Sentiment Analysis of Indian Political Tweets: A Comparative Study with LSTM and RNN Model

Om R Muddapur
School of Computer Science and Engineering
KLE Technological University, Hubli, India
01fe22bcs262@kletech.ac.in

Vidhi R Patel
School of Computer Science and Engineering
KLE Technological University, Hubli, India
01fe22bcs269@kletech.ac.in

Swati V Bhat
School of Computer Science and Engineering
KLE Technological University, Hubli, India
01fe22bcs273@kletech.ac.in

Chandanagouda H
School of Computer Science and Engineering
KLE Technological University, Hubli, India
01fe22bcs287@kletech.ac.in

Dr. Uday Kulkarni
School of Computer Science and Engineering
KLE Technological University, Hubli, India
uday_kulkarni@kletech.ac.in

Sunil Gurlahosur
School of Computer Science and Engineering
KLE Technological University, Hubli, India
svguralhosur@kletech.ac.in

***Abstract*—Sentiment analysis has become a key focus in computer science, especially with the rise of social media platforms like X (formerly Twitter), Reddit, Instagram, and Facebook. These platforms are now central to public conversations, including discussions on politics, generating massive amounts of data through tweets and comments. By analyzing this data, researchers can gain valuable insights into public opinion on important political topics, helping us better understand societal attitudes.The proposed work focuses on the sentiment analysis of Indian political tweets, with the objective of exploring public opinion on various political topics, leaders, and events. The study makes use of advanced natural language processing techniques, including feature extraction methods such as Term Frequency-Inverse Document Frequency(TF-IDF) and Word2Vec, for representing tweet text. To analyze these representations, Long Short-Term Memory (LSTM) models and Recurrent Neural Network (RNN) are used mainly because they capture sequential information as well as contextual information very well. According to the experimental results, TF-IDF embeddings, accompanied by LSTM and RNN models respectively, significantly outperform the Word2Vec model to classify sentiments at 83.02% and 81.06% . These findings highlight the effectiveness of LSTM with TF-IDF in sentiment classification tasks and provide a robust framework for analyzing political discourse on social media.**

***Index Terms*—Sentiment Analysis, Tweets, Politics, Machine Learning, Deep Neural Networks, Twitter, Social Networks.**

I. Introduction

The explosive growth of social media platforms such as Twitter, Reddit, Instagram and Facebook has fundamentally changed how people communicate, exchange information, and express their views [1]. These platforms have become integral to global conversations, enabling users to engage in real-time discussions on a wide range of topics [2]. Among them, Twitter stands out as a platform of choice for dynamic public discourse. Its simplicity, and accessibility allow users to express thoughts and opinions on various matters, including politics, a subject that consistently evokes strong reactions and vibrant debates [3].

Politics has always been a major topic in society. On X (formerly Twitter), people regularly discuss current affairs, express their opinions about policies, and assess the deeds and choices of political figures. This enormous number of conversations is a gold mine of public opinion, providing priceless insights into the opinions, worries, and goals of the population [4]. These insights can be used by analysts, researchers, and policymakers to better understand public sentiment, guide choices, develop governance plans, and even promote cross-border cooperation.

The problem lies in the current limitations of sentiment analysis research on social media platforms like Twitter. Most studies focus on specific events or use basic methods that classify opinions simply as positive or negative, missing the finer emotional details, layered trends, and complexities of public discussions. Additionally, many existing approaches fail to use advanced computational techniques to capture the context and subtleties of text data. This creates a gap in understanding the full spectrum of public sentiment, especially in the diverse and fast-evolving political landscape of India, where refined insights are crucial for informed decision-making and meaningful participation.

This study seeks to address these challenges and fill the research gaps by capturing refined sentiments by going beyond simple binary classifications to uncover the subtle layers of sentiment, emotional undertones, and evolving trends in political discourse. Leveraging advanced techniques by utilizing advanced text representation methods such as Term Frequency-Inverse Document Frequency (TF-IDF) [[5] , [6]] and Word2Vec [7] to extract meaningful features from textual data and applying deep learning models by employing Long Short-Term Memory (LSTM) [8] and Recurrent Neural Network (RNN) [9] to analyze sequential data, capturing contextual and temporal refinements in political tweets. Providing actionable insights to policymakers, researchers, and analysts a deeper understanding of public sentiment to inform decision-making and governance strategies.

The paper is structured to comprehensively address the objectives of the study, with Section II providing a detailed background study that reviews existing sentiment analysis techniques such as TF-IDF, Word2Vec [10], LSTM and RNN. This section also explores their applications in social media analysis, with a focus on prior research in political sentiment analysis, emphasizing the strengths and limitations of current approaches. Then in section III the proposed methodology is outlined, detailing the use of advanced text representation and deep learning techniques, such as TF-IDF for feature extraction, Word2Vec for contextual word embedding, and LSTM and RNN models for sequential data analysis. It also describes the process of data cleaning, preprocessing, and model training, ensuring accuracy and efficiency. And in section IV the experimental results are discussed in-depth, including a comparative analysis of the proposed methods using metrics like accuracy, recall, precision, and F1-score, accompanied by visualizations that provide insights into sentiment distribution and polarity trends. Then in the final section V the paper concludes with a summary of the key findings and their implications for understanding public sentiment in Indian politics,along with a discussion on potential future research directions, such as expanding the methods to other domains or exploring multimodal sentiment analysis.

II. Background Study

Several techniques have been developed for sentiment analysis, such as ensemble learning and machine learning, which are useful for improving the accuracy of sentiment classification across various domains, including politics, finance, and social media. Ensemble learning is majorly used for its adaptability and effectiveness in handling domain-specific challenges. For instance, [11] discusses its application with feature representations such as TF-IDF and Word2Vec, though it highlights limitations in addressing domain-specific challenges, particularly in politically charged datasets.

Machine learning approaches have also been extensively explored. [12] investigates Israeli political tweets, addressing challenges such as regional dialects and linguistic biases. Similarly, [13] employs ensemble methods for Arabic social media sentiment analysis, emphasizing the effectiveness of TF-IDF for sparse data, as elaborated in [14]. Word2Vec, another feature representation, captures semantic relationships between words, enhancing deep learning models like LSTM networks and RNNs, as demonstrated in [15].

Advanced neural network models, particularly LSTM and RNN, have proven highly effective for sequential data analysis. LSTMs excel in capturing long-term dependencies, making them ideal for tasks like sentiment analysis of long-form text. For example, [16] demonstrates the ability of LSTM to model temporal patterns, while [17] highlights its performance in analyzing sentiment within complex text structures.The architecture of an LSTM network, as detailed in [18], is shown in Figure 1. It visualizes the internal structure of a single LSTM cell, showcasing its gating mechanisms, including the input, forget, and output gates which regulate the flow of instructions.

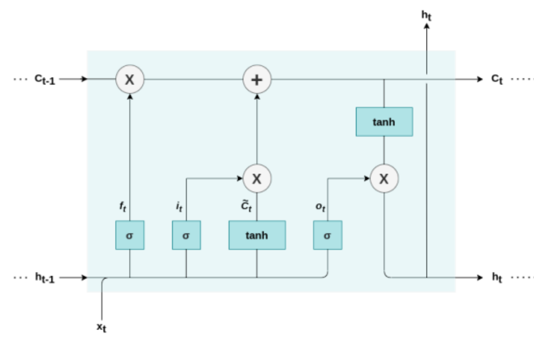

Fig. 1. The internal structure of a single hidden unit in an LSTM, visualizing the computation of h_t and C_t using an input x_t, the hidden state value of the previous unit h_{t-1}, and the cell state unit value of the preceding unit C_{t-1}.

RNNs, on the other hand, are well-suited for tasks involving shorter or fragmented text, such as tweets or reviews, due to their ability to model sequential dependencies. Studies like [19] and [20] emphasize the utility of RNNs in sentiment classification. The architecture of RNNs, as visualized in [21], is depicted in Figure 2. It highlights their sequential processing structure, where hidden states are passed from one time step to the next which enables RNNs to capture temporal patterns effectively

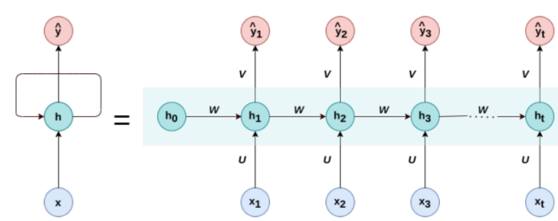

Fig. 2. A simple Recurrent Neural Network unfolded over t time steps, where U represents input-to-hidden weights, W represents hidden-to-hidden weights, and V represents hidden-to-output weights.

Despite these advancements, sentiment analysis of Indian political tweets remains underexplored. While techniques such as TF-IDF and Word2Vec have shown promise, their performance when paired with advanced models like LSTM. Further research could bridge this gap by evaluating and comparing the performance of different feature representations, such as TF-IDF and Word2Vec, when integrated with LSTM and RNN architectures to enhance sentiment analysis of Indian political tweets.

III. Proposed Work

To understand the sentiment behind political tweets in the Indian context, we need a thoughtful and structured approach. This section highlights the strategies and methods used to analyze and make sense of these sentiments effectively.Figure 3 illustrates the end-to-end workflow applied for sentiment analysis of Indian political tweets.

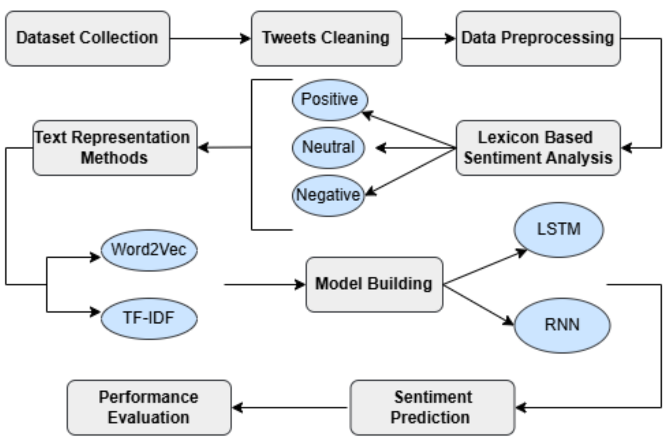

Fig. 3. Workflow: Step-by-step process for sentiment analysis of political tweets

A. Methodology and Implementation

This section provides a detailed overview of the models, methodologies, and their implementation strategies employed for analyzing sentiment in Indian political tweets. By delving into the computational techniques and tools used it aims to offer a comprehensive understanding of how sentiment analysis was conducted. The focus is on outlining the approaches used to process and interpret the vast amount of textual data, providing information about how these techniques were applied to extract meaningful insights from the public discourse on Indian politics.

1) Data Preprocessing: The dataset has been refined to enhance its suitability for analysis. This involved preparing plain text versions of the tweets by removing stopwords, extra whitespaces, usernames, and punctuation. Additionally, case folding was applied, duplicate entries were eliminated, and rows without tweets were removed. Irrelevant columns such as date and user details have been dropped to come up with a cleaner version of the dataset. The Natural Language Toolkit (NLTK) pre-trained 'punkt' English tokenizer was used to split longer tweets into individual words.

2) Lexicon Based Sentiment Analysis: The dataset was unlabeled, which consisted only of tweets without classification into positive, negative, or neutral categories. Sentiment analysis was performed to compute the sentiment in terms of the semantic orientation of words or phrases in the tweets.This was achieved using the python package TextBlob, which returns the polarity of the text. The polarity values range from -1 to 1, with negative sentiment at -1, neutral at 0, and positive at +1.

3) Text representation methods: In this subsection, we take a closer look at the techniques used to create word embeddings and evaluate the importance of words in a tweet.The methods include:

a) ***Word2Vec****:* In analyzing Indian political tweets, Word2Vec helps to represent a text in a way that captures the context and meaning of words, making it easier to identify subtle sentiments and topics in political discussions. Its ability to preserve the relationships between words allows machine learning models to perform better by utilizing the created word embeddings.

Figure 4 shows the Word2Vec architecture [22] applied to political tweets on the sample tweet "should we laugh or cry who is calling whom corrupt jokers of Indian politics." The tweet is tokenized and the Skip-gram approach is used to learn word embeddings by predicting the context of a target word. Each word is represented as a vector, capturing semantic relationships to reflect the meaning and context in political discourse.

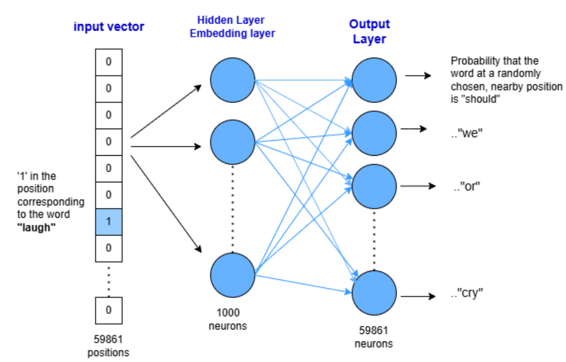

Fig. 4. Word2Vec Neural Network Architecture: Illustration of the model used to convert words into vector representations for semantic analysis.

b) ***TF-IDF****:* It is use to measure the importance of words in each tweet relative to the entire dataset. TF calculates how often a word appears in a tweet, using methods like raw frequency or logarithmic normalization. IDF identifies significant terms by reducing the weight of commonly used words (stop words) and emphasizing unique or less frequent words across tweets. This helps highlight key terms like political names and topics, enabling effective feature extraction for sentiment classification.

The TF-IDF weight of a term x in tweet y is:

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right) \tag{1}$$

where, $tf_{x,y}$ is the frequency of term x in tweet y, df_x is the number of tweets containing term x, N is the total number of tweets.

4) Model Building: The dataset is divided into two sets training set (80%) and testing set (20%). The training set is used to train the LSTM and RNN models, thus enabling them to learn from the data.

Both models take word embeddings as input. These word embeddings are generated using Word2Vec or TF-IDF. Each word in a tweet is represented as a vector, and the sequence of these word embeddings is passed through the RNN and LSTM models for sentiment analysis.

Algorithm 1 RNN Model

for $t = 1$ to T **do**
  $z_t \leftarrow Ux_t + Wh_{t-1} + b_h$ *Hidden State Computation*
  $h_t \leftarrow e(z_t)$ *Activation function* $(\tanh)$
  $o_t \leftarrow Vh_t + b_o$ *Output computation*
  $\hat{y}_t \leftarrow g(o_t)$ *Prediction using softmax*
end for
return $\{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_T\}$ =0

Algorithm 1 describes the working of the RNN model, where each word is processed sequentially over time steps $t = 1$ to T. At each step, the hidden state h_t is updated using the current word embedding x_t, the previous hidden state h_{t-1}, and a bias term with an activation function $(\tanh)$. The output o_t is computed from h_t using a weight matrix and bias, and the sentiment prediction $\hat{y}_t$ is obtained by applying the softmax function.

Algorithm 2 LSTM Model

Initialize parameters $W_i, W_f, W_o, W_g, b_i, b_f, b_o, b_g, W_{fc}, b_{fc}$
Process:
for $t = 1$ to T **do**
  $i_t \leftarrow \sigma(W_i[h_{t-1}, x_t] + b_i)$ *Input gate*
  $f_t \leftarrow \sigma(W_f[h_{t-1}, x_t] + b_f)$ *Forget gate*
  $o_t \leftarrow \sigma(W_o[h_{t-1}, x_t] + b_o)$ *Output gate*
  $\tilde{c}_t \leftarrow \tanh(W_g[h_{t-1}, x_t] + b_g)$ *Candidate values*
  $c_t \leftarrow f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$ *Cell state update*
  $h_t \leftarrow o_t \odot \tanh(c_t)$ *Hidden state update*
end for
$y \leftarrow W_{fc} \cdot h_T^{\text{drop}} + b_{fc}$ *Fully connected layer*
$y_{\text{pred}} \leftarrow \text{softmax}(y)$ *Prediction with softmax* =0

Algorithm 2 describes the working of LSTM model, where at each time step t, the input tweet is processed by updating the input gate i_t, forget gate f_t and output gate o_t. The candidate values $\tilde{c}_t$ are used to update the cell state c_t, while the hidden state h_t is updated to capture relevant context. The final prediction y_{pred} is made by feeding the hidden state through a fully connected layer followed by the softmax function.

The proposed LSTM model consists of an embedding layer, two Conv1D layers (32 and 64 filters, size 2) with ReLU activation, an LSTM layer for capturing word dependencies, and an attention layer for highlights key time steps. The GlobalMaxPooling1D layer reduces dimensions and the dense layer with 256 neurons extracts features. The output layer with 3 units and softmax activation gives the predicted class probabilities for three sentiment categories.

After training the model for 10 epochs, predictions are made using the unseen testing data based on the patterns learned. Both models use the same hyperparameters with an embedding dimension of 1000, a hidden dimension of 128, and a learning rate of 0.001.

IV. Experimental Results and Analysis

The dataset for the analysis of Indian political tweets in the study was obtained from Kaggle [23]. It contains 50,000 tweets along with information like posting date, user details, and engagement metrics such as likes and retweets. Figure 5 illustrates the distribution of polarity across the dataset after preprocessing, showcasing the sentiment breakdown of the tweets.

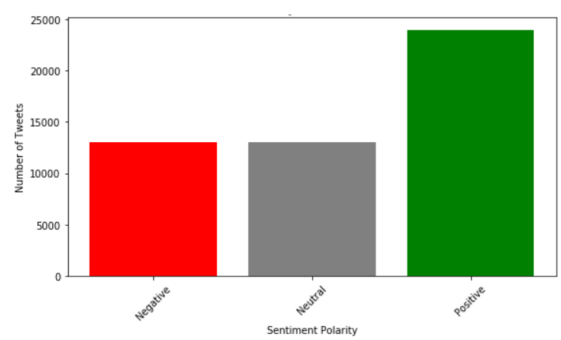

Fig. 5. Sentiment polarity vs Number of Tweets

The final training dataset includes about 24,000 positive tweets, 15,000 neutral tweets, and 13,000 negative tweets, which indicates most of the tweets are positive, followed by neutral and then negative.

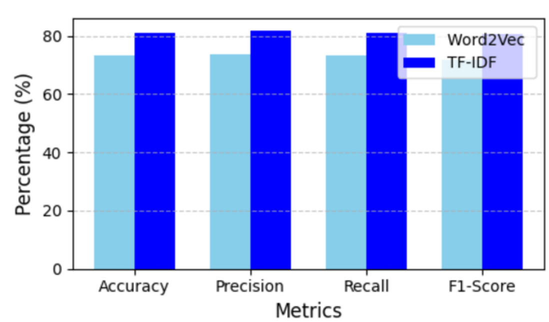

Fig. 6. Performance matrics of LSTM in %

Figure 6 shows how the LSTM model performs using TF-IDF and Word2Vec representations across different metrics. It is clear that the LSTM with TF-IDF performs much better. This suggests that TF-IDF captures the important features of the Indian political dataset more effectively. On the other hand, Word2Vec which creates distributed word representations, may not fully capture the specific details and context of this dataset.

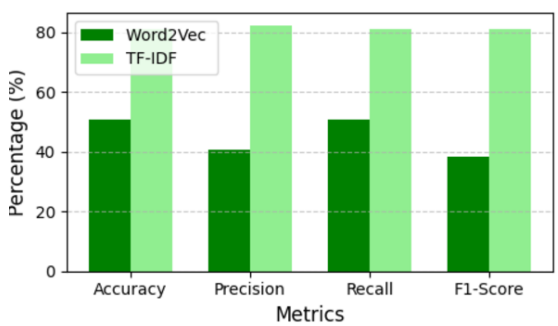

Fig. 7. Performance matrics of RNN in %

In Figure 7, the plot shows that the RNN model performs better with TF-IDF than with Word2Vec on all metrics. This suggests that, for the Indian political tweets dataset, TF-IDF is a better representation of features for sentiment analysis. With TF-IDF, it is easier for the RNN model to focus on important terms, which may not be the case if it uses Word2Vec. These results suggest TF-IDF as a more appropriate choice for this purpose.

TABLE I
TESTING ACCURACY OF MODELS WITH DIFFERENT TECHNIQUES

Models	**Accuracy (%)**	
	Word2Vec	**TF-IDF**
LSTM	73.08	83.02
RNN	50.7	81.06

As observed in Table I, the LSTM model with TF-IDF outperforms the same model with Word2Vec, achieving better performance across various metrics. This indicates that TF-IDF offers more effective feature representations for the Indian political tweets dataset when used with LSTM, highlighting its superiority over Word2Vec embeddings for this task. The enhanced performance of LSTM can be attributed to its ability to capture long-range dependencies in the text, which is essential for understanding the contextual and nuanced nature of political sentiments. Unlike RNNs, LSTMs are better equipped to retain important information across long sequences, making them more suitable for task like sentiment analysis of political tweets.

V. CONCLUSION

The research focuses on sentiment analysis of Indian politi cal tweets, comparing the performance of LSTM and RNN models. The LSTM model, paired with TF-IDF, achieves significantly higher accuracy (83.02%) compared to Word2Vec (73.08%), indicating TF-IDF is more effective for this dataset. Similarly, the RNN model performs better with TF-IDF (81.06%) than Word2Vec (50.7%), showcasing TF-IDF's ver satility. Notably, LSTM outperforms RNN, which shows that it is more suited to capture the sequential dependencies and contextual nuances of Indian political tweets. This consistency in superiority with both models suggests that TF-IDF captures the contextual importance of words more effectively, thus being a good choice for sentiment analysis.And the better ability of LSTM to capture complex word relationships further emphasizes its suitability for this task. This study therefore emphasizes the need to choose the right feature representation techniques to optimize model performance toward accurate predictions and better decision making for future political events.

VI. FUTURE WORK AND SCOPE

Future research in the sentiment analysis of Indian political tweets could focus on integrating more advanced models ca pable of handling multilingual data, which is crucial given the diversity of languages used in Indian political discourse. A sig nificant portion of tweets may not be in English, so developing robust models for regional languages would enhance accuracy and inclusivity. Additionally, incorporating features such as user metadata (e.g. political affiliation, location, and influence) could provide a richer context for sentiment prediction and improve model performance. Analyzing sentiment dynamics over time, particularly around key political events or elections, could offer deeper insights into public opinion trends and shifts. Another promis ing direction would involve improving the detection and interpretation of sarcasm, irony, and region-specific sentiment variations, which are prevalent in political discussions. Finally, addressing biases in sentiment classification models and en suring fairness in political sentiment analysis are essential for maintaining ethical standards and avoiding skewed interpreta tions of public sentiment.

REFERENCES

[1] C. Duncombe, "The politics of twitter: Emotions and the power of social media," *International Political Sociology*, vol. 13, pp. 409–429, 08 2019.

[2] S. Madakam and S. Tripathi, "Social media/networking: Applications, technologies, theories," *Journal of Information Systems and Technology Management*, vol. 18, 06 2021.

[3] C. Park, "Does twitter motivate involvement in politics? tweeting, opinion leadership, and political engagement," *Computers in Human Behavior*, vol. 29, p. 1641–1648, 07 2013.

[4] P. Research, "Political content on social media." https://www.pewresearch.org/political-content-social-media-2021/, 2021. [Online; accessed 30-Nov-2024].

[5] S. Qaiser and R. Ali, "Text mining: Use of tf-idf to examine the relevance of words to documents," *International Journal of Computer Applications*, vol. 181, 07 2018.

[6] C.-z. Liu, Y.-x. Sheng, Z.-q. Wei, and Y.-Q. Yang, "Research of text classification based on improved tf-idf algorithm," pp. 218–222, 08 2018.

[7] D. Jatnika, M. Bijaksana, and A. Ardiyanti, "Word2vec model analysis for semantic similarities in english words," *Procedia Computer Science*, vol. 157, pp. 160–167, 01 2019.

[8] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artificial Intelligence Review*, vol. 53, 12 2020.

[9] Z. Lipton, "A critical review of recurrent neural networks for sequence learning," 05 2015.

[10] S. Lei, "Research on the improved word2vec optimization strategy based on statistical language model," in *2020 International Conference on Information Science, Parallel and Distributed Systems (ISPDS)*, pp. 356–359, 2020.

[11] M. S. Başarslan and F. Kayaalp, "Sentiment analysis with ensemble and machine learning methods in multi-domain datasets," *Turkish Journal of Engineering*, vol. 7, no. 2, pp. 141–148, 2023.

[12] A. Gangwar and T. Mehta, "Sentiment analysis of political tweets for israel using machine learning," *arXiv preprint arXiv:2204.06515*, 2022.

[13] N. Hicham, S. Karim, and N. Habbat, "Customer sentiment analysis for arabic social media using a novel ensemble machine learning approach," *Int. J. Electr. Comput. Eng*, vol. 13, no. 4, p. 4504, 2023.

[14] S. Qaiser and M. Ali, "Text mining: Use of tf-idf to examine the relevance of words," *International Journal of Computer Applications*, vol. 181, no. 1, pp. 25–29, 2018.

[15] D. Jatnika and A. A. A. Setiawan, "Word2vec model analysis for semantic word similarities," *Indonesian Journal of Computer Science*, vol. 12, no. 3, pp. 45–57, 2020.

[16] A. R. Paduri *et al.*, "Stock price prediction using sentiment analysis and deep learning for indian markets," *arXiv preprint arXiv:2204.05783*, 2022.

[17] D. Murthy, S. Allu, B. Andhavarapu, and M. Bagadi, "Text based sentiment analysis using lstm," *International Journal of Engineering Research and*, vol. V9, 05 2020.

[18] H. Darji, "Investigating sparsity in recurrent neural networks," 2021. Unpublished.

[19] L. Kurniasari and A. Setyanto, "Sentiment analysis using recurrent neural network," in *Journal of Physics: Conference Series*, vol. 1471, p. 012018, IOP Publishing, 2020.

[20] M. Thomas and D. C A, "Sentimental analysis using recurrent neural network," *International Journal of Engineering and Technology(UAE)*, vol. 7, pp. 88–92, 08 2018.

[21] J. Stier, H. Darji, and M. Granitzer, "Experiments on properties of hidden structures of sparse neural networks," 2021.

[22] J. Zhang, "Skip-gram neural network from scratch," Oct 2019. Accessed: Nov. 30, 2024.

[23] Adritpal08, "Dataset of indian politics tweets and reactions," 2024. Accessed: 2024-12-24.