International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 03 | March - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Twitter Sentiment Analysis Android App

Dhanraj Taware, Jagruti Patil, Pranali Gawale, Darshan Vadnere

Prof. Vijayalaxmi Tadkal

Bachelor's Degree in Computer Engineering AIML & Bharat College of Engineering

---

ABSTRACT: Social networking websites like Twitter generate a vast volume of text data every day, and sentiment analysis is a critical task in Natural Language Processing (NLP). The Twitter Sentiment Analysis app is perfect for NLP and machine learning learners, providing a simple-to-use interface to conduct sentiment analysis from tweets. In contrast to other approaches that rely on API access, this app employs OCR (Optical Character Recognition) to pull text from tweet screenshots, which is more user-friendly and accessible.

Key Words: Sentiment Analysis, NLP & ML, Hackathons, Skill Development, Collaboration.

1. INTRODUCTION

The Twitter Sentiment Analysis app is aimed at NLP and machine learning beginners, giving a simple method to analyze tweet sentiments. Sentiment analysis is normally API-dependent, which can be limited, expensive, or difficult. This app gets around that by applying OCR (Optical Character Recognition) to get the text from tweets in screenshots, making analysis easy and within reach.

Moreover, the app accommodates multiple input types, such as text, images, and audio, and can undertake sentiment analysis in varying ways. It also comes with a grammar corrector bot, thus emerging as a powerful tool for sentiment recognition and language polishing..

2. Body of Paper

The primary findings are presented in numbered parts that make up the paper's body. The arrangement of these parts should best showcase the content.

Referring back (or ahead) to particular passages is frequently crucial. The section number is included when making such references, as in "In Sec. 2 we showed…" or "Section 2.1 contained a description…." The words Section, Reference, Equation, and Figure are written out if they begin a sentence. These terms can be shortened to Sec., Ref., Eq., and Fig. when they appear in the midst of a phrase.
When an acronym appears for the first time, spell it out and then put the acronym in parenthesis, such as charge-coupled diode (CCD).

1.1 Project Plan

The Coding Club Project is intended to be an intelligent and dynamic environment that encourages developer cooperation, career preparation, and problem-solving. The project makes use of contemporary online technologies to offer an organized and effective coding environment, giving users the resources they need to improve their abilities and tackle real-world problems.

This platform has a number of features, including job postings, problem-solving modules, interview preparation, user authentication, and parts specifically for practicing Data Structures and Algorithms (DSA). By tackling industry-relevant scenarios, the real-time problem-solving technique helps users improve their coding skills and get ready for technical care retrospect's. Utilizing a MongoDB database for effective storage and retrieval of user progress, interview experiences, job applications, and issue solutions, the project places a strong emphasis on intelligent data processing and optimization. This makes systematic learning and a smooth user experience possible. The portal also seeks to offer an intuitive user experience with well-organized sections so that users may quickly move between job advertisements, interview preparation materials, and DSA tasks. The project helps to improve skills and closes the gap between theoretical learning and practical application by providing an organized and participatory approach to problem-solving.

Through this initiative, the Coding Club Project envisions a smarter, more efficient**,** and collaborative ecosystem for developers, enabling them to upskill, prepare for technical roles, and contribute to solving industry challenges effectively.

2. Review of Literature

The Coding Club Project is built on insights from extensive research in web development, coding platforms, and technical interview preparation. Studies highlight the importance of efficient data management (MongoDB), user-centric design (React), gamification, and structured DSA practice in enhancing learning experiences.

Research on coding platforms emphasizes real-time problem-solving, mock interviews, and job-oriented challenges as key to improving technical skills. By integrating these best practices, the Coding Club ensures a seamless, engaging, and effective

environment for developers to learn, practice, and prepare **for** careers**.**

2.1 Existing Systems

- Numerous sentiment analysis applications already exist with a focus, mainly, on text-based analyses from social sites such as Facebook, Twitter, and Reddit. Some of the most popular programs are:
- MonkeyLearn & Lexalytics – The applications offer sentiment analysis of texts through NLP methods but no image or voice analysis.
- Google Cloud NLP & IBM Watson – The solutions employ machine learning for sentiment labelling but not at no charge and need specialist knowledge.
- Hugging Face Sentiment Models – Pre-trained models have decent accuracy for text sentiment analysis but lack support for multi-modal inputs like screenshots and audio.
- VADER & TextBlob – These are light-weight sentiment analysis libraries but lack decent accuracy when analyzing complex emotions, sarcasm, or multi-lingual text.
- Most current sentiment analysis tools are text-based, and they use APIs to retrieve data. There is no simple option to analyze tweets from screenshots or audio inputs. Moreover, most programs are paid, rely on API access, or need substantial ML knowledge, which makes it hard for newcomers to start with them.

2.2 Literature Survey of Similar Ideas

- There are already many sentiment analysis tools with a concentration, primarily, on text-based analysis from social media like Facebook, Twitter, and Reddit. Some of the most widely used software are:
- MonkeyLearn & Lexalytics – The tools provide sentiment analysis of texts using NLP techniques but not image or voice analysis.
- Google Cloud NLP & IBM Watson – The solutions utilize maSentiment analysis or opinion mining is a part of Natural Language Processing (NLP) dealing with discovering the opinions and sentiment existing in text, image, and audio information. Sentiment analysis is necessary for social media trend analysis, customer reviews, and public sentiment analysis. Various methods, including lexicon-based approaches, machine learning algorithms, and deep learning models, have been used for sentiment classification. With more individuals relying on social media sites like Twitter for communication, sentiment analysis has become an important tool for businesses, researchers, and policymakers to understand user behavior and trends.
- Traditional sentiment analysis is primarily concerned with text inputs, but recent advancements have brought forth multi-modal sentiment analysis, including text, images, and speech processing. Optical Character Recognition (OCR) assists in extracting text from images, and this may be utilized for sentiment analysis of scanned documents or screenshots. Speech recognition may also facilitate sentiment extraction from audio recordings. All these advancements add to the accuracy and usability of sentiment analysis for different data types.
- chine learning to label sentiment but not free of cost and require expert knowledge.
- Hugging Face Sentiment Models – Pre-trained models have fair accuracy for analyzing the sentiment of text but no support for multi-modal inputs such as screenshots and audio.
- VADER & TextBlob – Light-weight libraries for sentiment analysis but no fair accuracy when analyzing subtle emotions, sarcasm, or multi-lingual text.
- Most sentiment analysis tools that are currently available are text-based, and they fetch data using APIs. There is no direct option to analyze tweets from screenshots or audio input. Additionally, most applications are paid, dependent on API access, or require a lot of ML expertise, making it difficult for beginners to begin with them.

3. Proposed System

The proposed system offers an integrated, structured, and personalized approach to learning and preparing for technical careers, addressing the gaps in existing coding clubs. By combining real-time problem-solving**,** job readiness features, and data-driven insights, the project ensures users have the tools they need to grow their skills and succeed in the tech industry.

3.1 Analysis/Framework/Algorithm:

Our Twitter Sentiment Analysis project is suitable for Machine Learning and NLP beginners with an easy-to-use interface to analyze sentiment from tweets, text input, images, and voice. The system is designed to be efficient, real-time, and accurate and employs OCR to extract text, ML models for sentiment prediction, and an interactive UI for improved accessibility:

Step 1: Problem Identification Issue: Most sentiment analysis tools accept manual text input and do not have the capability to analyze tweets through screenshots.

Solution: Develop an OCR-based system that captures text from tweet screenshots and conducts sentiment analysis automatically.

Step 2: Data Collection & Preprocessing

Issue: Raw tweets have noise such as emojis, hashtags, and URLs that impact sentiment analysis accuracy.

Solution: Apply text-cleaning methods such as tokenization, stopword removal, and stemming to clean the text data.

Step 3: Sentiment Classification

Issue: Simple keyword-based sentiment classification does not have contextual awareness.

Solution: ML models (Naïve Bayes, SVM, or Transformers) should be integrated to classify sentiments more precisely based on contextual sense.

Step 4: Image-Based Sentiment Analysis

Issue: Most sentiment platforms do not cater to analyzing emotions from images that include text.

Solution: Use OCR (Tesseract) to pull text from images and use NLP-based sentiment analysis.

Step 5: Audio Sentiment Analysis

Issue: Speech sentiment detection is usually absent in sentiment analysis tools.

Solution: Use speech-to-text translation and then text-based sentiment categorization for content analysis of speech.

Step 6: Grammar Correction Bot

Issue: The users may input grammatically flawed text, influencing sentiment accuracy.

Solution: Add a grammar correction bot to pre-process input text prior to sentiment categorization.

Step 7: Data Visualization

Issue: Unprocessed text output is unengaging and hard to read.

Solution: Show results in pie charts and emojis for improved user comprehension with the AnyChart library.

Step 8: User Interaction & Experience

Issue: Sentiment analysis software tends to have limited interactive elements and intuitive UIs.

Solution: Create a simple and intuitive UI with XML and Java, making the application easy to use for beginners.

Step 9: Scalability & Performance

Issue: Processing large amounts of tweets may delay processing.

Solution: Utilize multithreading and caching methods to optimize the app to provide real-time performance.

Step 10: Security & Privacy

Issue: User privacy issue in analyzing personal tweets.

Solution: Make API calls secure, process data in an anonymized manner, and encrypt data to safeguard user details.

3.2 System Architecture (Challenges of Sentiment Anlaysis Tool)

Existing sentiment analysis tools face several challenges in processing tweets effectively, especially when dealing with images, audio, and grammatical errors. The following issues highlight architectural and functional limitations, along with their proposed solutions:

1. **Text Extraction and Preprocessing:**
   - **Challenge**: Many tools require manual text input and do not process tweets directly from screenshots, limiting usability.
   - **Solution**: Implement OCR (Tesseract API) to extract text from images and preprocess it by removing unnecessary elements like emojis, hashtags, and URLs.
2. **Sentiment Analysis Model:**
   - **Challenge**: Basic keyword-based sentiment analysis lacks accuracy and contextual understanding.
   - **Solution**: Use machine learning (ML) models like Naïve Bayes, SVM, or Transformers for enhanced text classification, providing context-aware sentiment evaluation.
3. **Multi-Modal Sentiment Analysis:**
   - **Challenge:** Most sentiment analysis apps focus solely on text and ignore emotional cues from images and audio.
   - **Solution**: For image sentiment analysis, extract captions and detect emotions using deep learning-based image recognition models.
4. **Grammar Correction & Text Refinement**:
   - **Challenge:** Poorly structured sentences can affect sentiment classification accuracy.
   - **Solution:** Integrate a grammar correction bot using NLP models (e.g., GrammerGPT) to refine user input before sentiment analysis.
5. **Data Visualization & User Interface**

   Challenge: Many tools provide raw sentiment results without proper visual representation, making it difficult for users to interpret data.

   Solution: Implement AnyChart to display results in pie charts, bar graphs, and emojis, making insights visually appealing and easier to understand.

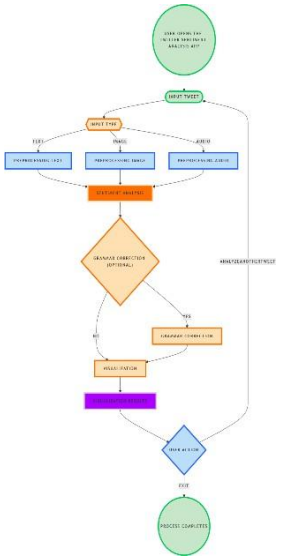


Fig -1: Figure

3.3 Data Model:

The data model for the Twitter Sentiment Analysis app follows an Entity-Relationship Model (ERM) to ensure structured and efficient data management. It consists of:

- User Profile: Stores user interactions and past sentiment analysis history.
- Tweet Data: Contains extracted tweet text, images, and audio files for analysis.
- Sentiment Results: Stores classified sentiments along with timestamps for tracking emotional trends.

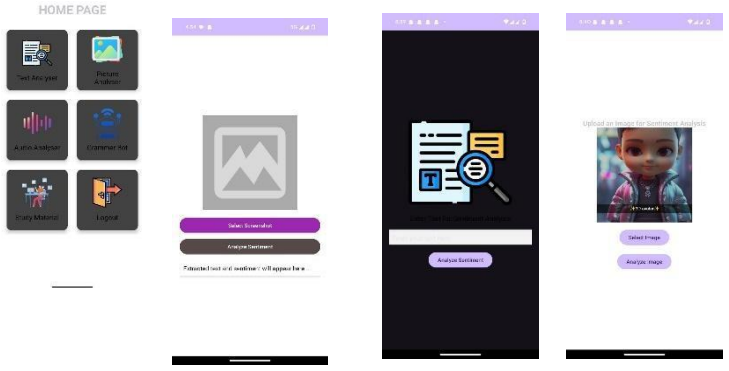
3.4 Methodology:

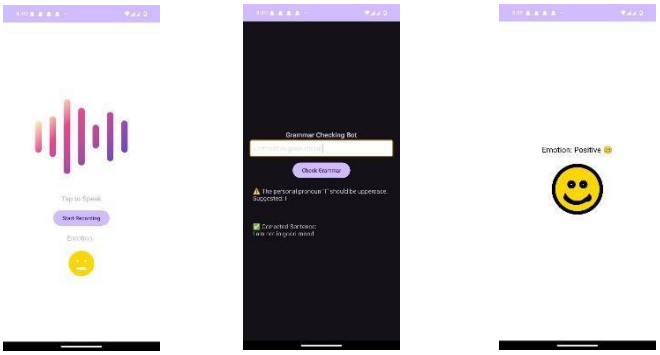
To guarantee effectiveness and scalability, the Twitter Sentiment Analysis project is developed using a systematic approach. The strategy consists of:

- API Integration: Utilizing external APIs like Sentiment140 API for sentiment analysis, OCR (Tesseract) for text extraction from images, and Speech-to-Text (STT) for audio sentiment conversion.
- Dataset Utilization: Using real-time Twitter data and public sentiment datasets to train and test the sentiment analysis model, ensuring high accuracy in classification.
- Machine Learning Models: Leveraging pre-trained NLP models and custom ML models to classify sentiments into Positive, Negative, or Neutral. GrammerGPT is used for grammar correction.
- Data Visualization & User Interface: Displaying results using pie charts, bar graphs, and emoji-based sentiment indicators with the AnyChart library for an intuitive user experience.
- Performance Optimization: Implementing efficient text preprocessing, removing noise from tweets, and handling different input formats (text, images, audio) for fast and accurate analysis.r enhance functionality and user engagement.

4.1 Proposed System Result:

The implemented Twitter Sentiment Analysis project provides a streamlined and efficient approach to analyzing sentiments from tweets using text, images, and audio inputs. The system is designed for NLP and machine learning beginners, offering an interactive and educational experience while ensuring high accuracy in sentiment detection..





5. Conclusion:

The Twitter Sentiment Analysis project provides a comprehensive multi-modal sentiment detection system using text, images, and audio. It helps NLP and ML beginners understand sentiment analysis, OCR, speech-to-text, and ML-based classification.The GrammerGPT-powered grammar correction bot enhances text processing.With real-time analysis, user-friendly visualization, and AI-driven insights, the system makes sentiment analysis accessible and engaging. Future improvements may include multilingual support, advanced sentiment contextualization, and improved ML models.

References:

- Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval, 2(1-2), 1–135.
- Cambria, E., Schuller, B., Xia, Y., & Havasi, C. (2013). New Avenues in Opinion Mining and Sentiment Analysis. IEEE Intelligent Systems, 28(2), 15–21.
- Go, A., Bhayani, R., & Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision. Stanford University.
- Liu, B. (2012). Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies, 5(1), 1-167.
- Hinton, G. (2012). Deep Learning for NLP and Speech Recognition. International Conference on Machine Learning.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780.
- Vaswani, A., et al. (2017). Attention is All You Need. Advances in Neural Information Processing Systems.
- Google Cloud Vision API Documentation. (n.d.). Retrieved from https://cloud.google.com/vision
- TensorFlow Library. (n.d.). Retrieved from https://www.tensorflow.org
- OpenAI. (2023). GPT-Based Language Models for NLP Tasks. Retrieved from https://openai.com/research
- AnyChart API. (n.d.). Data Visualization for Sentiment Analysis. Retrieved from https://www.anychart.com

Acknowledgment:

We express our gratitude to our project guide Prof. Vijayalakshmi Tadkal, Assistant Professor, Department of Computer Engineering AIML for her valuable suggestions, cooperation, and support in the working of this paper.