IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2025

DOI: 10.17148/IJIREEICE.2025.13518

Sentiment Analysis of Social Media Content

Roni Bhakta[1], Samarth Hatte[2], Tanishq Dasari[3], Vikas Budhyal[4], Ms. Aarti Valsang[5]

Student, CSE, A. G. Patil Institute of Technology, Solapur, India[1,2,3,4]

Assistant Professor, CSE, A. G. Patil Institute of Technology, Solapur, India[5]

Abstract: This document presents Sentiment Analysis, an AI-powered sentiment analysis system that leverages advanced large language models (LLMs) via the LangChain framework. The primary objective of this project is to analyse and classify textual input into sentiment categories such as Positive, Neutral, and Negative, with potential applications in content moderation, user feedback analysis, and social media monitoring. The system is built using a modular Python backend, integrated with LangChain to streamline prompt engineering and model interaction. By utilizing APIs from state-of-the-art language models (e.g., Gemini or GPT), SENTIMENT ANALYSIS delivers high-accuracy, context-aware sentiment classification. This paper describes the architectural components, implementation methodology, and output results of the SENTIMENT ANALYSIS framework. The proposed approach demonstrates the effectiveness of combining LLMs with LangChain's orchestration layer for building adaptable, intelligent sentiment analysis tools.

Sentiment Analysis is built around the idea of creating a communication platform that values thoughtful expression, emotional context, and user control. One of the core features we're working on is an intelligent flow for handling user posts, called Echoes. When a user writes an Echo, it goes through a validation process using Zod both on the frontend and at the Cloudflare middleware layer to ensure the data is clean and safe. Instead of immediately storing the Echo in the database, it's first passed through backend functions that handle specific logic as needed.

The Echo is then temporarily saved with a private flag and sent to an Azure-based API powered by FastAPI, where the data is validated again using Pydantic. From there, the Echo enters a custom sentiment analysis pipeline built using LangChain. This model, trained specifically for the kind of conversations expected on Sentiment Analysis, classifies the Echo as positive, neutral, or negative. If it's found to be positive, the system updates the Echo in the database, changes its visibility from private to public, and reflects the change on the user interface in real time. For Echoes that come back as neutral or negative, the system holds them in private, generates alternative phrasings using the LangChain suggestion engine, and saves those suggestions in the database for the user to review. The user can choose to rephrase and publish, or keep it as is. Throughout this process, LangMemo keeps track of context to ensure a smooth and consistent experience. This entire flow helps users communicate more thoughtfully, while giving them tools to refine their messages and maintain control over what they share. It's a step toward building a platform where expression feels safe, supported, and emotionally aware.

Keywords: Sentiment Analysis, LangChain, Large Language Models (LLMs), Text Classification, PromptEngineering, Artificial Intelligence

I. INTRODUCTION

This research paper is for introducing **Sentiment Analysis**, a modular sentiment analysis framework that utilizes large language models (LLMs) through the LangChain orchestration library to classify user input based on sentiment. In today's digital landscape, the ability to automatically detect and categorize sentiments from textual data has become increasingly important in areas such as content moderation, customer feedback analysis, and social media monitoring.

Sentiment Analysis aims to address the limitations of traditional sentiment analysis models by leveraging the power of modern LLMs like Gemini and GPT, integrated through LangChain to enable dynamic prompt engineering and structured output parsing. The system supports real-time processing and classification of textual data submitted by users, providing accurate sentiment insights with context awareness.

The backend of SENTIMENT ANALYSIS is built using Python and is deployed through a Cloudflare-based API layer. It integrates **PostgreSQL** for structured data storage and **Weaviate** as a vector database to manage semantic embeddings generated by LangChain, enabling efficient similarity searches and deeper contextual understanding.

This paper outlines the architecture, methodology, and practical implementation of Sentiment Analysis. It highlights the integration of LangChain and vector-based search mechanisms for enhanced AI-driven sentiment classification. Additional information about the system's deployment, evaluation, and applications is provided in the sections that follow.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2025

DOI: 10.17148/IJIREEICE.2025.13518

II. OBJECTIVES

To design and develop a scalable sentiment analysis framework that can process multimodal inputs such as text, images, and video thumbnails to detect user sentiment with high accuracy.

To integrate advanced language models through LangChain for enhanced contextual understanding and dynamic prompt handling, enabling more reliable sentiment classification.

To implement a hybrid data architecture combining vector-based semantic storage (Weaviate) and structured relational storage (PostgreSQL) for efficient information retrieval and sentiment mapping.

To ensure input validation and type safety using schema-driven tools such as Zod, thereby maintaining system integrity and minimizing erroneous data processing.

To address issues related to online harm and digital toxicity by classifying content into actionable categories such as Harmful, Neutral, or Good, with potential use in content moderation systems.

To evaluate the performance of the system through real-world test data and measure accuracy, latency, and system reliability in different use-case scenarios.

III. LITERATURE REVIEW

Sentiment analysis has evolved significantly over the last decade, transitioning from traditional rule-based approaches to deep learning and transformer-based models. Earlier methods relied heavily on lexicon-based techniques such as SentiWordNet and VADER, which offered simplicity but lacked the ability to capture contextual nuances in natural language.

With the advent of deep learning, models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) began to improve sentiment classification by learning features from large datasets. Despite their improvements, earlier models struggled to maintain coherent understanding across lengthy text inputs and often misinterpreted emotionally nuanced language.

The development of transformer architectures, such as BERT and GPT, represented a significant leap forward in natural language processing capabilities. Transformer models leverage self-attention mechanisms to better understand how words relate within a sentence, enabling them to detect sentiment with greater accuracy across diverse linguistic structures.

Advanced language models like OpenAI's GPT series and Google's Gemini have pushed the boundaries further by enabling zero-shot and few-shot reasoning, allowing them to handle diverse tasks with minimal training examples.

Parallel to advancements in modeling, vector databases like Weaviate and Pinecone have enabled efficient semantic search and retrieval using embeddings. It has proven useful in storing and comparing content based on meaning rather than surface form. Combining LLMs with vector search offers a hybrid solution for intelligent data analysis, enabling systems to understand both linguistic context and semantic similarity.

LangChain is one such framework that facilitates the coordination of large language model workflows by streamlining prompt management, linking outputs across multiple stages, and enabling integration with external tools and APIs. LangChain allows developers to build complex pipelines involving document parsing, reasoning, and interactive AI behavior, making it ideal for modular systems like Sentiment Analysis.

This project builds on recent advancements by combining them into a unified system that not only performs accurate sentiment classification but also suggests improved versions for potentially harmful content, thereby promoting healthier and more constructive online interactions.

IV. METHODOLOGY

In this project, we have implemented a primary working model for sentiment analysis, leveraging LangChain as a core tool in its appropriate context. The objective of this project is to analyze the social tendencies of users who are actively engaged on social media platforms. The nature and tone of comments made on posts serve as indicators of user intent and emotional state.

To avoid social dilemmas such as regional conflicts, hate speech, cyberbullying, and other forms of online harm, the system categorizes content into three sentiment classes: **Harmful**, **Neutral**, and **Good**. We use **LangChain** to interface with the **Gemini 2.0 Flash** model, which processes and interprets the input data. The data pipeline is built to handle text, image, and video inputs, ensuring multimodal coverage.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2025

DOI: 10.17148/IJIREEICE.2025.13518

User-generated content is first captured via a FastAPI backend and passed through preprocessing filters. The processed data is then embedded into a vector space using **Weaviate**, which serves as the semantic search and vector database engine. Alongside, structured metadata is stored in **PostgreSQL**, enabling efficient data retrieval and user-specific sentiment tracking.

Validation of inputs and schema consistency is maintained using **Zod**, a TypeScript-first schema declaration library. This ensures all user inputs including form data and media files meet the structural requirements before being sent for processing.

The modular architecture allows for scalability and parallel processing, supporting future extensions such as real-time analysis, integration with content moderation systems, or multilingual support. This hybrid model of combining traditional storage, vector embedding, and LLM-based interpretation allows for a robust and flexible sentiment analysis framework suited for modern online platforms.

The Sentiment Analysis works in three ways

1st It checks if the tweet includes any bad/negative Sentiment comments if yes then it will send a notification to user for that the message is negative for some reason and it will give it some time to change his comment or delete it.

2nd If the user does not modify or delete the post, it will be automatically removed

3rd If user want to modify it quickly it will suggest better alternatives to them.

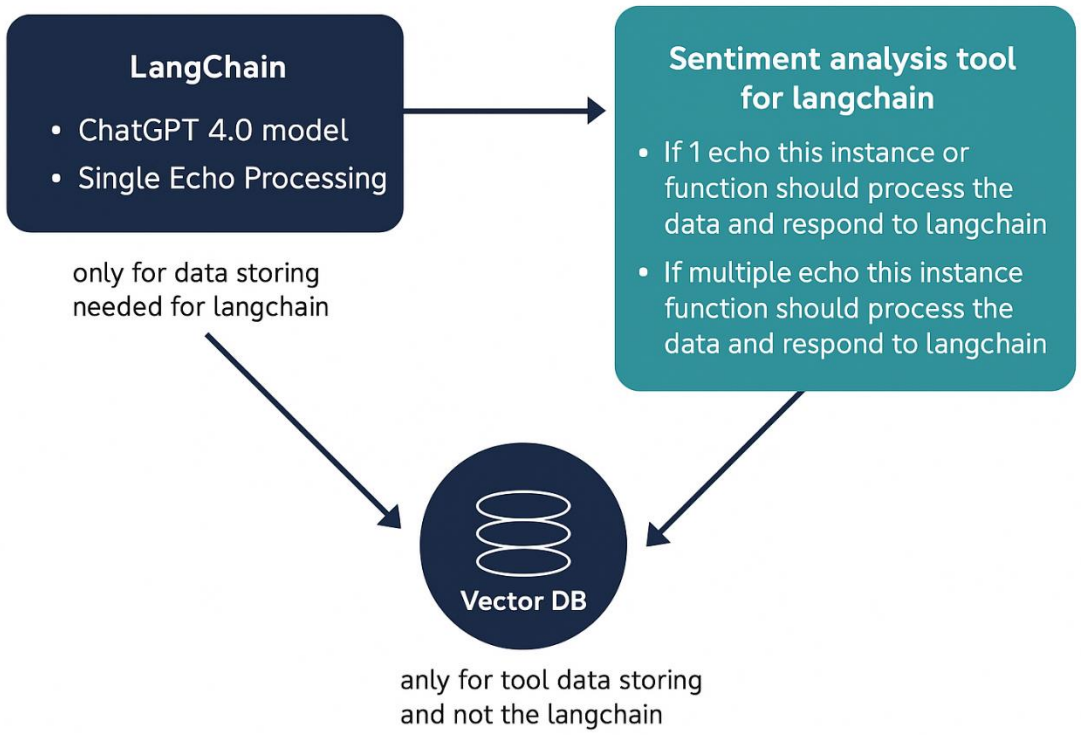

Fig. 1 Tool architecture for analysis tool

V. WORKFLOW FOR SENTIMENT ANALYSIS

User feed ECHO→ ZOD check → Cloudflare Middleware with ZOD Check→ functions calling to process the things as required logic → retrieve the echoed data through the function and not DB & just Put the ECHO into DB with ECHO "FLAGGED as Private" → push it to the AZURE ENDpoint (FastAPI & Pydantic for data validation (works like a zod)) → langchain pipeline (Sentiment Analysis tool (custom model, not LLM)) classified as Positive, Neutral, Negative → respond it to Cloudflare layer, LangMemo stores context, where it's waiting in the function to complete the PROMISE (If "Success" with "positive flag") → Update the echo & Save it in the DB using prisma→ update the user interface with the updated echo → make the ECHO "flag" from Private to Public in DB.

If Negative/Neutral→ LangMemo feeds the context to LangChain pipeline→ LangChain suggests reworded alternatives → Save suggestions to DB for user interaction→ Flag remains Private unless accepts suggestion

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2025
DOI: 10.17148/IJIREEICE.2025.13518

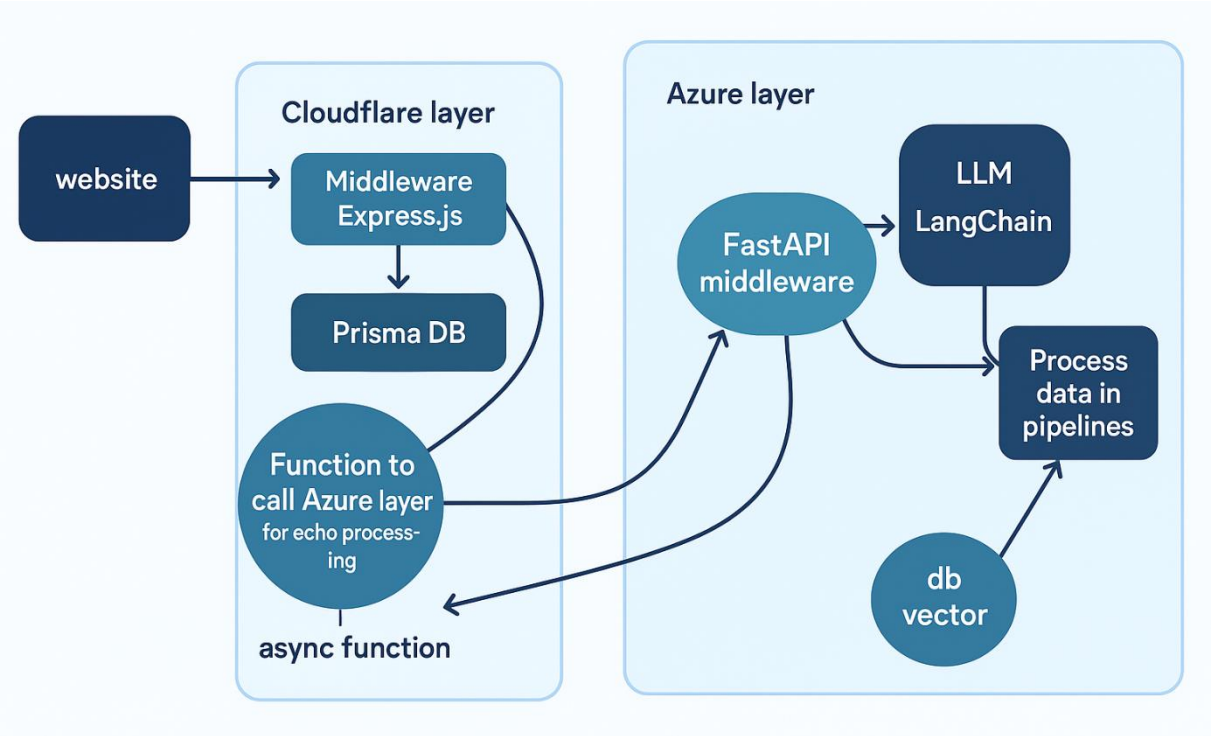

Fig. 2 Sentiment Analysis Work flow.

VI. CONCLUSION

The proposed Sentiment Analysis presents a scalable and modular approach to sentiment analysis using large language models and vector embeddings. By integrating technologies such as LangChain, Weaviate, PostgreSQL, and Gemini 2.0 Flash, the system effectively classifies multimodal user inputs as Harmful, Neutral, or Good. This implementation addresses the limitations of traditional lexicon-based sentiment tools by offering contextual and semantic understanding. The architecture also supports real-time processing and type-safe validation, making it suitable for applications in content moderation, feedback analysis, and online safety. Future enhancements may include multilingual support, GPU acceleration, and integration with real-time data streams to improve performance and applicability. The version of this template is V2. Most of the formatting instructions in this document have been adapted to follow IEEE conference paper standards using Microsoft Word format..

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the open-source communities and developers of LangChain, Weaviate, Gemini, and FastAPI for providing robust tools that significantly contributed to the success of this project. We also acknowledge the guidance and support from our faculty members, whose feedback helped shape the direction and quality of the research. Special thanks to Google Cloud and OpenAI for access to advanced APIs and documentation that enabled us to build and evaluate our model effectively

REFERENCES

[1] H. Thoppilan et al., “LaMDA: Language Models for Dialog Applications,” arXiv preprint arXiv:2201.08239, 2022.
[2] H. S. Park, A. Tamkin, and P. Liang, “LangChain: A Framework for Composable LLM Applications,” LangChain Documentation, Accessed: May 2025. [Online]. Available: https://docs.langchain.com/
[3] Weaviate B.V., “Weaviate: An Open Source Vector Search Engine,” [Online]. Available: https://weaviate.io/, Accessed: May 2025.
[4] Google AI, “Gemini: Multimodal Capabilities in Large Language Models,” Google Research Blog, Dec. 2023. [Online]. Available: https://ai.googleblog.com/
[5] Zod Documentation, “TypeScript-first schema validation with static type inference,” [Online]. Available: https://zod.dev/, Accessed: May 2025.

DOI: 10.17148/IJIREEICE.2025.13518

[6] K. Goyal et al., "Sentiment Analysis of Social Media Data Using Deep Learning," IEEE Access, vol. 9, pp. 123456-123467, 2021.
[7] T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," Proc. EMNLP 2020: System Demonstrations, pp. 38–45, 2020.
[8] S. Ruder, "An Overview of Multi-Task Learning in Deep Neural Networks," arXiv preprint arXiv:1706.05098, 2017.

BIOGRAPHY

Roni Bhakta is a full-stack MERN developer passionate about building innovative and impactful solutions. He leads the Google Developer Group (GDG) on Campus, CodeClub AGPIT, and the E-Cell at A. G. Patil Institute of Technology, actively fostering student innovation and entrepreneurship. He has showcased an AI-powered ERP solution at Startup Mahakumbh and contributes to the startup ecosystem as a Campus Ambassador for IIT Bombay E-Cell. In 2025, he successfully completed Google Summer of Code (GSoC) with the Internet Archive. His technical expertise spans backend development (Python, Node.js), frontend frameworks (React, Next.js), and DevOps tools (Docker, AWS). Roni is driven by a vision to collaborate and develop scalable technologies that create meaningful impact.

Samarth Hatte is currently pursuing a Bachelor's degree in Computer Science and Engineering. He has a keen interest in Artificial Intelligence, Natural Language Processing, and Human-Centered Computing. His recent work focuses on building AI-powered systems that ensure online safety and ethical content moderation using large language models and vector-based retrieval systems. Samarth has also participated in various national-level hackathons and academic projects related to Android development and machine learning. He aims to contribute towards scalable, secure, and human-aligned AI applications.

Vikas Budhyal is currently pursuing a Bachelor's degree in Computer Science and Engineering from Dr. Babasaheb Ambedkar Technological University (DBATU). He is passionate about web development, ethical technology, and AI-driven digital platforms. His recent work centers around the Sentiment Analysis project—an AI-enhanced microblogging system that leverages large language models and real-time content moderation to promote safe and personalized user experiences. As the secretary of his college's code club, Vikas has led multiple technical events and participated in national-level hackathons. His goal is to build secure, inclusive, and impactful web technologies that align with human values and digital well-being.

Tanishq Dasari is a third-year Computer Science undergraduate at A. G. Patil Institute of Technology, Solapur, with a strong focus on Full-Stack Python Development, Artificial Intelligence, Machine Learning, and System Design. He plays a key role in the Code Club at AGPIT and is currently leading the backend and architecture development of the SENTIMENT ANALYSIS project. His work integrates FastAPI, LangChain, Sentiment Analysis, Weaviate, and Azure OpenAI to build intelligent, scalable systems. His technical interests lie in vector databases, LLM integration, and designing end-to-end AI pipelines for real-world deployment.

Ms. Aarti Valsang is currently serving as an Assistant Professor in the Department of Computer Science and Engineering at A. G. Patil Institute of Technology, Solapur, India. She has presented over five research papers at international conferences and four at the national level, reflecting her active engagement in academic research. Ms. Valsang has participated in more than 20 workshops aimed at enhancing pedagogical and technical competencies. She is a lifetime member of the Indian Society for Technical Education (ISTE) and contributes to institutional development as a member of the Internal Quality Assurance Cell (IQAC) at AGPIT. Her areas of interest include artificial intelligence, data analytics, and engineering education practices.