www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 03, March 2025, pp : 2465-2468

e-ISSN : 2583-1062
Impact Factor : 7.001

SOCIAL MEDIA SENTIMENT ANALYSIS FOR BRAND MONITORING

Pranav. R[1], Chithra. S[2]

[1]UG Student, Computer Science with Data Analytics, Sri Ramakrishna College of Arts & Science, Coimbatore, Tamilnadu, India.

[2]Assistant Professor, Computer Science with Data Analytics, Sri Ramakrishna College of Arts & Science, Coimbatore, Tamilnadu, India.

ABSTRACT

This study explores sentiment analysis for social media brand monitoring using the BERT model. By analyzing tweets, the model classifies sentiments as positive or negative, providing businesses with insights into public perception. Data is collected via the Twitter API, preprocessed through text cleaning and tokenization, and analyzed for sentiment trends. Results highlight the effectiveness of deep learning in monitoring brand reputation. Future enhancements could include multilingual analysis and aspect-based sentiment classification.

Keywords: Sentiment Analysis, Social Media Monitoring, Deep Learning, NLP, Brand Reputation.

1. INTRODUCTION

The Research highlights the importance of sentiment analysis in brand monitoring. Social media platforms have become powerful tools for understanding customer opinions. Analysing sentiments from posts, tweets, and reviews helps businesses improve their strategies. This study utilizes machine learning and deep learning models to classify sentiments related to brand mentions. The objective is to extract meaningful insights from large datasets and provide real-time brand monitoring solutions. Furthermore, sentiment analysis plays a crucial role in decision-making for businesses by enabling proactive measures to address customer concerns. The increasing volume of social media data presents challenges in extracting relevant insights, making advanced machine learning techniques essential. With advancements in Natural Language Processing (NLP), sentiment analysis models have achieved higher accuracy and efficiency. By leveraging transformer-based models such as BERT, the study aims to enhance sentiment classification performance. The findings of this research can be applied across various industries, including marketing, customer support, and brand reputation management.

2. METHODOLOGY

The proposed sentiment analysis system is designed to monitor brand perception on social media using the BERT model. The methodology consists of five main steps:

1. Data Collection and Preprocessing
2. Feature Extraction and Encoding
3. Model Training and Fine - Tuning
4. Real-Time Sentiment Analysis
5. Evaluation and Performance Optimization

2.1 Data Collection and Preprocessing

Tweets related to a specific brand are fetched using the Twitter API. Data is retrieved in real-time based on user-defined queries. The API allows access to historical and live tweets, ensuring comprehensive data coverage. Filters such as language, location, and hashtags are applied to refine the dataset. To maintain data quality, duplicate tweets and bot-generated content are removed before further analysis. Additionally, metadata such as timestamps, user engagement metrics, and sentiment scores are stored for deeper insights.

2.2 Feature Selection and Normalization

In this step, Preprocessing Text preprocessing includes removing URLs, mentions, special characters, and converting text to lowercase. Tokenization is performed using the BERT tokenizer to prepare the data for sentiment classification. Stop words are removed to enhance feature extraction, and lemmatization is applied to standardize word forms. The cleaned text is then transformed into input embeddings compatible with deep learning models. Furthermore, noise reduction techniques such as spell correction and slang translation are applied to improve model accuracy.

2.3 Model Training and Implementation

The BERT-based sentiment analysis model classifies the sentiment of each tweet as positive or negative. The model is fine-tuned for better accuracy.

The analysis includes sentiment distribution visualization, word cloud generation for frequently used words, and graphical representation of sentiment trends. These insights help identify emerging sentiment patterns and track brand perception over time. Additionally, the model performance is evaluated using accuracy, precision, and recall metrics to ensure reliable sentiment classification results.

2.4 Real-Time Data Integration and Prediction

After training, the model integrates with a live API to fetch real-time data. This enables accurate sentiment analysis based on current trends.

The system processes incoming data and predicts sentiment shifts, helping brands respond proactively.

2.5 Evaluation and Performance Optimization

The final stage involves evaluating the model's performance using accuracy metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

The system is continuously optimized by fine-tuning hyperparameters and retraining with updated data to enhance its prediction accuracy and adaptability to dynamic market conditions.

3. MODELING AND ANALYSIS

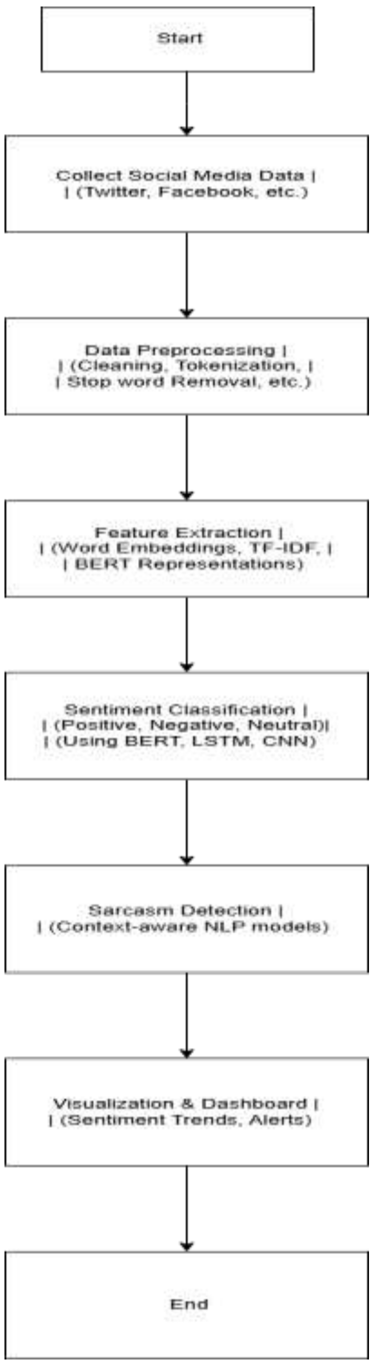
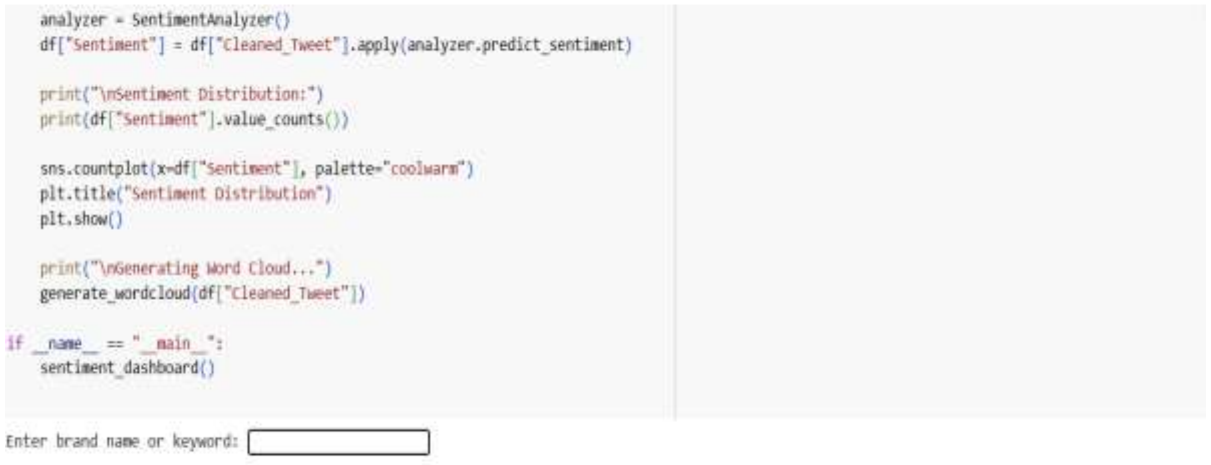

Start
Collect Social Media Data | | (Twitter, Facebook, etc.)
Data Preprocessing | | (Cleaning, Tokenization, | | Stop word Removal, etc.)
Feature Extraction | | (Word Embeddings, TF-IDF, | | BERT Representations)
Sentiment Classification | | (Positive, Negative, Neutral)| | (Using BERT, LSTM, CNN)
Sarcasm Detection | | (Context-aware NLP models)
Visualization & Dashboard | | (Sentiment Trends, Alerts)
End

Figure 1: System Work Flow

4. RESULTS AND DISCUSSION

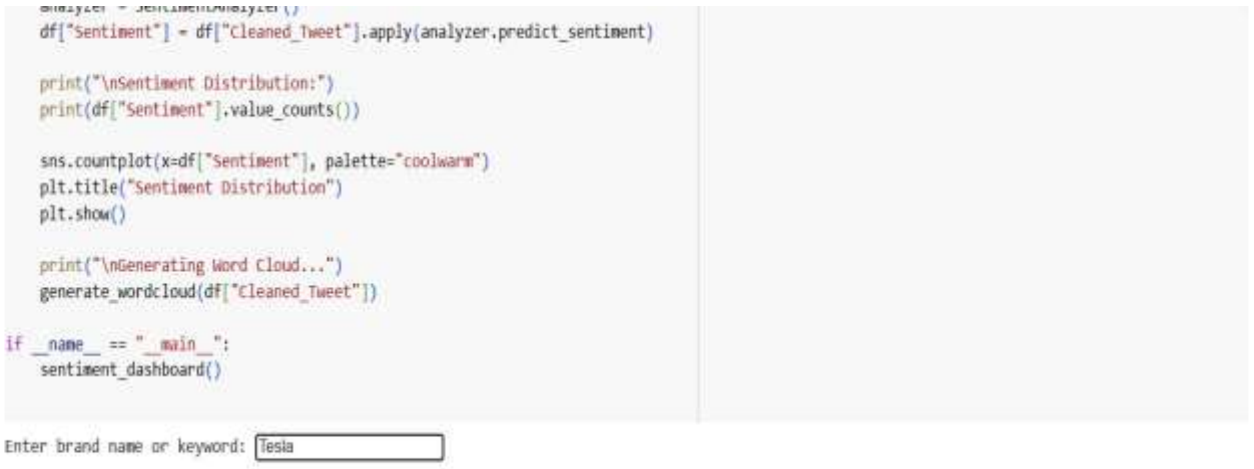
DATASET: The results indicate the proportion of positive and negative sentiments associated with the monitored brand. A graphical representation helps interpret public perception. The accuracy of sentiment classification is evaluated based on benchmark datasets. Observations include trends in brand perception and common keywords influencing sentiment.



```
    analyzer = SentimentAnalyzer()
    df["Sentiment"] = df["Cleaned_Tweet"].apply(analyzer.predict_sentiment)

    print("\nSentiment Distribution:")
    print(df["Sentiment"].value_counts())

    sns.countplot(x=df["Sentiment"], palette="coolwarm")
    plt.title("Sentiment Distribution")
    plt.show()

    print("\nGenerating Word Cloud...")
    generate_wordcloud(df["Cleaned_Tweet"])

if __name__ == "__main__":
    sentiment_dashboard()

Enter brand name or keyword:
```

Figure 2: Starting Interface

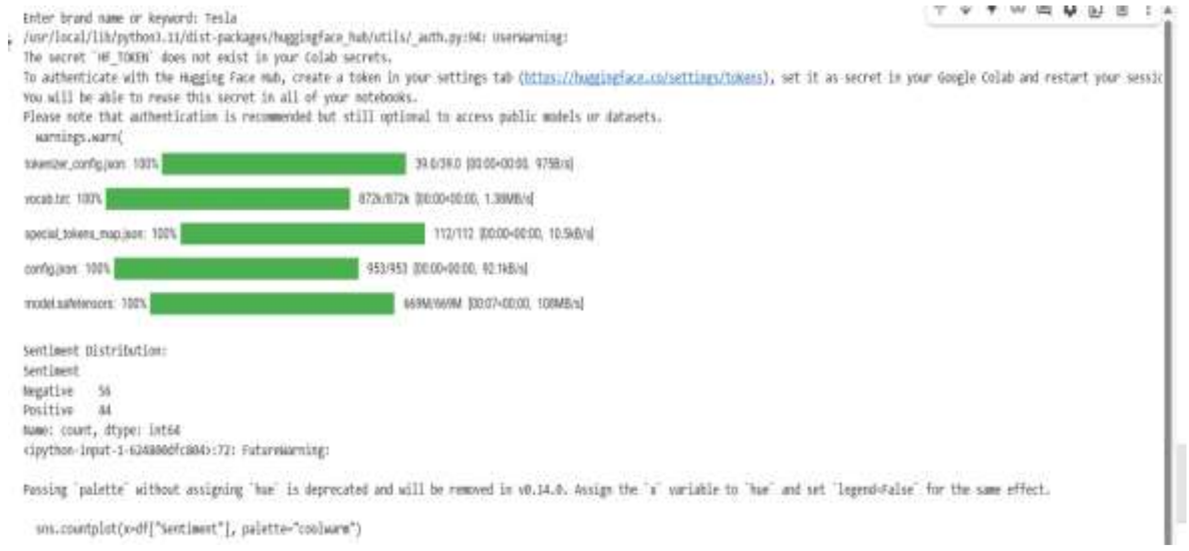
Enter a brand name to get sentiment analysis results for recent tweets, including sentiment distribution and keyword insights.



```
    analyzer = SentimentAnalyzer()
    df["Sentiment"] = df["Cleaned_Tweet"].apply(analyzer.predict_sentiment)

    print("\nSentiment Distribution:")
    print(df["Sentiment"].value_counts())

    sns.countplot(x=df["Sentiment"], palette="coolwarm")
    plt.title("Sentiment Distribution")
    plt.show()

    print("\nGenerating Word Cloud...")
    generate_wordcloud(df["Cleaned_Tweet"])

if __name__ == "__main__":
    sentiment_dashboard()

Enter brand name or keyword: Tesla
```

Figure 3: Input image

After entering a brand name, the system analyzes recent tweets and displays sentiment analysis results, including positive and negative sentiment percentages, graphical representation, and a word cloud of frequently used terms.



```
Enter brand name or keyword: Tesla
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your sessic
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
tokenizer_config.json: 100%  39.0/39.0 [00:00<00:00, 975B/s]
vocab.txt: 100%  872k/872k [00:00<00:00, 1.38MB/s]
special_tokens_map.json: 100%  112/112 [00:00<00:00, 10.5kB/s]
config.json: 100%  953/953 [00:00<00:00, 92.1kB/s]
model.safetensors: 100%  669M/669M [00:07<00:00, 108MB/s]

Sentiment Distribution:
Sentiment
Negative    56
Positive    44
Name: count, dtype: int64
<ipython-input-1-624800dfc804>:72: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(x=df["Sentiment"], palette="coolwarm")
```

Figure 4: Output image day 1

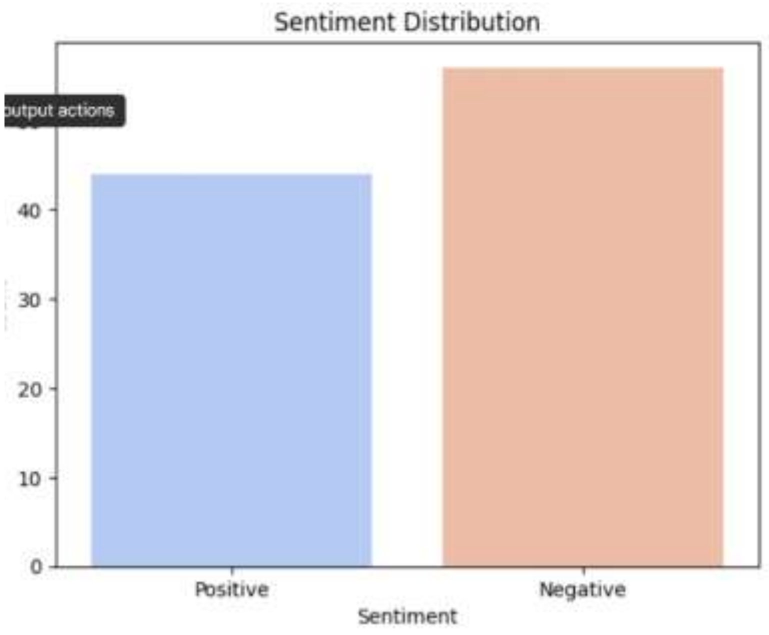


Figure 5: Output image day 2

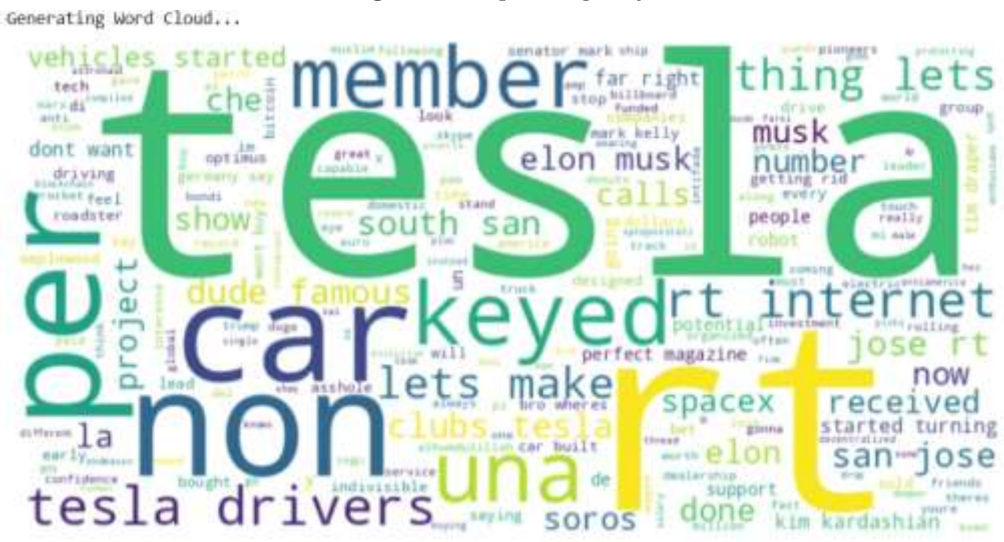


Figure 6: Output image day 3

5. CONCLUSION

This study successfully implements sentiment analysis using BERT for brand monitoring. The results highlight the effectiveness of deep learning in extracting insights from social media data. Future improvements can include multilingual support and aspect-based sentiment analysis for more precise brand monitoring.

6. REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv preprint arXiv:1810.04805, 2018.
[2] Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu, "Sentiment Analysis using Word Embeddings and Deep LearningTechniques,"InternationalConferenceonComputationalLinguistics,2018.
[3] A. Vaswani et al., "Attention Is All You Need," Advances in Neural Information Processing Systems, 2017
[4] Twitter Developer API Documentation, https://developer.twitter.com/en/docs.
[5] K. Clark, M. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training Text Encoders as Discriminators Rather Than Generators," ICLR, 2020.