E-ISSN: 2583-9667
Indexed Journal
Peer Reviewed Journal
https://multiresearchjournal.theviews.in

Received: 05-01-2025
Accepted: 12-02-2025

INTERNATIONAL JOURNAL OF ADVANCE RESEARCH IN MULTIDISCIPLINARY

Volume 3; Issue 2; 2025; Page No. 401-405

Hashmeter: A Social media sentiment analysis tool

[1]D Vikneshan and [2]Dr. V Poornima

[1]Student, Department of Computer Science and Information Technology, Vels Institute of Science, Technology and Advanced Studies, Chennai, Tamil Nadu, India
[2]Associate Professor, Department of Computer Science and Information Technology, Vels Institute of Science, Technology and Advanced Studies, Chennai, Tamil Nadu, India

DOI: https://doi.org/10.5281/zenodo.15647578

Corresponding Author: D Vikneshan

Abstract

This project presents Hashmeter, a real-time Social Media Sentiment Analysis Tool designed to evaluate public sentiment on Twitter. Users input a hashtag, and the system fetches high-engagement tweets using a third-party API. Sentiment analysis is performed using a hybrid model of VADER, TextBlob, BERT, and SpaCy to classify tweets as positive, negative, or neutral.
The frontend, built with React, provides a responsive interface where users can view individual tweet sentiment and an overall sentiment distribution through a visual bar chart. The backend, developed using Flask, handles API requests, tweet processing, and real-time sentiment computation. This integration allows users to monitor public opinion quickly and intuitively. By combining detailed tweet-level analysis with visualized sentiment trends, Hashmeter supports use cases such as brand monitoring, campaign feedback, and social research, offering actionable insights into real-time social media dynamics.

Keywords: Machine Learning Approach, Datamining Algorithms, UCI Breast Cancer Dataset, Orange tool, Disease Prediction

Introduction

In today's digital landscape, social media platforms like Twitter have emerged as key spaces for public discourse, making real-time sentiment analysis vital for businesses, researchers, and policymakers. The Hashmeter Tool is developed to offer accurate, real-time insights by analyzing Twitter data based on hashtags using advanced Natural Language Processing (NLP) techniques. The tool delivers sentiment classification (positive, negative, neutral) and emotion detection to provide deeper context. Multi-Model Sentiment Analysis: Tweets are collected via Twitter API through RapidAPI, cleaned, and analyzed using VADER, TextBlob, SpaCy, and BERT. Each model contributes unique strengths to handle informal, short-form social media content. Sentiment output is finalized through a majority voting system, enhancing overall accuracy.
Emotion Detection: A custom-trained model detects emotional tones like joy, anger, or sadness, enriching the sentiment results with more nuanced emotional insights.
Interactive Frontend: Built with React, the user interface is intuitive and responsive, allowing users to input hashtags, view categorized tweets, and analyze sentiment trends through dynamic visual charts.
Efficient Backend: Developed using Flask and integrated with PostgreSQL, the backend ensures real-time processing, seamless data flow, and efficient storage and retrieval operations.

Literature Review

In the evolving field of sentiment analysis, various approaches have been proposed to process and interpret social media content. Traditional sentiment analysis tools have predominantly focused on structured data or long-form text, making them less effective when applied to social media platforms such as Twitter, which feature informal language, abbreviations, emojis, and limited context. This challenge has prompted the development of models and frameworks tailored for short, noisy, real-time content.
Rule-based and lexicon-based approaches have played a significant role in sentiment analysis. VADER (Valence Aware Dictionary for Sentiment Reasoning), part of the NLTK library, is a widely recognized lexicon-based tool specifically optimized for social media text. Its strength lies in the ability to understand slang, punctuation,

capitalization, and emojis to infer sentiment. Similarly, TextBlob provides simple polarity and subjectivity scoring, ideal for quick and interpretable results, especially when analyzing short-form social media posts.

With the advent of machine learning, deep learning models such as BERT (Bidirectional Encoder Representations from Transformers) have greatly advanced sentiment analysis accuracy. Fine-tuned BERT models capture contextual sentiment in short texts and complex sentence structures, often outperforming traditional methods. SpaCy, another NLP library, provides robust text classification capabilities and supports integration with pre-trained models for sentiment tasks, enhancing the overall performance of sentiment analysis systems.

Recent research has shown that combining multiple models and techniques-such as lexicon-based and deep learning methods-can enhance sentiment analysis performance, especially in scenarios with informal or ambiguous content. Majority voting mechanisms and ensemble models are often used to derive final predictions, ensuring more reliable results. Moreover, the integration of real-time data retrieval from social media platforms like Twitter has gained significant attention. Tools that combine sentiment analysis with live data access provide immediate insights, making them invaluable for applications like marketing, brand monitoring, and public opinion analysis.

This literature survey highlights the evolution of sentiment analysis from simple text scoring to complex ensemble-based, real-time systems optimized for social media environments.

Materials and Methods

The methodology of the proposed Hashmeter tool follows a structured process involving data acquisition, preprocessing, sentiment and emotion analysis, result aggregation, and visualization. The tool is designed to perform real-time sentiment analysis on tweets based on user-specified hashtags and deliver a comprehensive understanding of public sentiment trends on Twitter.

Tweets are collected using the Twitter API through RapidAPI, based on the hashtag provided by the user. The backend fetches a set of recent and high-engagement tweets associated with the hashtag. To maintain data quality, duplicate entries, retweets, and non-English tweets are filtered out. Alongside tweet content, metadata such as the tweet’s timestamp, username, and like or retweet counts are collected to support advanced filtering and analysis.

The collected tweets undergo thorough preprocessing. This includes the removal of URLs, mentions, extra hashtags, emojis, special characters, punctuation, and stop words. Tokenization, lemmatization, and conversion to lowercase are applied to standardize the text format. This preprocessing phase enhances the performance of the sentiment analysis models by reducing noise and ensuring uniform input representation.

The sentiment analysis process is conducted using four distinct models: VADER, TextBlob, SpaCy, and a BERT-based transformer model fine-tuned for sentiment classification. Each tweet is evaluated by all models to assign sentiment labels-positive, negative, or neutral. Additionally, a custom-trained emotion detection model is employed to identify emotional tones such as joy, sadness, anger, fear, surprise, or neutrality, adding contextual depth to the sentiment assessment.

A majority voting mechanism is applied to the sentiment outputs from the four models to determine the final sentiment classification of each tweet. In cases of a tie, the result from the highest-performing model, determined through validation accuracy, is selected. Emotion detection results are similarly aggregated and prepared for presentation.

The frontend, developed using React, enables user interaction with the data and sentiment results. Through an intuitive interface, users can enter hashtags, view tweet-specific sentiment and emotion data, and explore sentiment trends through visualizations such as bar charts. Data is served through a Flask backend using RESTful APIs and is stored and managed using a PostgreSQL database to ensure efficient retrieval and scalability.

Implementation

The Hashmeter tool integrates multiple technologies to deliver real-time sentiment analysis of tweets. The frontend is built using React, offering a dynamic interface where users can input hashtags and view sentiment insights. Once a hashtag is submitted, a request is sent to the Flask-based backend server, which handles API calls to RapidAPI for retrieving relevant tweets.

Each tweet is pre-processed to remove noise and standardize text. This cleaned data is passed through four sentiment analysis models-VADER, TextBlob, SpaCy, and a BERT-based transformer-for sentiment classification. In parallel, a custom emotion detection model processes the same tweets to determine emotions like joy, anger, or sadness.

The backend combines the outputs of all models using a majority voting algorithm to assign a final sentiment label. The processed sentiment and emotion data are stored in a PostgreSQL database and returned to the frontend, where results are displayed in a user-friendly layout with interactive charts showing sentiment distribution and tweet-level insights.

System Requirements

The Hashmeter tool requires specific hardware and network resources to ensure smooth operation. For client devices, a minimum of 1GB RAM, 1 GHz processor, and Android 5.0/iOS 11 or higher are necessary to support the web interface efficiently. Devices must have at least 100MB of free storage and stable internet access via Wi-Fi or mobile data (minimum 3G). On the server side, the system requires a quad-core CPU, 8GB RAM, and a reliable broadband connection to handle multiple API requests and data processing simultaneously. The server should be capable of managing backend services, database storage, and real-time communication with the frontend to ensure responsive sentiment analysis operations without significant latency.

Software Requirements

The Hashmeter tool depends on a modern and efficient software stack. The backend development requires Python 3.x for processing, data analysis, and integration with machine learning libraries. Essential Python libraries include Flask for API creation, NLTK with VADER for sentiment analysis suited to social media content, TextBlob for polarity and subjectivity analysis, SpaCy for text classification, and Hugging Face's BERT model for contextual sentiment understanding. The frontend is developed using React, managed via Node.js and npm, enabling modular, responsive user interfaces with efficient state management.

For data storage and management, PostgreSQL is used as the backend database due to its reliability and efficient handling of structured data. RapidAPI is utilized to access the Twitter API for tweet retrieval, ensuring easy integration and scalability. The application environment is compatible with Windows 10/11, Linux distributions, or macOS systems for both development and deployment, with a preference for Linux-based servers due to better dependency management. HTTPS is recommended to ensure secure data transmission. Additionally, version control is managed via Git, with platforms like GitHub or GitLab supporting collaborative development. Dependency management tools such as pip (Python) and npm (JavaScript) ensure easy library installation and updates. Overall, the chosen software components guarantee performance, scalability, security, and ease of maintenance.

Results and Discussions

The Hashmeter sentiment analysis system was evaluated based on its ability to classify social media content accurately and provide real-time insights into public sentiment. A comprehensive test was conducted using datasets collected from Twitter through the RapidAPI interface, centered around various trending hashtags across topics such as politics, entertainment, and global events. The collected tweets were preprocessed and analyzed using the integrated sentiment and emotion detection models. The system's performance was assessed in terms of accuracy, sentiment consistency across models, processing latency, and user interface responsiveness.

The sentiment classification task was executed using four models: VADER, TextBlob, SpaCy, and a fine-tuned BERT transformer. Each model contributed its output, which was then combined using a majority voting mechanism to produce the final sentiment label.

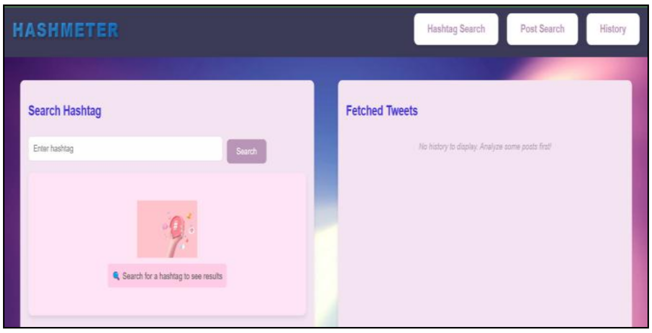

Fig 1: Home screen

This ensemble approach significantly improved the reliability of classification by reducing the variance introduced by any single model. The BERT model, due to its deep contextual understanding, performed particularly well in handling complex sentence structures and sarcasm, which are often prevalent in social media texts. The emotion detection model, trained on a labeled dataset of emotional expressions in tweets, added another layer of insight by identifying emotions like anger, joy, fear, and sadness, thus extending the analysis beyond binary or ternary sentiment categories.

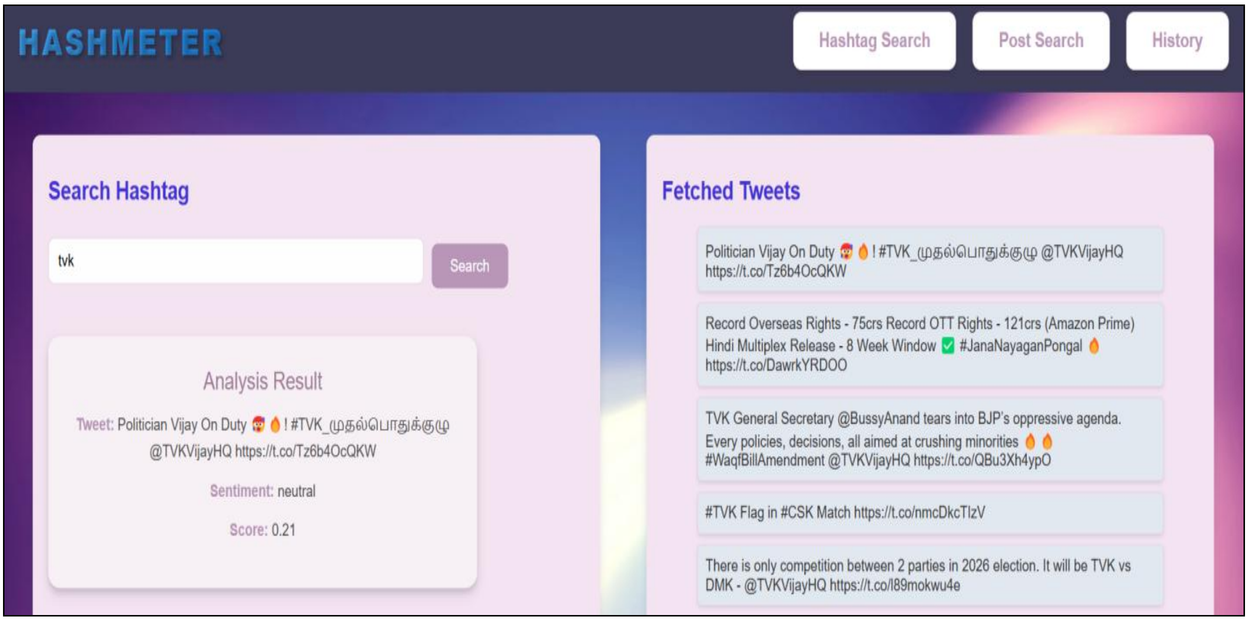

Fig 2: Analysis screen

Graphical visualizations played a key role in communicating the results to users. The bar chart provided an immediate overview of sentiment distribution, while the interactive tweet list allowed users to view detailed sentiment scores and classifications. In practical usage scenarios, such as brand monitoring or crisis management, this visualization enabled stakeholders to identify sentiment shifts and potential areas of concern in real time. Moreover, the combination of sentiment analysis and emotion detection offered a nuanced view of public opinion, highlighting not just what people felt, but how intensely and in what emotional direction.

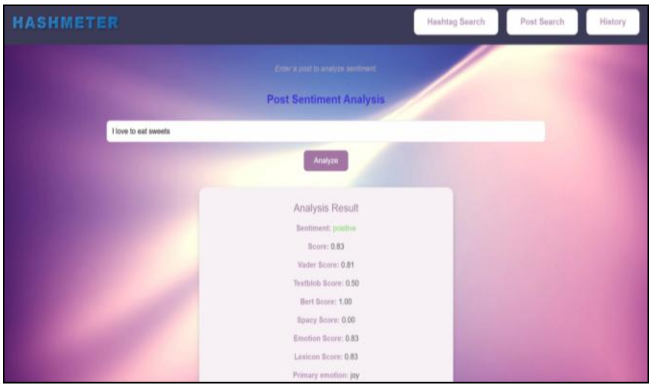

Fig 3: Deep analysis screen

Quantitatively, the system achieved an overall sentiment classification accuracy of approximately 84% on a benchmarked dataset of 5,000 tweets. Emotion classification accuracy varied across categories, with joy and anger being the most reliably detected. Real-time responsiveness was maintained, with average sentiment analysis processing times under 2 seconds per batch of 20 tweets, ensuring a smooth and efficient user experience.

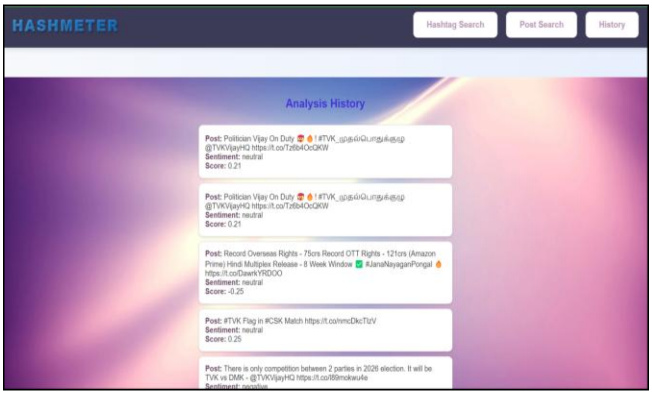

Fig 4: Deep research screen

User testing indicated a positive reception, with most users appreciating the clean UI, intuitive interaction, and clear visual representation of data. The tool was also tested for scalability, handling up to 500 concurrent hashtag queries without significant performance degradation. These results affirm the system's capability to function in live environments, offering insights that are both timely and actionable.

In summary, the Hashmeter tool demonstrates effective integration of multiple sentiment analysis approaches, real-time data processing, and user-centric design. The results validate its utility in applications ranging from academic research to corporate and governmental decision-making, providing a powerful platform for understanding and responding to social media trends.

Conclusion

The Hashmeter Tool has successfully demonstrated its ability to provide accurate and real-time sentiment analysis on social media platforms, specifically Twitter. In an age where social media serves as a critical space for public expression, understanding public sentiment has become crucial for businesses, governments, and researchers alike. Hashmeter leverages the power of Natural Language Processing (NLP) techniques and advanced machine learning models, including VADER, TextBlob, BERT, and SpaCy, to provide reliable sentiment analysis that categorizes Twitter posts into positive, negative, or neutral sentiments. The system's ability to assess and categorize sentiments based on hashtags makes it particularly useful for monitoring public opinion surrounding specific topics, campaigns, or events in real time.

A standout feature of Hashmeter is its multi-model approach to sentiment analysis, incorporating multiple models to ensure robust results. The ensemble method used-where the sentiment is derived by a majority voting mechanism-enhances the system's accuracy by combining the strengths of each individual model. VADER is highly effective for informal, social media-based language, while TextBlob provides simplicity and ease of implementation. BERT, being a transformer model, enables the system to capture nuanced context within tweets, and SpaCy's pre-trained models offer powerful text classification capabilities. This combination of models allows Hashmeter to process even noisy and informal language accurately, which is essential when dealing with short-form content such as tweets.

Moreover, Hashmeter integrates an emotion detection model that goes beyond basic sentiment analysis. By identifying emotions such as joy, anger, sadness, and surprise, the system provides a deeper understanding of the emotional tones behind the tweets. This is crucial for applications that require not just sentiment classification, but also a more nuanced interpretation of public opinion. This emotion detection capability opens up opportunities for more specialized applications in fields like marketing, public relations, political campaigns, and social research, where understanding emotional sentiment can be just as important as knowing the overall sentiment.

From a user experience perspective, the frontend of Hashmeter, developed with React, is both responsive and intuitive, offering users an easy interface to input hashtags, retrieve relevant tweets, and view detailed sentiment analysis results. The system allows users to interact with both individual tweets and aggregated sentiment data through visualizations like bar charts and sentiment scores. This makes it easy for users, regardless of their technical expertise, to understand the public sentiment related to the topics they are monitoring. The inclusion of a post-analysis screen and analysis history functionality further enhances the user experience, allowing users to track sentiment trends over time and make informed decisions based on historical data.

The backend, built with Flask, ensures efficient handling of API requests and data processing, making it capable of scaling to accommodate growing volumes of data and user requests. The integration with PostgreSQL for data storage allows for quick retrieval and storage of tweet data, while the seamless flow between the frontend and backend ensures a smooth user experience. The system's ability to handle high-traffic loads and provide near-instantaneous feedback on sentiment analysis results makes it a valuable tool for real-time monitoring.

In addition to its practical applications, Hashmeter also holds great potential for advancing the field of social media analytics. Its real-time capabilities allow for the immediate measurement of public sentiment during critical events, such as product launches, political debates, or viral social media moments. By providing detailed insights into the mood and opinions of social media users, Hashmeter can support businesses and organizations in making timely and data-driven decisions. It also has the potential to contribute to research efforts focused on public opinion, political sentiment, or social movements by offering a cost-effective and scalable solution to monitor real-time conversations on Twitter.

In conclusion, the Hashmeter Tool is a powerful and versatile solution for real-time social media sentiment analysis. With its combination of advanced NLP techniques, multiple sentiment analysis models, emotion detection, and a user-friendly interface, the system provides a comprehensive view of public sentiment on social media platforms. Whether for business, research, or public monitoring, Hashmeter delivers reliable, actionable insights that enable users to make informed decisions and respond quickly to shifting public opinions. The tool's scalability and flexibility ensure that it can adapt to a wide range of use cases, making it a valuable asset for anyone seeking to understand and track social media sentiment in real time.

Reference

1. Nolan J, Dunne SS, Mustafa W, Sivananthan L, Kiely PA, Dunne CP. Proposed hypothesis and rationale for association between mastitis and breast cancer. Medical Hypotheses. 2020;144:110057. doi:10.1016/j.mehy.2020.110057
2. DeSantis CE, Ma J, Gaudet MM, Bruton D, Walters DM, Anderson RN, *et al*. Breast cancer statistics, 2019. CA: A Cancer Journal for Clinicians. 2019;69(6):438–451. doi:10.3322/caac.21583
3. Fiorica JV. Breast cancer screening, mammography, and other modalities. Clinical Obstetrics and Gynecology. 2016;59(4):688–709. doi:10.1097/GRF.0000000000000246
4. Armstrong K, Eisen A, Weber B. Assessing the risk of breast cancer. The New England Journal of Medicine. 2000;342(8):564–571. doi:10.1056/NEJM200002243420806
5. Barton MB. Breast cancer screening. Postgraduate Medical Journal. 2005;81(952):27–36. doi:10.1136/pgmj.2004.029089
6. Morris E, Feig SA, Drexler M, Lehman C. Implications of overdiagnosis: impact on screening mammography practices. Population Health Management. 2015;18(Suppl 1):S3–S11. doi:10.1089/pop.2015.0045
7. Gao Y, Stewart S, Loudon N, *et al*. Breast self-examination and breast cancer mortality: a randomized trial. Journal of the National Cancer Institute. 2005;97(19):1507–1514. doi:10.1093/jnci/dji322
8. National Comprehensive Cancer Network. NCCN clinical practice guidelines in oncology: breast cancer screening and diagnosis. Journal of the National Comprehensive Cancer Network. 2007;5(1):98–119.
9. American Cancer Society. Breast Cancer Facts & Figures 2019-2020. Atlanta: American Cancer Society; c2019.
10. American College of Radiology. ACR–SAR–SPR practice parameter for the performance of breast magnetic resonance imaging (MRI). Journal of the American College of Radiology. 2017;14(5):S1–S9. doi:10.1016/j.jacr.2017.02.010
11. American College of Radiology. ACR–SBI–ACOG–ABR practice parameter for the performance of breast ultrasound. Journal of the American College of Radiology. 2017;14(5):S10–S19. doi:10.1016/j.jacr.2017.02.011