

GLOBAL NEXT CONSULTING INDIA PRIVATE LIMITED

PROJECT REPORT

REAL-TIME SOCIAL MEDIA SENTIMENT ANALYSIS

1.1 INTRODUCTION

In the digital age, social media platforms such as Twitter, Facebook, and Instagram have become key arenas for public expression, where users share opinions, emotions, and reactions in real time. This vast and continuously growing stream of user-generated content offers valuable insights into public sentiment across diverse topics, including politics, brands, events, and social issues. Real-time sentiment analysis is a technique that combines natural language processing (NLP), machine learning, and data streaming technologies to monitor, analyze, and classify the emotions or opinions expressed in social media posts as they happen. By evaluating whether the sentiment is positive, negative, or neutral, organizations and researchers can make timely decisions, track public opinion trends, manage brand reputation, and respond proactively to emerging crises.

This project/paper focuses on the development and implementation of a real-time social media sentiment analysis system, addressing key challenges such as handling high-velocity data, processing informal language, and ensuring scalable performance. The system aims to extract actionable insights by capturing and analyzing sentiment from live social media feeds, providing a valuable tool for businesses, governments, and researchers in today's fast-paced information landscape.

1.2 ABSTRACT

With the rapid growth of social media platforms, users continuously express opinions and emotions about various topics, creating a vast source of real-time data. Analyzing this data can provide critical insights for businesses, governments, and researchers to understand public sentiment and make informed decisions. This study/project presents a real-time sentiment analysis system that captures, processes, and classifies user sentiments from live social media streams.

The system leverages Natural Language Processing (NLP) techniques and machine learning algorithms to analyze the polarity (positive, negative, or neutral) of social media posts, primarily focusing on platforms such as Twitter. Real-time data acquisition is achieved using streaming tools, while text preprocessing, sentiment classification, and visualization are handled efficiently to ensure minimal latency and high accuracy. This implementation addresses challenges such as noisy text, slang, and high data velocity, offering a scalable and robust solution for sentiment monitoring insights, making it a valuable tool for real-time decision-making, trend detection, and public opinion tracking.

1.3 TOOLS TO BE USED

1. Google Colab

- Used as the main development environment for data preprocessing, model building, and sentiment analysis using Python.
- Provided access to cloud-based GPUs and Python libraries like NLTK, TextBlob, or VADER for Natural Language Processing tasks.

2. Microsoft Excel

- Used for organizing, cleaning, and performing initial analysis or formatting of the collected sentiment data.
- Helpful in preparing data before visualization or exporting results from Colab.

3. Power BI

- Used to create detailed, interactive visualizations and dashboards of the sentiment analysis results.
- Enabled advanced data exploration through charts, filters, and live updates.

4. Streamlit

- Used to build a real-time, web-based dashboard to showcase sentiment analysis results in an interactive and user-friendly way.
- Integrated with Python scripts from Google Colab for deploying the analysis as a live web application.

1.4 STEPS INVOLVED

1. EXTRACT DATA:

Extracting the contents of the provided zip file (archive.zip).

2. LOAD DATA:

Loading the extracted data (from sentimentdataset.csv) into a pandas DataFrame.

3. EXPLORE DATA:

Displaying the first few rows and information about the DataFrame to understand its structure and data types.

4. STORE DATA IN DATABASE:

Creating a SQLite database and storing the 'Text' and 'Sentiment' columns from the DataFrame into a table.

5. VERIFY DATABASE CONTENT:

Reading data back from the SQLite database to confirm successful storage.

6. PREPARE DATA FOR ANALYSIS:

Cleaning the text data by removing unnecessary columns and performing basic text cleaning (lowercase, removing URLs, special characters, extra whitespace).

7. ANALYZE SENTIMENT:

Applying TextBlob to the cleaned text to get sentiment polarity and compare it with the existing sentiment labels.

8. ANALYZE TRENDS AND DISTRIBUTIONS:

- Analyzing sentiment trends over time (daily and monthly).
- Visualizing the distribution of sentiment labels.
- Visualizing sentiment distribution by country.
- Analyzing and visualizing historical sentiment proportions over time.
- Visualizing the distribution of platforms.
- Visualizing average retweets and likes by sentiment.

9. CREATE STREAMLIT APP:

Writing a Python script (app.py) to create a web application using Streamlit to visualize the data and analysis results interactively.

10. RUN STREAMLIT APP:

Starting the streamlit application in the background.

11. EXPOSE STREAMLIT APP WITH NGROK:

Using ngrok to create a public URL to access the Streamlit application running in the Colab environment.

1.5 CONCLUSION

- The data was successfully loaded and preprocessed, including handling of missing values and text cleaning.
- Sentiment distribution was analyzed, revealing the prevalence of different sentiment labels in the dataset.
- Sentiment trends over time were visualized, showing how the frequency of various sentiments changed monthly.
- Geographic distribution of sentiment by country provided insights into regional sentiment patterns.
- Historical sentiment proportions highlighted shifts in the overall sentiment landscape over time.
- Platform distribution showed where the sentiment data originated.
- Average retweets and likes by sentiment offered a glimpse into how different sentiments correlate with engagement.

- A Streamlit application was created and exposed via ngrok, providing an interactive dashboard to explore the data and visualizations.

Next Steps:

- Advanced Text Preprocessing: Explore more advanced techniques like stemming, lemmatization, and handling of emojis or slang for improved sentiment analysis accuracy.
- Named Entity Recognition (NER): Identify and categorize named entities in the text to understand the context of the sentiment.
- Topic Modeling: Discover prevalent topics within the text data and analyze sentiment associated with each topic.
- Machine Learning Models: Train and evaluate machine learning models for automated sentiment classification.
- More Sophisticated Visualization: Create more interactive and detailed visualizations within the Streamlit app, potentially incorporating filtering or drill-down capabilities.
- Real-time Data Integration: Explore methods to integrate real-time data streams for continuous sentiment monitoring.

GITHUB LINK

<https://github.com/priyadharshinibd-collab/social-media-segment-analysis-streamlit-app.py>

STREAMLIT LINK

<https://kelli-vadose-colourably.ngrok-free.dev/>

