Enhanced Sentiment Analysis Web Application using VADER Model

Prof. Yogeshri Chaudhary
Information Technology
KDK College of Engineering
Nagpur, India
yogeshri.c@gmail.com

Jyoti Rangu
Information Technology
KDK College of Engineering
Nagpur, India
jyotirangu657@gmail.com

Sakshi Malve
Information Technology
KDK College of Engineering
Nagpur, India
malvelatal@gmail.com

Ritik Urkude
Information Technology
KDK College of Engineering
Nagpur, India
ritikurkude@gmail.com

Vedant Patel
Information Technology
KDK College of Engineering
Nagpur, India
vedantpatel07756@gmail.com

***Abstract*—The proposed application aims to automatically detect and analyze sentiments—positive, negative, and neutral—from social media comments. Leveraging a sophisticated sentiment analysis model based on Natural Language Processing (NLP) techniques, specifically the VADER model, comments will be classified into distinct sentiment categories. The user interface design utilizes the Figma tool for intuitive and visually appealing interaction. Frontend components are implemented using frontend technologies, facilitating seamless navigation and visualization of results. A PostgreSQL database is integrated using a Flask class in Python to develop a Json API, ensuring efficient data management and scalability. This comprehensive approach to sentiment analysis in social media comments contributes to the advancement of automated analytics in the digital landscape.**

***Keywords* : Vader Model, Json API, Flask, Sentiment.**

I. Introduction

In today's digital age, social media platforms serve as ubiquitous spaces for individuals to express their opinions and sentiments. However, analyzing this vast array of comments manually is impractical. To address this challenge, we propose the development of an application aimed at automatically detecting and analyzing sentiments—positive, negative, and neutral—from social media comments.

The core functionality of our application involves processing user-uploaded data, including comments containing emojis, in CSV format. The application then employs a sophisticated sentiment analysis model based on Natural Language Processing (NLP) techniques, specifically leveraging the VADER model. This model enables the classification of comments into distinct sentiment categories.

For the user interface design, we utilize the Figma tool, ensuring an intuitive and visually appealing experience. The frontend components are implemented using HTML, CSS, and JavaScript, allowing for seamless interaction and visualization of results. Users can easily navigate through the application and interpret sentiment analysis outcomes through graphical representations such as pie charts or graphs.

To ensure scalability and efficient data management, we integrate a PostgreSQL database using a Flask class in Python to develop a robust API. This API facilitates seamless communication between the frontend interface and the backend database, enabling the storage and retrieval of analyzed comment data.

Our proposed solution not only simplifies sentiment analysis for end-users but also demonstrates the synergy between cutting-edge UI design tools, frontend technologies, and backend frameworks. Through this integration, we present a comprehensive approach to sentiment analysis in social media comments, contributing to the advancement of automated analytics in the digital landscape.

II. Related Work

As we know that VADER stands for Valence Aware Dictionary and Sentiment Reasoner. This algorithm makes use of the valence of each word i.e.; nothing but the strength of each word towards that sentence. Even though this algorithm has this advantage its accuracy is less compared to that of the other sentiment analysis algorithms. So, we go for the improvement of VADER[1].

Liu (2012) and his team are seminal works in sentiment analysis, offering comprehensive insights into theoretical foundations and practical applications. Liu's work delves into various approaches, including lexicon-based methods, machine learning techniques, and sentiment classification algorithms. Meanwhile, Pang and Lee emphasize computational challenges and opportunities in sentiment analysis, highlighting the significance of feature selection, sentiment representation, and classification algorithms. Together, these works provide a robust foundation for understanding sentiment analysis methodologies and their applications[2].

The VADER algorithm achieved a maximum accuracy of 0.84%. Then we removed the information about the verb, and how it is used VADER We learned the spirit of the sentence by considering the verb and its indefinite form. We achieved an accuracy of 0.74%. sentences with observed verbs and 0.54% as accuracy for A sentence with no action[1].

Shayaa, Wai, Chung, Sulaiman, Jaafar, & Zakaria (2017) conducted a study on social media sentiment analysis regarding employment in Malaysia. Employing a lexicon-based approach, the research revealed a predominantly negative sentiment score associated with employment. The analysis was conducted across multiple social media channels, providing insights into public perceptions and attitudes towards employment issues in Malaysia[3].

C.J. Hutto and Eric Gilbert introduced Vader, a Parsimonious Rule-based Model for sentiment analysis of social media text. Their paper outlines the application of Vader on tweets and presents a comparative study with other sentiment analysis algorithms, including SentiWordNet and Support Vector Machines (SVM). The study concludes that Vader outperforms these algorithms in sentiment analysis tasks. However, it is noted that Vader does not support aspect-based analysis, which may limit its applicability in certain contexts[2].

Mandsberg (2019) and his team investigated the utility of social media as a resource for sentiment analysis of Airport Service Quality (ASQ). Utilizing machine learning techniques, the study aimed to analyze the quality of airport services. The research specifically focused on Twitter accounts to gather data for sentiment analysis. This approach provided valuable insights into public perceptions of airport services and demonstrated the potential of social media platforms as sources of sentiment data for evaluating service quality[3].

A. Sharada and P. Preethi Krishna proposed a model for verb-based sentiment representation in their research. This model offers a structured approach for developing sentiment analysis specifically tailored for movie reviews. The study employs various interfaces, tools, and Natural Language Processing (NLP) libraries to facilitate sentiment analysis. Notably, the authors identify Parts-of-Speech (POS) tags and utilize Text Blob for determining polarity and sentiment. This approach provides a comprehensive framework for analyzing sentiments expressed through verbs in movie reviews[1].

Jihang Mao and Wanli Liu present a BERT-based approach for automatic humor detection and scoring in their research. The study focuses on tweets from a corpus, aiming to predict whether a tweet contains a joke. To achieve this, they employ BERT assessment examination and pre-process humor analysis based on human annotation tasks, including Bidirectional Long Short-Term Memory (BiLSTM) and Long Short-Term Memory (LSTM) networks. This method is applied for multiple text classification tasks and includes error analysis of labeled and predicted scores, offering insights into the effectiveness of the proposed approach for humor detection in textual data. The study's findings contribute to advancing the field of computational humor understanding and provide a valuable framework for automated humor detection systems. Further research in this area may explore extending the approach to other forms of humor and linguistic nuances, enhancing its applicability in diverse contexts. Proposed approach for humor detection in textual data[1].

III. *METHODOLOGY*

In this research paper, we present a comprehensive methodology for sentiment analysis, focusing on the intricate data flow involved in the process. Beginning with data collection through the upload of CSV files, we delve into the preprocessing of textual data, integration of analytical models, and visualization of sentiment analysis results. By elucidating each phase's significance and intricacies, this paper aims to provide a detailed understanding of the data flow in sentiment analysis systems, laying the groundwork for effective sentiment analysis implementations is demonstrate in figure 2.

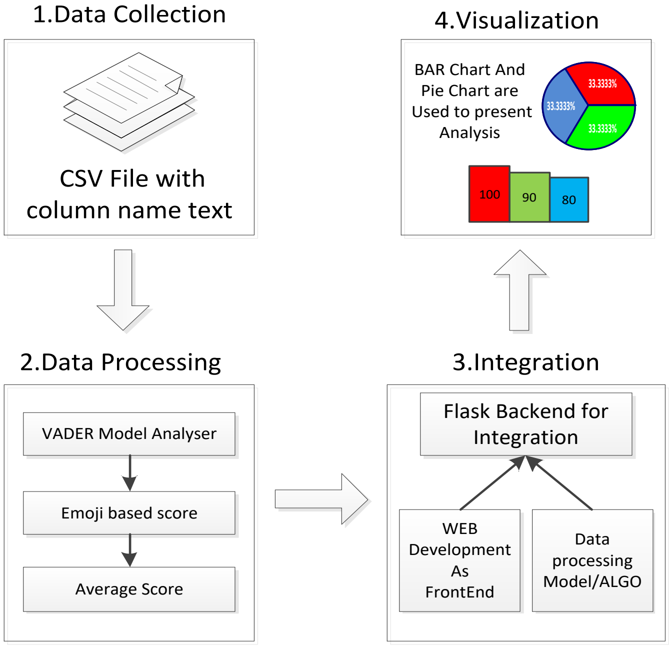
1.Data Collection
CSV File with column name text
4.Visualization
BAR Chart And Pie Chart are Used to present Analysis
33.3333%
33.3333%
33.3333%
100
90
80
2.Data Processing
VADER Model Analyser
Emoji based score
Average Score
3.Integration
Flask Backend for Integration
WEB Development As FrontEnd
Data processing Model/ALGO

Fig. 1. Data Flow

Data collection is crucial for sentiment analysis systems, providing the raw textual data needed for analysis. In our approach, we use CSV files for structured data collection. This allows users to upload text from various sources like customer reviews or social media posts, ensuring versatility. CSV files simplify data processing, enabling easy ingestion and organization. We incorporate validation checks to maintain data integrity, ensuring high-quality inputs. This systematic approach establishes a robust foundation for accurate sentiment analysis outcomes.

In Data Processing, we refine and prepare the raw textual data for sentiment analysis. Our methodology employs several key techniques:

1. **Vader Model Analysis**: We utilize the Vader sentiment analysis model to assess the sentiment of the text. This model considers lexical features such as punctuation, capitalization, and conjunctions to provide sentiment scores.

Algorithm:

```
def analyze_vader_sentiment(text):
    vader_scores = analyze_sentiment_with_vader(text)
    return vader_scores['compound'], vader_scores['pos'], vader_scores['neg'], vader_scores['neu']
```

2. **Emoji-Based Score**: Emojis often convey sentiment in textual data. Therefore, we incorporate emoji-based scoring to capture the sentiment expressed through emojis, enhancing the accuracy of our analysis.

   Algorithm:

```
def analyze_emoji_sentiment(text):
    emoji_scores = 0
    for char in text:
        if char in emoji_sentiments:
            emoji_scores += emoji_sentiments[char]
    sentiment_label = "Positive" if emoji_scores > 0 else ("Negative" if emoji_scores < 0 else "Neutral")
    return emoji_scores, sentiment_label
```

3. **Average Calculation**: To synthesize the sentiment scores obtained from the Vader model and emoji-based analysis, we calculate the average sentiment score for each piece of text. This approach provides a comprehensive understanding of the overall sentiment expressed.

   Algorithm:

```
def sentiment_analysis(text):
    vader_sentiment, vader_positive, vader_negative, vader_neutral = analyze_vader_sentiment(text)
    emoji_sentiment_score, emoji_sentiment_label = analyze_emoji_sentiment(text)
    average_sentiment = (vader_sentiment + emoji_sentiment_score) / 2
    return {
        "Vader Sentiment Score": vader_sentiment,
        "Vader Positive Score": vader_positive,
        "Vader Negative Score": vader_negative,
        "Vader Neutral Score": vader_neutral,
        "Emoji Sentiment Score": emoji_sentiment_score,
        "Emoji Sentiment Label": emoji_sentiment_label,
        "Average Sentiment Score": average_sentiment
    }
```

In the research paper, this programmatic representation of the sentiment analysis algorithm can be presented as a concise and executable solution for sentiment analysis tasks. It provides clear functions for analyzing sentiment using both the Vader model and custom emoji sentiment dictionary. The main algorithm orchestrates these functions to compute comprehensive sentiment analysis results, including Vader sentiment scores, emoji sentiment scores, and the average sentiment score. This concise representation can serve as a practical reference for researchers and developers interested in implementing similar sentiment analysis techniques.

In the Integration Phase, of our sentiment analysis system, we seamlessly merge various components to ensure efficient functionality. Our approach involves the following steps:

1. **Flask Backend**: We employ Flask, a lightweight and versatile web framework for Python, to develop the backend of our system. Flask allows us to create RESTful APIs that handle incoming requests from the frontend and interact with the sentiment analysis modules.
2. **Web Frontend**: The frontend of our system provides an intuitive user interface for interacting with the sentiment analysis functionalities. We develop the frontend using modern web technologies such as HTML, CSS, and JavaScript, ensuring a responsive and user-friendly experience.
3. **Integration with Vader Model**: The Flask backend integrates with the Vader sentiment analysis model to process incoming text data and generate sentiment scores. This integration enables real-time analysis of user-provided text through API endpoints exposed by the backend.

By integrating these components, we establish a cohesive system architecture that facilitates the seamless flow of data and interactions between the frontend, backend, and sentiment analysis modules. This ensures a robust and scalable solution for sentiment analysis tasks across various domains.

In the Visualization Phase of our sentiment analysis system, we leverage JavaScript libraries like D3.js or Chart.js to create dynamic and interactive visualizations, including bar charts and pie charts. Bar charts effectively illustrate sentiment distribution, while pie charts provide a clear breakdown of sentiments within the analyzed text data. By harnessing the capabilities of JavaScript libraries, we ensure that our visualizations are responsive, user-friendly, and seamlessly integrated into the web frontend. This approach enables users to intuitively explore and interpret sentiment analysis results, empowering them to make informed decisions based on the analyzed data.

The seamless integration of data collection, processing, integration, and visualization components forms the backbone of our sentiment analysis system. Through a structured approach to data collection facilitated by CSV file uploads, we ensure accessibility and ease of use for users, while maintaining data quality standards through validation checks. In the data processing phase, the utilization of advanced techniques such as the Vader sentiment analysis model and emoji-based scoring enhances the accuracy and depth of sentiment analysis outcomes. Integration with Flask backend and JavaScript libraries enables a cohesive system architecture, allowing for real-time analysis and seamless interaction with the sentiment analysis functionalities. Finally, through dynamic visualizations generated using JavaScript libraries, users can intuitively explore and interpret sentiment analysis results, gaining actionable insights into sentiment trends and patterns. By integrating these components effectively, our sentiment analysis system provides a comprehensive solution for understanding and analyzing textual data, empowering users to make informed decisions across various domains and applications.

IV. *RESULT AND DISCUSSION*

The results in Table I show the effectiveness of our proposed method in accurately analyzing sentiment from textual data. As the given example, "It's a good way of learning! 😍 👍 ", our algorithm assigned a VADER score of 0.8172 and an emoji score of 0.8, and an average score of 0.8086. This shows a great feeling , similar to the warm words and positive phrases used in the text. In contrast, in the second example, "I Hate the New way of learning! 😡 ", the algorithm returned a VADER score of -0.6114 and an emoji score of -0.8, and an average score of -0.7057.

TABLE I RESULT BASED ON PROPOSED ALGORITHM

Sr.	**Text**	**VADER Score**	**Emoji Score**	**Average Score**
1.	It's a good way of learning! 😍 👍	0.8172	0.8	0.8086
2.	I Hate the new way of learning! 😡	-0.6114	-0.8	-0.7057

This shows negative emotions followed by the use of the word "hate" and the angry emoji. These results demonstrate the ability of our algorithm to accurately capture emotions from multiple input data, providing better information for studying different emotions. However, further testing and evaluations are needed to evaluate the effectiveness of the algorithm on large and diverse datasets..

V. *CONCLUSION*

In summary, our research illustrates the valuable application of sentiment analysis in social media data, facilitating insights across various domains from business strategies to disaster response. Leveraging Flask, VADER model, and PostgreSQL API, we've developed an effective method for analyzing sentiment, including emojis. Future research should focus on analyzing beyond text to include images, videos, and audio, providing a more comprehensive understanding of user sentiments on social media platforms. Refining universal sentiment analysis models, exploring multimodal data analysis, and extending analysis to diverse social media platforms are areas that warrant attention. This work contributes to enhancing decision-making processes and understanding public sentiment in today's digital landscape.

.

REFERENCES

[1] Veena G, Aadithya Vinayak and Anu J Nair, "Sentiment Analysis using Improved Vader and Dependency Parsing," 2021 2nd Global Conference for Advancement in Technology (GCAT) Bangalore, India. Oct 1-3, 2021.

[2] C.J. Hutto and Eric Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text"., · January 2015 at ResearchGate Conference.

[3] Zulfadzli Drus and Haliyana Khalid, "Sentiment Analysis in Social Media and Its Application," at The Fifth Information Systems International Conference 2019.