

Hate Speech Detection in Marathi and Code-Mixed Languages using TF-IDF and Transformers-Based BERT-Variants

Sakshi Kalra¹, Kushank Maheshwari¹, Saransh Goel¹ and Yashvardhan Sharma¹

¹Department of CSIS, BITS Pilani, 333031, Rajasthan, INDIA

Abstract

People now express their ideas on social media on a global scale. Online attacks against others can be made without fear of repercussions due to the increased sense of freedom provided by the anonymity feature, which eventually leads to the spread of hate speech. The current attempts to filter online information and stop the propagation of hatred are insufficient. Regional languages' popularity on social media and the lack of hate speech detectors that can be used in multiple languages are two aspects that contribute to this. This paper discusses two aspects of fake news detection namely: Identification of Conversational Hate-Speech in Code-Mixed Languages like Hindi, English and German, while second part discusses about Offensive Language Identification in Marathi. Our approach uses TF-IDF word embedding combined with Machine Learning models and transformer based BERT models for the classification of hate speech in each of the two sub tasks. The MuRIL-BERT model produces the best results, with an accuracy of 73.1% and a Macro-F1 score of 0.727 for the code-mixed language and a macro F1-score of 0.8306 on Marathi data, which is 6% more from previous year.

Keywords

Cyber hate, Social Media, MuRIL, HASOC, BERT, Distil-BERT, Code Mixed, Transformers model, Text Classification, Tokenizer, TF-IDF, Multilingual BERT, Machine Learning

1. Introduction

In the past few years, academics have become more interested in the topic of hate speech. This is shown by the fact that the number of Web of Science (WOS)-indexed publications went from 42 in 2013 to 162 in 2018 [1]. According to the Encyclopedia of the American Constitution, "Hate speech is speech that attacks a person or group on the basis of attributes such as race, religion, ethnic origin, national origin, sex, disability, sexual orientation, or gender identity." [2]. The hate speech on social media is becoming the new normal and is devastating for our society. Hate speech divides society and sometimes even leads to communal disharmony and violence. In recent years, it has been seen that some terrorist attacks motivated by hate had a long history of hateful posts on social media, which led to radicalization [3]. In some cases, social media even plays a more direct role, such as in the 2019 attack in Christchurch, New Zealand, and the recent shooting in a mall in the USA, where the suspect live broadcast the shootings on social media platforms [3]. The only way to stop this spread of hatred is to quickly identify the hate

FIRE 2022: Forum for Information Retrieval Evaluation, December 9-13, 2022, India

✉ p20180437@pilani.bits-pilani.ac.in (S. Kalra); f20180679@pilani.bits-pilani.ac.in (K. Maheshwari); f20190988@pilani.bits-pilani.ac.in (S. Goel); yash@pilani.bits-pilani.ac.in (Y. Sharma)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

speech, which is impossible to do manually and must instead be done computationally.

By setting up assignments and seminars, online communities, social media businesses, and technology firms are making significant investments and promoting research in this field of Hate Speech Detection. FIRE is one such group, and it has been actively managing the HASOC responsibilities since 2019 [4]. HASOC 2022 is looking for technology that can detect inflammatory language and hate speech without human intervention. The competition is broken up into two subtracks.

For the first task, the dataset contains code-mix tweets in more than one language (Hinglish and German), along with comments and replies to those comments. When the language is coded, it is difficult to tell what is hate speech. Code mixed text uses the vocabulary and grammar of more than one language [5]. For example, in the dataset used, the Hinglish data has Hindi written in both roman and devanagari script, which makes it harder to find hate speech in this data. The proposed model uses two methods for text classification one is machine learning approach using TF-IDF feature extraction and other is deep learning approach using different BERT variants, which are based on the transformers model; BERT has been shown to be the best in understanding the right and left context in a text up to this point.

For the second task of Hate Speech and Offensive Content Identification in Marathi Language aims at Binary classification to classify a tweet by a user as either offensive and hate or not offensive. The overview of FIRE 2022 subtasks is presented in [6] and [7]. We approached the task using the Transformers-based models namely MuRIL, Distil-BERT and Multilingual-BERT which have displayed impressive outcomes in NLP tasks like text classification. The provided Marathi dataset is fine-tuned using a pre-trained transformer model from the HuggingFace library¹. We demonstrate that using transfer learning on pre-trained BERT models is preferable to using conventional machine learning algorithms. The code is available from the github repository².

2. Related Work

For the code-mixed languages, various approaches in the past have been used. The authors of [8] explain how we can extract features from text data using TF-IDF. They examined the performance of the TF-IDF implementation using 1400 papers from the United Nations Parallel Text Corpus for LDCs and only returned the top 100 relevant texts. In a further study [9], researchers went into greater detail about TF-IDF feature extraction and compared character n-grams to word n-grams, concluding that character n-grams were more useful for detecting hate speech. Another paper by [10] describes how the BERT model can be used for text classification; this paper covers the architecture of the BERT model, which is trained on a large corpus of data and input tokenized text, as well as an attention mask. They achieved GLUE scores of 80.5%, 86.7% MultiNLI accuracy, 93.2% on the SQuAD v1.1 question-answering test, and 83.1%

¹<https://huggingface.co/>

²https://github.com/Kushank24/Marathi_akenews

on the SQuAD v2.0 test. The approach of utilising BERT for classification in [10] is further explained by using the output corresponding to the [CLS] token and adding a Feed Forward Network above it. Another study [11] used soft voting technique on three transformer-based architectures (urduhack, BERT, and XLM-RoBERTa) to achieve an accuracy of 93.6%. The authors in [12] make an attempt to identify threatening posts using deep learning based models on transformers, they essentially employed the pretrained BERT model (RoBERTa) for classifying text as threatening and non-threatening and obtained an F1 score of 53.46% and ROC AUC of 81.99%.

Another paper in [13] fine-tuned monolingual and multilingual transformers over Urdu text, and used ensembling techniques to combine the results of RoBERTa-urdu-small, XLM-RoBERTa, bert-based-multilingual-case, and Alberta-urdu-large, yielding an accuracy of 0.596 and an F1 score of 0.449. In another attempt by [14] got the highest F1 score of 0.7993 by using pre-trained BERT models with a fine-tuning classification layer over them. They also used data augmentation to make the models generalise better and used both machine learning and deep learning techniques for the task of recognising hate and offensive speech. The effectiveness of several pre-trained multilingual BERT models in the detection of threats and hate speech, which are also types of emotions, was discussed in [13] and [14]. [3] used a variety of datasets, the majority of which were based on data from Twitter, including TRAC, hatebase Twitter, Kaggle, etc., and suggested an SVM-based model called mSVM, which on the TRAC dataset produced state-of-the-art results with 80% accuracy and a 53.68% macro F1 score. They also employed the BERT model, which produced results that were 2 percent better but could not explain the interpretability of the choice.

For the Marathi Language, Automated offensive and hate speech detection has been tested using a variety of machine learning and deep learning techniques [15]. The bulk of conventional machine learning techniques extract features from voice text, such as lexical and linguistic features, n-grams, and bags of words [16]. Word embedding techniques have also recently been presented for these tasks [17]. However, these methods fall short of capturing the speech's whole context. Deep learning methods [18] are currently becoming more and more popular in a variety of fields, including machine translation, sentiment analysis, text classification, and language modelling. Recurrent Neural Networks (RNNs) [19], Convolutional neural networks (CNNs) [20], long short-term memories (LSTMs) [21], and the newest approach, bidirectional encoder representations (BERT) [22], are a few of these methods. A combination of Machine Learning models and transformers based models is presented in [23].

In [11] both ML models as well as Transformer based models have been applied for Urdu Language. Additionally, BERT models for Hate Speech detection for Urdu Language has also been applied in FIRE 2021 [12]. Another study [24] for identifying hate speech phrases on Twitter was done. In order to comprehend semantics, the deep convolutional neural network model and GloVe embedding vectors have been combined. With an F1-score of 0.92, the findings explain that their model performed better than the other models. In [14] techniques like TF-IDF weightings as well as word embeddings are used, which is then fed into machine learning algorithms namely random forest, logistic regression and support vector classifier.

Table 1
Dataset Statistics

| Data Type | HOF | NOT | Total Entries |
|---------------|------|------|---------------|
| Training Data | 2612 | 2609 | 5221 |

3. Dataset

Task A (Code-Mixed Language):

The dataset used in this task is collected from HASOC (2022)³ which is one of the subtracks of the Forum for Information Retrieval Evaluation (FIRE)⁴ 2022. It is a collection of tweets; each instance of the dataset [25], [26] includes a main tweet that is labelled as HOF or NOT. Additionally, each tweet may obtain multiple comments, each of which is also labelled "HOF" or "NOT." Finally, each comment may receive multiple replies, each of which is also labelled "HOF" or "NOT."

- **Non Hate Offensive(NOT)** - Tweets, Comments or Replies with this label does not include Hate Speech.
- **Hate and Offensive(HOF)** - Tweets, Comments or Replies with this label include Hate or offensive speech

The dataset differs in that the determination of whether a comment or a reply falls under the category of hate speech depends on both the main tweet and the comment in the case of a reply. For instance, a comment of "yes" is meaningless by itself, but if it is made in response to a main tweet that is hate speech, then it is considered hate speech, while a comment of "no" for the same tweet is not. Therefore, the modification we made to get it ready for the model (to capture the context of the tweet, comment, and reply) is that the text for the main tweet remains the same, the main tweet is appended to the comment, and the main tweet as well as the comment are appended to the reply text (separated by blank space). This way, it will be able to capture the context of the comment and reply.

- **Main tweet:** <main tweet>
- **Comment:** <main tweet> <comment>
- **Reply:** <main tweet> <comment> <reply>

Table 1 shows the dataset statistics. The graphical representation of statistics for Hinglish+German twitter dataset and the exact data distribution is shown in Figure 1.

Task B (Marathi Language):

The datasets for the tasks are provided by the organizers of HASOC'22⁵. The subtask A in the HASOC challenge for Marathi Language is a binary classification task. We need to categorize the sentences in the Marathi Language dataset into the following classes:

³<https://hasocfire.github.io/hasoc/2022/index.html>

⁴<http://fire.irs.res.in/fire/2022/home>

⁵<https://hasocfire.github.io/hasoc/2022/index.html>

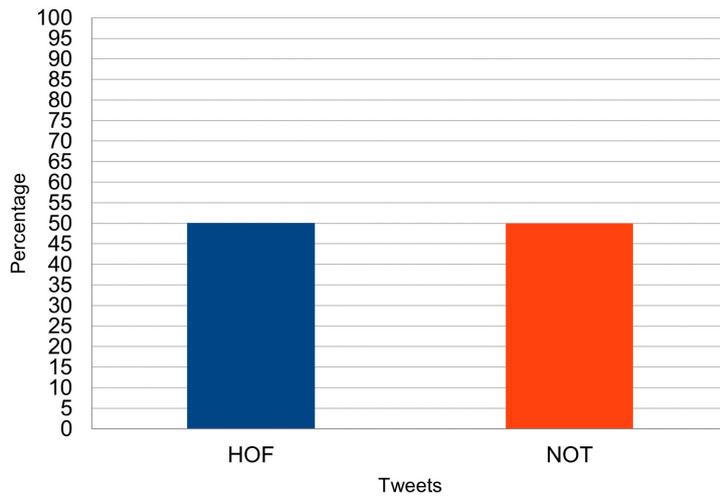


Figure 1: Training set distribution in the Hinglish+German Dataset

- **Non-Offensive(NOT)** - Tweets containing this label do not contain hate speech, foul language, or other offensive material.
- **Offensive(OFF)** - Hateful, offensive, and profane content can all be found in tweets with this label.

The data statistics are as follows:

Table 2

Dataset Statistics on the basis of Binary Label Data

| Data | NOT | OFF | Total Entries |
|----------------------|------|------|---------------|
| Training Data | 2034 | 1069 | 3103 |

The graphical representation of statistics for the dataset are listed in Figure 2. Twitter’s definition of the term ”Offensive” refers to abusive remarks made to people or groups with the intention of intimidating them or silencing their voice.

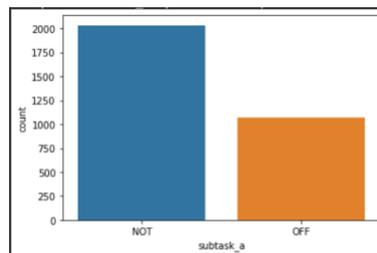


Figure 2: Training set distribution in the Marathi Dataset

4. Handling the Class Imbalanced Issue

For the Task A, the dataset was balanced while for Task B the dataset was imbalanced. To solve this issue, Resampling the training dataset randomly is one way to deal with the issue of data imbalance. The dataset can be resampled using two different techniques: undersampling, which involves removing examples from the majority class, and oversampling, which involves repeating samples from the minority class [14]. We oversampled the dataset using the imblearn [27] library because the training instances are already rather few and removing examples from the majority class will further reduce them. Making the ratio of the minority to the majority class 0.5 by using RandomOverSampler with a sampling method of 0.5.

5. TF-IDF for Text Classification

TF(term frequency) explains the importance of a word for a particular document [8].

$$(\text{Term Frequency}) TF(m) = \frac{\text{Number of times term } m \text{ in doc}}{\text{total terms in doc}}$$

IDF (inverse document frequency) describes the relevance of a word for a corpus. For instance, stopwords are included in every document, making them the least relevant for classifying the whole corpus. As a result, their IDF value will be lower. On the other hand, a word's IDF value will be high if it appears in a small number of documents.

$$(\text{Inverse Document Frequency}) IDF(m) = \log\left(\frac{\text{Total number of docs}}{\text{number of docs with term } m}\right)$$

Then finally we combine both TF and IDF to form TF-IDF:

$$TF - IDF(m) = TF(m) * IDF(m)$$

For the classification of an input tweet, the voting method is used. For each word in the input text, we calculated the <HOF score> and <NOT score>. The code iterates over the entire training set tweet by tweet. For each word in the input text; if the word is present in the tweet, then check for the tweet's label. If label is 1, then the tf-idf value of the word for that tweet is added to its <HOF score> otherwise to its <NOT score>. The <HOF score> and <NOT score> values of all words thus calculated are added to the full input text, and the label with the higher score is the predicted label.

6. BERT Model and its Variants for Text Classification

[10] There are two main steps related to the BERT architecture for classification: pre-training and fine-tuning. Pre-training involves training the model on unlabeled data using several pretrained tasks. An English teacher teaches a language to a child by using "fill in the blanks", "question and answer" types of exercises. The BERT model is pre-trained in a similar way by giving it tokenized text and masking part of the text's tokens; the model's job is to discover the

missing word. Another method used for pre-training BERT is next sentence prediction. It starts with choosing two sentences A and B, 50% of the time B is the actual sentence following A and 50% of the time it is a random sentence from the corpus. This teaches the model to identify the relationship between two sentences, which will help in "question answering" tasks. The next step is the fine-tuning of various tasks, such as classification and question answering, for which two sentences are appended with a [SEP] token between them and only one sentence is passed as input. The fine-tuning task will require some additional layers over the output from the BERT model for training for a particular task, for example, for classification, the output corresponding to the [CLS] token is taken as input for the Feed Forward Network (FFN). This Feed Forward Network is called the fine tuning layer, and during fine tuning, the weights of this classification layer are trained without changing the weights inside the BERT model. So, we can say that the fine tuning layer is using the knowledge of the BERT model to train for classification; in our case, there are two nodes in the output layer for binary classification. BERT architecture is shown in Figure 3. The following BERT variants are used in the proposed task:

- **MuRIL**⁶ - MuRIL [28] is a BERT based model trained over 17 Indian languages using Wikipedia data.
- **Multilingual-BERT**⁷ - M-BERT [29] has 104 languages pre-trained from large wikipedia data. WordPiece is used to tokenize and lowercase the texts, and a common vocabulary with a size of 110,000 is used. This model is case sensitive.
- **Distil-BERT**⁸ - DistilBERT [30] model is based on small, cheap and fast transformers used knowledge distilling during pre-training and reduced the size of BERT by 40%

7. Proposed Techniques and Algorithms

Task A (Code-Mixed Language):

The suggested model, as shown in Figure 4, first takes the multilingual and code mixed text as input and preprocesses it by deleting stopwords(sklearn library provides the list of stopwords for English and German language and Kaggle provided for Hindi language, a custom function is used to remove the stopwords from dataset one by one using the lists of stopwords) for the dataset presented in Figure 1. The hyperlinks, emojis and hashtags are also removed. The text is made lowercase to handle names in the text. Following that, the preprocessed data is used to train two different models, the TF-IDF feature extraction model and the BERT model (all models are trained independently). The HOF and NOT scores for test data are determined using the TF-IDF feature extraction approach, which is described in the next section. The following are the phases related to the text classification using TF-IDF:

- **Data Pre-Processing**
- **Extracting TF-IDF features**
- **Calculating TF-IDF score for classification**

⁶<https://huggingface.co/google/MuRIL-base-cased>

⁷<https://huggingface.co/bert-base-multilingual-cased>

⁸<https://huggingface.co/distilroberta-base>

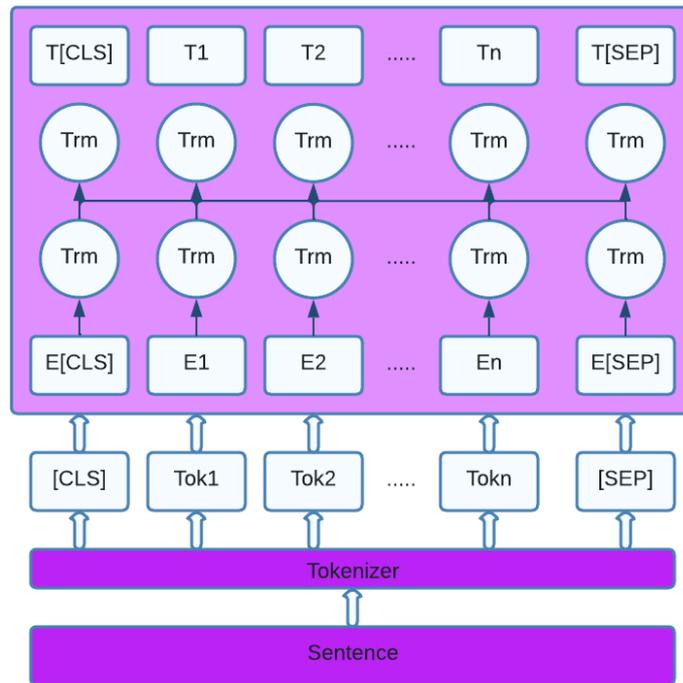


Figure 3: BERT Model

Table 3
Various Hyperparameters and its Descriptions

| Hyperparameter | Description |
|------------------|-------------|
| Learning Rate | 1e-05 |
| Number of Epochs | 4 |
| Batch Size | 2 |

To determine whether text input is HOF or NOT, a tokenizer is applied first, followed by a fine-tuning layer over the four pre-trained BERT models (Distill BERT, Multilingual BERT, RoBERTa, and Murlil BERT). Figure 4 shows the proposed architecture for the hate speech classification. The four key phases of the process are:

- **Data Pre-Processing**
- **Tokenization**
- **Using Pre-Trained BERT Model**
- **Fine-Tuning Classifier for the Pre-Trained Model**

Table 3 lists the various hyperparameters used while training of the proposed models.

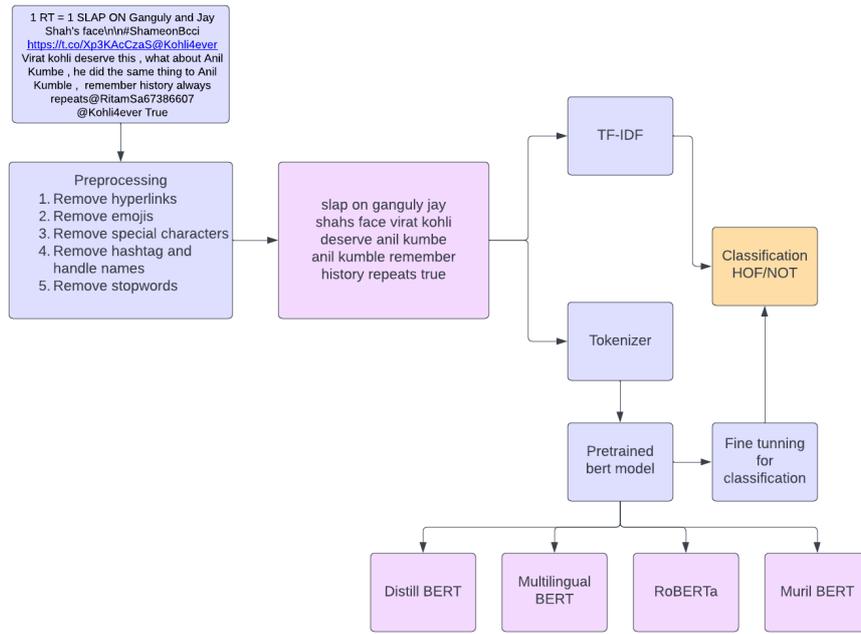


Figure 4: The Proposed Architecture

Task B (Marathi Language):

Transformers-based models offer state-of-the-art implementation for several NLP related tasks such as fake news detection, question answering systems, machine translation, rumour detection etc. As a result of their bidirectional training and greater language comprehension, they outperform other ML approaches. First-step in the Transformer-based model creation is pre-training, which is then followed by fine-tuning. Large language datasets (monolingual) or datasets in several languages (multilingual) are used to train the model in the initial stages. To obtain the word embeddings, just the encoder component of the transformer design is used. To calculate the probability for binary classes, an additional output layer is implemented. The different word embedding models that have been used are mentioned above in the BERT explanation part.

The Flowchart in Figure 5 shows the complete approach. In Brief, the main 4 steps of the process are:

- **Data Pre-Processing**
- **Tokenization**
- **Using Pre-Trained BERT Model**
- **Fine-Tuning Classifier for the Pre-Trained Model**

The hyperparameters for training the model are mentioned in Table 4.

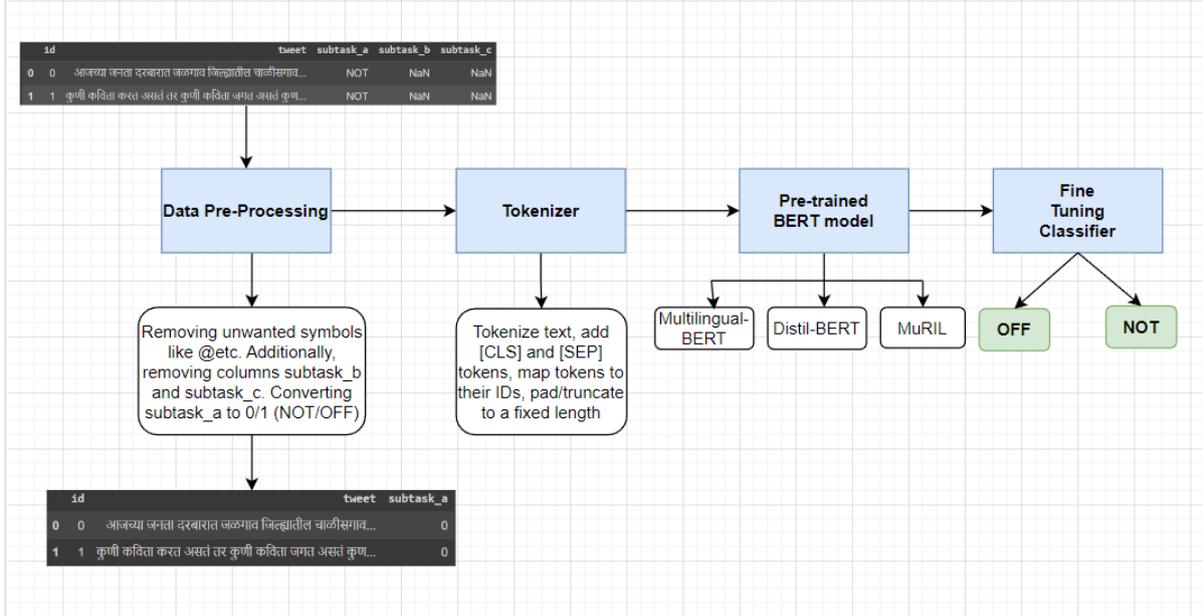


Figure 5: Flowchart of our methodology and techniques

Table 4

Hyper-parameters used in Training

| Hyper-parameter | Description |
|------------------|-------------|
| Learning Rate | 1.00742e-05 |
| Number of Epochs | 4 |
| Batch Size | 2 |

8. Results and Evaluations

Task A (Code-Mixed Language):

The performance of each model is evaluated using various evaluation metrics. Table 5 lists the accuracy, precision, recall, and F1-measure using the TF-IDF model. Table 6 lists the accuracy for Micro-F1 and Macro-F1 using BERT and its variants, Of the three BERT versions, MuRIL produced the best outcomes. Distil-BERT and Multilingual-BERT produced nearly identical results, but Multilingual-BERT performed better. The code is available from the github repository⁹

Task B (Marathi Language):

Accuracy and Macro F1 are used to evaluate each model's performance. MuRIL gave the best results among all 3 BERT models. While MuRIL and Multilingual-BERT almost gave similar results, but MuRIL performed better than Multilingual-BERT. While Distil-BERT performed the worst. The test data provided by HASOC is only for the following Hyperparameters: **Number**

⁹<https://github.com/saransh-goel/HASOC.git>

Table 5

Performance Evaluatuion using TF-IDF

| Model | Accuracy | Precision | Recall | F1-Measure |
|--------|----------|-----------|--------|------------|
| TF-IDF | 0.685 | 0.676 | 0.698 | 0.687 |

Table 6

Performance Evaluatuion using BERT Variants

| BERT Variants | Accuracy | Micro-F1 | Macro-F1 |
|-------------------|----------|----------|----------|
| MuRIL | 0.731 | 0.695 | 0.727 |
| Multilingual-BERT | 0.69 | 0.676 | 0.69 |
| Distil-BERT | 0.69 | 0.67 | 0.69 |

of Epochs = 4, Batch size = 2 and Learning Rate = 1.00742e-05. The results are shown in the following below tables namely Table 7, Table 8, Table 9 and Table 10:

Table 7

Final Results for given Hyper-parameters

| Epochs = 4, batch size = 2, Learning Rate = 1.00742e-05 | | | | | | |
|---|---------------|---------------|-----------------|---------------|-----------------|---------------|
| Data | Training Data | | | Testing Data | | |
| Metrics | Accuracy | Macro-F1 | Macro-Precision | Macro-F1 | Macro-Precision | Macro-Recall |
| MuRIL | 0.9198 | 0.9197 | 0.9202 | 0.9450 | 0.9464 | 0.9446 |
| Multilingual BERT | 0.9103 | 0.9103 | 0.9103 | 0.9291 | 0.9332 | 0.9285 |
| Distil-Bert | 0.7724 | 0.7712 | 0.7777 | 0.8015 | 0.8145 | 0.8021 |

Table 8

Final Results for given Hyper-parameters

| Epochs = 4, batch size = 4, Learning Rate = 1.00742e-05 | | | |
|---|---------------|---------------|-----------------|
| Data | Training Data | | |
| Metrics | Accuracy | Macro-F1 | Macro-Precision |
| MuRIL | 0.9198 | 0.9197 | 0.9205 |
| Multilingual BERT | 0.9021 | 0.9020 | 0.9031 |
| Distil-Bert | 0.8549 | 0.8549 | 0.8552 |

The results show that MuRIL gives the best results in all the scenarios. When the Learning Rate is decreased the accuracy of all three models increases, while when the Learning Rate is increased accuracy of MuRIL and mBERT decreases while accuracy for Distil-BERT increases. At the same time the changes seen when changing Batch size is similar to Learning Rate.

Table 9

Final Results for given Hyper-parameters

| | Epochs = 4, batch size = 2, Learning Rate = 1.1e-05 | | |
|--------------------------|--|-----------------|------------------------|
| Data | Training Data | | |
| Metrics | Accuracy | Macro-F1 | Macro-Precision |
| MuRIL | 0.9186 | 0.9186 | 0.9186 |
| Multilingual BERT | 0.9009 | 0.9009 | 0.9010 |
| Distil-Bert | 0.8136 | 0.8128 | 0.8192 |

Table 10

Final Results for given Hyper-parameters

| | Epochs = 4, batch size = 4, Learning Rate = 1e-05 | | |
|--------------------------|--|-----------------|------------------------|
| Data | Training Data | | |
| Metrics | Accuracy | Macro-F1 | Macro-Precision |
| MuRIL | 0.9233 | 0.9233 | 0.9238 |
| Multilingual BERT | 0.9127 | 0.9127 | 0.9130 |
| Distil-Bert | 0.7853 | 0.7853 | 0.7854 |

9. Conclusion and Future Work

Task A (Code-Mixed Language):

The proposed results demonstrate that the BERT model performs better than the TF-IDF feature extraction model. This is because the BERT model takes into account the right and left context in the text, allowing it to detect hate speech more accurately by taking into account the context of each sentence; additionally, BERT takes subwords as tokens as well; for example, "playing" is broken into "play" and "ing," and then separate embeddings are calculated for each token; this extra quality also helps the BERT model perform better. In this scenario, MuRIL-BERT outperforms multilingual-BERT and Distil-BERT. The next stage for detecting hate speech would be viewed as a multimodal technique. Some social context-based features can also be investigated in future research. One could even go much farther in the TF-IDF feature extraction process to employ character and word n-grams for hate speech detection. There must be a BERT model trained over a large dataset that performs better for code mixed languages, particularly Hindi written in roman script.

Task B (Marathi Language):

The results presented above show that pre-trained BERT models perform better and are better able to grasp the meaning of a given sentence, serving as better learning representations. Therefore, compared to conventional feature extraction approaches, the transfer learning strategy using pre-trained BERT models is better suitable for identifying offensive and hate

speech. The MuRIL performed the best among the three models. On the public leaderboard rankings, we came in fourth place. Additionally, By focusing on both images and text and obtaining the visual components for better feature extraction, we may approach this hate speech detection issue from a multimodal perspective. With better word tokenization and specific tokens for Marathi language, the performance could be enhanced. In the future, models can be trained on a larger corpus to improve accuracy even further. Further, future experiments with deeper transformer architectures may be conducted.

References

- [1] M. A. Paz, J. Montero-Díaz, A. Moreno-Delgado, Hate speech: A systematized review, *Sage Open* 10 (2020) 2158244020973022.
- [2] J. T. Nockleby, L. W. Levy, K. L. Karst, D. J. Mahoney, *Encyclopedia of the american constitution*, Detroit, MI: Macmillan Reference 3 (2000).
- [3] S. MacAvaney, H.-R. Yao, E. Yang, K. Russell, N. Goharian, O. Frieder, Hate speech detection: Challenges and solutions, *PloS one* 14 (2019) e0221152.
- [4] P. Mehta, T. Mandl, P. Majumder, S. Gangopadhyay, Report on the fire 2020 evaluation initiative, in: *ACM SIGIR Forum*, volume 55, ACM New York, NY, USA, 2021, pp. 1–11.
- [5] N. Choudhary, R. Singh, I. Bindlish, M. Shrivastava, Sentiment analysis of code-mixed languages leveraging resource rich languages, *arXiv preprint arXiv:1804.00806* (2018).
- [6] S. Modha, T. Mandl, G. K. Shahi, H. Madhu, S. Satapara, T. Ranasinghe, M. Zampieri, Overview of the hasoc subtrack at fire 2021: Hate speech and offensive content identification in english and indo-aryan languages and conversational hate speech, in: *Forum for Information Retrieval Evaluation*, 2021, pp. 1–3.
- [7] T. Mandl, S. Modha, G. K. Shahi, H. Madhu, S. Satapara, P. Majumder, J. Schäfer, T. Ranasinghe, M. Zampieri, D. Nandini, et al., Overview of the hasoc subtrack at fire 2021: Hate speech and offensive content identification in english and indo-aryan languages, *arXiv preprint arXiv:2112.09301* (2021).
- [8] J. Ramos, et al., Using tf-idf to determine word relevance in document queries, in: *Proceedings of the first instructional conference on machine learning*, volume 242, Citeseer, 2003, pp. 29–48.
- [9] A. Schmidt, M. Wiegand, A survey on hate speech detection using natural language processing, in: *Proceedings of the fifth international workshop on natural language processing for social media*, 2017, pp. 1–10.
- [10] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* (2018).
- [11] S. Kalraa, Y. Bansala, Y. Sharmaa, Detection of abusive records by analyzing the tweets in urdu language exploring transformer based models (2021).
- [12] S. Kalraa, M. Agrawala, Y. Sharmaa, Detection of threat records by analyzing the tweets in urdu language exploring deep learning transformer-based models (2021).
- [13] S. Kalraa, P. Vermaa, Y. Sharmaa, G. S. Chauhanb, Ensembling of various transformer based models for the fake news detection task in the urdu language (2021).

- [14] S. Kalraa, K. N. Inania, Y. Sharmaa, G. S. Chauhanb, Applying transfer learning using bert-based models for hate speech detection (2020).
- [15] T. Davidson, D. Warmsley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, in: Proceedings of the international AAAI conference on web and social media, volume 11, 2017, pp. 512–515.
- [16] A. Gaydhani, V. Doma, S. Kendre, L. Bhagwat, Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach, arXiv preprint arXiv:1809.08651 (2018).
- [17] R. Kshirsagar, T. Cukuvac, K. McKeown, S. McGregor, Predictive embeddings for hate speech detection on twitter, arXiv preprint arXiv:1809.10644 (2018).
- [18] P. Badjatiya, S. Gupta, M. Gupta, V. Varma, Deep learning for hate speech detection in tweets, in: Proceedings of the 26th international conference on World Wide Web companion, 2017, pp. 759–760.
- [19] G. K. Pitsilis, H. Ramampiaro, H. Langseth, Effective hate-speech detection in twitter data using recurrent neural networks, Applied Intelligence 48 (2018) 4730–4742.
- [20] Z. Zhang, L. Luo, Hate speech detection: A solved problem? the challenging case of long tail on twitter, Semantic Web 10 (2019) 925–945.
- [21] A. Bisht, A. Singh, H. Bhadauria, J. Virmani, et al., Detection of hate speech and offensive language in twitter data using lstm model, in: Recent trends in image and signal processing in computer vision, Springer, 2020, pp. 243–264.
- [22] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
- [23] M. Zampieri, T. Ranasinghe, M. Chaudhari, S. Gaikwad, P. Krishna, M. Nene, S. Paygude, Predicting the type and target of offensive social media posts in marathi, Social Network Analysis and Mining 12 (2022) 77. URL: <https://doi.org/10.1007/s13278-022-00906-8>. doi:10.1007/s13278-022-00906-8.
- [24] P. K. Roy, A. K. Tripathy, T. K. Das, X.-Z. Gao, A framework for hate speech detection using deep convolutional neural network, IEEE Access 8 (2020) 204951–204962.
- [25] Satapara, Shrey and Majumder, Prasenjit and Mandl, Thomas and Modha, Sandip and Madhu, Hiren and Ranasinghe, Tharindu and Zampieri, Marcos and North, Kai and Premasiri, Damith, Overview of the HASOC Subtrack at FIRE 2022: Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages, in: FIRE 2022: Forum for Information Retrieval Evaluation, Virtual Event, 9th-13th December 2022, ACM, 2022.
- [26] S. Modha, T. Mandl, P. Majumder, S. Satapara, T. Patel, H. Madhu, Overview of the HASOC Subtrack at FIRE 2022: Identification of Conversational Hate-Speech in Hindi-English Code-Mixed and German Language, in: Working Notes of FIRE 2022 - Forum for Information Retrieval Evaluation, CEUR, 2022.
- [27] S. Prabhu, M. Mohamed, H. Misra, Multi-class text classification using bert-based active learning, arXiv preprint arXiv:2104.14289 (2021).
- [28] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. T. Nagipogu, S. Dave, et al., Muril: Multilingual representations for indian languages, arXiv preprint arXiv:2103.10730 (2021).
- [29] T. Pires, E. Schlinger, D. Garrette, How multilingual is multilingual bert?, arXiv preprint arXiv:1906.01502 (2019).

- [30] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108 (2019).