# Real-Time Integrated Social Media Feedback Analysis System

**Dr. Ch. Vijay Kumar,** Professor & HOD Department of Computer Science and Engineering
ACE Engineering College, Ankushapur, Hyderabad

Shaik Basheer
Department of Computer Science and Engineering
ACE Engineering College
Hyderabad, 501301, India
shaikbasheer3364@gmail.com

Viswanath Akshay
Department of Computer Science and Engineering
ACE Engineering College
Hyderabad, 501301, India
mail2akshay14@gmail.com

Veerala Sathwik Kumar
Department of Computer Science and Engineering
ACE Engineering College
Hyderabad, 501301, India
vsathwikkumar29@gmail.com

**Abstract**: This research presents a Real-time Integrated Social Media Feedback Analysis System that extracts, analyzes, and summarizes user opinions from Reddit and YouTube. With rapid growth of user-generated content on these platforms, understanding public sentiment supports decision-making in marketing, product development, and social analysis. The system collects data through official and public APIs, processes unstructured text, and applies the VADER (Valence Aware Dictionary and sentiment Reasoner) lexicon model for positive, neutral, and negative classification—well suited to informal social text. For interpretability, the system integrates hosted large language models via the Groq API to produce concise summaries of collected feedback. The architecture uses a React frontend and Flask backend for interactive dashboards (Chart.js visualizations), scalable separation of concerns, and deployable REST endpoints. Query relevance filtering, duplicate removal, and HTML entity normalization improve the quality of retrieved posts and comments. Under typical API conditions, end-to-end analysis completes in approximately 3–8 seconds for representative queries. Limitations include sarcasm and irony, API rate limits, English-oriented filtering, and sampling bias across platforms. The work demonstrates a practical intelligent systems pipeline from ingestion to visualization.

***Keywords:*** *Social media analysis; sentiment analysis; VADER; Groq; Reddit; YouTube; Flask; React; NLP; real-time analytics.*

## 1. Introduction

The expansion of social media has produced large volumes of unstructured opinions on products, services, and events. Reddit and YouTube are significant sources of such text; manual review does not scale. Sentiment analysis (opinion mining) classifies text into categories such as positive, negative, and neutral and supports monitoring of public perception.

Traditional batch-oriented or single-platform tools often fail to combine live retrieval, robust sentiment, and human-readable summaries in one workflow. This paper describes a Real-Time Integrated Social Media Feedback Analysis System that:

The proposed system is designed as an interactive pipeline that enables users to analyze social media feedback in near real time. It begins by accepting a keyword or topic from the user, which serves as the basis for data collection. The system then fetches relevant posts from Reddit and comments from YouTube using their respective APIs. Once the data is collected, it undergoes preprocessing steps such as cleaning, normalization, and relevance filtering to ensure consistency and accuracy. The processed text is then analyzed using the VADER sentiment analysis model to classify it into positive, negative, or neutral categories, with an optional enhancement through Groq-based large language models for improved sentiment understanding. Following sentiment classification, the system generates an abstractive summary using Groq, providing a concise and human-readable overview of the collected feedback. Finally, the results are presented through an interactive React-based dashboard supported by a Flask backend, where users can view sentiment distribution charts,

platform-wise comparisons, and filterable lists of posts and comments.

The system is optimized for near real-time performance, delivering results within a few seconds rather than relying on continuous stream processing. However, it faces several challenges, including handling sarcasm and mixed sentiments in user-generated content, managing API rate limits, and ensuring stable connectivity during data retrieval and processing. The structure of the paper is organized such that Section 2 reviews the related work in sentiment analysis and social media analytics, Section 3 describes the proposed methodology and system design, Section 4 presents the results and discussion based on experimental and qualitative analysis, and Section 5 concludes the study while suggesting directions for future work.

The structure of the paper is organized to clearly present the research and implementation details. The section on related work reviews existing approaches in sentiment analysis and social media mining, highlighting their strengths and limitations. The

methodology and system design section explains the architecture, data flow, and technologies used in building the system. The results and discussion section provides both experimental findings and qualitative observations based on system performance. Finally, the conclusion summarizes the contributions of the work and outlines potential directions for future improvements, such as incorporating multilingual support, expanding to additional platforms, and enhancing model accuracy through the advanced machine learning models. .

## 2. Related works

Sentiment analysis has evolved from classical machine learning (e.g., Naïve Bayes, SVM) with heavy feature engineering to lexicon-based and deep learning approaches [1][3]. VADER is a lightweight, rule-augmented lexicon method designed for social media short text, emoticons, and informal language [1] and is widely used when interpretability and low latency matter.

Summarization of large text collections supports decision-makers; large language models (LLMs) can produce concise narratives, though grounding in source text is required to limit fabrication [9][22]. Commercial and research systems differ in platform coverage, bias, and reproducibility. Reddit and YouTube expose APIs for programmatic access; public fallbacks exist where credentials are unavailable [4][5]. Works on YouTube and Reddit highlight noise, thread structure, and rate limits [14][16][17].

Visualization and dashboards improve usability of analytics [11]. Web stacks combining SPA frontends (e.g., React [12]) and Python web frameworks (e.g., Flask [19]) are standard for deployable prototypes.

Google Gemini and other multimodal LLM families represent prominent general-purpose models [6]; this implementation uses Groq-hosted inference for summarization and optional sentiment to match the deployed system. Prior surveys cover LLM-based summarization challenges [9] and social opinion mining at scale [16][17][18].

Many descriptions remain single-platform or omit end-to-end deployment details. This work contributes a documented, modular pipeline: multi-source collection, VADER + optional LLM sentiment, Groq summarization, relevance and deduplication, and a deployable React– Flask architecture.

## 3. Proposed Methodology

### 3.1. System overview

The system follows a three-layer architecture:

- Presentation layer: React SPA—search, charts (sentiment distribution, platform comparison), summary card, keyword filters, platform-specific post/comment lists.

- Application layer: Flask REST API—`GET /analyze?query=` orchestrates collection and NLP; `GET /health` reports connectivity and key configuration.

- Data & intelligence layer: Module for multi-platform collection; module for sentiment, keywords, and summary (Groq).


**Figure 1.** system overview

### 3.2. Data acquisition

Reddit: Search across all subreddits using PRAW when client credentials are configured; otherwise a public JSON search path may be used. Parameters favor relevance and a recency window (e.g., month-scale). Title and self-text are concatenated for each submission; length may be capped.

YouTube: The YouTube Data API retrieves videos for the query and top-level comments for selected videos; duration and relevance heuristics reduce low-information shorts.

Language: Optional English detection filters non-English items for consistent lexicon behavior.

### 3.3. Preprocessing, relevance, and deduplication

- HTML entity decoding (e.g., `&#39;` → apostrophe) after stripping markup where applicable.
- Duplicate suppression using normalized text keys to avoid repeated lines in statistics and summaries.
- Query relevance: For multi-word queries, phrase-oriented search and post-hoc checks require the full query phrase (case-insensitive, normalized spaces) in the combined title and body where applicable, reducing off-topic hits from loose token matching (e.g., isolated numbers).

### 3.4. Sentiment analysis

VADER assigns compound score \(c \in [-1,1]\) and component scores. Labels use thresholds \(\tau^+\) and \(\tau^-\):

Worked Example: "poco f7 is great phone"

**Step 1: Lexicon Lookup**
VADER checks each word against its pre-defined lexicon. Most words in this sentence are "neutral" (carrying 0 valence), while "great" is a high-valence word.

| WORD | VALENCE SCORE |
|---|---|
| Poco | 0 |
| F7 | 0 |
| Is | 0 |
| Great | 3.1 |
| phone | 0 |

**Total Sum (sum_valence) = 3.1**

**Step 2: Apply the Normalization Formula**

We now plug the sum into the VADER normalization formula. We use the standard smoothing constant alpha = 15.

$$x = \frac{sum\_valence}{\sqrt{sum\_valence^2 + \alpha}}$$

Substitute the values:

$$x = \frac{3.1}{\sqrt{3.1^2 + 15}}$$

$$x = \frac{3.1}{\sqrt{9.61 + 15}}$$

$$x = \frac{3.1}{\sqrt{24.61}}$$

$$x = \frac{3.1}{4.9608467}$$

$$x \approx 0.6249$$

**Final Compound Score: 0.6249**

**Step 3: Classification against Thresholds**

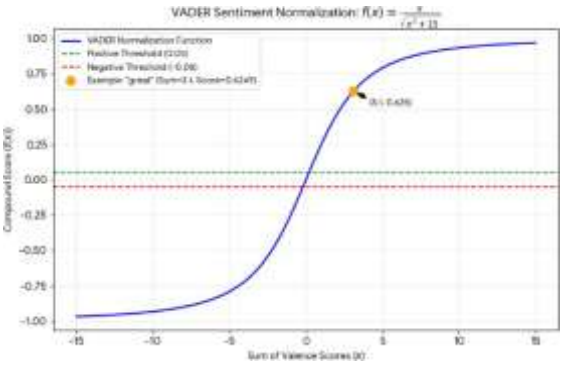
Now, we compare the calculated score to the established thresholds:

- **Positive:** ≥ 0.05
- **Neutral:** between -0.05 and 0.05
- **Negative:** ≤ -0.05

Since **0.6249 > 0.05**, the sentence is mathematically classified as **Positive**.

**Visualization of the Normalization Function**

The graph below illustrates how the compound score behaves. You can see your example sentence (the orange dot) sits comfortably in the "Positive" region. Note how the curve flattens as it approaches 1.0, which is the "asymptotic" property that prevents scores from ever exceeding the [-1, 1] range.



Optional LLM sentiment: Batches of texts are classified via Groq with a structured (e.g., JSON) response; labels are mapped to positive / neutral / negative. On failure, the system falls back to VADER-only labels.

### 3.5. Keyword extraction

A frequency-based extractor removes stop words and short tokens; top terms are aggregated across platforms for global keywords and per-platform highlights, supporting interactive filtering in the UI.

### 3.6 Summarization via Groq

Collected items are formatted as lines with sentiment tags (e.g., `[positive] text`). A Groq chat completion model generates a short summary (e.g., 150–200 words) in plain language, instructed to use only the provided content. This yields qualitative context alongside quantitative sentiment counts.

### 3.7 Frontend–backend integration and deployment

The frontend calls the backend with the user query. CORS is configured via environment variables (e.g., allowed origins for production hosts). For cloud deployment (e.g., Render), the frontend build sets `REACT_APP_API_BASE` to the HTTPS API URL **at build time. Figure 2 shows the operational dashboard.

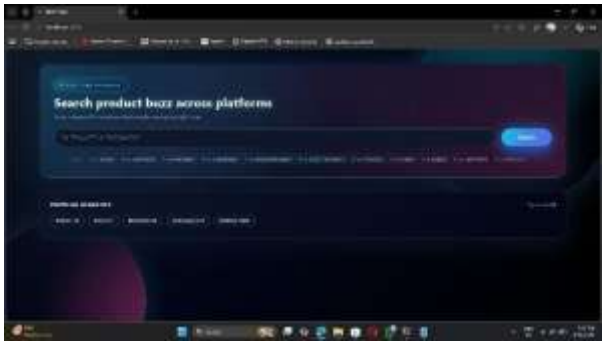



**Figure 2.** System operational Dashboard

## 4. Results and discussions

### 4.1. Functional workflow

The user enters a keyword (e.g., “MacBook Air”, “Poco F7”) on the React dashboard. The client sends a request to Flask, which concurrently obtains Reddit and YouTube text. VADER (and optionally Groq) assigns polarity per item; counts feed doughnut and bar charts. Groq returns a narrative summary displayed in the analysis summary panel. Keyword tags support drill-down views.


**Figure 3.** summary analysis

The image shows a web application interface designed for searching product buzz across various platforms. The interface has a dark-themed design with a search bar prominently displayed at the top, where the user has entered "MacBook Air." Below the search bar, there are category tabs such as "Phones," "Stickers," "Cars," "Bikes," "Laptops," "Phones," "Cameras," "Headphones," "Electronics," and more, allowing users to filter their search by product type.

The main section of the interface displays an "Analysis Summary" for the searched product "MacBook Air." The summary provides insights about the product's popularity and user opinions. It mentions that many people are happy with the MacBook Air, highlighting its lightweight design, good battery life, and performance. It also compares the MacBook Air to the MacBook Pro, noting that while the Pro is more powerful, the Air is a good choice for everyday use and portability.

Overall, the image depicts a tool aimed at aggregating and summarizing consumer feedback and buzz about various products to help users make informed decisions.


**Figure 3.** Detailed Analysis

The image shows a detailed analysis dashboard for product sentiment analysis. Key elements include:

1. **Total Items:** The total number of items analyzed is 100.
2. **Sentiment Breakdown:**
   - Positive sentiments dominate with 82 items (82%).
   - Neutral sentiments account for 9 items (9%).
   - Negative sentiments also account for 9 items (9%).
3. **Sentiment Distribution:** A colorful donut chart visually represents the proportions of positive (green), neutral (yellow), and negative (red) sentiments.
4. **Platform Comparison:** A bar chart compares sentiment across different platforms, such as Reddit and Twitter. Reddit shows a higher number of positive sentiments compared to Twitter, which has fewer positive and some negative sentiments.

Overall, this dashboard provides a clear visual summary of how a product or topic is perceived across different platforms, highlighting the predominance of positive feedback.

**Figure 4.** Platform Breakdown & Keywords

The image shows the continuation of a product sentiment analysis dashboard with a focus on platform-specific breakdowns and keyword insights:

1. **Platform Breakdown:**
   - **Reddit:** Displays sentiment icons indicating positive, neutral, and negative sentiments. It also lists top keywords associated with Reddit discussions, such as "macbook," "air," "laptop," "fans," and "apple."
   - **YouTube:** Similarly shows sentiment icons and top keywords like "macbook," "pro," "battery," "energy," "mac," and "screen."
2. **Top Keywords Overall:** A horizontal list of popular keywords related to the product or topic analyzed, including terms like "macbook," "air," "pro," "apple," "laptop," "battery," "screen," "fans," and others.

This section provides insights into where conversations are happening and what specific terms are most frequently mentioned, helping to understand the context and focus of user discussions across platforms.

### 4.2. Functional workflow

**Table 1. Illustrative end-to-end latency**

| Query (example) | Reddit items (approx.) | YouTube comments (approx.) | Response time (s) |
|---|---|---|---|
| **MacBook Air** | 50 | 50 | 4-8 |
| **POCO F7** | 48 | 50 | 3-8 |
| **Iphone 16** | 49 | 49 | 4-7 |

Discussion: Latency depends on API round-trips, Groq calls, cold start on free hosting, and result limits. Rate limits may throttle or fail requests; retry and caching are future improvements.

### 4.3. Qualitative observations

**Table 2.** ***Sentiment distribution example for one query***

| Class | Count | Percentage |
|---|---|---|
| **Positive** | 82 | 82 |
| **Neutral** | 9 | 9 |
| **Negative** | 9 | 9 |

Relevance filtering reduces irrelevant subreddit hits compared with unfiltered search. Sarcasm and irony remain misclassified in some cases—consistent with known NLP limits [3][8].

### 4.4. Limitations

- Platform bias (who posts on Reddit vs. who comments on YouTube).
- English focus if language filtering is enabled.
- LLM summaries may over-generalize if prompts are not strict—mitigated by source-only instructions.
- Ethics and ToS: Public data use must respect terms and privacy.

### 4.5. Conclusion and future scope

This paper presented a real-time integrated social media feedback system combining Reddit and YouTube ingestion, VADER-based sentiment with optional Groq refinement, Groq-based summarization, and a React–Flask dashboard with visual analytics. The design emphasizes modularity, deployability, and interpretable lexicon scores alongside LLM narrative. Measured interactive latencies on the order of seconds are achievable under normal API usage.

Future work includes: multilingual models; Twitter/X or other sources; streaming or scheduled collection; user studies (SUS, task time); formal evaluation against manual labels; bias and fairness analysis; and on-premise models for stricter data governance.

**References**

[1] C. J. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," *Proc. Int. AAAI Conf. Web Social Media*, Ann Arbor, MI, USA, 2014, pp. 216–225.

[2] B. Liu, *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*, 2nd ed. Cambridge, UK: Cambridge University Press, 2020.

[3] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Egyptian Informatics Journal*, vol. 15, pp. 109–123, 2014.

[4] Reddit, Inc., Reddit API documentation. [Online]. Available: https://www.reddit.com/dev/api/

[5] Google, YouTube Data API v3. [Online]. Available: https://developers.google.com/youtube/v3

[6] Google DeepMind, "Gemini: A family of highly capable multimodal models," *arXiv*:2312.11805, 2023. *(Related work; summarization in this study uses Groq-hosted LLMs.)*

[7] A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[8] J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019.

[9] H. Zhang et al., "A review of LLM-based text summarization: Methods, applications and challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 36, pp. 1242–1260, 2024.

[10] R. Batool et al., "Precise real-time opinion mining of social media," in *Proc. Int. Conf. Cloud & Ubiquitous Computing & Emerging Technologies*, Pune, India, 2013.

[11] A. Agarwal et al., "Sentiment analysis of Twitter data," in *ACL Workshop on Languages in Social Media*, 2011.

[12] A. Banks and E. Porcello, *Learning React*, 2nd ed. Sebastopol, CA, USA: O'Reilly, 2020.

[13] R. Jain and A. Sharma, "Real-time social media analytics framework for sentiment tracking," *Int. J. Comput. Sci. Inf. Secur.*, vol. 20, pp. 45–58, 2022.

[14] S. S. Patil et al., "Review of sentiment analysis in social media using big data," *Int. J. Comput. Appl.*, vol. 175, pp. 17–24, 2020.

[15] M. Z. Asghar et al., "Sentiment analysis on YouTube: A brief survey," *arXiv*:1511.09142, 2015.

[16] K. Cortis and B. Davis, "Over a decade of social opinion mining: A systematic review," *Artif. Intell. Rev.*, vol. 54, pp. 4873–4965, 2021.

[17] M. Rodríguez-Ibánez et al., "A review on sentiment analysis from social media platforms," *Expert Syst. Appl.*, vol. 223, 119861, 2023.