Sentiment Analysis of Indian Political Tweets using BERT Model

[1, *]Ayushi Pati, [2]Ayan Mondal

[1] Department of Computer Science and Engineering, Silicon University, Bhubaneswar, India
[2] Department of Computer Science and Engineering, Indian Institute of Technology Indore, India
{[1]ayushipati2006@gmail.com,[2]ayanm@iiti.ac.in}

Abstract. Social media has drastically changed the political discussion in India, allowing people to interact and share political content instantly. These platforms generate large volumes of real-time discussions, opinions, and reactions. This study helps in predicting sentiments in Indian political tweets to understand public sentiment using three deep learning (DL) models, Convolutional Neural Networks (CNN), Bidirectional Long Short-Term Memory (Bi-LSTM), and Bidirectional Encoder Representations from Transformers (BERT). The dataset was taken from Kaggle dataset repository, and pre-processing steps were carried out to reduce the noise present in the dataset. Each considered model was trained and evaluated for sentiment classification. The results noted from the experimental study states that the BERT model outperformed among others with 95% accuracy. The accuracies obtained for CNN and Bi-LSTM were found to be 92% and 93% respectively. The experimental findings from the study demonstrate that BERT model can help in accurately analyzing political discussions on social media and offer insights into public opinion as well as electoral trends in India.

Keywords: CNN, BiLSTM, BERT, Political tweets, Sentiment analysis

For Personal Use Only

1. Introduction

Social networks platforms have become an important forum for modern political discussion. It is influencing the way information is produced, broadcasted, and accepted by the audiences [1]. In the Indian context, both political awareness and digital connectivity are rapidly increasing day by day. Twitter (now called as *X*) is now accepted as one of the important space for instant political interaction. Now a days, users actively share their opinions, engage with campaign narratives, re-

spond to policy decisions, and contribute to politically driven discussions in Twitter. As a result, the platform generates huge amount of contents which can be analyzed to discover insights into public sentiments [2, 3]. Analyzing user driven political discussions data in the Indian context generated from social media may have many challenges [4].

Users generally use multiple languages, commonly Hindi and English in a single post along with abbreviations, hashtags, emojis, and informal grammar [5]. This results in reducing the effectiveness of conventional text analysis techniques [6]. Recent developments in natural language processing (NLP), especially the evolvement of transformer-based architectures, have proven to be effective in handling multilingual and code-mixed datasets [6]. Transformer based DL architectures, BERT, XLM-RoBERTa, and IndicBERT are capable of capturing complex linguistic patterns across diverse languages, enabling more accurate sentiment detection and thematic classification [5, 6]. This study analyzes political tweets using a transformer based model, BERT to predict sentiments across a diverse set of political conversations and also compares traditional DL models such as CNN and Bi-LSTM with BERT to measure the effectiveness of BERT in predicting sentiments using various evaluation parameters as well as accuracy.

The rest of the paper is outlined as follows: The literature review on the topic presented in this study is included in Section 2. The proposed work and the results are given in Section 3 and Section 4 respectively. Section 5 includes the conclusion along with future scope of this work.

2. Background Study

Twitter is now become the key place for political communication which enables the users to participate in public debate as well as in sharing their opinions [7]. Earlier research findings state that traditional ML models and lexicon-based methods were used for social media text sentiment prediction [8]. However, the performance of these techniques was found to have limitations as the user generated content are informal in nature. However, recent studies demonstrate that DL based models offer considerable improvements [9, 12, 13].

Bhat *et al.* [8] proposed a technique for sentiment analysis of Indian political tweets using LSTM and Recurrent Neural Network. The dataset used by the authors was taken from Kaggle. They compared and evaluated these models using two feature representation methods such as Word2Vec and TF-IDF. Their results illustrates that TF-IDF outperforms Word2Vec for this task with the LSTM model

For Personal Use Only

and produces the accuracy of 83.02% and is higher as compared to Recurrent Neural Network with 81.06% accuracy. In [9], authors proposed a method which uses LSTM architecture with Kalman filters to perform sentiment analysis on tweets about political parties in India. In their work, they demonstrated that Kalman filter is used to smooth the noisy sentiment signal and LSTM handles long-term dependencies in the sequential data. They reported that the hybrid model achieved higher performance as compared to using LSTM only. In [10], authors proposed a technique using ML to analyze Israeli political tweets to overcome challenges such as regional dialects and linguistic biases.

Beyond sentiment classification, some studies have explored the emotional dimensions of political communication. Sharma and Vegad [11] conducted emotion analysis on a large corpus of Indian political tweets collected during the 2019 general election with 117,545 tweets. For modeling, they employed a Support Vector Machine classifier to categorize tweets into different sentiment classes. Their work highlights the complexities of emotion detection in political Twitter data. By applying ML models to such a large and politically relevant dataset, the authors demonstrated how emotion analysis can be used to understand public opinion and political sentiment on social media.

In [13], authors had illustrated the ability of LSTM model in analyzing the sentiment within complex text structures while in [12], Paduri *et al.* demonstrated that how LSTM can model temporal patterns. In [14], author performed a literature study emphasizing that the effective political tweet analysis must account for linguistic diversity, data imbalance, and rapid change of political narratives. Author identified that a major technical challenge in the Indian context is the presence of code-mixed i.e. Hindi-English (Hinglish) tweets.

In summary, the existing research indicates that effective analysis of Indian political tweets requires robust handling of code-mixed text and emotion-aware modelling to capture political polarization. These works provide directions to the researchers for developing more suitable models for understanding political discourse in India.

3. Proposed Work

The proposed work focuses on using DL architectures to accurately predict sentiments in Indian political tweets. For this we used a dataset downloaded from Kaggle, consisting of Indian political tweets. The dataset was then pre-processed and then DL architectures were applied on the cleaned dataset and the models

For Personal Use Only

were compared to choose the best model among them. This section presents the methodology used to predict the users' sentiments effectively.

3.1. Methodology and implementation

This section covers the overall description of the steps followed in the methodology and implementation. The methodology used in this work consists of four major steps such as data collection, pre-processing, model development, and analysis. The implementation of this study was done using Python version 3 in Google Colab environment.

3.1.1 Dataset description

The dataset [15] is a collection of public tweets on Indian politics and their associated reactions. This dataset helps to gain insights into public opinion and sentiment on various political issues and events in India. The dataset consists of various categories of tweets from users of India that covers various political contexts. The information about the considered dataset is presented in Table 1.

Table 1: About the Dataset

Dataset Name	"Indian Politics Tweets & Reactions" [15]
No. of instances	50,000
No. of features	06

The word cloud of the text present in the considered dataset is displayed in Fig. 1. The size of each word represents the number of occurrences of a word in the tweets.



Fig. 1: Word cloud of the dataset

For personal Use Only

3.1.2 Data Pre-processing

During this step, first we removed all irrelevant columns except the tweets column and the pre-processing was done using "re" library of Python to remove punctuations, special symbols, links, and spaces etc. to clean the dataset. This step helps us to ensure that the dataset is cleaned and suitable for DL tasks.

3.1.3 Sentiment polarization and label generation using TextBlob

The original dataset was unlabelled and contained only raw tweets without any predefined sentiment classes. To generate sentiment labels, the TextBlob Python library was used. TextBlob computes the polarity score of each tweet based on its semantic orientation. The polarity value ranges from −1 to +1. Here, a value less than 0 indicates a negative sentiment, 0 represents a neutral sentiment, and greater than 0 value indicates a positive sentiment. Based on these polarity scores, the tweets were automatically categorized into three sentiment classes: positive, neutral, and negative [16]. This automated labelling process transformed the raw textual dataset into a supervised learning dataset suitable for training DL models.

After sentiment polarization using TextBlob, 23,600 positive, 13,548 neutral, and 12,852 negative tweets were labeled in the dataset. The sentiment polarity distribution present in the dataset in shown in Fig. 2. Fig. 2 shows that there is higher number of positive tweets present in the dataset as compared to negative and neutral tweets and the percentage of negative and neutral tweets are found to be present in nearly equal amount.

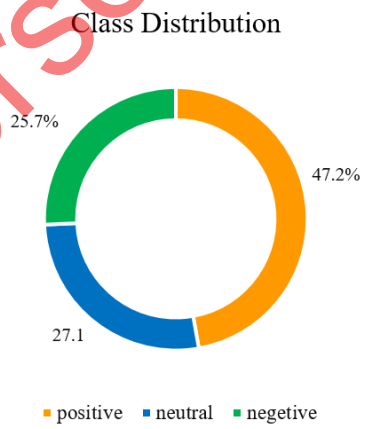


Fig 2: Polarity distribution of tweets in the dataset

For Personal Use Only

3.1.4 Tokenization and feature representation

After sentiment labelling, the textual data were converted into numerical form so that they can be used by the DL models. For the CNN and Bi-LSTM models, text tokenization was done using the TensorFlow/Keras Tokenizer whose vocabulary was limited to 20,000 frequently occuring words. An out-of-vocabulary (OOV) token is used to handle words that are not seen during training. The input sequences were then padded to a fixed length of 100 tokens which ensures uniform input size.

For the BERT model, tokenization was done using the BERT tokenizer, which breaks text into smaller sub-word units when needed. The input sequences in this model were padded and truncated to a maximum length of 128 tokens, and attention masks were created to differentiate real words from padded values. This process ensured that the text data were properly prepared for transformer-based BERT model.

3.1.5 Parameter tuning for DL models

In this step, DL architectures such as CNN, Bi-LSTM, and BERT models were trained and various hyper parameters such as optimizer, number of epochs, and batch size were customized to improve the accuracy. For training and testing purpose, the dataset was divided in 80:20 ratio. A batch size of 32 for five epochs was considered for training of all considered models. For the CNN and Bi-LSTM models, the *Adam* optimizer was used with *ReLU* activation in the hidden layers and *Softmax* in the output layer. The loss function used for CNN and Bi-LSTM was *sparse categorical cross-entropy*. The BERT model is a pre-trained transformer-based architecture, therefore it inherently uses the *GELU* activation function and is optimized using the *AdamW* optimizer with *cross-entropy* loss to preserve its pre-trained learning capability. Here, different training configurations were adopted to ensure fair and optimal performance based on the architectural nature of each model rather than enforcing identical hyperparameters across all models.

4.Results and Discussion

The obtained results from the experimental study carried out in this work are presented in this section. Different DL models were applied on the dataset to predict sentiments. After the model evaluation, it was found that BERT had given higher accuracy of 95.42% whereas CNN and Bi-LSTM had 92% and 93% accuracy respectively. The accuracy obtained from different models is displayed in Fig. 3. The accuracy graph indicates that BERT model has outperformed all other models in predicting sentiments. On the other hand, CNN was found to be less efficient model in terms of accuracy for analyzing political tweets.

For Personal Use Only

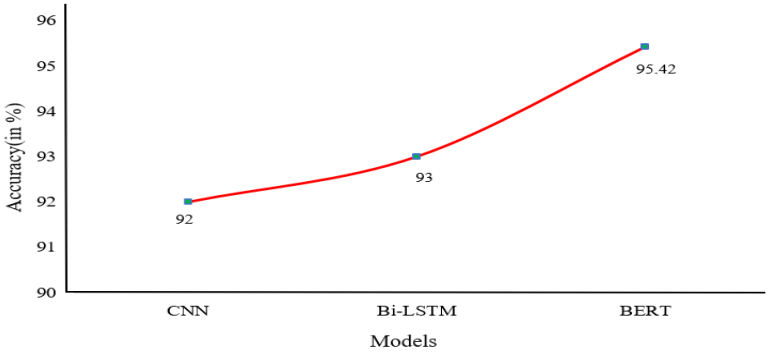


Fig. 3: Accuracy of different models

In addition to accuracy, BERT model was analysed using several performance metrics such as Precision (Pre), Recall (Rec), F1-Score (Fsc), and Support (Sup). The values obtained for these evaluation parameters are listed in Table 2. The Confusion Matrix (ConM) was presented in Fig. 4 which shows the classification results of the model. The ConM of BERT model indicates that it achieves a very high number of true positives which is required for reliable sentiment prediction.

Table 2: Models performance parameters values

Models	**Classes**	**Metrics**			
		Pre	**Rec**	**Fsc**	**Sup**
CNN	Negative	0.92	0.83	0.87	2597
	Neutral	0.95	0.93	0.94	2615
	Positive	0.90	0.96	0.93	4788
Bi-LSTM	Negative	0.88	0.91	0.90	2597
	Neutral	0.93	0.95	0.94	2615
	Positive	0.95	0.92	0.94	4788
BERT	Negative	0.95	0.92	0.93	2593
	Neutral	0.98	0.95	0.96	2587
	Positive	0.94	0.98	0.96	4820

For Personal Use Only

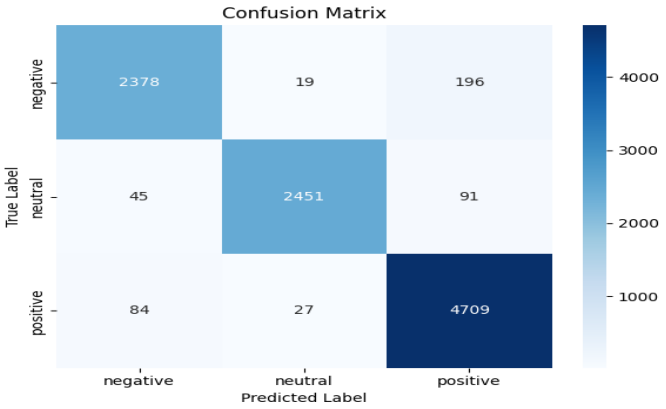


Fig. 4: Confusion matrix of BERT model

The ROC curves per class for BERT model is depicted in Fig. 5 and is plotted between the true positive rate vs. false positive rate. The comprehensive performance of the model is observed from the area under the curve (AUC).

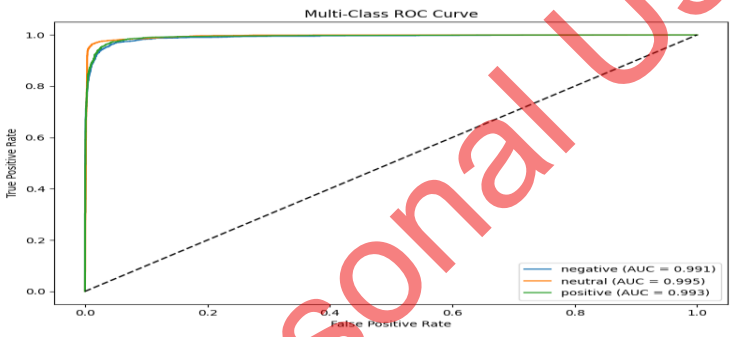


Fig. 5: ROC curves per class for BERT model

Fig. 5 shows that the proposed model achieve excellent class-wise discrimination, with AUC values of 0.991, 0.995, and 0.993 for negative, neutral, and positive classes respectively. This indicates that the model is capable of effectively distinguish between positive, neutral, and negative political opinions with high reliability.

5. Conclusion and Future work

In the current study, we applied three DL based models, CNN, Bi-LSTM, and BERT on a dataset downloaded from Kaggle repository for accurate sentiment

For Personal Use Only

prediction. Then the performance of these considered models were analysed. The BERT model was found to give higher accuracy of 95% as compared to other models.

Future research can incorporate domain specific models to improve contextual understanding and reduce linguistic bias. Since Indian political tweets often are a mix of English with Hindi or regional languages, future work can focus on multilingual transformers and Hinglish code-mixed models to enhance sentiment classification accuracy.

References

1. Duncombe, C. (2019). The politics of twitter: Emotions and the power of social media. International Political Sociology, 13(4), pp. 409-429.
2. Kumar, S., & Shah, N. (2018). False information on web and social media: A survey. Social Media + Society, 4(1), pp. 1-11.
3. Dubois, E., & McKelvey, F. (2020). Influencers, the elite, and the 2016 U.S. election: Twitter's real impact on politics. Political Communication, 37(2), pp. 1-15.
4. Kakwani, D., et al. (2020). IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 4948–4961, Online. Association for Computational Linguistics.
5. Conneau, A., et al. (2020). Unsupervised cross-lingual representation learning at scale, In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8440–8451, Online. Association for Computational Linguistics.
6. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1, pp. 4171-4186. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423
7. Madakam, S. and Tripathi, S. (2021). Social media/networking: Applications, technologies, theories. Journal of Information Systems and Technology Management, 18, https://doi.org/10.4301/S1807-1775202118007.
8. Bhat S., Patel V.R., Muddapur O.R., Chandanagouda H., Kulkarni U., Gurlahosur S. (2025). Sentiment Analysis of Indian Political Tweets: A Comparative Study with LSTM and RNN Model. In: INCOFT 2025. SciTePress, pp. 701–706. https://doi.org/10.5220/0013584000004664
9. Negi D., Abhinandan, Jasveer, Taneja S. (2023). Sentiment Analysis of Twitter Data Using Kalman Filters and LSTM for Political Opinion Prediction. IJETMR, 10(6), 58–67. https://doi.org/10.29121/ijetmr.v10.i6.2023.1604
10. Gangwar, A. and Mehta, T. (2022). Sentiment analysis of political tweets for israel using machine learning. arXiv preprint arXiv:2204.06515.

For Personal Use Only

11. Sharma P.P., Vegad M.T. (2024). The Emotion Analysis of Indian Political Tweets Using Machine Learning. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 10(2), pp.462-468.
12. Paduri, A. R. et al. (2022). Stock price prediction using sen timent analysis and deep learning for indian markets. arXiv preprint arXiv:2204.05783.
13. Murthy, D., Allu, S., Andhavarapu, B., and Bagadi, M. (2020). Text based sentiment analysis using lstm. International Journal of Engineering Research and, 9(5), pp. 299-303.
14. Ramesh A. (2022). Algorithmic Analysis of Political Tweets: A Systematic Literature Review. Journal of Information and Knowledge, 59(4), pp. 185–195. https://doi.org/10.17821/srels/2022/v59i4/170345
15. Adritpal08 (2024). Dataset of indian politics tweets and reactions. Accessed: 2025-09-28.
16. Python | TextBlob.sentiment() method – GeeksforGeeks. https://www.geeksforgeeks.org/python/python-textblob-sentiment-method/. Accessed: 2025-09-28.

For Personal Use Only