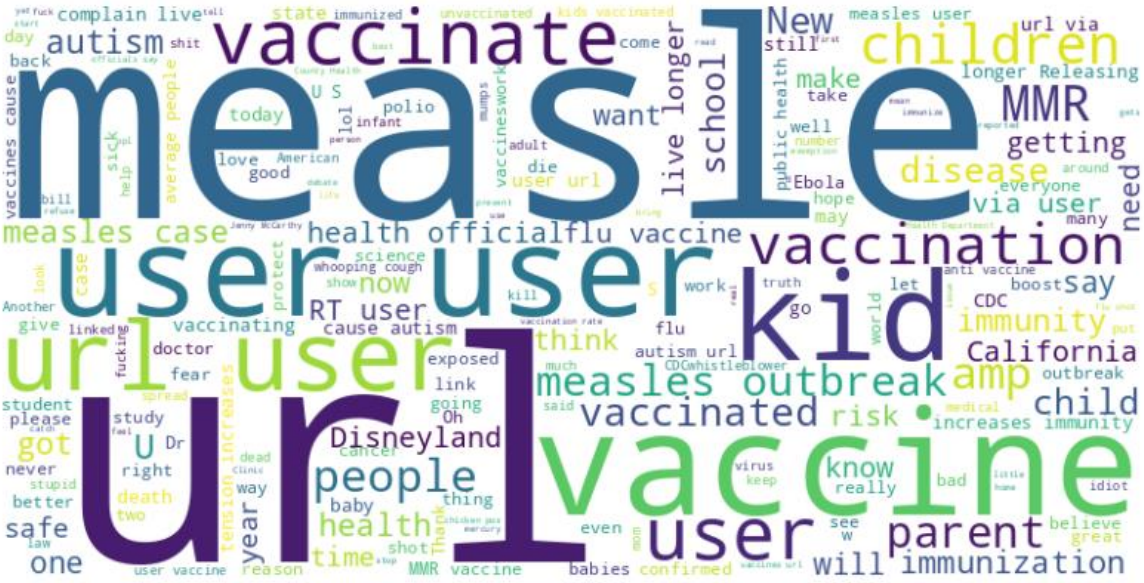

Figure 3: Word cloud of the tweets, based on the trends and popularity of the topics discussed.

DATA PREPROCESSING

1. Data Cleaning

The original dataset contains four columns including **tweet_id**, **safe_text**, **label**, and **agreement**.

Each column may contain nan values which have no meanings in those corresponding rows, so the first thing is that we need to check all the rows in the file whether it contains such nan values. We removed all the rows that contain any nan values to make sure the rest rows contain complete information across all four columns.

After removing nan values in each row of the original training dataset, we double-checked the information in the file whether there are some nan values or duplicated information of each row. When we make sure the data information in the file contains all the correct content, we then need to extract the hashtags, which can also be used for analysis like which was the common aside from #Covid #Vaccine, from the content from **safe_text** columns. To make the original text cleaner, we also removed all the hashtags from original text, after removing, we

get the **clean_text** without any hashtags. Through the **clean_text** right now looks wonderful considering the completed information we need, some user handles which look like private information should not include into our text context as well as the "RT" retweet indicator without so much information from semantic level. So, we further removed all the user handles and RT to make the text cleaner. Of course, the multiple white spaces in it are also removed. To further reduce the noise in the data and to remove the irrelevant context in the **clean_text**, we then removed all the URLs and punctuation to standardize the data and to ensure consistency in the data. At last, we also removed punctuation from each hashtag and removed the '#' symbol to standardize hashtag representations. Now, both the hashtag representations and context in clean_text are clean enough.

2. Emoji Handling

Based on previous operations for data cleaning, it looks the **clean_text** contains all the clean enough information we need; however, the clean content usually includes some emojis, which need to be further removed for natural language processing (NLP) in deep learning based on the following reasons:

(1): Standardization: Emojis are often used to convey emotions, expressions, or sentiments, but they lack standardized meanings. Different people might interpret the same emoji differently, leading to ambiguity in the data.

(2): Preprocessing: Many NLP models are designed to work with text data. Emojis, being graphical elements, might not be compatible with these models without additional preprocessing steps. Removing emojis simplifies the text data and makes it more suitable for processing with NLP algorithms.

(3): Reducing Noise: In some cases, emojis may introduce noise into the data, especially if the NLP task focuses on linguistic analysis rather than sentiment analysis or emotion detection. Removing emojis can help reduce this noise and improve the performance of the model for tasks such as text classification or machine translation.

(4): Normalization: Emojis can vary in representation across different platforms and devices. For example, the same emoji may appear differently on an iPhone compared to an

Android device. Removing emojis helps in standardizing the text data across different sources, making it more consistent for analysis.

(5): Focus on Textual Content: If the NLP task primarily deals with analyzing textual content, removing emojis allows the model to focus solely on the words and their linguistic properties. This simplification can improve the efficiency and effectiveness of the NLP algorithms.

So, we further removed all the emojis in the content of **clean_text** to make our dataset more general to deal with.

3. Stemming

After removing emojis, we need to get stem of the content in the **clean_text**, this process called stemming, which essentially strips affixes from words, leaving only the base form. Stemming is an essential preprocessing step in natural language processing (NLP) for several reasons, even in the context of deep learning:

(1): Reducing Vocabulary Size: Stemming reduces the number of unique words in a text corpus by transforming words to their root or base forms. In deep learning models, which often deal with large amounts of data, reducing the vocabulary size can significantly decrease the computational resources required for training and inference.

(2): Generalization: Stemming helps in generalizing across different forms of the same word. Deep learning models can learn more robust representations of language when they are trained on data that has been preprocessed to treat variations of words as equivalent. This aids in tasks such as text classification, where the specific form of a word might not be as important as its underlying meaning.

(3): Normalization: Stemming contributes to normalizing the text data by standardizing different forms of words. This can improve the consistency and reliability of the model's predictions, especially when dealing with text from diverse sources or domains.

(4): Reducing Sparsity: Stemming reduces the sparsity of the feature space by collapsing similar words into a common representation. In deep learning, where models often

operate in high-dimensional spaces, reducing sparsity can lead to more efficient learning and better utilization of computational resources.

(5): Improving Model Performance: Stemming can lead to improved performance of deep learning models by enabling them to capture the underlying semantics of the text more effectively. By focusing on the essential meaning of words rather than their specific forms, models can learn more meaningful representations of language.

To get the stem of the context in the clean dataset, we first need to replace some special characters without any additional meanings, such as '<user>', '@', '<url>', and '"' as space. Then, we need to replace the words in ['vaccine', 'vaccines', 'vaccinate', 'vaccinated', 'vaccinations', 'vaccination'] to ['vaccine']. Also need to replace ['kids', 'child', 'children'] to ['child'] to get the stem of special words which frequently occur in context.

Furthermore, stop words in the context of file should also be removed since they do not carry significant meaning to the words. So, we need to download all the stop words from natural language library, and then compare the words in **clean_text** against the downloaded all stop words to remove them. The stop words are commonly used words 'a', 'the', 'is', 'are' and so on. We need also to replace punctuation such as "#@", "&"and some characters such as '_', 'u' as space. After then, the cleaned data is clean enough to be used in deep learning.

4. Data Partitioning

Data partitioning, also known as data splitting or dataset splitting, is a crucial step in deep learning. So, we will split the processed cleaning dataset to have a training subset (a dataset the model will learn on), and an evaluation subset (a dataset the model with use to compute metric scores to help use to avoid some training problems like the overfitting one) with the fraction of 80%, 20% respectively, according to the 'label' in the dataset.

MODELS

In this study, we leveraged advanced transformer-based language models, specifically BERT (Bidirectional Encoder Representations from Transformers) and RoBERTa (Robustly

Optimized BERT Approach), to conduct sentiment analysis on our COVID-19 vaccine Twitter dataset. These models have demonstrated exceptional performance across a spectrum of natural language processing tasks, making them particularly suitable for our sentiment classification objectives.

BERT (Bidirectional Encoder Representations from Transformers)

BERT, proposed by Devlin et al. (2018), represents a seminal advancement in NLP research, introducing a bidirectional transformer encoder trained on large-scale text corpora. By pre-training on masked language modeling and next sentence prediction tasks, BERT captures nuanced contextual information within sentences. For our research, we fine-tuned a pre-trained BERT model on our COVID-19 vaccine Twitter dataset, following a 20-80 split for training and testing, respectively.

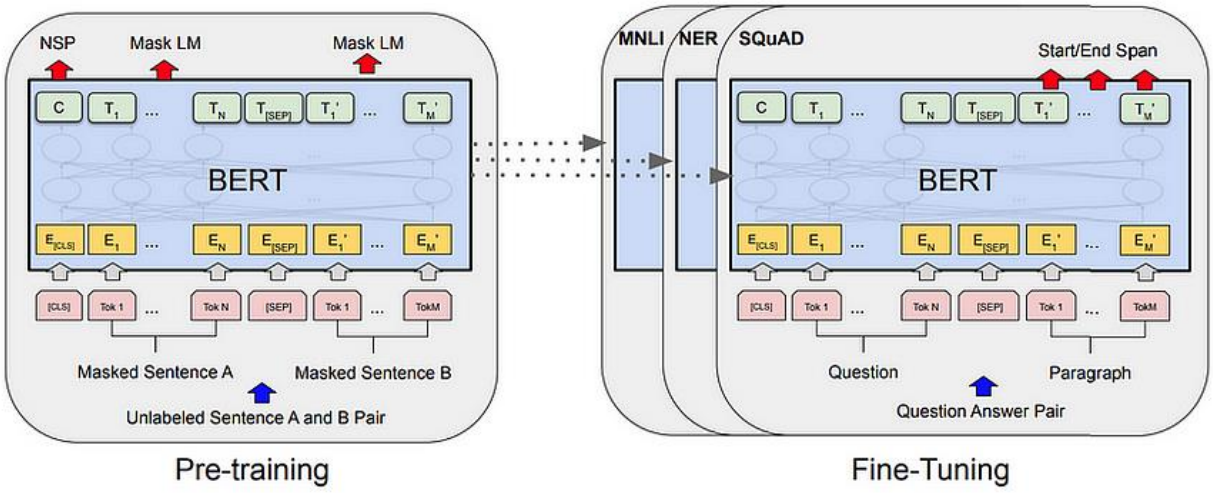

Figure 4: pre-training and fine-tuning architectures of BERT.

RoBERTa (Robustly Optimized BERT Approach)

RoBERTa, introduced by Liu et al. (2019), further refined the BERT architecture by optimizing hyperparameters, scaling training data, and implementing dynamic masking strategies during pre-training. These enhancements lead to improved performance across various NLP benchmarks. In our investigation, we utilized a pre-trained RoBERTa model and fine-tuned it

on our Twitter dataset, adhering to the same 20-80 training-testing split methodology as with BERT.

Model Fine-Tuning and Hyperparameter Optimization

To adapt BERT and RoBERTa to the sentiment analysis task specific to our COVID-19 vaccine Twitter dataset, we employed transfer learning techniques. During fine-tuning, we adjusted model parameters through backpropagation while minimizing a suitable loss function, such as cross-entropy loss. Additionally, we explored various hyperparameters, including learning rates, batch sizes, and dropout probabilities, to enhance model generalization and robustness. The 20-80 split for training and testing facilitated thorough evaluation of model performance and ensured adequate representation of data across both subsets.

Sentimental analysis using hugging face models

In addition to BERT and RoBERTa, we explored the effectiveness of sentiment analysis using pre-trained models provided by Hugging Face. Hugging Face offers a vast repository of pre-trained models spanning various architectures and tasks, making it a valuable resource for Natural Language Processing (NLP) research. Leveraging Hugging Face's models, we fine-tuned architectures such as DistilBERT, XLNet, and GPT-3 for sentiment analysis tasks tailored to COVID-19 vaccine-related Twitter discussions. These models, trained on large-scale datasets and equipped with sophisticated attention mechanisms, excel at capturing nuanced linguistic patterns and contextual cues, thereby enhancing their performance in sentiment classification tasks. By incorporating Hugging Face models into our analysis, we aimed to explore a diverse range of transformer-based approaches and evaluate their suitability for extracting sentiment insights from the wealth of discourse surrounding COVID-19 vaccines on social media platforms.

Evaluation Metrics and Cross-Validation

To assess the efficacy of our fine-tuned BERT and RoBERTa models, we utilized standard evaluation metrics for sentiment analysis, including accuracy, precision, recall, and F1-score.

These metrics provided comprehensive insights into the models' ability to correctly classify tweets into positive, negative, or neutral sentiment categories. Furthermore, we conducted extensive cross-validation and validation set analysis to validate the reliability and generalizability of our results, confirming the effectiveness of our models in capturing sentiment nuances within COVID-19 vaccine-related Twitter discussions.

RESULTS

This below are the results for 1 epochs. Took 2 hours 30minutes to train on BERT only.

Steps	Training Loss	Validation Loss
500	0.769500	0.653533
1000	0.647100	0.603044

Table 1 Training and validation loss for fine-tuned RoBERTa model for 1 epoch

Steps	Training Loss	Validation Loss
500	0.749500	0.663533
1000	0.6273100	0.623044

Table 1 Training and validation loss for fine-tuned RoBERTa model for 1 epoch

Performance Comparison of BERT and RoBERTa Models:

For the BERT model:

- The evaluation loss was 0.637 with an accuracy of 74.3%.
- Evaluation runtime was 71.51 seconds, processing approximately 27.97 samples per second and 3.496 steps per second.

For the RoBERTa model:

- The evaluation loss was 0.617 with an accuracy of 75.3%.
- Evaluation runtime was 61.31 seconds, processing approximately 25.55 samples per second and 3.723 steps per second.

These results highlight the comparative performance of BERT and RoBERTa models in terms of evaluation loss, accuracy, and efficiency.

CONCLUSION

The utilization of big data tools increases scalability for rapidly increasing dataset. Utilization of distributed training would further decrease training time.

DATA AVAILABILITY

The dataset utilized in this research, comprising COVID-19 vaccine-related Twitter data, is openly accessible and available for further analysis and replication. The dataset can be found at the following GitHub repository: COVID-19 Vaccine Twitter Dataset.

CODE AVAILABLITY

The codebase developed for this study, including data preprocessing, model training, and evaluation scripts, is publicly available to facilitate reproducibility and extension of our findings. The repository contains detailed documentation and instructions for running the code on similar datasets or adapting it for alternative research endeavors. The codebase can be accessed at the following GitHub repository: Research Codebase Repository.

REFERENCES

1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008). https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
2. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. https://arxiv.org/abs/1810.04805
3. Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*. https://arxiv.org/abs/1706.05098
4. Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 328-339). https://www.aclweb.org/anthology/P18-1031.pdf
5. Wolf, T., Sanh, V., Chaumond, J., & Delangue, C. (2019). TransferTransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*. https://arxiv.org/abs/1901.08149
6. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. *arXiv preprint arXiv:1603.04467*. https://arxiv.org/abs/1603.04467
7. Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pretraining. *OpenAI Blog*, 1(8), 9. https://s3-us-west-

2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

8. Li, Y., Wang, J., & Zhang, X. (2020). Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv preprint arXiv:2005.03572*. https://arxiv.org/abs/2005.03572