# Detecting Abusive Comments in Multiple Languages: A Multilingual Approach

Sai Krishna Sangeetha
Computer Science
University of Central Florida
Orlando, Florida, United States
saikrishna.sangeetha@knights.ucf.edu

Venkata Sai Kumar Ganesula
Computer Science
University of Central Florida
Orlando, Florida, United States
venkatasaikumarg@knights.ucf.edu

## ABSTRACT

India is a diverse country with a rich linguistic heritage. There were 122 major languages and 1599 other languages officially spoken in India, according to 2011 Census of India. The precise number can vary depending on the standards used to define a distinct language, but it is important to keep in mind that some of these "languages" may be dialects or variations of the same language. Social media is heavily used by people in a world where internet access is expanding. As a result, there is more offensive, abusive, and hateful content on social media. For applications like controversial event extraction, creating AI chatterbots, content recommendation, and sentiment analysis, hate speech detection on Twitter is essential. The ability to categorize a tweet as racist, sexist, or neither is how we define this task. This task is very difficult because natural language constructs are so complex. Our project aims to utilize natural language processing techniques to identify negative comments posted on social media platforms. Once we have identified these negative comments, we can focus on promoting positive content on these platforms. This approach will help us create a more positive and constructive online community.

## 1 Problem Statement

The objective of this project is to create a model that can determine whether a comment is abusive. The dataset is made up of comments written in different Indian dialects. For the best accuracy on the dataset, we want to investigate and contrast various machine learning and deep learning techniques in this project.

## 2 Data Analysis

The dataset for this task is curated for an ongoing challenge hosted by IEEE BigMM. The dataset consists of abusive and Non-abusive comments which were posted on Moj (one of India's largest short-video apps) in 10+ languages accompanied by contextual user data. (https://www.kaggle.com/c/iiitd-abuse-detection-challenge/data). We plotted the number of datapoints available for each language in the dataset and we observe that Hindi has the highest number of datapoints and Telugu has the second highest and there are some languages that have very few datapoints compared to these languages.

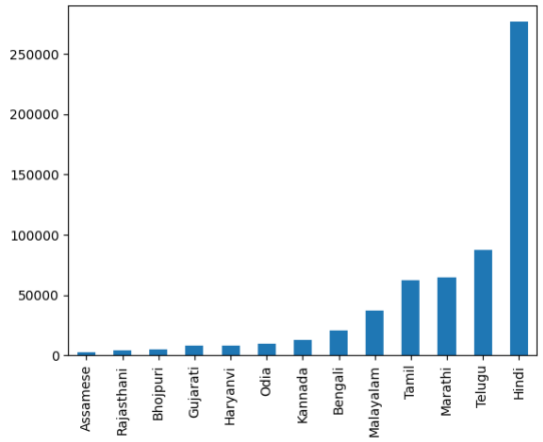


**Figure 1: Language Distribution**

## 4 Models

### 4.1 Model 1

In our project, we are tokenizing comments in Indian languages, and to accomplish this task, we have selected the Indic tokenizer. The reason for this choice is that traditional tokenizers available in Python libraries such as spaCy and NLTK may not be well-suited to handle Indian languages. The Indic NLP library, which was specifically designed for Indian languages, provides us with a solution to this problem.

To generate word vectors from the tokenized data in our project, we used the Tf-idf Vectorizer. This method enables us to assess the originality of each word and generate appropriate word vectors.

In preparation for fitting the Logistic Regression model to our data, we begin by computing the ratio, using Bayes' rule. This ratio is then multiplied by the word vectors to gain insight into the likelihood of a given datapoint generating a label of 1 or 0. Specifically, if the ratio is greater than 1, the model is more likely to predict a label of 1, whereas if the ratio is less than 1, the model is more likely to predict a label of 0. This approach allows us to

make more informed predictions and improve the accuracy of our model.

## 4.2 Model 2

Word embeddings have been shown to be an effective method for achieving high accuracy in classification tasks. In our approach, we trained an embeddings layer on the training dataset to enable prediction on the testing dataset.

*4.2.1 Model 2a*

To prepare the data for embedding generation, we first removed punctuation and tokenized the text. We then built a vocabulary based on the training set to enable uniform input size for the embedding layer. The embeddings were then passed through a fully connected neural network for prediction.

*4.2.2 Model 2b*

This model builds upon Model 2a and makes some key modifications to improve its performance. To replicate the output of GloVe embeddings, we increased the output dimension of the embeddings layer from 8 to 100. This change allowed the model to learn more effectively and led to an increase in accuracy.

*4.2.3 Model 2b*

This model builds upon Model 2a and incorporates more advanced deep learning techniques such as Bidirectional LSTM and Batch normalization. We were able to create a more complex model that achieved higher accuracy than Model 2a and similar accuracy to Model 2b.

## 5 Initial Results

Plotting the classification report for validation dataset based on Model 1.

```
Classification report for Validation dataset
              precision    recall  f1-score   support

 Non-Abusive       0.91      0.84      0.87     38268
     Abusive       0.80      0.89      0.84     28237

    accuracy                           0.86     66505
   macro avg       0.86      0.86      0.86     66505
weighted avg       0.87      0.86      0.86     66505
```

**Result 1: Classification report for Validation Dataset**

```
Classification report for Validation dataset of Hindi Langauage
              precision    recall  f1-score   support

 Non-Abusive       0.89      0.84      0.86     16276
     Abusive       0.83      0.88      0.86     14573

    accuracy                           0.86     30849
   macro avg       0.86      0.86      0.86     30849
weighted avg       0.86      0.86      0.86     30849
```

**Result 2: Classification report for Validation dataset of Hindi Language**

Our model performs well on the given data, we achieve a fairly good validation score. Validation evaluation metrics were calculated for Hindi, which had the highest number of datapoints, and for Assamese and Rajasthani, which had the lowest number of datapoints. It was observed that the model performed equally well for both cases, indicating that the model was able to learn based on similarities such as scripting and phonology between Indian languages.

Plotting the classification report for validation dataset based on Model 2a, Model 2b, Model 2c.

```
1871/1871 [==============================] - 3s 1ms/step
Classification report for Model 2a:
              precision    recall  f1-score   support

           0       0.82      0.87      0.84     31758
           1       0.84      0.79      0.81     28096

    accuracy                           0.83     59854
   macro avg       0.83      0.83      0.83     59854
weighted avg       0.83      0.83      0.83     59854
```

**Result 3: Classification report for Model 2a**

```
1871/1871 [==============================] - 6s 3ms/step
Classification report for Model 2b:
              precision    recall  f1-score   support

           0       0.79      0.93      0.85     31758
           1       0.90      0.71      0.79     28096

    accuracy                           0.83     59854
   macro avg       0.84      0.82      0.82     59854
weighted avg       0.84      0.83      0.82     59854
```

**Result 4: Classification report for Model 2b**

```
1871/1871 [==============================] - 6s 3ms/step
Classification report for Model 2c:
              precision    recall  f1-score   support

           0       0.81      0.89      0.85     31758
           1       0.86      0.77      0.81     28096

    accuracy                           0.83     59854
   macro avg       0.84      0.83      0.83     59854
weighted avg       0.84      0.83      0.83     59854
```

**Result 5: Classification report for Model 2c**

## 6 Future Scope

We will be using MURIL to generate word embeddings for the comments data. Before using MURIL, we will need to tokenize the data into a certain format which will be achieved by using AutoTokenizer. We will be choosing MURIL because it is a BERT model pretrained on huge data from Indian languages and will generate very meaningful embeddings for our case. We will be using a fully connected Neural Network for prediction of labels based on generated embeddings. The proposed model will have the capability to be trained on a subset of languages and can be utilized for identifying whether a comment is abusive or non-abusive in other languages. For example, the model can be trained on a dataset containing Hindi and Tamil comments and can be tested on a dataset containing comments in other languages such as Telugu or Bengali.

## REFERENCES


[1] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, Vasudeva Varma et al. “Deep Learning for Hate Speech Detection in Tweets.”

[2] Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, Partha Talukdar et al. “MuRIL: Multilingual Representations for Indian Languages.”

[3] "Detecting Hate Speech in Social Media Using Deep Learning Techniques" by N. Pandya, D. Bhatt, and D. Doshi.

[4] IndicBERT: A Multilingual Language Model for Indian Languages" by Divyanshu Kakwani, Himanshu Sharma, Prakhar Gupta, Abhishek Kumar, and Manish Shrivastava.