Transformer Ensemble System for Detection of Offensive Content in Dravidian Languages

B S N V, Chaitaya[1], Karri, Anjali[1]

[1]*Indian Institute of Information Technology, SriCity, India*

Abstract

Hate speech is a form of oral, written or physical activity that criticizes or uses derogatory language in correspondence to a person or a community discriminating their identity factors. Hate speech or the use of offensive language can endanger democratic principles and societal stability. The growing usage of social media is also increasing the number of people being affected by hate speech. Online hate speech moderation has been significantly increasing, especially through social media platforms like Facebook, Twitter, YouTube, and Instagram. It is high time to take appropriate actions to curb the intensifying online hate speech by supporting the detection of hate speech or offensive language texts in social media. The work presented to Hate Speech and Offensive Content Identification in Dravidian-CodeMix (HASOC) 2021, a joint assignment under Forum for Information Retrieval Evaluation (FIRE) 2021, is described in this paper. In this paper, we proposed an ensemble system of transformer models (mBERT, DistilBERT and MuRIL) to achieve the task of identifying social media code-mixed comments/posts in Dravidian Languages (Malayalam-English and Tamil-English) as offensive or not-offensive texts. The motivation behind this was to use the power of transformers in combination with ensembling to enhance the prediction quality. For sub-task 2, the proposed ensemble method received 3rd and 6th positions in Malayalam and Tamil languages, respectively. The code is publicly available at https://github.com/chaitnayabasava/HSU_TransEmb.

Keywords

Hate speech, Offensive Language, BERT Transformers, Ensemble, CodeMix

1. Introduction

Social media platforms offer users freedom of expression. Simultaneously, they also bring up new challenges in terms of freedom of expression, speech, and human dignity. Hate speech on the internet is the expression of tensions between various groups and can also have a detrimental impact on society. Hate speech expressed through social media is not inherently different from hate expressed outside, but it could have specific difficulties stemming from its indefiniteness, durability, and anonymity. Hate speech in online venues may persist in many formats across several platforms, and it can be connected multiple times. Counteracting hate speech in the internet world demands more thought and innovative strategies. Social media platforms such as Youtube, Facebook, and Twitter each have algorithms for identifying hate speech. Nonetheless, identifying and classifying hate speech is still a significant issue for social media firms alongside researchers.

FIRE 2021: Forum for Information Retrieval Evaluation, December 13-17, 2021, India

viswachaitanya.b16@iiits.in (B. S. N. V. Chaitaya); anjalipoornima.k16@iiits.in (K. Anjali)

India being a diverse country, most of the Indians mix up different languages with English while communicating. In the multilingual community, code-mixing is common, and code-mixed writings are occasionally produced in non-native scripts. Due to the convenience of using local languages alongside English, code-mixed languages are becoming increasingly popular on different social media platforms. However, ambiguity is introduced by spelling variances and the absence of grammatical standards, making it increasingly arduous to automate text analysis. We can observe a growing demand for offensive language identification, especially on social media messages, which are mostly code-mixed. Many researchers have been looking into varying algorithms for detecting hate speech, and most of the studies concentrated on monolingual text data. But, due to the intricacy of code-mixing, models trained on monolingual data commonly fail when tried on code-mixed data.

Therefore, as part of HASOC 2021 [1], we developed a classification model to identify offensive texts in code-mixed Dravidian languages. HASOC 2021 has two sub-tasks and this paper provides the working notes on sub-task 2, which involves categorizing the given code-mixed tweet as offensive or non-offensive. The evaluation metric reported and considered for model selection in this paper is the weighted average F1-score. The competition page and reference document [1] provide further information on the challenges. We organized the rest of the paper as follows: section 2 highlights the relevant work, section 3 details the proposed technique, section 4 depicts the experiments and outcomes, and section 5 concludes the article and summarises our findings.

2. Related Work

The task of hate-speech detection is often treated as a text classification task. Using machine learning or deep learning approaches to detect offense, hostility, and hate speech in user-generated content is one of the most effective strategies for combating this problem. As indicated by recent articles, this topic has got a lot of attention recently. Few survey articles that describe significant areas that have been investigated for this task include are as follows. [2] represents a survey covering the important areas that were investigated for employing natural language processing to automatically recognize various types of utterances. [3] looked at strategies for detecting hate speech in social media and separating it from ordinary obscenities. The findings showed that the most difficult part is distinguishing between profanity and hate speech. [4] examined the complexities of the concept of hate speech, which is defined differently across platforms and settings, and offers a unified definition.

In the literature, a number of distinct classifiers have been used in various works. [5] was one of the earliest research in the problem of hate speech detection. The authors developed a prototype for detection abusive messages using a decision-tree generator with 47-features corresponding to the syntax and semantics. Later, machine learning classification methods like SVM and logistic regression were used to tackle the task of hate speech detection. For instance, [6] used logistic regression to perform obscenity-related offensive tweets detection. [7] constructed machine learning models like a Support Vector Machine with a linear kernel, and a Random Forest with 100 trees to identify cyber hate for a range of protected traits such as race, disability, and sexual orientation to facilitate the automatic detection of cyber hate online, specifically on Twitter. The feature set used by them included Bag of Words, features obtained

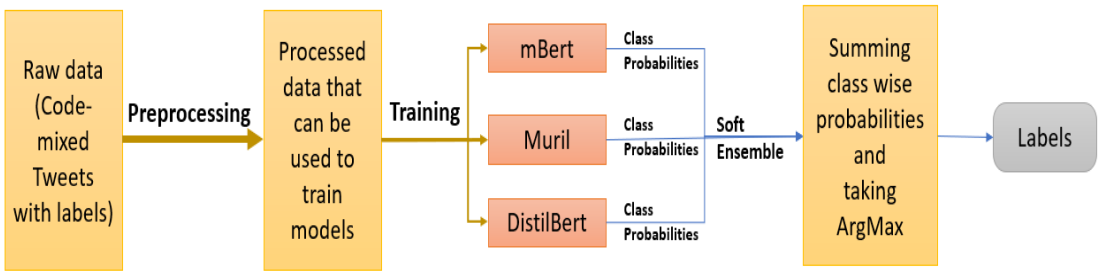

Figure 1: The architecture of the proposed transformers ensemble with all the training steps included. The Raw data is firstly pre-processed as explained in section 3.1 and used to train 3.3 the models described in 3.2. The class probabilities from each of these models are further used to build a soft ensemble to predict the final class labels.

by identifying hostile words and phrases for hate speech and typed dependencies. Although bag-of-words methods have a high recall rate, they also have a high incidence of false positives. [8] and [9] developed Convolution network-based models to achieve Hate-speech detection. [8] trained different CNN models with different sets of features like 4-grams, word2vec word vectors, word vectors which are randomly generated and combination of word vectors with n-grams. With 78.3% F-score, CNN model with word2vec features vectors performed best. [9] experimented with 16k annotated dataset and used features by coupling the embedding learned by deep neural network models and gradient boosted decision trees.

[10] and [11] employed Recurrent Neural networks namely LSTM and BiLSTM. The former authors implemented SVM and LSTM for Hate-speech detection in Italian language, using morpho-syntactic and syntactic features, sentiment polarity, and lexical text features. The later, experimented with Convolutional Networks, BiLSTM and Convolutional Networks with BiLSTM to identify postings indicating the user's use of medicine. Frequently, a single solution to a complicated problem does not apply to all possible circumstances. As a result, researchers employ ensemble methods to solve such issues. Thus, [11] and [12] addressed this classification task using ensembles with stacked deep learning CNN ensembles and an ensemble of Recurrent Neural Network classifiers respectively. Therefore, taking inspiration of using deep learning techniques and ensembles, in this paper we proposed an ensemble of transformers [13]. The power of pre-trained transformers was harnessed by BERT. BERT is a pre-trained model on unlabelled text corpus which can further be fine-tuned for specific tasks like classification. [14] presented an overall idea of all the methods and results for Offensive Language Identification in Dravidian Languages-EACL 2021. [15] provided an overall idea of the task of hate speech recognition in Tamil, Malayalam, Hindi, English and German as part of the HASOC track at FIRE 2020. The authors of [16] worked to compare different pretrained text embeddings to classify hate speech in Indian Code-Mixed sentences.

3. Methodology

To achieve the task of classifying the code-mixed tweet as an offensive or not-offensive tweet, we proposed an ensemble model of transformers. This section elaborates on the dataset and its pre-processing steps, subsequently explaining the ensemble setting. Figure 1 depicts the architecture of the proposed Transformer Ensemble model for classifying offensive tweets using the dataset given by the organizers of HASOC 2021 [1].

3.1. Pre-Processing

The phase of pre-processing is very crucial, especially while working with tweets. Unprocessed tweets are unstructured, often containing redundant information and noise that could mislead predictions. We processed the tweets by transforming them to lower case and subsequently tokenized each tweet. Tokenization converts a tweet into words, punctuation marks, numeric digits, and other symbols. These tokenized tweets were further processed by removing the punctuation's since they do not add much information to the underlying content. Tweets mostly go with the # and @ handles, which would not help us much in modelling and may lead to biases that ultimately hamper the predictions. We removed digits, URLs, # and @ handles using regex expressions. Emojis and emoticons have become an integral part of our everyday lives and frequently appear in social media texts. We also removed these symbols and characters during pre-processing. The categories of the given dataset are also not uniform. We dropped data points with labels: not-Tamil and not-Malayalam. Finally, we trained different models using cleaned tweets with two labels, namely 'NOT' and 'OFF'.

3.2. Models

To build our ensemble model, we majorly worked with three different transformer-based models, namely multilingual BERT (mBERT), Multilingual Representations for Indian Languages (MuRIL) and Distilled BERT (DistilBERT). [17] has marked the use of transformer models with encoder-decoder blocks using attention maps for long sequence tasks. The goal of transformers is to completely manage the dependencies between input and output using attention maps and recurrent networks. Bidirectional Encoder Representations from Transformers, Google's BERT [13] has paved the way for a new era of using transfer learning in NLP. This language model is built with a multi-layer bidirectional Transformer encoder along with bi-directional self-attention layers. It enables the users to fine-tune the pre-trained language model to achieve state-of-the-art performance in many NLP-related tasks like question answering, translation, classification, etc. BERT's pre-training objectives, Masked Language Modelling (MLM) and Next Sentence Prediction, are straightforward yet effective. The MLM masks tokens in the input randomly and, the goal of the model is to predict the masked tokens. The next-sentence prediction makes sure that the model understands the connection between consecutive sentences. Thus these unsupervised pre-training objectives made BERT a powerful pre-trained model for language representations.

The original pre-trained models of Google BERT have been trained on lower-cased English text. Since our task was the classification of tweets in code-mixed Dravidian languages, we tried

Table 1
Different transformer models have been trained and evaluated using dev and test sets. Weighted F1-score on dev and test sets of each considered model are tabulated here. Among the five models, the top three best performed models (mBERT, MuRIL and DistilBERT) are used to build the soft ensemble model.

Model	Tamil dev	Tamil test	Mal dev	Mal test
mBERT	0.92	0.65	0.76	0.73
distillBERT	0.90	0.63	0.75	0.69
MuRIL	0.92	0.66	0.78	0.73
indicBERT	0.84	0.62	0.70	0.72
xlm-Roberta	0.83	0.63	0.72	0.65

Table 2
The weighted F1-score on dev and test sets of top performing models and the proposed ensemble model (HSU_TransEmb) are tabulated. An increase in the scores for dev set is observed.

Model	Tamil dev	Tamil test	Mal dev	Mal test
mBERT	0.92	0.65	0.76	0.73
distillBERT	0.90	0.63	0.75	0.69
MuRIL	0.92	0.66	0.78	0.73
HSU_TransEmb	0.93	0.65	0.80	0.73

to use other BERT models from HuggingFace [18] that were pre-trained in different languages. The mBERT is the original BERT base model pre-trained on the top 102 languages, including Tamil and Malayalam. The model is pre-trained with the same MLM objective as BERT with the Wikipedia corpus. mBERT develops complex cross-lingual representations that enable language transfer of code-mixed tweets more efficiently. [19] proposed a lighter version of BERT which reduced the number of parameters by 40% preserving 97% of the language representations knowledge and increasing the computation speed by 60%. DistilBERT is a lighter and faster transformer model with a triple loss combining the language modelling, distillation of the BERT base, and cosine-distance. For the proposed ensemble model, we used the multilingual DistilBERT model, having 6 transformer layers with 12 attention heads and 134M parameters in total and is a distilled version of mBERT. [20] proposed MuRIL, a multilingual Language model that was trained specifically on a large corpus of 17 Indian Languages. This model was designed to perform a range of fine-tuned NLP tasks in Indian languages. This model is also trained on transliterated data, which is a regular occurrence in the Indian environment and can help in improving the performance of the classification task in Dravidian languages.

3.3. Training

Hugging Face pre-trained transformer models have been used to build models for the fine-tuning task of tweet classification. The outputs of the last hidden layer of the corresponding models are averaged and used as the final feature representation of the tweet. This representation is finally passed through an output layer with output dimensions equal to the number of classes,

two. We used a batch size of 32 with a max sequence length of 256 and trained the classifier by monitoring the cross-entropy loss, which increases when the prediction diverges from the ground truth.

The dataset provided by Chakravarthi et al. [1] has a slight imbalance issue between the two available classes ('OFFENSIVE', 'NOT-OFFENSIVE'). The Malayalam dataset has 2047 not-offensive and 1953 offensive tweets whereas, the Tamil dataset has 2020 not-offensive and 1980 offensive tweets. To address this imbalance, we used inverse weighting to penalize the incorrect predictions of the lower-represented class more in the cross-entropy loss function. Finally, we trained the models with a learning rate of 1e-5 for 30 epochs.

3.4. Ensemble of Transformers

We employed a voting soft ensemble model for getting the final predictions. As discussed in section 3.3, we fine-tuned each considered model using the provided dataset. The motivation behind using an ensemble voting mechanism is to have a system that combines the outputs of various BERT based models to give the final predictions. The base models were trained using varying amounts of data and transformer layers, resulting in each model identifying different patterns from the text. By using the ensemble setting, we can capture and use these multiple patterns to give the final prediction. This setting has helped us improve the performance above the F1-score of the best performing model amongst the considered one's.

In the proposed soft voting ensemble setting, the prediction probabilities of each model are averaged as shown in eq 1. The final prediction then comes from using eq 2, where p^i_{not} is the probability of the comment being 'NOT-OFFENSIVE' predicted by model i and n is the total number of models considered for the ensemble setting.

$$p^{ensemble}_{not} = \frac{1}{n}\sum_{i=0}^{n-1} p^i_{not} \tag{1}$$

$$pred = \begin{cases} NOT, & \text{if } p^{ensemble}_{not} > 0.5 \\ OFF, & \text{else} \end{cases} \tag{2}$$

4. Experiments and Results

We considered various transformer-based multilingual models, trained with datasets containing the two languages (Tamil and Malayalam) in focus and fine-tuned them using the provided dataset using the training setup described in section 3.3. To apply the pre-trained models of BERT, we first need to tokenize the input using the Bert Tokenizers. These tokenizers split the input text into tokens and add tokens like [CLS] and [SEP] used to indicate the start and end of sentences. We considered the max length as 256 so that the input sentences are padded or truncated to this length. Lastly, the attention mask is created and returned along with the tokenized input. The classifier is fed the average of features from the last hidden layer of the BERT model and fine-tuned using Adam optimizer with weight decay with a learning rate of 1e-5. We trained the models with a batch size of 32 for 30 epochs each.

Table 1 summarizes the individual model's performance using the weighted F1-score. The three considered models (mBERT, MuRIL & DistilBERT) were the top performers on the dev set in both the Dravidian languages and so were used in the ensemble setting, described in section 3.4.

Table 2 compares the weighted F1-score obtained by using the three considered models both individually and in the ensemble setting. We observe that the proposed ensemble model improved the overall performance in both the Dravidian languages on the dev set. But the performance of all the models has deteriorated drastically on the test set, especially in Tamil. We may address this by using a vast dataset that covers varying patterns in the code-mixed text.

5. Conclusion

We proposed an ensemble transformer model that utilized various transformers trained on multilingual data to identify hate speech and offensive language in the Dravidian languages, Tamil and Malayalam. The proposed ensemble model was able to outperform the standalone models on the dev set. Yet, the F1-score of all the models is very low on the provided test set. The poor performance may be mainly be attributed to the change in distribution from the train and dev sets. In future work, we will consider using multiple open-sourced Hate speech recognition code-mix datasets along with the provided dataset to cover various possible data patterns. We will also explore the effects of using language-specific LSTM based models like ULMFit [21].

References

[1] 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.

[2] 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, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 1–10. URL: https://aclanthology.org/W17-1101. doi:10.18653/v1/W17-1101.

[3] S. Malmasi, M. Zampieri, Detecting Hate Speech in Social Media, in: Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, INCOMA Ltd., Varna, Bulgaria, 2017, pp. 467–472. URL: https://doi.org/10.26615/978-954-452-049-6_062. doi:10.26615/978-954-452-049-6_062.

[4] P. Fortuna, S. Nunes, A Survey on Automatic Detection of Hate Speech in Text, ACM Computing Surveys (CSUR) 51 (2018) 1 – 30.

[5] E. Spertus, Smokey: Automatic Recognition of Hostile Messages, in: Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence, AAAI'97/IAAI'97, AAAI Press, 1997, p. 1058–1065.

[6] G. Xiang, B. Fan, L. Wang, J. Hong, C. Rose, Detecting Offensive Tweets via Topical Feature Discovery over a Large Scale Twitter Corpus, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12, Association for Computing Machinery, New York, NY, USA, 2012, p. 1980–1984. URL: https://doi.org/10.1145/2396761.2398556. doi:10.1145/2396761.2398556.

[7] P. Burnap, M. L. Williams, Us and them: identifying cyber hate on Twitter across multiple protected characteristics, EPJ Data science 5 (2016) 1–15.

[8] B. Gambäck, U. K. Sikdar, Using Convolutional Neural Networks to Classify Hate-Speech, in: Proceedings of the First Workshop on Abusive Language Online, Association for Computational Linguistics, Vancouver, BC, Canada, 2017, pp. 85–90. URL: https://aclanthology.org/W17-3013. doi:10.18653/v1/W17-3013.

[9] 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.

[10] F. Del Vigna12, A. Cimino23, F. Dell'Orletta, M. Petrocchi, M. Tesconi, Hate me, hate me not: Hate speech detection on facebook, in: Proceedings of the First Italian Conference on Cybersecurity (ITASEC17), 2017, pp. 86–95.

[11] D. Mahata, J. Friedrichs, R. R. Shah, et al., # phramacovigilance-Exploring Deep Learning Techniques for Identifying Mentions of Medication Intake from Twitter, arXiv preprint arXiv:1805.06375 (2018).

[12] G. K. Pitsilis, H. Ramampiaro, H. Langseth, Detecting offensive language in tweets using deep learning, arXiv preprint arXiv:1801.04433 (2018).

[13] 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).

[14] 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.

[15] 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.

[16] S. Banerjee, B. Raja Chakravarthi, J. P. McCrae, Comparison of Pretrained Embeddings to Identify Hate Speech in Indian Code-Mixed Text, in: 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 2020, pp. 21–25. doi:10.1109/ICACCCN51052.2020.9362731.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[18] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-Art Natural Language Processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural

Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.
[19] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, 2020. arXiv:1910.01108.
[20] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. T. Nagipogu, S. Dave, S. Gupta, S. C. B. Gali, V. Subramanian, P. Talukdar, MuRIL: Multilingual Representations for Indian Languages, 2021. arXiv:2103.10730.
[21] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, arXiv preprint arXiv:1801.06146 (2018).