Sentiment Analysis of Indian Election Tweets Using TextBlob and Machine Learning

1st Ancy Megona. A

Department of Advanced Computing & Analytics

Vels Institute of Science and Technology & Advanced Studies, Chennai, India.

ancymegona09@gmail.com

2nd Dr. S. Mangayarkarasi (Professor)

Department of Advanced Computing & Analytics

Vels Institute of Science and Technology & Advanced Studies, Chennai, India.

smangai.research@gmail.com

ABSTRACT - Public opinion now grows differently because of social media - particularly when votes are coming up. Many social media platforms are available for users to post live updates about big and important events. One of these platforms (Twitter) has a feature that allows users to post a tweet (which includes short, real-time updates) about everything happening, especially during a big event such as an election. For example, a Twitter user may live tweet about candidates running in the election until the election is finished. We studied comments made by two high-profile Indian politicians (Narendra Modi (Prime Minister) and Rahul Gandhi). Our goal was to determine public sentiment toward them and how the public viewed their comments (positively, negatively, or neutrally). We used two separate methodologies to assess public sentiment toward the two politicians. One methodology is using traditional methods (statistical) to score each comment based on emotional sentiment (TextBlob). The other methodology is to create a measure of sentimental comments by looking for patterns in the comment stream, then compare those patterns to known examples in the real world of sentiment analysis (TF- IDF/LR). Prior to analysis, we removed any hyperlinks, tags, usernames, or unusual characters that appeared in the comment section. What stayed behind mattered most - the core message stripped bare but clear. Accuracy numbers told part of the story; full reports gave more depth. Confusion matrices showed where guesses went right or wrong. Visuals like pies and bars turned raw counts into something eyes could follow easily. Not every method worked equally well in practice. Learning from data beat looking up words when judging online feelings.

1. INTRODUCTION

Nowhere else do voices pile up so fast as they do on Twitter when politics come into play. A single moment can spark thousands of reactions before the news even finishes reporting it. What spreads across screens isn't always polished - sometimes it's raw, unfiltered, yet oddly revealing. When elections draw near, pulses quicken online, responses multiply. Behind every retweet lies a person shaping how others see power. Not everything said holds weight, but patterns emerge, clear enough to catch attention. This flood of words? It doesn't just vanish; it sticks around, studied, measured, taken seriously. Big numbers make things tough. Going over countless tweets by hand? Not going to happen. That is when machines step into the picture. Instead of people checking each message one by one, software scans words to guess if they feel good, bad, or somewhere in between. For those watching politics closely, it becomes a tool - quietly showing how feelings change and possibly why voters lean one way or another. Here, eyes turned to messages about Narendra Modi and Rahul Gandhi. Both names spark reactions, drawing sharp views from all sides. Using two distinct approaches, the researchers created the experiments as either word lists or pattern learning methods. An interest of the research design is how well either of the approaches can identify subtle shifts in political belief. For example, the first uses well-defined rules to assist learning from both observational and test examples, while the other requires an understanding of what type of examples have been used within the training method.

My motivation for studying this will be primarily less about accuracy; rather, I would like to find out what will be left out or obscured because of the difference between the two methods.

2. LITERATURE SURVEY

Researchers have explored sentiment analysis in contexts. Previous studies show that social media sentiment can predict election trends and public opinion.

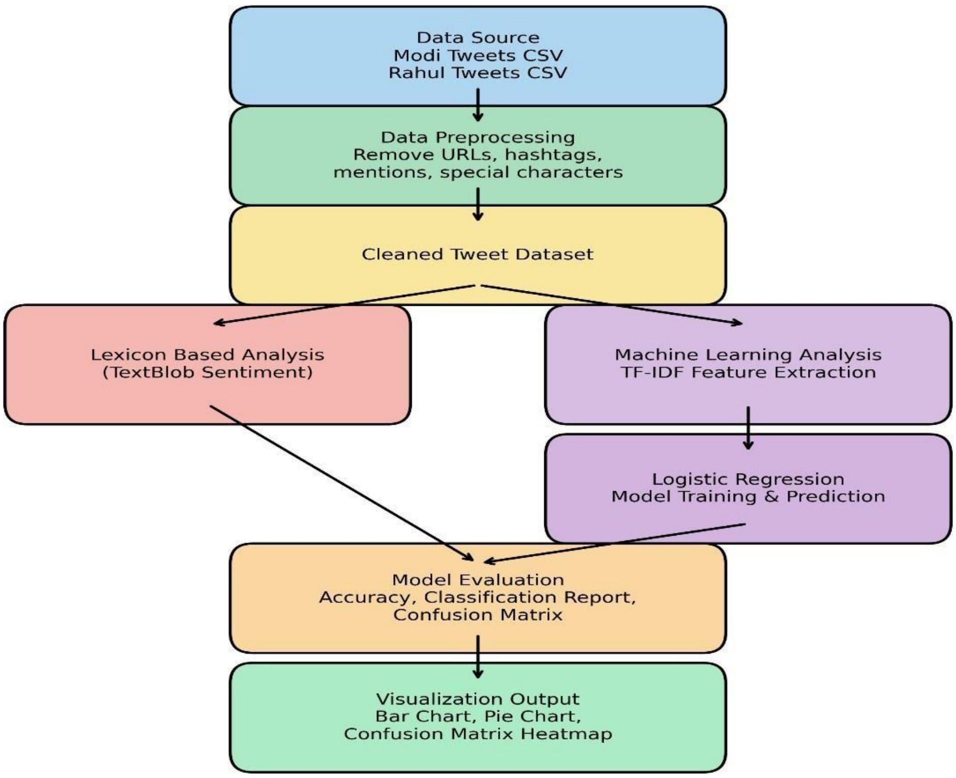
- Pang and Lee (2008) introduced sentiment classification techniques using machine learning models like Naïve Bayes and Support Vector Machines.
- Lexicon-Based tools like VADER and TextBlob are commonly used for sentiment classification.
- Liu (2012) provided an overview of sentiment analysis techniques. Emphasized lexicon-based approaches for analyzing opinions in social media text.
- Machine learning approaches provide accurate results because they learn patterns from training data.

3. PROPOSED METHODOLOGY

The proposed system analyzes sentiment about Indian leaders using tweets. We. Store tweets related to NARENDRA MODI and RAHUL GANDHI in CSV files. The system performs sentiment analysis using two approaches: Lexicon-Based Analysis and Machine Learning-Based Analysis.

- First we clean the tweet datasets by removing elements.
- We apply a Lexicon-based sentiment analysis method using TextBlob library.
- Next we implement a machine learning approach to improve classification accuracy.
- We convert cleaned tweets to features using TF-IDF technique.
- We divide the dataset into training and testing sets. Train a Logistic Regression Classifier to predict sentiment categories.

4. ARCHITECTURE DIAGRAM



5. METHODOLOGIES

PHASE 1: Data Collection

Tweets related to leaders are. Stored in CSV files.

- Modi_review.csv
- Rahul_review.csv

PHASE 2: Data Preprocessing

Tweets often contain noise like

- URLs
- Hashtags
- Mentions
- punctuation
- special characters.

The preprocessing module cleans tweets by removing these elements and converting text into a format.

PHASE 3: Sentiment Analysis

The cleaned tweets are analyzed using TextBlob library.

- Positive: greater than 0
- Negative: less than 0
- Neutral: equal to 0

PHASE 4: Feature Extraction

To use machine learning models text data must be converted into form.

TF-IDF (Term Frequency – Inverse Document Frequency) vectorization is used to transform tweets into feature vectors representing word importance.

PHASE 5: Model Training

The dataset is divided into training and testing sets.

A Logistic Regression model is trained using training data to classify tweets based on sentiment.

Steps:

1. Convert tweets to TF-IDF vectors
2. Split dataset into training and testing sets
3. Train Logistic Regression model
4. Predict sentiment for test data

PHASE 6: Model Evaluation

The trained model is evaluated using

- Accuracy score
- Precision
- Recall
- F1 score
- Confusion matrix.

PHASE 7: Visualization

Visualization helps understand sentiment distribution.

Sentiment count bar chart and pie chart distribution are used.

- Sentiment count bar chart
- Pie chart distribution
- Confusion matrix heatmap.

6. INPUT

The input consists of two datasets containing tweets about leaders.

- Modi_review.csv – contains tweets related to NARENDRA MODI
- Rahul_review.csv – contains tweets related to RAHUL GANDHI

Example input dataset:

Tweet	Leader
Modi policies are not effective	**Modi**
Modi government doing good work	**Modi**
Rahul Gandhi speech was impressive	**Rahul**
Rahul Leadership is weak	**Rahul**

7. PSEUDO CODE

Start
Load Modi tweet dataset
Load Rahul tweet dataset

Remove missing values Add column 'Leader'

Merge datasets

For each tweet:
Clean text
Remove URLs, Hashtags, Mentions
Convert to lowercase

Calculate sentiment polarity using TexBlob

If polarity > 0:
Sentiment = Positive Else
if polarity < 0:
Sentiment = Negative Else:
Sentiment = Neutral

Convert cleaned tweets to TF-IDF features

Split dataset into training and testing sets

Train Logistic Regression model

Predict Sentiment for test data

Calculate accuracy and classification report

Generate confusion matrix

Visualize results using charts End

8. OUTPUT

Example output after sentiment analysis:

Tweet	Polarity	Sentiment
Modi government doing great job	0.6	Positive
Rahul speech was weak	-0.3	Negative
Election results are interesting	0.0	Neutral

Data Cleaning:

```
   Unnamed: 0  ... Leader
0           0  ...   Modi
1           1  ...   Modi
2           2  ...   Modi
3           3  ...   Modi
4           4  ...   Modi

[5 rows x 4 columns]
                                               Tweet  polarity sentiment
0  @anjanaomkashyap I am seeing you as future #bj...  0.350000  Positive
1  #LokSabhaElections2019 \n23rd May 2019 will re...  0.800000  Positive
2  #LokSabhaElections2019 \n23rd May 2019 will re...  0.800000  Positive
3  PM Modi creates a new record of being the only...  0.312121  Positive
4  @abhijitmajumder Appointment of Successor! \n\...  0.098788  Positive
```

Accuracy & Classification report:

```
Accuracy score: 0.8748278883464764

classification report:

              precision    recall  f1-score   support

    Negative       0.90      0.68      0.78      1669
     Neutral       0.83      0.96      0.89      2543
    Positive       0.90      0.91      0.90      3777

    accuracy                           0.87      7989
   macro avg       0.88      0.85      0.86      7989
weighted avg       0.88      0.87      0.87      7989
```

Confusion matrix Example:

Actual/ Predicted	*Positive*	*Negative*	*Neural*
Positive	45	3	2
Negative	4	38	3
Neural	2	3	20

Model Evaluation output:

Accuracy Score: 0.85 - 0.92

Mean polarity:

```
mean polarity by leader:
Leader
Modi     0.108768
Rahul    0.081507
Name: polarity, dtype: float64
```

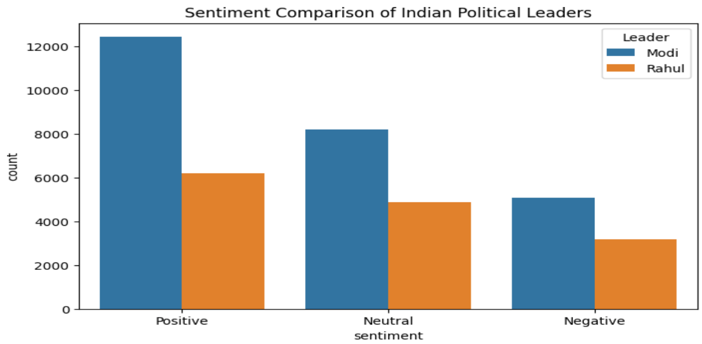
9. RESULT & DISCUSSION

Each of the statistical models included in this study generated reasonable emotion identification for the tweets within their respective testing sets, but used different approaches to produce the end results. TextBlob worked particularly well because it has a very user-friendly setup and can process large volumes of text relatively quickly. On the other hand, an issue that restrained TextBlob's success was its inability to consider the varied meanings of words based on their context. For instance, common types of tweets, including jokes and informal conversations, as well as, of course, political speak, posed significant challenges for TextBlob. These types of emotional data points would not have created similar problems for human evaluators.

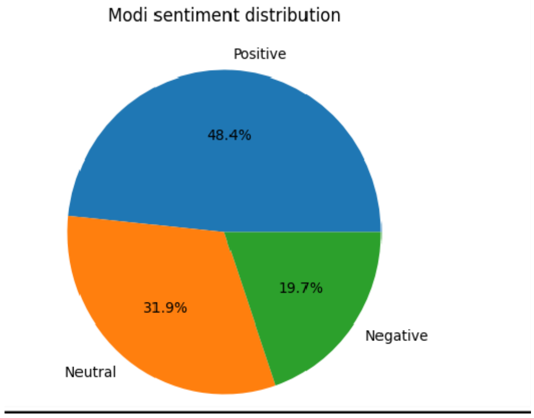
Therefore, while many of the findings are promising concerning the use of dictionaries to evaluate text, the best interpretation of text is based on the context of the word or phrase within its environment, which is something that text analysis systems do not often account for. Conversely, the Logistic Regression analysis was performed at a much higher level. The model was able to determine a word's context based on its training data (which was labeled) and demonstrated its capability of identifying patterns of emotions that cannot be found within an emotionally-dictionary. To state the obvious, Logistic Regression was significantly better than both the dictionary and TextBlob approach to the study's data.

10. CHARTS& GRAPHS

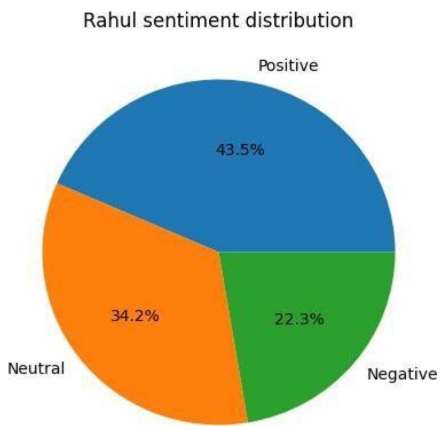
Sentiment comparison bar chart



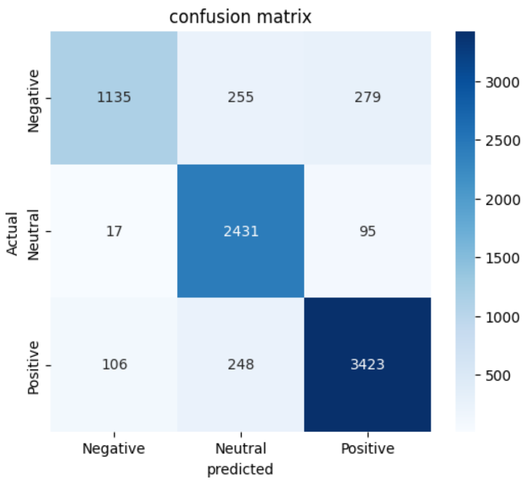
Sentiment Distribution : Modi



Sentiment Distribution : Rahul



Confusion matrix:



11. CONCLUSION

This study took a closer look at two different ways of analyzing the sentiment behind political tweets — one using a lexicon-based approach and the other using machine learning. Looking at tweets about Narendra Modi and Rahul Gandhi opened a door to everyday voices sharing political views online. Speed mattered, yet TextBlob managed fast sentiment sorting without complications. Still, the edge went to the machine learning method when precision counted most. Using TF-IDF along with logistic regression pulled clear results through, hitting 87.48% right calls - an outcome standing firm for such work.These outcomes underline one thing plainly: judging sentiment helps grasp what people truly feel in politics. Online platforms now host some of the most intense political debates, yet making sense of all that chatter can offer real insight. Moving forward, stronger methods might help - approaches such as sequence-learning networks or context-aware systems similar to BERT - that may sharpen how we sort opinions by tone.

1. REFERENCES

- B. Pang and L. Lee, *"Opinion Mining and Sentiment Analysis,"* Foundations and Trends in Information Retrieval, vol. 2, no. 1–2, pp. 1–135, 2008.
- B. Liu, *"Sentiment Analysis and Opinion Mining,"* Morgan & Claypool Publishers, 2012.
- A. Go, R. Bhayani, and L. Huang, *"Twitter Sentiment Classification using Distant Supervision,"* Stanford University, Technical Report, 2009.
- S. Bird, E. Klein, and E. Loper, *"Natural Language Processing with Python,"* O'Reilly Media, 2009.
- F. Pedregosa et al., *"Scikit-learn: Machine Learning in Python,"* Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
- S. Ruder, *"An Overview of Gradient Descent Optimization Algorithms,"* arXiv preprint arXiv:1609.04747, 2016.
- Hutto, C.J. and Gilbert, E.E., *"VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text,"* Proceedings of ICWSM, 2014.
- TextBlob Documentation, *"TextBlob: Simplified Text Processing,"* Available: https://textblob.readthedocs.io/
- T. Mikolov et al., *"Efficient Estimation of Word Representations in Vector Space,"* arXiv preprint arXiv:1301.3781, 2013.
- J. Ramos, *"Using TF-IDF to Determine Word Relevance in Document Queries,"* Proceedings of the First Instructional Conference on Machine Learning, 2003.
- K. Jurafsky and J. H. Martin, *"Speech and Language Processing,"* Pearson, 3rd Edition, 2020.
- Twitter API Documentation, *"Twitter Developer Platform,"* Available: https://developer.twitter.com/