Twitter Sentiment Analysis using Machine Learning

Arun Kumar Verma · Divyansh Upadhyay· Santosh Parmar · Abhineet Singh

Noida Institute of Engineering and Technology

0231dcsai308@niet.co.in · 0231dcsai309@niet.co.in · 0231dcsai300@niet.co.in ·0231dcsai001@niet.co.in

Title - Twitter Sentiment Analysis

Abstract

Social media platforms like Twitter offer a valuable stream of real-time public opinion. This paper presents a sentiment analysis system that classifies tweets as positive, negative, or neutral using machine learning algorithms. The methodology includes data preprocessing, feature extraction via TF-IDF, and training of models like Logistic Regression and Naive Bayes. The system achieves an accuracy of approximately 85%, demonstrating its potential in understanding user sentiment. Future improvements may include deep learning models like BERT and real-time analysis capabilities.

Keywords

Sentiment Analysis, Twitter, Machine Learning, NLP, TF-IDF, Social Media Mining

1. Introduction

With the exponential rise of social media platforms, Twitter has emerged as a powerful medium for expressing opinions. Sentiment analysis, also known as opinion mining, involves analyzing text to determine the emotional tone behind it. This research aims to build a machine learning pipeline to classify sentiments from tweets.

2. Literature Review

Early sentiment analysis relied on rule-based systems and lexicons. With machine learning advancements, algorithms like Naive Bayes, SVM, and Logistic Regression improved classification tasks. Recent models like LSTM and BERT show promise in handling context and complex sentiment structures. Notable studies include:
- Pak & Paroubek (2010): Twitter corpus for sentiment analysis.

- Go et al. (2009): Distant supervision using emoticons.
- Rosenthal et al. (2017): SemEval Twitter sentiment benchmarking.

3. Research Gaps

- Sarcasm Detection: Difficult for traditional models.
- Slang/Informal Language: Tweets often include emojis, abbreviations, and misspellings.
- Multilingual Support: Focus is mostly on English.
- Data Integration: Limited use of external context (e.g., news).
- Real-time Tracking: Most systems analyze data post hoc.

4. Objectives

- Build a sentiment analysis model using ML.
- Improve classification accuracy through preprocessing.
- Enable real-time tweet classification.
- Explore multilingual and cross-domain applications.

5. Methodology

5.1 Data Collection:
Tweets were gathered using Twitter API and Nitter. Pre-labeled datasets from Kaggle were also utilized.

5.2 Preprocessing:
- Removal of URLs, mentions, emojis.
- Tokenization, stopword removal.
- Lemmatization.

5.3 Feature Extraction:
TF-IDF vectorization was used to convert text into numeric form.

5.4 Model Selection:
Models: Logistic Regression, Naive Bayes, Random Forest.
Evaluation Metrics: Accuracy, Precision, Recall, F1-Score.

5.5 Tools:
Python, Pandas, NumPy, Scikit-learn, NLTK, Matplotlib, Seaborn.

6. Results

The Logistic Regression model yielded the best results with an accuracy of 85%. Misclassifications were mostly between neutral and positive sentiments. Visualization techniques like word clouds and bar charts were used for interpretability.

Metric	Value
Accuracy	85%
Precision	84%
Recall	83%
F1-Score	83.5%

7. Conclusion

The project successfully demonstrates how machine learning can be used to derive sentiment from tweets. Despite informal language and brevity, simple models perform well with proper preprocessing. Applications include political analysis, brand monitoring, and crisis management.

8. Future Work

- Integrate deep learning models (e.g., BERT, LSTM).
- Real-time streaming and prediction using Kafka/Spark.
- Multilingual sentiment analysis.
- Dashboard interface for sentiment visualization.
- Ethical considerations in automated opinion mining.

9. References

1. Liu, B. (2012). Sentiment Analysis and Opinion Mining.
2. Pak, A., & Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis.
3. Go, A., Bhayani, R., & Huang, L. (2009). Twitter Sentiment Classification Using Distant Supervision.
4. Devlin, J. et al. (2019). BERT: Pre-training of Deep Bidirectional Transformers.
5. scikit-learn Documentation. https://scikit-learn.org
6. NLTK Documentation. https://www.nltk.org
7. Twitter Developer API. https://developer.twitter.com