# mBERT based model for identification of offensive content in south Indian languages

Shankar Biradar, Sunil Saumya and Arun Chauhan

*Indian Institute of Information Technology Dharwad,*
*Indian Institute of Information Technology Dharwad*
*Graphic Era University Dehradun*

**Abstract**

In recent years, there has been a lot of focus on offensive content. The amount of offensive content generated by social media is increasing at an alarming rate. It created a greater need to address this issue than ever before. To address these issues, the organizers of "Dravidian-Code Mixed HASOC-2021" have created two challenges. Task 1 involves identifying offensive content in Malayalam data, whereas Task 2 includes Malayalam and Tamil Code Mixed Sentences. Our team participated in Task 2. We used multilingual BERT to extract features in our proposed model, and we used two different classifiers, Support Vector Machine (SVM) and Deep Neural Network (DNN), on the extracted features. In addition, we used the proposed data to evaluate the performance of a monolingual BERT classifier. Our best performing model monolingual Bert received a weighted F1 score of 0.70 for Malayalam data, ranking fifth; we also received a weighted F1 score of 0.573 for Tamil Code Mixed data, ranking twelfth.

**Keywords**

Offensive, mBERT, CodeMixed, SVM

## 1. Introduction

The availability of smartphones and the internet has created a lot of interest in social media among today's youth. These applications give a huge platform for users to connect with the outside world and share their ideas and opinions with others. With these benefits comes a disadvantage: many people misuse the platform under the name of freedom of expression to publish inflammatory content on social media. This inflammatory information typically targets a single person, a group of people, a particular faith, or a community [1]. People generate objectionable content and aggressively propagate it on social media. This type of material is produced for a variety of reasons, including commercial and political gain [2]. This type of content can disrupt social harmony and cause riots in society. Also, it has the potential to have a detrimental psychological influence on the readers. It can harm people's emotions and conduct. Therefore, identifying this type of content is critical; as a result, researchers, policymakers, and investors (stakeholders) are attempting to develop a dependable technique to identify offensive content on social media.

*Forum for Information Retrieval Evaluation, December 13-17, 2021, India*
shankar@iiitdwd.ac.in (S. Biradar); sunil.saumya@iiitdwd.ac.in (S. Saumya); aruntakhur@gmail.com (A. Chauhan)

Various studies on hate speech, harmful content, and abusive language identification in social media have been conducted during the previous decade. The majority of these studies were focused on monolingual English content, and a large amount of English language cuprous has been created [3]. But, people in countries with a complex culture and history, such as India, frequently use regional languages to generate inappropriate social media posts. Users typically mix their regional languages with English while creating such content. This type of text is known as code mixed text on social media. Hence we require an efficient method to classify offensive content in Code-Mixed Indian languages. In this context, the "Dravidian-CodeMixed HASOC-2021" shared task provider has organized two tasks for detecting hate speech in Dravidian languages such as Malayalam and Tamil code-mixed data. Our team took part in Task No. 2, and this paper presents the working notes for our suggested model.

The rest of the article is arranged in the following manner: Section 2 provides a brief summary of previous work, while Section 4 describes the proposed model in full. Section 5 concludes by providing information on the outcome.

## 2. Literature review

Many researchers and practitioners from industry and academia have been attracted to the subject of automatic identification of hostile and harmful speech. [4] Provided a high-level review of the current state-of-the-art techaniques in offensive language identification and related issues, such as hate speech recognition. [5] Developed a publicly accessible dataset for identifying the offensive language in tweets by categorizing them as hate speech, offensive but not hate speech or neither. Various machine learning models, such as Support Vector Machine (SVM) and logistic regression, were created utilizing various data properties, such as n-grams, TF-IDF, readability, etc., for this purpose. [6] Built a model with deep neural networks in combination with SVM for the detection of offensive content with the accomplishment of F1 score of 90%.

Offensive content detection from tweets is part of some conferences as well as competition tasks. Offensive 2020 was provided by SemEval in 2020 as a task in five languages: English, Arabic, Danish, Greek, and Turkish [7]. In FIRE 2019, a similar task was achieved for Indo-European languages such as English, Hindi, and German. The data set was created using samples obtained on Twitter and Facebook in all three languages. Various models, including LSTM with attention, Word2vec embedding with CNN, and BERT, were used for this task. In several cases, traditional learning models outperformed deep learning methods for a language other than English [8]. Shared task on offensive language detection in Dravidian languages was provided by [9] [10].

## 3. Task and Data set description

We have taken data set from HASOC subtask, offensive language identification of Dravidian CodeMix[11]. Challenges provided by the organizers are as follows.

Task 1: A binary classification problem with message-level labeling for offensive and non-offensive information in Malayalam CodeMixed YouTube comments.

**Table 1**
data set description

| Task | Data set | Offensive | Not-offensive | Total |
|---|---|---|---|---|
| **Tanglish** | Train | 1980 | 2020 | 4000 |
| | Test | 475 | 465 | 940 |
| **Manglish** | Train | 1953 | 2047 | 4000 |
| | Test | 512 | 488 | 1000 |

Task 2: Given Romanized Tanglish and Manglish tweeter or YouTube comments, the system must classify them as offensive or non-offensive.

Our team took part in Task 2 for identifying offensive information in the Tanglish and Manglish data sets. According to the organizer, Tanglish data is collected from Twitter tweets and comment on the hello APP. Whereas Manglish data is taken from YouTube comments [11]. A detailed description of the data set is provided in Table 1, both Tanglish and Manglish data contain ID, Tweet, and Label fields.

## 4. Methdology

Our team has proposed three submissions based on three different models. In the first two models, mBERT embeddings are passed through SVM and DNN classifiers, while in the third model, monolingual BERT is employed as a classifier. Each of them is designed using the general architecture shown in Figure 1. Thus, our model consists of three stages, each of which is discussed in the preceding subsections.

### 4.1. Data processing

The data set provided by the organizer contains many unwanted information. A few data preprocessing steps were undertaken on both text and label fields to convert the data suitable for model building. Digits, special characters, hyperlinks, and Twitter user handles were omitted from the data set because they were not helping us improve the performance of our model. Furthermore, the social media data provided by the organizer did not follow grammatical norms; hence, data lemmatization is performed to convert the data to its usable base form. For example, the word ate, eaten, and eating were converted to their base form eat. Converting text to lower case is also done to eliminate redundant terms. All of this preprocessing was done with the help of the NLTK toolbox from the Python library [12]. The preprocessed data is then fed into a tokenizer, which divides the tweet into several tokens. The mBERT tokenizer [1] is used for this purpose. Padding and masking were also used to handle variable-length sentences.

[1]https://huggingface.co/bert-base-multilingual-cased

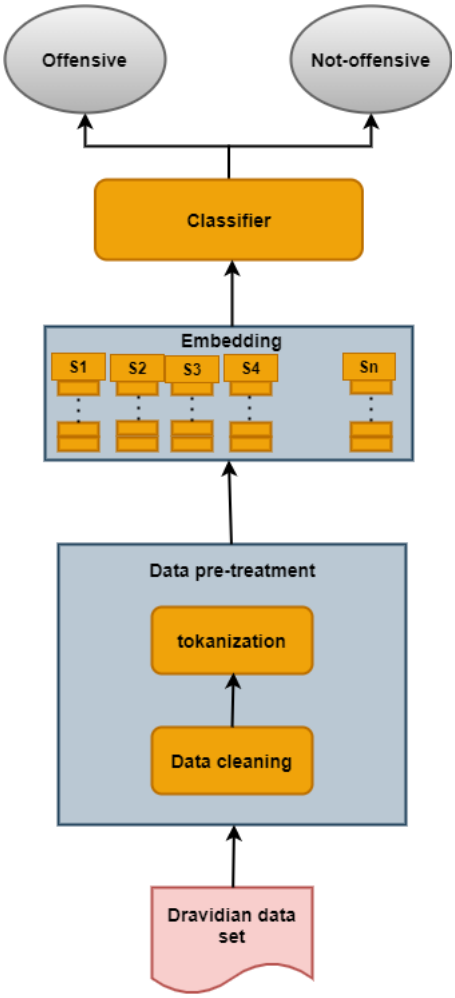


**Figure 1:** General architecture of classifier model

### 4.2. Feature extraction

To obtain contextual embeddings from Code-Mixed data, we used the multilingual Bidirectional Encoder Representation (mBERT) model [13] in models 1 and 2, and monolingual BERT in model 3. The architecture of the mBERT model is largely based on the original monolingual BERT architecture [14], which has 12 transformer blocks, 12 attention heads, and 768 hidden layers. Furthermore, the vector dimension of mBERT embeddings is 768. This model was trained using the same pre-training technique as the BERT, namely Masked Language Modeling (MLM) and Next Sentence Prediction. The only distinction is that multilingual BERT is trained on Wikipedia data from 104 different languages to handle languages other than English. We only draw embeddings from the CLS token at the beginning for our classification purposes because it gives whole sentence embeddings.

### 4.3. Classification

Our proposed model experimented with three different classifiers: SVM and DNN classifiers with mBERT embedding in models 1 and 2 and pre-trained language model BERT in model 3. The descriptions of these models are presented in the subsections that follow. The intuition behind selecting these proposed models is that they outperformed other models such as Logistic Regression (LR), Random Forest (RF), and Naive Bayes (NB) in our preliminary trials.

#### 4.3.1. Traditional machine learning based classifier

We experimented with traditional machine learning algorithms such as Support Vector Machine (SVM) with ten-fold cross-validation. Experiment results for the suggested model demonstrate that kernel value "1" and solver "lbfgs" produce the best results. Experimental trials are used to determine these hyper-parameter values. This model accepts mBERT embeddings as input and produces labels that are either offensive or non-offensive. The model was developed using Python's sci-kit-learn library [15].

#### 4.3.2. Deep neural network based model

Later, we experimented with the Deep Neural Network (DNN) model, a second model in our proposed methodology. The DNN model comprises several dense layers that are designed to extract more significant features from input embeddings. We used dense layers of 1000, 500, 100, and 50 neurons in our model. Each dense layer follows a dropout rate of 0.4 to prevent the overfitting problem. The optimum grid search value determines the dropout rate of 0.4, and it remains constant throughout the experiment. To normalize activation data, we additionally employed a batch normalization layer. The output from these layers is then classified using the sigmoid layer.

#### 4.3.3. Transformer model

In our last model, we experimented with transformer-based language models such as BERT. Transformer architectures are trained on generic tasks such as modeling language and then fine-tuned for classification. The underlying model for our classification is Bert-base-uncased [2], which BERT developers provide. We did not use ten-fold cross-validation to evaluate monolingual BERT since it is more computationally expensive. Implementation details of all three proposed models are provided in GitHub repository[3].

## 5. Experimental Results

To evaluate the presented models, the organizers have provided a weighted F1 score. Among the proposed models, our top-performing monolingual BERT received a sixth-place for offensive content recognition in the Mangalish data set and eleventh in the Tanglish data set. Table 2 and Table 3 provide the list of top-performing models with weighted F1 scores for Manglish

[2]https://huggingface.co/transformers/model_doc/bert.html
[3]https://github.com/shankarb14/dravidian-codemix

**Table 2**
Top performing models on Manglish data set

| Team name | Precision | Recall | F1 score | Rank |
|---|---|---|---|---|
| AIML | 0.776 | 0.762 | 0.766 | 1 |
| MUCIC | 0.764 | 0.76 | 0.762 | 2 |
| HSU | 0.744 | 0.73 | 0.735 | 3 |
| IIIT Surat | 0.752 | 0.727 | 0.734 | 4 |
| IRLab | 0.754 | 0.705 | 0.714 | 5 |
| **IIITD-ShankarB** | **0.715** | **0.693** | **0.7** | **6** |

**Table 3**
Top performing models on Tanglish data set

| Team name | Precision | Recall | F1 score | Rank |
|---|---|---|---|---|
| MUCIC | 0.679 | 0.685 | 0.678 | 1 |
| AIML 2 | 0.67 | 0.67 | 0.67 | 2 |
| SSN_IT_NLP | 0.685 | 0.688 | 0.668 | 3 |
| ZYBank AI | 0.671 | 0.676 | 0.654 | 4 |
| IRLab | 0.654 | 0.662 | 0.65 | 5 |
| **IIITD-ShankarB** | **0.599** | **0.568** | **0.573** | **11** |

**Table 4**
Comparative results of proposed models

| Model name | Data set | Accuracy(%) | F1-score OFF(%) | F1-score NOT(%) |
|---|---|---|---|---|
| mBERT+SVM | Malayalam | 53 | 41 | 61 |
| mBERT+DNN | Malayalam | 55 | 43 | 63 |
| **BERT classifier** | **Malayalam** | **70** | **58** | **77** |
| mBERT+SVM | Tamil | 50 | 43 | 54 |
| mBERT+DNN | Tamil | 51 | 44 | 55 |
| **BERT classifier** | **Tamil** | **57** | **53** | **60** |

and Tanglish data set respectively (The result of our proposed model is shown in bold letters). Among our proposed models, BERT outperformed other classifiers, reaching 70% accuracy for the Mangalish data set and 57% accuracy for the Tanglish data set. Finally, we compared the results of our proposed models in Table 4. We trained our proposed models on a Tanglish data set comprising 4000 comments from the training set and tested them on 940 comments from the test set. For the Manglish data set, 4000 train comments and 1000 test comments are used.

## 5.1. Error analysis

We investigated the behavior of proposed models on sample test sentences to evaluate their performance. We discovered that our best-performing model monolingual BERT classifier could accurately classify all of the test samples based on our experimental observations. However,

**Table 5**
Test case result for sample test sentences

| Tweet | Data set | mBERT+SVM | mBERT+DNN | BERT | Target |
|---|---|---|---|---|---|
| athe beharyku deputationil pokam pinarai vijayanu chinayilum pokam. | Malayalam | NOT | NOT | NOT | NOT |
| ithu vallathum nadakkumo shajan sir kettittu kothiyakunnu. | Malayalam | NOT | OFF | NOT | NOT |
| Indha movie ku award tharlanA avanga mansanay illa bro. | Tamil | OFF | NOT | OFF | OFF |
| kritheeck Kookaburra en unaku enachu? Cbsc ah?? | Tamil | NOT | NOT | NOT | NOT |

multilingual BERT models such as mBERT+SVM and mBERT+DNN could not classify test samples 3 and 2, respectively. Table 5 summarises the results of the findings.

## 6. Conclusion and future enhancement

In our work, we presented a model submitted by our team IIITD-ShankarB for offensive content identification in the shared task "Dravidian-CodeMixed HASOC-2021". Our proposed work experimented with three distinct models: a machine learning-based model, a Deep Neural Network model, and a transformer-based language model. Our model is one of the top-performing models, ranking sixth on the Manglish data set and eleventh on the Tanglish data set. In future work, we can improve the efficiency of the suggested model by including domain-specific embeddings.

## References


[1] S. A. Chowdhury, H. Mubarak, A. Abdelali, S.-g. Jung, B. J. Jansen, J. Salminen, A multi-platform Arabic news comment dataset for offensive language detection, in: Proceedings of the 12th Language Resources and Evaluation Conference, 2020, pp. 6203–6212.

[2] C. N. d. Santos, I. Melnyk, I. Padhi, Fighting offensive language on social media with unsupervised text style transfer, arXiv preprint arXiv:1805.07685 (2018).

[3] H. Mubarak, K. Darwish, W. Magdy, Abusive language detection on Arabic social media, in: Proceedings of the first workshop on abusive language online, 2017, pp. 52–56.

[4] P. Fortuna, S. Nunes, A survey on automatic detection of hate speech in text, ACM Computing Surveys (CSUR) 51 (2018) 1–30.

[5] 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.

[6] H. Al-Khalifa, W. Magdy, K. Darwish, T. Elsayed, H. Mubarak, Proceedings of the 4th workshop on open-source Arabic corpora and processing tools, with a shared task on offensive language detection, in: Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, 2020.

[7] M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak, L. Derczynski, Z. Pitenis, Ç. Çöltekin, Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020), arXiv preprint arXiv:2006.07235 (2020).

[8] T. Mandl, S. Modha, P. Majumder, D. Patel, M. Dave, C. Mandlia, A. Patel, Overview of the HASOC track at FIRE 2019: Hate speech and offensive content identification in Indo-European languages, in: Proceedings of the 11th forum for information retrieval evaluation, 2019, pp. 14–17.

[9] B. R. Chakravarthi, R. Priyadharshini, N. Jose, A. Kumar M, T. Mandl, P. K. Kumaresan, R. Ponnusamy, H. R L, J. P. McCrae, E. Sherly, Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada, in: "Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages", Association for Computational Linguistics, Kyiv, 2021, pp. 133–145. URL: https://aclanthology.org/2021.dravidianlangtech-1.17.

[10] T. Mandl, S. Modha, A. Kumar M, B. R. Chakravarthi, Overview of the HASOC track at FIRE 2020: Hate speech and offensive language identification in Tamil, Malayalam, Hindi, English and German, in: Forum for Information Retrieval Evaluation, 2020, pp. 29–32.

[11] B. R. Chakravarthi, P. K. Kumaresan, R. Sakuntharaj, A. K. Madasamy, S. Thavareesan, P. B, S. Chinnaudayar Navaneethakrishnan, J. P. McCrae, T. Mandl, Overview of the HASOC-DravidianCodeMix Shared Task on Offensive Language Detection in Tamil and Malayalam, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.

[12] S. Bird, E. Klein, E. Loper, Natural language processing with Python: analyzing text with the natural language toolkit, " O'Reilly Media, Inc.", 2009.

[13] T. Pires, E. Schlinger, D. Garrette, How multilingual is multilingual bert?, arXiv preprint arXiv:1906.01502 (2019).

[14] 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).

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python. the journal of machine learning research 12 (2011).