Interim-Project Report

Sentiment Analysis of Covid19 Vaccine Twitter dataset with fine-tuned BERT and RoBERTa LLM models

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

Introduction

This project tackles a critical public health challenge: understanding public sentiment towards COVID-19 vaccinations. This project tackles sentiment analysis of COVID-19 vaccination tweets using a Machine Learning model. The vast amount of textual data on Twitter, categorized as "big data," necessitates the use of Large Language Models (LLMs) for efficient processing.

We implemented a machine learning solution that leverages Large Language Models (LLMs) to analyze this data and extract valuable insights. However, raw social media data requires pre-processing to bridge the gap between the unstructured format and the needs of LLMs. This pre-processing will transform the tweets into a format suitable for LLM analysis.

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.

Dataset

The data comes from tweets collected and classified through Crowdbreaks.org [Muller, Martin M., and Marcel Salathe. "**Crowdbreaks**: Tracking Health Trends Using Public Social Media Data and Crowdsourcing." Frontiers in public health 7 (2019).]. Tweets have been classified as pro-vaccine (1), neutral (0) or anti-vaccine (-1). The tweets have had usernames and web addresses removed.

Data Analytics

Exploratory Data Analysis (EDA)

In the initial phase, we conducted an Exploratory Data Analysis (EDA) to become familiar with the data's characteristics. This helped us identify patterns, trends, and get a sense of the data's quality and suitability for our analysis.

Data Preprocessing

Next, we performed Data Preprocessing. This involved cleaning and preparing the data for analysis. We addressed missing values, corrected inconsistencies, and removed irrelevant information. This step ensured the quality and integrity of our data, leading to more reliable results.

Emoji Handling

We also addressed the presence of emojis in the data. This process involved deciding how to manage them. Depending on our project goals, we might have chosen to remove them, convert them to text descriptions, or create categories to group similar emojis.

Stemming

To further prepare the data for analysis, we employed a technique called Stemming. This process reduces words to their root form, allowing us to group related words together. This can be particularly useful for tasks like text classification or information retrieval.

Data Partitioning

Finally, we split the data set into two subsets through a process called Data Partitioning. A common split is into training and testing sets. We used the training set to build our model, and the testing set will be used to evaluate its performance on unseen data. This helps ensure our model generalizes well to new data.

Models

Leveraging BERT and RoBERTa for COVID-19 Twitter Analysis

This project explores the use of pre-trained language models (PLMs) for analyzing COVID-19 related information on Twitter. Here's a breakdown of how we'll utilize BERT and RoBERTa:

- **Pre-trained Powerhouses:** Both BERT (Bidirectional Encoder Representations from Transformers) and RoBERTa (A Robustly Optimized BERT Pretraining Approach) are powerful PLMs trained on massive amounts of text data. This training imbues them with the ability to understand the nuances of language and relationships between words.
- **Fine-Tuning for COVID-19 Tweets:** We won't be training these models from scratch. Instead, we'll leverage their pre-trained knowledge as a starting point. We'll then "fine-tune" them on a specifically curated COVID-19 Twitter dataset. This fine-tuning process adjusts the models' internal parameters to excel at the specific task of understanding COVID-19 related tweets.

Utilization of big data 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. Tools like Apache Ambari or Cloudera Manager can be used for this purpose.
  - **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.
  - **MapReduce Programming Model:** We plan to utilize the MapReduce programming model, a core concept in Spark, which allows us to parallelize data processing tasks by dividing them into separate "map" and "reduce" phases, leveraging the power of the distributed cluster.
  - **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.

Task distribution

Akshata (Team Member 1):

- **Task:** Dataset Acquisition and Exploratory Data Analysis (EDA)
- **Tools:** Familiarity with data sources and retrieval methods
- **Description:** Akshata will be responsible for obtaining the dataset we will be using for this project. Once acquired, she will perform Exploratory Data Analysis (EDA) to understand the data's characteristics, identify patterns and trends, and assess its suitability for our analysis.

Yanli (Team Member 2):

- **Task:** Data Preprocessing and Emoji Handling
- **Tools:** Pyspark, sparkNLP (or related tools)
- **Description:** Yanli will focus on cleaning and preparing the data for analysis using Pyspark. This includes addressing missing values, correcting inconsistencies, and removing irrelevant information. Additionally, she will handle emojis within the data, deciding how to manage them (e.g., removal, conversion to text descriptions, or categorization).

Ajay (Team Member 3):

- **Task:** Stemming and Data Splitting
- **Tools:** BERT/ROBERTA on Spark
- **Description:** Ajay will utilize techniques like Stemming to reduce words to their root forms, allowing for better grouping of related words. He will then split the data set into training and testing sets using tools like BERT or ROBERTA on Spark. The training set will be used to build our model, and the testing set will evaluate its performance on unseen data.

Thank you!