



A Transfer Learning Framework for Sentiment Analysis in Indian Vernaculars

Kunal Kumar^{1,*} Shivam Kumar¹

¹ Mizoram University, Aizawl-796004, India

Emails: kunal9900fice@gmail.com · kingshivam854@gmail.com

Received: October 15, 2023 Revised: January 22, 2024 Accepted: April 17, 2024 ★ Corresponding author

ABSTRACT

This paper explores sentiment analysis in Indian languages through a deep learning approach, combining machine learning techniques with natural language processing (NLP). Three neural network architectures—CNN, LSTM, and GRU—are employed to construct sentiment analysis models. Additionally, transfer learning is utilized via FastText, MURIL, and IndicBERT embeddings. The models are trained and evaluated on a translated dataset derived from the Sentiment140 dataset from Kaggle. Performance metrics such as accuracy, precision, recall, and F1-score are used to evaluate the models. The study addresses the challenges of sentiment analysis in Indian languages by leveraging deep learning techniques and linguistic diversity, providing insights into sentiment analysis across diverse languages and cultures. Furthermore, this project extends its analysis to include work on Gujarati, Marathi, and Sindhi languages, contributing to the understanding of sentiment analysis in a broader spectrum of Indian languages.

Keywords: Sentiment Analysis ▪ Convolutional Neural Network (CNN) ▪ Long Short-Term Memory (LSTM) ▪ Gated Recurrent Units (GRUs) ▪ Deep Learning

1. INTRODUCTION

1.1 Sentiment Analysis

Sentiment analysis, a key domain in Natural Language Processing (NLP), is vital for discerning subjective information and emotions from text. Its significance extends to gauging public opinion, analyzing customer feedback, and evaluating social media sentiments. This field employs computational methods to categorize text as positive, negative, or neutral, providing valuable insights into user attitudes. Applications span diverse sectors, including marketing, brand management, and political analysis. Sentiment analysis techniques leverage machine learning algorithms, lexicon-based approaches, and deep learning models to decode the nuanced sentiment expressed in text, contributing to a comprehensive understanding of user sentiments across various platforms and communication channels [1, 2].

1.2 Problem Statement and Challenges

As models already exist to analyse the sentiments of comments in English and other major languages of the world, there is no comparable model created so far to perform the same task for many Indian languages. Therefore, this work creates such models for Indian languages by implementing techniques that have previously produced successful results. Sentiment analysis in Indian languages poses unique challenges due to the linguistic diversity and syntactical variations across the languages spoken in India. Unlike sentiment analysis in English, where a substantial amount of research and resources are available, sentiment analysis in Indian languages is relatively under-explored.

Some of the primary challenges include the lack of labeled datasets, because building accurate sentiment analysis models requires large labeled datasets; translation accuracy, because

translating text from English to Indian languages introduces varying sentence structures, idiomatic expressions, and linguistic nuances; and linguistic diversity, because India is home to numerous languages, each with its own linguistic characteristics, cultural contexts, and sentiment expressions. Developing models that can generalize across this linguistic diversity requires careful consideration and adaptation.

1.3 Objectives

The objectives of this paper are twofold. First, it addresses the challenges of sentiment analysis in Indian languages, including the lack of labeled datasets, translation accuracy, and linguistic diversity, by developing effective methods that bridge the gap between sentiment analysis in English and sentiment analysis in Indian languages. Second, it evaluates the performance of different deep learning architectures and transfer learning techniques in capturing sentiment across diverse Indian languages, thereby providing valuable insights into their suitability and effectiveness for sentiment analysis in varied linguistic contexts.

2. METHODS FOR SENTIMENT ANALYSIS

Sentiment analysis can be approached through various methods, and in this paper a deep learning-based approach is employed to tackle sentiment analysis in Indian languages. The following methods and techniques are utilized.

2.1 Neural Network Architectures

Three different neural network architectures, namely Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU), are selected for building the sentiment analysis models. Each architecture possesses unique characteristics that enable it to capture and analyze sentiment in textual data effectively.

CNN architecture utilizes convolutional layers to detect local patterns and features in text data. It excels in identifying important textual cues for sentiment classification, making it particularly suitable for capturing sentiment-related information. LSTM, a type of recurrent neural network (RNN), is capable of modeling long-term dependencies in sequential data. By incorporating memory cells and gating mechanisms, LSTM can effectively capture the contextual information necessary for sentiment analysis. GRU, also an RNN variant, is similar to LSTM but has a simpler structure with fewer parameters. It can capture dependencies in sequential data while being computationally efficient, making it a valuable choice for sentiment analysis tasks.

2.2 Transfer Learning with Word Embeddings

To enhance model performance and capitalize on existing knowledge, transfer learning techniques are employed using pre-trained word embeddings. Specifically, three widely used word embedding methods, FastText, MURIL, and IndicBERT, are utilized to generate embeddings tailored to Indian languages. These embeddings serve as a foundational source of linguistic information and semantic relationships, enabling sentiment analysis models to better grasp the context and sentiment expressed in Indian languages.

FastText, developed by Facebook AI Research, is a library renowned for its efficiency in text classification and repre-

sentation learning. By utilizing character n-grams to handle word variations, FastText offers fast and scalable solutions, particularly suited for sentiment analysis tasks, especially on large-scale datasets. Moreover, FastText provides pre-trained models for multiple languages, enhancing its applicability.

MURIL, a derivative of the BERT architecture, is a language model trained on a vast corpus of diverse Indian language texts. Its specialized component, IndicBERT, focuses on comprehending and generating content in languages like Hindi, Bengali, Tamil, Telugu, among others. MURIL empowers various natural language processing tasks, including sentiment analysis, language understanding, machine translation, and named entity recognition, thanks to its adaptability and understanding of linguistic nuances across Indian languages.

IndicBERT, another BERT-based architecture, is trained on a large corpus of text from diverse domains in Indian languages. Like MURIL, IndicBERT excels in understanding and generating text in languages such as Hindi, Bengali, Tamil, and Telugu. Its versatility makes it suitable for various natural language processing tasks, including sentiment analysis, language understanding, machine translation, and named entity recognition. By leveraging these pre-trained word embeddings, sentiment analysis models can effectively transfer linguistic knowledge, thereby improving their ability to analyze sentiment in Indian languages proficiently.

3. RELATED WORKS

The research on sentiment analysis (SA) in Indian languages has been extensive, with significant work done in languages like Hindi, Bengali, Tamil, Malayalam, and Urdu, among others [3, 4, 5]. In Hindi, researchers have used various approaches, including lexicon-based methods, machine learning techniques, and deep learning models such as convolutional neural networks. They have achieved accuracies ranging from 64% to 87% in different studies. Similar efforts have been made in Bengali, Tamil, and Malayalam, with accuracies ranging from 70% to 98.7% using techniques such as SVM, NB, and RNN.

In languages with minor research work such as Punjabi, Oriya, Nepali, Marathi, Konkani, and Manipuri, the focus has been on developing lexicons, using rule-based and machine learning approaches, and achieving accuracies up to 93.6%. For example, in Nepali, researchers have developed the Nepali Sentiment WordNet and compared it with machine learning approaches like NB, LR, and SVM, showing the superiority of machine learning techniques. Overall, research in sentiment analysis in Indian languages has shown promising results, with researchers exploring various approaches to address the unique challenges posed by these languages. However, there is still room for improvement, particularly in languages with minor research work, where more efforts are needed to develop robust sentiment analysis systems [6].

4. ARCHITECTURE

The investigation initiated with the acquisition of the Sentiment140 dataset, which was subsequently translated into various regional languages using the Google Translate API. Following translation, the dataset underwent a series of pre-processing steps, encompassing tokenization, stop-word elim-

ination, and normalization, to ensure its suitability for analysis. For the sentiment analysis models, the study meticulously opted for three distinct neural network architectures: CNN, LSTM, and GRU, chosen for their established effectiveness in natural language processing tasks. To augment the models' proficiency in understanding sentiment nuances in regional languages, FastText and MURIL word embedding models were incorporated for transfer learning. The preprocessed and embedded dataset was subsequently employed to train and evaluate the CNN, LSTM, and GRU models. Lastly, the models' efficacy was comprehensively assessed using precision, accuracy, F1-score, and recall metrics to gauge their performance in capturing sentiment across diverse regional languages.

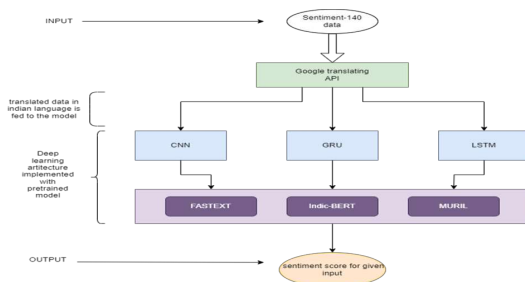


Figure 1. Proposed system.

5. METHODOLOGY

5.1 Dataset Description

Sentiment140 is a widely used dataset for sentiment analysis. It consists of 1.6 million tweets labeled with positive or negative sentiment. Each tweet is annotated based on the presence of positive or negative emotions. The dataset is balanced and commonly employed as a benchmark for sentiment analysis tasks. It helps researchers evaluate and compare sentiment analysis models, particularly for short, informal text such as tweets. However, the dataset lacks neutral labels and may contain noise inherent to Twitter data. Despite its limitations, Sentiment140 serves as a valuable resource for training and evaluating sentiment analysis models in social media contexts.

5.2 Preprocessing Methods

First, the dataset is translated into Indian languages using the Google API, and then further preprocessing steps are applied to the data. In the Sentiment140 dataset, typical data preprocessing steps include removing noise such as HTML tags and URLs, tokenization, lowercasing, stop-word removal, handling emoticons and emoji, expanding abbreviations and contractions, lemmatization or stemming, handling negations and context, removing rare or frequent words, and data balancing. These steps help improve data quality, reduce noise, and enhance sentiment analysis model performance.

5.3 Model Selection and Description

For this paper, three different model architectures are selected: CNN, LSTM, and GRU. Each model is trained on the translated Indian language dataset.

- **CNN:** The CNN model employs convolutional layers to capture local patterns in the text data. It learns to

recognize important features for sentiment classification.

- **LSTM:** The LSTM model is a type of recurrent neural network that can effectively model long-term dependencies in sequential data. It is capable of capturing contextual information in the text.
- **GRU:** The GRU model is another variant of the RNN architecture that can capture dependencies in sequential data. It has fewer parameters than LSTM, making it computationally efficient.

5.4 Model Parameters

The model parameters are set as follows. The CNN model consists of multiple convolutional layers followed by fully connected layers. The number of filters, filter sizes, and pooling techniques are tuned for optimal performance. The LSTM model comprises multiple LSTM layers with adjustable hidden units and dropout rates. The output of the LSTM layers is fed to a dense layer for sentiment classification. The GRU model has similar parameters to LSTM, including the number of hidden units and dropout rates.

6. EXPERIMENTAL SETUP

6.1 Experimental Environment

The paper used Google Colab, leveraging its GPU resources for training deep learning models. The environment was set up with TensorFlow, Keras, and scikit-learn. The dataset, translated into regional languages, was preprocessed and loaded into Colab. Three architectures (CNN, LSTM, GRU) were used with FastText and MURIL embeddings. Training and validation sets were created, and models were trained on the GPU. Model performance was evaluated with metrics like accuracy and F1-score. The Colab setup included an Intel Xeon CPU with 2 vCPUs and 13GB RAM.

7. RESULT AND DISCUSSION

7.1 Performance Evaluator

The performance of sentiment analysis models is evaluated using various metrics, such as accuracy, precision, recall, and F1-score, which provide valuable insights into their predictive effectiveness. These metrics include True Positive (TP), representing the accurate classification of positive reviews; True Negative (TN), denoting the correct identification of negative reviews; False Positive (FP), indicating the incorrect labeling of negative reviews as positive; and False Negative (FN), referring to the mistaken categorization of positive reviews as negative.

Precision measures the ratio of correctly identified positive samples to the total predicted positive samples:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (1)$$

Recall, also known as sensitivity, quantifies the ratio of actual positive instances to the total positive instances:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

The F1-score represents the harmonic mean of precision and

recall:

$$F1 = \frac{2(\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}. \quad (3)$$

Accuracy signifies the ratio of correct classifications to total predictions:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \quad (4)$$

7.2 Results

Table 1. Gujarati results using FastText and MURIL embeddings.

Embedding	Model	Precision	Recall	Accuracy	F1
FastText	CNN	77.60	77.88	77.56	77.74
FastText	LSTM	75.48	82.19	77.61	78.69
FastText	GRU	80.21	77.16	78.93	78.65
MURIL	CNN	57.11	62.28	57.50	59.59
MURIL	LSTM	60.52	39.56	56.61	47.84
MURIL	GRU	54.83	51.72	57.12	54.83

Table 2. Marathi results using FastText and MURIL embeddings.

Embedding	Model	Precision	Recall	Accuracy	F1
FastText	CNN	85.15	58.18	74.07	69.13
FastText	LSTM	76.65	79.21	77.59	77.91
FastText	GRU	77.21	78.31	77.65	77.76
MURIL	CNN	57.36	41.67	55.40	48.28
MURIL	LSTM	58.99	25.39	53.92	35.50
MURIL	GRU	56.74	36.74	54.42	44.58

Table 3. Sindhi results using FastText and MURIL embeddings.

Embedding	Model	Precision	Recall	Accuracy	F1
FastText	CNN	58.67	96.38	64.25	72.94
FastText	LSTM	76.76	72.45	75.26	74.55
FastText	GRU	75.93	77.56	76.49	76.74
MURIL	CNN	53.07	67.09	53.83	59.26
MURIL	LSTM	52.59	61.93	53.00	56.88
MURIL	GRU	55.49	32.91	53.26	41.31

7.3 Discussion

This study explores the effectiveness of transformer-based learning using pre-trained models (FastText, MURIL, and IndicBERT) for sentiment analysis in Indian languages. By implementing these models with deep learning architectures such as CNN, GRU, and LSTM, the study advances the understanding of sentiment analysis in linguistically diverse contexts. The findings suggest that transformer-based learning with pre-trained models can significantly improve sentiment analysis in Indian languages.

These models excel in capturing the nuances and complexities of sentiment, showcasing their potential for applications requiring sentiment analysis in diverse linguistic environments. Overall, this research contributes to the field of sentiment analysis by demonstrating the efficacy of advanced deep learning techniques and transfer learning in addressing the challenges of sentiment analysis in Indian languages. The study underscores the importance of leveraging pre-existing linguistic

resources and advanced models for more accurate sentiment analysis across different languages and cultures.

8. CONCLUSION

This paper outlines a comprehensive approach to conducting sentiment analysis in Indian languages. By employing machine learning techniques, translation methods, and diverse model architectures, sentiment analysis can be effectively carried out in non-English languages. The evaluation of model performance using metrics like accuracy, precision, recall, and F1-score illustrates the suitability of these models for sentiment analysis tasks. Through this endeavor, our aim is to contribute to the advancement of sentiment analysis in Indian languages and foster the development of precise NLP models tailored for non-English languages. The paper underscores the significance of linguistic diversity and lays the groundwork for sentiment analysis applications that cater to a broad spectrum of languages and cultures.

FUNDING

This research received no external funding.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

- [1] S. Rani and P. Kumar, "A journey of indian languages over sentiment analysis: A systematic review," *Artificial Intelligence Review*, vol. 52, pp. 1415–1462, 2019.
- [2] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artificial Intelligence Review*, vol. 55, pp. 5731–5780, 2022.
- [3] S. Rani and P. Kumar, "A journey of indian languages over sentiment analysis: A systematic review," *Artificial Intelligence Review*, 2018.
- [4] R. S. Patil and S. R. Kolhe, "Supervised classifiers with tf-idf features for sentiment analysis of marathi tweets," *Social Network Analysis and Mining*, 2022.
- [5] A. Saroj, A. Thakur, and S. Pal, "Sentiment analysis on hindi tweets during covid-19 pandemic," *Computational Intelligence*, 2023.
- [6] D. Chakraborty, D. Rudrapal, and B. Bhattacharya, "A study on the research progress of multimodal sentiment analysis in indian languages," in *Chapter 17*. Springer Science and Business Media LLC, 2023.