professionals@DravidianLangTech-EACL2021: Malayalam Offensive Language Identification - A Minimalistic Approach

Srinath Nair*
International Institute of Information Technology, Hyderabad
srinath.nair@research.iiit.ac.in

Dolton Fernandes*
International Institute of Information Technology, Hyderabad
dolton.fernandes@research.iiit.ac.in

Abstract

This paper is a description of the system that was designed by team "professionals" for the task Offensive Language Identification in Dravidian Languages-EACL 2021. Our system Dravidian Offensive Language Identifier Classifier (DrOLIC) uses Indic-BERT to generate word embeddings which is then fed into a 4-layer Multi Layer Perceptron (MLP) which does the multi-class classification task. The system helped us achieve an F1 score of 85% in the shared task for the Malayalam language.[1]

1 Introduction

Offensive language identification is a very popular multi-class classification research problem in NLP. A proper solution to the research problem could give the world ways and means to identify and restrict offensive content on the internet, especially on public platforms and social media. The rising demand for a system to identify offensive language on social media and in multiple languages is a testimony to the relevance of the Offensive Language Identification in Dravidian Languages task (Chakravarthi et al., 2021).

Dravidian languages are spoken in South Asia, especially in the Indian subcontinent and are a family of multiple languages including Tamil, Malayalam and Kannada, and more. Research on NLP in Indian languages was largely focused on Hindi which is spoken by a large section of people in India. Dravidian languages have now recently started getting attention from the research community which is viewed as a positive change.

In this work, we propose Dravidian Offensive Language Identifier Classifier $(DrOLIC)$, a 4-layered Multi Layer Perceptron (MLP) to do the multi-class classification task of identifying offensive language for the Malayalam language. The reason for choosing a simple MLP over more complex neural networks was because of the simplicity that we see in an MLP. Our initial experimentation with MLP gave us pretty good results which were much better than what we were initially expecting out of a simple system like a Multi Layer Perceptron. This made us want to stick to the system and explore how well it would perform under different conditions.

2 Related Work

A lot of work has been put into identification of offensive language task in recent years with an aim to enable machines and platforms to automatically filter content, especially in social media platforms. SemEval Task-6 (Zampieri et al., 2019) is a notable shared task done in this direction where participants identified and categorized offensive language in social media. The dataset contained over 14,000 tweets in English and was divided into three sub-tasks. The sub-task A required participants to identify offensive language while sub-task B was an automatic categorization and sub-task C required participants to identify the offense target. The top teams experimented extensively with LSTM (Hochreiter and Schmidhuber, 1997), pre-trained BERT (Devlin et al., 2018) models, pre-trained GloVe vectors (Pennington et al., 2014) to achieve the results.

With the evolution of technology it has been made possible to prepare and circulate content in different languages and recent years have seen a lot of importance being given to Indian languages. The availability of content in Indian languages brought in a need to expand the task of offensive language identification into Indian languages. This resulted in much work being done in the domain, but we can

*Indicates equal contribution

[1]Link to code: https://github.com/snath99920/Professionals-EACL-2021

Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pages 175–179

see most work clearly leaning towards Hindi and a few other languages spoken mostly in the Northern and Central parts of India. One such work does an interesting comparative study of offensive and aggressive language in Hindi, Bangla and English (Kumar et al., 2021). They have used SVM, BERT and its derivatives like ALBERT (Lan et al., 2019) and DistilBERT (Sanh et al., 2019) to develop the classifiers. The performance of the classifiers were judged based on F1 score where they managed to achieve an F1 score as high as 0.80 using BERT.

HASOC (Hate Speech and Offensive Content identification in Indo-European Languages) track at FIRE 2019 (Mandl et al., 2019) and Fire 2020 (Mandl et al., 2020) is another such interesting initiative. The HASOC dataset was prepared from publicly available posts from Twitter Facebook and allows the participants to develop supervised models. The limitation to this task again lies in the fact that the task and the dataset are specific to three languages: English, Hindi and German.

3 System Description

We approach the problem as a multi-class classification of embeddings and for this we present the *DrOLIC* architecture as depicted in figure 1. For every comment represented as an indic-BERT embedding, we pass it through a series of dense layers with ReLU activation to learn the representations. In order to make the layers do the learning more independently and avoid overfitting of the model, we introduce batch normalization and dropout layers respectively. To do the classification we introduce a dense softmax layer to output class probabilities.

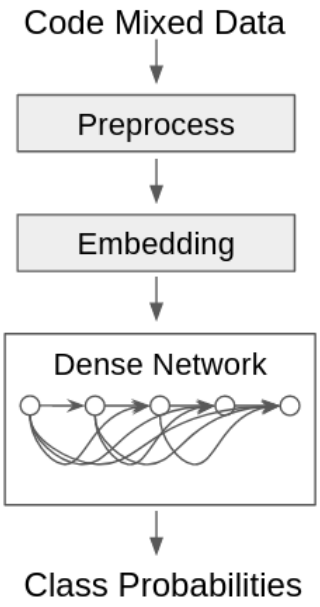

Figure 1: Overview of our method.

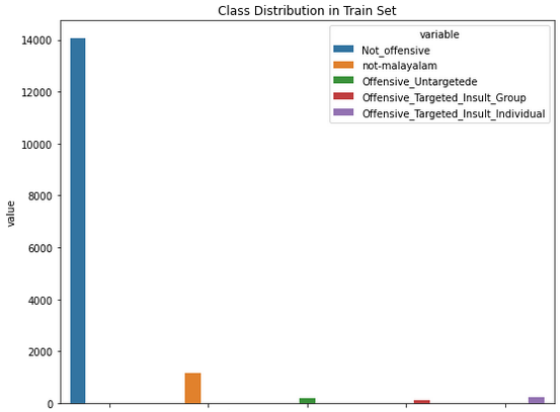

Figure 2: Visualization of train data.

4 Data and Experiments

4.1 Dataset

The data (Chakravarthi et al., 2021) (Chakravarthi et al., 2020a) (Chakravarthi et al., 2020b) (Hande et al., 2020) that was made available by the organizers of the task consisted of English-Malayalam code-mixed text. The data was prepared by collecting comments appearing on the trailers of various Malayalam movies on YouTube. The data was present in the form of a CSV file with the sentences belonging to the classes "Not offensive", "Offensive targeted insult group", "Offensive targeted insult individual", "Offensive untargeted" and "Not Malayalam". As one can see in figures 2, the training set has a noticeable class imbalance with most of the sentences belonging to the "Not offensive" class.

4.2 Data Preprocessing

The sentences in the dataset are code-mixed in Malayalam-English. We first transliterated the code-mixed sentences to English with the help of indic transliterator (Bhat et al., 2015). The sentences were then processed to remove hashtags, emojis and other unrecognised symbols. Now, out of these processed sentences, those whose length had significantly reduced (we kept the threshold as 4 words) were removed from the train and validation sets.

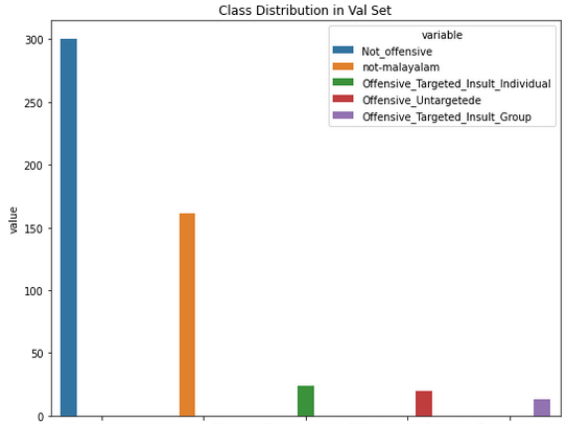

Figure 3: Visualization of validation data.

4.3 Generating Word Embeddings using Indic-BERT

IndicBERT (Kunchukuttan et al., 2020) (Kakwani et al., 2020), a multilingual NLU model that was pretrained on 12 Indian languages and evaluated on IndicGLUE. IndicBERT can be used for some of the most popularly spoken Dravidian languages including Malayalam, Tamil, Kannada, Telugu, etc. Notably, the model used here is ALBERT, a much more compact version of BERT with fewer parameters making it much more convenient to use.

For a comment X, we extract embeddings using indicBERT to get a 768 sized vector which is then used for training the MLP.

4.4 Model

We used a 4-layer Multi Layer Perceptron (MLP) to do the classification. The dimension and configuration of each layer of the model can be seen in table 1. We have 3 dense layers with ReLU activation followed by Batch Normalization layers to help the layers learn more independently. We add a dropout of 0.2 to two of the layers to reduce overfitting. At the end we have a dense layer with softmax activation to output the class probabilities. Our model has a total of 470,789 parameters out of which 469,381 are learnable.

4.5 Training

We first perform min-max scaling on the generated embedding vectors to not let any bias due to huge/small values propagate further into the network. The class labels are one hot encoded to make it suitable for the loss function. A random stratified split of 0.1 is done on the training data to get a validation set for our model. We don't use the original validation set provided for training in any way. It was only used to check our model's performance before submitting the final results.

layer	input size	output size	activation	parameters
$dense_1$	768	512	$relu$	393728
BN_1	512	512	–	2048
$dense_2$	512	128	$relu$	65664
BN_2	128	128	–	512
$dropout_1$	128	128	–	0
$dense_3$	128	64	$relu$	8256
BN_3	64	64	–	256
$dropout_2$	64	64	–	0
$dense_4$	64	5	$softmax$	325

Table 1: The detailed architecture of our model used for the multiclass classification of embeddings.

As mentioned earlier, there was a huge class imbalance in the data provided. To not let this affect our model, we kept only 300 samples from the class with the highest samples in the validation set and transferred them to the the training set. The final training and validation set distribution are shown in figures 2 and 3 respectively.

The model is trained using Adam optimizer with a learning rate of $1e-5$ and a decay of $1e-4$ for 100 epochs. The learning rate was modified according to the following rule:

$$lr_i = \frac{initialLR}{1 + decay * i} \tag{1}$$

where lr_i is the learning rate at the i^{th} iteration, $initialLR$ is the initial learning rate, $decay$ is the decay factor and i is the iteration number.

We use the categorical cross-entropy loss for evaluating how good our method is. The model computes categorical cross-entropy loss L between the predicted class probabilities and the correct class for the sample, as given below:

$$L(y, \bar{y}) = \sum y_i.log(\bar{y}_i) \tag{2}$$

where y_i is the i^{th} value in the model output y and $\bar{y}_i$ is the i^{th} value in the target $\bar{y}$.

4.6 Evaluation Metric

The organisers of the shared task used weighted average F1 score for getting an overall quality for the classification, so we used the same metric to seek a balance between precision and recall.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3}$$

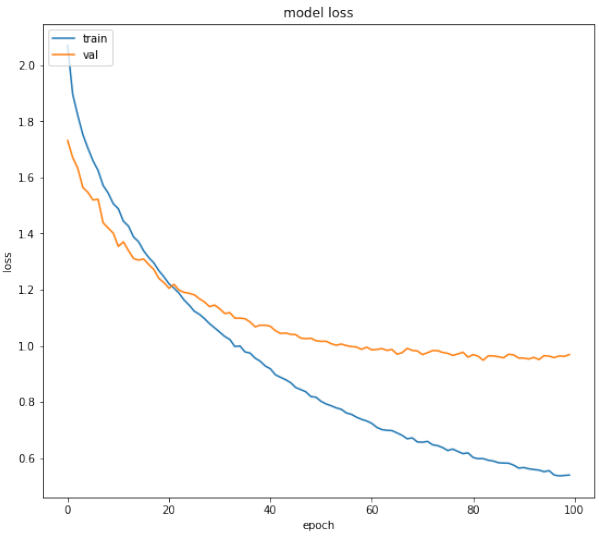

Figure 4: Loss

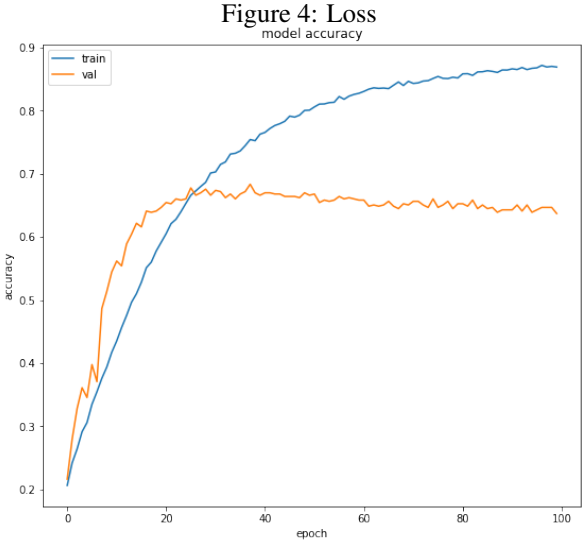

Figure 5: Accuracy

So the weighted F1 score is:

$$\tilde{F1} = \sum_{i=1}^{n} p_i x_i \quad (4)$$

where p_i is the class probability for class i and n is the total number of classes.

5 Results and Analysis

The results of our method stands at 85% in the contest for the Malayalam language, but we have fine tuned our model since then and the latest results are reported here. The original test set was used for evaluation. The loss and accuracy curves for the training can be seen in figures 4 and 5 respectively. We noticed that MLP stops learning after some epochs, which is bound to happen because MLP's don't take care of vanishing gradients unlike for eg. LSTM.

Sr. No.	Model	Precision	Recall	F1
1.	$MLP + Pr$	0.90	0.87	0.88
2.	$MLPw/oPr$	0.88	0.83	0.85

Table 2: The model descriptions and the results arranged in descending order of test F1 score. Here Pr stands for preprocessing.

From table 2 we can see the importance of text processing. Without preprocessing we get a F1 score of 85%, but on preprocessing the text we get a F1 score of 88%.

6 Conclusion

This shared task is one of the first in the area of offensive language identification in Dravidian languages. With the advent of a need for localization of the internet by introducing content in multiple languages, this shared task is a positive step.

The most interesting aspect of our work lies in the fact that we have used a simple 4-layered MLP to train the multi-class classifier. While keeping the system, we managed to get an F1 score as high as 0.85 and achieve a shared 11^{th} position in the shared task. The sentences in the dataset were represented as IndicBERT embeddings. The use of IndicBERT gives us an added advantage by letting us use the same setup for around 12 popularly spoken Indian languages. To overcome the challenges of using a codemixed dataset, we transliterated the sentences into Malayalam before generating the IndicBERT embeddings. The system proposed by us can be further improved especially because, like we have mentioned earlier, this is a very simple setup. The first step that can be taken is replacing the MLP with an LSTM or an SVM which could give a significantly better result.

7 Acknowledgments

We would like to begin by thanking Dr. Radhika Mamidi from Language Technology Research Center (LTRC), IIIT Hyderabad for her support and guidance throughout the course of the shared task in both building the system and writing the paper for review. We would also like to thank the organizers of the Offensive Language Identification in Dravidian Languages for bringing us this opportunity. Lastly, we would like to thank the anonymous reviewers for their quality suggestions that helped us shape the paper better and for all their constructive feedback.

References

Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shrivastava. 2015. Iiit-h system submission for fire2014 shared task on transliterated search. In *Proceedings of the Forum for Information Retrieval Evaluation*, FIRE '14, pages 48–53, New York, NY, USA. ACM.

Bharathi Raja Chakravarthi, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John Philip McCrae. 2020a. A sentiment analysis dataset for code-mixed Malayalam-English. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 177–184, Marseille, France. European Language Resources association.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John Philip McCrae. 2020b. Corpus creation for sentiment analysis in code-mixed Tamil-English text. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 202–210, Marseille, France. European Language Resources association.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Navya Jose, Anand Kumar M, Thomas Mandl, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Hariharan V, Elizabeth Sherly, and John Philip McCrae. 2021. 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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Adeep Hande, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. 2020. KanCMD: Kannada CodeMixed dataset for sentiment analysis and offensive language detection. In *Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media*, pages 54–63, Barcelona, Spain (Online). Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of EMNLP*.

Ritesh Kumar, Bornini Lahiri, and Atul Kr. Ojha. 2021. Aggressive and offensive language identification in hindi, bangla, and english: A comparative study. *SN Computer Science*, 2(1):26.

Anoop Kunchukuttan, Divyanshu Kakwani, Satish Golla, Gokul N. C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Thomas Mandl, Sandip Modha, Anand Kumar M, and Bharathi Raja Chakravarthi. 2020. 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*, FIRE 2020, page 29–32, New York, NY, USA. Association for Computing Machinery.

Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. 2019. 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*, FIRE '19, page 14–17, New York, NY, USA. Association for Computing Machinery.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.