Classification of Tweets in Hindi Language

Akash Rawat
MT21005

Parul Sikri
MT21065

Shubham Rana
MT21092

Abstract

Social media plays an essential role in today's life. It is an online platform that brings people together. It enables them to share their thoughts, views, and opinions. Today's life is unimaginable without social media. It has become part and parcel of our day-to-day life. Twitter is one such helpful social media site. The following article analyzes the Twitter post written in the Hindi Language. We collected the data-set which consists of around 8000 tweets. These tweets cover four hostility dimensions-: hate, offensive, defamation, and fake tweets, and one non-hostile label. The hostile tweets can be multi-labels as they can share common attributes among themselves. In this article, we have shared various classification techniques to solve the problem of the classification of tweets.

1 Introduction

In the recent Covid-19 pandemic situation, the people were restricted to their homes. The various social media platforms were the only hope for them to connect with each other and share information. So, there was a significant number of new users on these platforms. Twitter is currently home to 22.1 million users in India. It is considered a valuable platform for companies to advertise their products and make users aware of their brand. Also, some companies use Twitter as a medium to provide support to their customers and receive valuable consumer feedback and suggestions to make their services better. It also serves as a great source of news. It not only acts as a powerful means of communication but also shapes the thinking of an individual. It is a flexible, open, and multi-use platform that grabs people's attention all across the globe.

However, with the good part of Twitter comes its bad part. In 2018, Amnesty International, an organization that works for human rights, regarded Twitter as a ”toxic place for women” based on its analysis of almost 15 million tweets. According to the findings of Amnesty International, it turned out that women using Twitter often become the target of online abuse. Online abuse in hostile tweets, troll messages, death, and rape threats primarily impacts women's mental health, making Twitter an unsafe and toxic place for women. As a result, this also affects the participation of women in keeping their views on Twitter. It also leads to an increase in levels of stress and anxiety among women who face online abuse. Twitter fails when it comes to racist abuse. Often black people suffer from racist abuse on Twitter. When Black English footballers lost to Italy in a UEFA European Football Championship, many racist tweets were made against the players. There is no doubt that social defamation can destroy one's personal and professional relationships. Surprisingly, Twitter is one such social media platform where one can expect to find many defamation cases. Defamation cases and false statements made about a business or a person not only damages one's reputation but also negatively impact one's work life. It can make people lose their job and mental peace. According to a study, false news travels faster than a true story on Twitter. The spread of false statements on Twitter can have harmful consequences. During the lockdown, Twitter served as one of the virtual platforms to learn information about Covid-19. But, the false and misleading information on various social media often led to false beliefs and panic among the audience. Twitter serves as a residence for online abuse, hostile communication, defamation, and fake news. It is a matter of concern that needs special attention. Thus, there is a dire need to stop abusive, offensive, defamatory, and fake tweets on Twitter. It is crucial to identify all such tweets because not all negative tweets are hostile, harsh, or objectionable. We will try to develop a model that labels Hindi tweets as offensive, hate, defamation,non-hostile or fake(multi-label classification). Hindi is the most commonly spoken language in India . Not much work has been previously done on Hindi tweet classification. Our model will aid to achieve the main goal of controlling abusive tweets and fake news on Twitter (India).

Here are the meanings of the labels assigned to a tweet:

- **Offensive:** Offensive tweets are those which try to insult a person or a business using disrespectful ,hurtful and rude words.

- **Hate:** Hate tweets are those which show hatred towards a person or a specific group on the grounds of their race,religious beliefs,etc.

- **Defamation:** Defamation tweets are those which try to destroy/damage the reputation of an individual or a group in public.

- **Fake:** Fake tweets are those which are false or not genuine.

- **Non-hostile:** Non-hostile tweets are those which do not involve any hostility.

2 Literature Survey

For the baseline models, the different problems can be reduced to the classification of tweets in the Hindi language and the problem can be reduced as the multi-label classification problem. For each baseline, the F1 score is taken as the primary performance measure to compare the variety of baseline models.

Mohit Bhardwaj et al. proposed a multilingual Bert model for hostility detection dataset in Hindi using baseline models such as Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Logistic Regression (LR). The authors have used the Bert model for embedding the features on the baseline models. The author compared the models using the F1 score. Upon evaluation, they found that SVM performs best among all the other models with an F1 score of 84.11%.

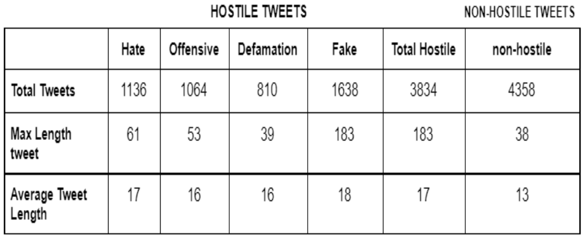
Neeraj Vashistha et al. proposed a multilingual hate speech detection in Hindi and English using baseline models like logistic regression for each language and Bert models. The author presented the TF-IDF as a feature extraction technique. Upon evaluation, the author found that the accuracy using logistic regression is 95% for the Hindi language and using the Bert model, the accuracy was the same 95%.

Vikas Kumar Jha et al. proposed a classification of offensive tweets in the Hindi language using feature extraction models: Word2vec and fastText. Word2vec is a neural network model used to vectorize the textual representation of data. fastText is the extension of Word2vec in which it calculates the vector by summing up all the characters in word by giving them n-gram weight. Upon evaluation of the models, the best precision achieved by the fastText model is 0.922.

Satyajit Kamble et al. proposed Hate Speech Detection from Code-mixed Hindi-English Tweets Using Deep Learning Models. The author presented a Qualitative Evaluation of domain-specific word embeddings using deep learning models like LSTM, CNN, etc. Upon evaluation of the models, the best working model with accuracy 82.62% is CNN-1D.

Sawinder Kaur et al. proposed a multi-level ensemble approach using the baseline models like Logistic Regression, Naïve Bayes, Decision Trees, and Support Vector Machines based on a voting mechanism. The authors compared the proposed model's accuracy using three feature extraction techniques: TF-IDF and Count-Vectorization. Upon model evaluation, TFIDF outperformed the Count-Vectorization feature extraction technique.

Ojasv Kamal et al. proposed a multilingual Bert embedding through which all the input sentences were embedded into vectors. The authors compared the proposed model's accuracy using MLC: Multi-Label Classification, MTL: Multitask Learning, BC: Binary Classification and AUX: Auxiliary Model. Upon model eval-



	HOSTILE TWEETS					NON-HOSTILE TWEETS
	Hate	Offensive	Defamation	Fake	Total Hostile	non-hostile
Total Tweets	1136	1064	810	1638	3834	4358
Max Length tweet	61	53	39	183	183	38
Average Tweet Length	17	16	16	18	17	13

Figure 1: Dataset Statistics and Label distribution

uation, AUX and Indic Bert embedding outperformed all other techniques.

3 Exploratory Data Analysis

The dataset being used for this project is taken from the CONSTRAINT 2021 contest. The Training set, Validation set, and Test set all are taken from CONSTRAINT 2021. The given dataset consisted of tweets, and the multi-class labels suggesting whether the tweet is fake, hate, offensive, defamation, or non-hostile.

3.1 Class Distribution

We observed that the samples were evenly distributed between the classes and had no class imbalance over the dataset on analyzing the dataset. The brief statistics of the dataset are shown in Figure 1. Out of the total 8192 tweets, 4358 belong to the non-hostile class, while the rest 3834 tweets belong to one of the hostile dimensions. For the hostile dimensions, 1136 belong to hate, 1064 to offensive, 810 to defamation, and 1638 tweets belong to the fake class.

3.2 Tweet Length Analysis

On analysing the length of tweets based on their label in the training set it was observed that the tweets which were labelled as ‘real’ were of longer length than the tweets which were labeled as ‘fake’. This has been pictorially represented in the Figure 2. The mean length of the real tweets is around 215 characters while the mean length of the fake tweets is around 144 characters. Even there is a significant difference between the standard deviation of the length of the real tweet and length of the fake tweets. These all stats have been displayed in more detail in Figure 3.

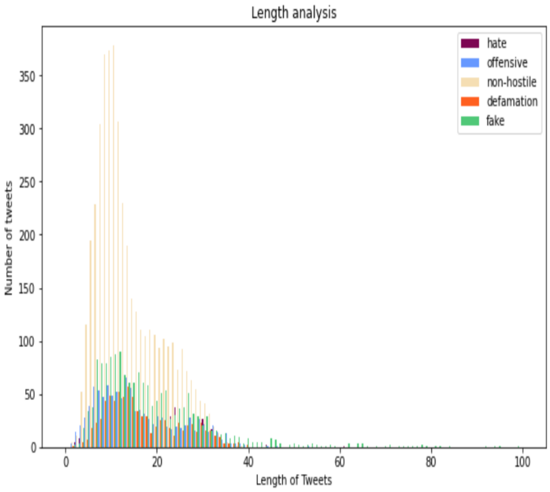


Figure 2: Length Analysis

3.3 Word Count Analysis

There are many similar words in all the class labels. On extracting the most frequently occurring words in the dataset, words like 'देश', 'भारत', 'मोदी', 'सरकार are being used in the majority of the tweets. The bar plot of hostile dimensions and non-hostile label for the most frequent word can be seen in Figure 3 and 4. So, to make our model more accurate, it is good to remove such words from our dataset that are more frequent in the dataset as they will not impact the classification of the tweets. We have drawn the word cloud for non-hostile and hostile tweets for the pictorial representation of the most frequent words. This representation can be seen in Figure 5.

4 Methodology

4.1 Preprocessing

Data pre-processing is done prior to training the models.It is a crucial step because it cleans the data to make it useful.Also the performance of the models on the clean data is good.

- Removal of url links: All the URL links starting with either https or www. were removed from the tweets since they do not have any role in deciding the nature of the tweet.

- Removal of @ mention: All the mentions using @ were removed because they do not have any role in deciding the nature of the tweet.

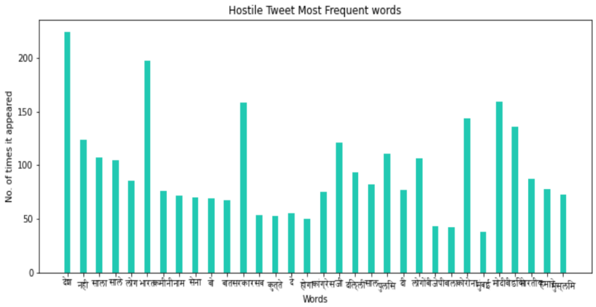


Figure 3: Hostile Tweets Most Frequent Words

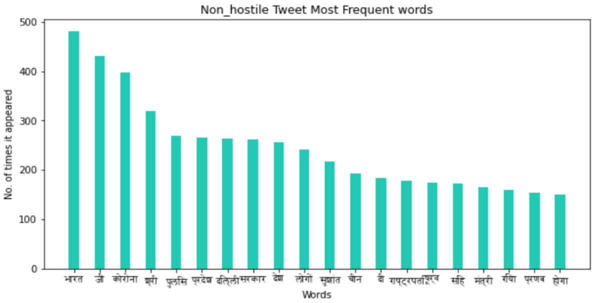


Figure 4: Non-Hostile Tweets Most Frequent Words

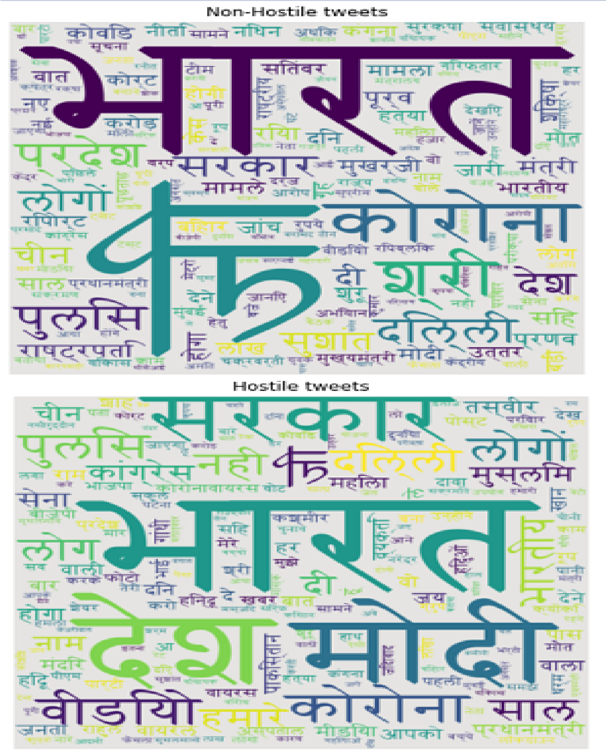


Figure 5: Word Cloud

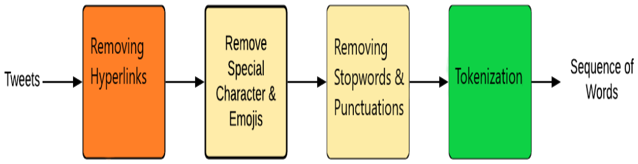

Tweets
Removing Hyperlinks
Remove Special Character & Emojis
Removing Stopwords & Punctuations
Tokenization
Sequence of Words

Figure 6: Phases in Pre-processing of Tweets.

- Removal of all characters other than Devanagari characters: All the characters other than Devanagari characters were removed because they are not a part of Hindi language.

- Removal of all the special characters: All the special characters were removed because they do not have any role in deciding the nature of the tweet.

- Removal of emojis: All the emojis were removed because they are not text.

- Removal of stop-words: All the stop words were removed. That is all the commonly used words of Hindi language were removed.

4.2 Feature Extraction

Feature extraction is the extraction of features from the textual representation of data (Tweets in our case) in the form of numerical features. It contains the following steps to convert the textual data into numerical features:

i. **Tokenization**: In tokenization, the textual representation is split into strings, and it assigns some unique identification numbers to each of the strings. So, the strings with specific id's are called tokens. The process of converting textual data into tokens is called tokenization.

ii. **Counting**: After producing the tokens, the number of occurrences of each token in the whole dataset is assigned in this process.

iii. **Normalization**: After getting the frequency of each token, normalisation is now performed not to emphasise those tokens whose occurrence is very high or very low because it can affect the accuracy of the model. There are two techniques that can be used are:

- **Count Vectorizer**: It is the normalization technique in which it first creates the list of vocabulary or features based on the whole dataset. Now, it stores the frequency of each feature for each of the instances regardless of the importance of the features.

- **TF-IDF**(Term Frequency – Inverse Document Frequency): Now, this technique is used to find the importance of the features. It first builds the list of vocabulary by applying specific rules, and then it finds the TF-IDF values for each of the words in the vocabulary of each of the instances in the dataset.

4.3 Word2Vec

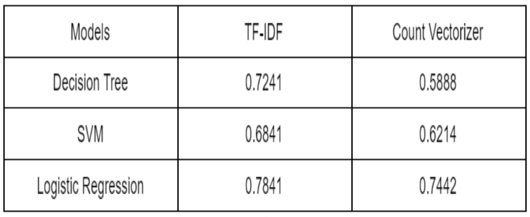
Word2Vec is a neural network group of models that are used to generate word embedding for the set of words to intact the linguistic context of the word. It maps each word of the sentence to a vector of high dimensional space and gives the similarity report for each word in the corpus. It maps word vectors so that the word that shares familiar contexts is placed close to each other in the higher dimension and helps find the dissimilarity of words within the corpus.

4.4 RNN

RNN is a neural network model which several layers along with hidden layers to perform the sentiment analysis in our case. We have used Keras embedding to convert the sentences into a group of vectors, and after that, we have trained this group of vectors on the RNN model in which we have used 'sigmoid' as an activation function. We have also used different classification models with a one vs rest approach to classifying the multilabel tweets in the same embedding.

4.5 M-Bert

Multilingual Bert (M-Bert) extends the Bert model that Google developed. It provides the word embedding for 104 different languages; Hindi is one of them. It gives the pre-trained model for word embedding to generate the vectors for the words. We have used the Hindi Electra model, which is a small model and has comparable results with the M-Bert model. We have fine-tuned the model and adjusted the learning rate to get better results. After getting the embedding for tweets, we have used several classification models on this embedding matrix and used one vs rest classification.



Models	TF-IDF	Count Vectorizer
Decision Tree	0.7241	0.5888
SVM	0.6841	0.6214
Logistic Regression	0.7841	0.7442

Figure 7: F1-Score of Baseline Models

4.6 Baseline Models

The above-mentioned techniques performance was evaluated against some standard Machine Learning models like Decision Trees, Logistic Regression, SVM. The classification models were trained using the features extracted by the TF-IDF and Count-Vectorizer. The models were evaluated using the weighted F1 score.

5 Results

Several models which we used for classification are discussed below. Evaluation metric used to calculate the performance of the model is Weighted F1-Score.

5.1 Decision Tree Classifier

We have used hyper-parameters as: Max_depth=20, criterion='gini', class_weight='balanced'.
F1-score using Decision Tree Classifier on Test Set using RNN embedding : 0.724
F1-score using Decision Tree Classifier on Test Set using Word2Vec : 0.794
F1-score using Decision Tree Classifier on Test Set using M-Bert : 0.776

5.2 SVM classifier

We have used hyper-parameters as: kernel=rbf, C=0.01, gamma=1, class_weight='balanced'.
F1-score using SVM Classifier on Test Set using RNN embedding : 0.57
F1-score using SVM Classifier on Test Set using Word2Vec : 0.698
F1-score using SVM Classifier on Test Set using M-Bert : 0.688

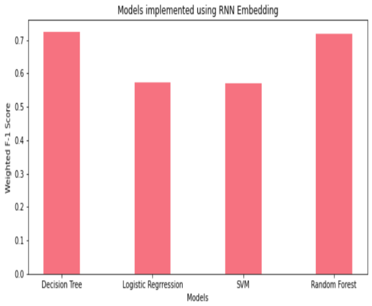


Figure 8: F1-Score of different Models using RNN Embedding.

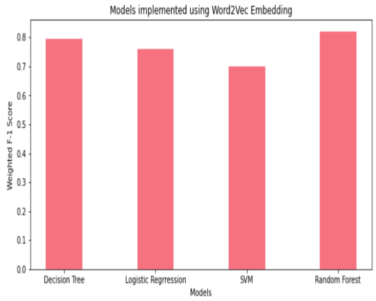


Figure 9: F1-Score of different Models using Word2Vec Embedding.

5.3 Logistic Regression Classifier

We have used hyper-parameters as: penalty=L2, solver=lib-linear, class_weight='balanced'.
F1-score using Logistic Regression Classifier on Test Set using RNN embedding : 0.572
F1-score using Logistic Regression Classifier on Test Set using Word2Vec : 0.760
F1-score using Logistic Regression Classifier on Test Set using M-Bert : 0.696

5.4 Random Forest Classifier

We have used hyper-parameters as: n-estimator=400, class_weight='balanced'.
F1-score using Random Forest Classifier on Test Set using RNN embedding : 0.718
F1-score using Random Forest Classifier on Test Set using Word2Vec : 0.820
F1-score using Random Forest Classifier on Test Set using M-Bert : 0.884

5.5 LSTM

We have used LSTM and bi-directional LSTM with activation function as sigmoid and found the following results in terms of accuracy:
LSTM:- Accuracy = 77.8%
Bi-directional LSTM:- Accuracy = 85.6%

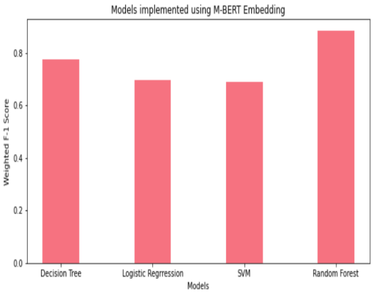
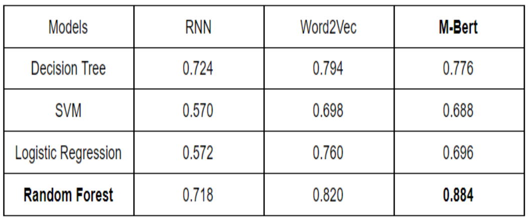


Figure 10: F1-Score of different Models using M-BERT Embedding.



Models	RNN	Word2Vec	**M-Bert**
Decision Tree	0.724	0.794	0.776
SVM	0.570	0.698	0.688
Logistic Regression	0.572	0.760	0.696
Random Forest	0.718	0.820	**0.884**

Figure 11: Weighted F1-Score of different Models

6 Conclusion

This article concludes that M-Bert with random forest classifier outperformed RNN, Word2vec and LSTM with different models. With M-Bert and random forest classifier, we got the highest weighted F1-score = 0.884. For M-Bert, we have used the "Hindi Electra Model", it is very lightweight and is pre-trained on a vast corpus of data and works well on low memory. However, Word2Vec and LSTM models also performed significantly good.

7 Future Work

Tweet analysis for Non-hostile and hostile such as fake, defamation, hate and offensive news detection has a lot of scopes and is becoming more prevalent these days. On a larger scale, this paper's idea can be used to classify the multi-label comments on various social media platforms such as Instagram, Twitter, Facebook, etc., especially for the Hindi language, as it is becoming more prevalent.

8 Contribution

- Akash Rawat: Exploratory data analysis, data pre-processing,and implemented LSTM.
- Parul Sikri: Baseline model, RNN and

Word2vec implementation.

- Shubham Rana: Literature Survey, Baseline model, and M-Bert implementation.

References

M. Bhardwaj, M.S. Akhtar, A. Ekbal, A. Das, T. Chakraborty. *Hostility Detection Dataset in Hindi.* URL: https://arxiv.org/abs/2011.03588

N. Vashistha, A. Zubiaga. *Online Multilingual Hate Speech Detection: Experimenting with Hindi and English Social Media.* URL: https://www.mdpi.com/2078-2489/12/1/5

V. K. Jha, H. Poroli, Vinu P. N, V. Vijayan , Prabaharan P. *DHOT-Repository and Classification of Offensive Tweets in the Hindi Language.* https://doi.org/10.1016/j.procs.2020.04.252

S. Kamble, A. Joshi. *Hate Speech Detection from Code-mixed Hindi-English Tweets Using Deep Learning Models.* URL: https://arxiv.org/abs/1811.05145

S. Kaur, P. Kumar, P. Kumaraguru. *Automating fake news detection system using multi-level voting model.* URL: https://link.springer.com/article/10.1007/s00500-019-04436-y

H. Ahmed, Hadeer, I. Traore, S. Saad. *Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques.*URL: https://link.springer.com/chapter/10.1007%2F978-3-319-69155-8_9

O. Kamal, A. Kumar, T. Vaidhya. *Hostility Detection in Hindi leveraging Pre-Trained Language Models.*URL: https://arxiv.org/pdf/2101.05494.pdf

Drive link for codes and other data:
URL: https://drive.google.com/drive/folders/16t1Vwek6focjQEJxDiqiETV4Uks0wb0O?usp=sharing