OFFLangOne@DravidianLangTech-EACL2021: Transformers with the Class Balanced Loss for Offensive Language Identification in Dravidian Code-Mixed text.

Suman Dowlagar
LTRC
IIIT-Hyderabad
suman.dowlagar
@research.iiit.ac.in

Radhika Mamidi
LTRC
IIIT-Hyderabad
radhika.mamidi
@iiit.ac.in

Abstract

The intensity of online abuse has increased in recent years. Automated tools are being developed to prevent the use of hate speech and offensive content. Most of the technologies use natural language and machine learning tools to identify offensive text. In a multilingual society, where code-mixing is a norm, the hate content would be delivered in a code-mixed form in social media, which makes offensive content identification, further challenging. In this work, we participated in the EACL task to detect offensive content in the code-mixed social media scenario. The methodology uses a transformer model with transliteration and class balancing loss for offensive content identification. In this task, our model has been ranked 2^{nd} in Malayalam-English and 4^{th} in Tamil-English code-mixed languages.

1 Introduction

Language is a social phenomenon. It is through language that day-to-day interactions and interpersonal relations are possible (Barnali et al., 2017). Languages keep on changing and adapting. In a multilingual scenario, the languages influence each other in certain ways. Normally, this interaction is reflected in language convergence, borrowing, and replacement. It also leads to the emergence of hybrid languages, such as pidgins, creoles, and other mixed languages. This form of language interaction is known as Language contact (Thomason, 2001). Language contact is considered to be an important phenomenon, especially in multilingual societies. In bilingual or multilingual communities, speakers use their native tongue and their second language in different domains. This form of alternation of two or more languages is called 'code-mixing' (CM) (Muysken et al., 2000).

With the increase in social media access, Offensive content and hateful material on the internet has increased in the recent past (Thavareesan and Mahesan, 2019, 2020a,b). The internet harbors a variety of hateful and offensive statements, and nowadays, social media is a hotbed of such conversations. Recently, countries across the world have already begun to address hate speech and offensive content and how it affects society's functioning (Chakravarthi, 2020). Research and technologies worldwide are utilizing natural language and machine learning tools to detect and curb the use of offensive content on social media.

In a multilingual society, code-mixing has become a norm. The hateful and offensive content is delivered in a code-mixed form (Jose et al., 2020; Priyadharshini et al., 2020). Automatic hate speech detection on code-mixed data has faced quite many challenges due to the non-standard variations in spelling and grammar (Bali et al., 2014). The typical hate-speech and offensive language tools developed for monolingual data will not work for code-mixed data. So there is a need for more research and analysis to be done to identify the offensive content in code-mixed social media data.

To encourage research on code-mixing and restrain the use of offensive texts on social media, the NLP community has organized several workshops such as Workshops on Computational Approaches to Linguistic Code-Switching, SentiMix (Patwa et al., 2020), Dravidian CodeMix (Chakravarthi et al., 2020d), HASOC Dravidian CodeMix [1] (Chakravarthi et al., 2020b; Mandl et al., 2020). Similarly, the European Association of Computational Linguistics 2021's DravidianLangTech (Chakravarthi et al., 2021) was also devoted to identifying offensive content on Kannada-English, Tamil-English, and Malayalam-English code-mixed languages. This task aims to classify the given code-mixed com-

[1] https://sites.google.com/view/dravidian-codemix-fire2020/overview

Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pages 154–159
April 20, 2021

ments into one of the six predefined categories: Not-offensive, offensive-untargeted, Offensive-Targeted-Insult-Individual, Offensive- Targeted-Insult-Group, Offensive-Targeted-Insult-Other, or Not-in-indented-language.

This paper presents a pre-trained BERT model with the class balanced loss for offensive content identification on the Dravidian code-mixed text.

The paper is organized as follows. Section 2 provides related work on offensive content identification on CM social media text. Section 3 provides information on the task and datasets. Section 4 describes the proposed work. Section 5 presents the experimental setup and the performance of the model. Section 6 concludes our work.

2 Related Work

This section describes the related work on hate speech detection and offensive content identification in the code-mixed scenario.

Bohra et al. (2018) created a dataset for hate speech detection from Hindi-English tweets. They collected around 4575 Hindi-English tweets and used traditional machine learning models with feature engineering for hateful and offensive content identification. Mandl et al. (2019) created a Hindi-English dataset for a hate speech and offensive content identification Task (HASOC), organized at FIRE 2019. It consists of 4665 Hindi-English annotated posts collected from social media sites. They used Twitter API for crawling an unbiased dataset. The top models used the Long Short Term Memory (LSTM) (Schmidhuber and Hochreiter, 1997) with attention mechanism, pre-trained Bidirectional encoder representations with transformers (BERT) (Devlin et al., 2018) models, and convolutional neural networks (CNN) for hate speech and offensive content identification. Kumar et al. (2018) created the dataset for an aggression detection task. The dataset was annotated for a comparison task by (Rani et al., 2020). This data set consists of 3367 posts and tweets collected from social media sites. The authors used traditional machine learning classification models for hate speech detection. A shared task called Dravidian Code-Mix (Chakravarthi et al., 2020b) was organized to identify the offensive language from comments/posts in code-mixed Dravidian Languages (Tamil-English and Malayalam-English) collected from social media. Each comment/post is annotated with the offensive language label at the comment/post level. The data set has been collected from YouTube comments and Tweets. The dataset contains 4000 annotated Youtube comments and 4952 annotated tweets for the Malayalam language, and 4940 annotated tweets for the Tamil language. The top models used deep learning models like LSTM, CNN, and BERT for hate speech and offensive content detection on the given Dravidian data.

3 Task and Dataset information

The goal of offensive language identification is to identify the offensive language content of the code-mixed dataset of comments/posts in Tamil-English (Chakravarthi et al., 2020c), Malayalam-English (Chakravarthi et al., 2020a), and Kannada-English (Hande et al., 2020) Dravidian languages collected from social media. Each comment or post is annotated with offensive, not-offensive, and not-in-intended-language labels. Where the offensive label is fine-grained into further categories. The description of each label is given below

- Offensive-targeted-individual: offensive text delivered to an individual or person
- Offensive-targeted-group: offensive text delivered to a group of people
- Offensive-targeted-other: offensive text delivered to topics such as films, elections, sports, and so on.
- Offensive-untargeted: the offensive text is delivered but without targeting any person or a topic.
- Not-in-indented-language: the given content is not in the intended language.
- Not-offensive: the post does not contain any offensive content

The dataset for offensive content identification into divided into train, development(dev), and test sets for the given languages.

The details of the dataset are given in table 1.

4 Our work

In this section, we start with pre-processing of code-mixed text. Later we describe the pre-trained multilingual BERT model with the class balanced loss for offensive content identification.

Data	#train	#dev	#test	#total
Kannada-English CM	6217	777	778	7772
Malayalam-English CM	16010	1999	2001	20010
Tamil-English CM	35138	4388	4392	43919

Table 1: Data Statistics

4.1 Pre-processing

The given code-mixed dataset depicts real-time scenarios of variations in the spelling, script changes, use of hashtags, mentions, and emoticons in the text and has imbalance problems. So pre-processing is necessary for such a dataset. During pre-processing,

- To resolve the ambiguities resulting from script change, we back-transliterated the script to the native language. As the data has Dravidian and English comments, we used the NLTK[2] English word corpus to detect if the word is in English or not. Later we back-transliterated the word to its native script. We applied linguistic rules not to transliterate the tweet/comment "not-Kannada/not-Tamil/not-Malayalam" as they are not in the intended language.
- We removed all the punctuations, URLs, mentions, unwanted numbers, and emoticons from the given dataset. We accepted repetitions of characters up to a length of 2 and removed others.

4.2 Our proposed model

In our approach, we have used multilingual pre-trained BERT with the class balanced loss on the transliterated data.

We have used multilingual pre-trained BERT (Devlin et al., 2018) in this work because it is a transformer-based self multi-headed attention model that is pre-trained on a huge collection of data and can be finetuned for our offensive content classification task. This kind of transfer learning is very successful when we want to learn a classifier from a small set of data by taking advantage of pre-trained embeddings.

BERT is a non-regressive model that reads the whole string of terms present in the text in a single stretch. BERT analyzes the meaning of a term depending on its context given on both sides. As they are pre-trained on a large corpus, the semantic and syntactic information is well modeled and can be directly finetuned for a specific task.

The transformer part in the BERT works like an attention mechanism capable of learning the contextual relationships between the terms in a sentence. The basic form of transformer consists of an encoder and a decoder. The encoder part reads the text as the input, and the decoder part gives the corresponding predictions.

4.3 Class balanced loss to handle dataset imbalance

While handling an imbalanced dataset (one with most of the samples belonging to very few of the classes and many other classes with very few instances), loss calculation can be tricky. The most common approach to balance the loss is assigning weights to the loss. The weights are calculated as the inverse of the number of class instances or inverse of the square root of the number of class instances. This form of weighing scheme creates the problem by shifting focus entirely to the classes with very few instances.

To handle the shift, the authors Cui et al. (2019) proposed a Class-Balanced Loss based on Effective Number of Samples. A framework to measure data overlap by associating each sample to a small neighboring region rather than a single point. The effective number of samples is defined as the volume of samples and can be calculated by a simple formula $(1-b^n)/(1-b)$, where n is the number of samples and b is a hyper-parameter, and it takes values between [0,1]. The authors designed a re-weighting scheme that uses sufficient samples for each class to re-balance the loss, thereby yielding a class-balanced loss.

$$\mathrm{CB}(\mathbf{p}, y) = \frac{1}{E_{n_y}}\mathcal{L}(\mathbf{p}, y) = \frac{1-\beta}{1-\beta^{n_y}}\mathcal{L}(\mathbf{p}, y) \tag{1}$$

Here, $\mathcal{L}(\mathbf{p}, y)$ can be any loss function.n_y is the number of estimated samples for each labels y.

[2] https://www.nltk.org/

Malayalam-English CM data	Accuracy	macro-F1	Weighted-F1
SVM	0.96	0.78	**0.96**
mBERT	0.68	0.38	0.70
mBERT 'balanced'	0.68	0.38	0.70
mBERT 'transliterated'	0.96	0.78	**0.96**
Our approach	**0.97**	**0.80**	**0.96**

Table 2: Classification metrics for Malayalam-English CM data

Tamil-English CM data	Accuracy	macro-F1	Weighted-F1
SVM	0.77	0.43	**0.80**
mBERT	0.76	0.45	0.78
mBERT 'balanced'	0.76	0.46	0.78
mBERT 'transliterated'	**0.78**	0.47	**0.80**
Our approach	**0.78**	**0.48**	**0.80**

Table 3: Classification metrics for Tamil-English CM data

Kannada-English CM data	Accuracy	macro-F1	Weighted-F1
SVM	0.70	0.43	0.73
mBERT	**0.74**	0.44	**0.77**
mBERT 'balanced'	**0.74**	**0.45**	**0.77**
mBERT 'transliterated'	0.66	0.42	0.64
Our approach	0.66	0.43	0.65

Table 4: Classification metrics for Kannada-English CM data

5 Experiments

The section presents the baselines, hyper-parameter settings, and analysis of observed results.

The baselines used for the proposed work is:

1. **SVM with TF-IDF** Term frequency and inverse document frequency-based vectorization is used to represent the text data, and the support vector machine is used to classify the data.

2. **Pre-trained multilingual BERT (mBERT)** A pre-trained multilingual BERT model with a feed-forward network for classification.

3. **mBERT with class balanced loss (mBERT "balanced")** A pre-trained multilingual BERT model with a feed-forward network and class balanced loss is used for classification.

4. **mBERT with transliteration (mBERT "transliterated")** A pre-trained multilingual BERT model with a feed-forward network with transliterated data is used for classification

5.1 Hyperparameters and libraries used

During pre-processing, we have used a deep transliteration tool known as ai4bharat-transliteration[3] library. We have used SVM with TF-IDF vectorization from the scikit-learn library (Pedregosa et al., 2011). The default parameters are used to train the SVM for multi-class classification. The multilingual pre-trained BERT is obtained from huggingface transformers library (Wolf et al., 2019) and is finetuned for this sentence classification task. The optimizer used is weighted Adam with the learning rate of 2e-5 and epsilon value equal to 1e-8. The loss function used is a cross-entropy loss. The number of epochs used for training the model is 30.

5.2 Results and Analysis

Tables 2, 3 and 4 presents the f1-score and accuracy of the models on the Dravidian code-mixed datasets.

From the above results, for Malayalam and Tamil datasets, it is clear that our approach of the multilingual pre-trained BERT model with the class

[3] https://pypi.org/project/ai4bharat-transliteration/

balanced loss and transliteration works best for the given datasets. It is due to the effectiveness of the pre-processing and class balanced loss. Removing URLs, punctuations and emojis features helped the BERT model to focus on the relevant information. As the script is associated with the embeddings, back-transliteration helped the BERT model distinguish between the native script and English words that improved the model's accuracy. The class balanced loss statistically estimated the weightage of each label and helped the model not to favour the label with maximum instances.

Our approach for Kannada-English CM dataset didn't give best results. It might be due to the problems given below,

1. The transliteration tool didn't function well for the Kannada data.

2. There were more English words in the data, with small spelling variations or abbreviations, that are not detected by the NLTK corpus and transliterated to Malayalam script.

5.3 Conclusion

We used pre-trained multilingual bi-directional encoder representations using transformers (BERT) for offensive content identification given the Kannada-English, Malayalam-English, and Tamil-English code-mixed datasets. We compared the BERT with traditional machine learning classification methods with and without class balanced loss. The results showed that using the back-transliteration helped the module to obtain the nativeness of script and class balanced loss handled the problem of imbalanced data. During back-transliteration, we observed that the data has many spelling variations for the same word. So the back-transliteration had many instances of the same word with small variations. In the future, we wish to normalize these transliterations based on the context, such models, if developed, will help in handling the code-mixed real-time data better. It would also be an interesting study to analyze the effects of different loss functions in our model on given imbalanced data.

References

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. "I am borrowing ya mixing?" An Analysis of English-Hindi Code Mixing in Facebook. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 116–126.

Chetia Barnali et al. 2017. Code-Switching and Mixing in Communication- A Study on Language Contact in Indian Media. In *The Future of Ethics, Education and Research*, pages 110–123. Scientia Moralitas Research Institute.

Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A dataset of Hindi-English code-mixed social media text for hate speech detection. In *Proceedings of the second workshop on computational modeling of people's opinions, personality, and emotions in social media*, pages 36–41.

Bharathi Raja Chakravarthi. 2020. HopeEDI: A multilingual hope speech detection dataset for equality, diversity, and inclusion. In *Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media*, pages 41–53, Barcelona, Spain (Online). Association for Computational Linguistics.

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, M Anand Kumar, John Philip McCrae, Premjith B, Soman KP, and Thomas Mandl. 2020b. Overview of the track on HASOC-Offensive Language Identification-DravidianCodeMix. In *Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2020). CEUR Workshop Proceedings. In: CEUR-WS. org, Hyderabad, India*.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John Philip McCrae. 2020c. 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.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Shardul Suryawanshi, Navya Jose, Elizabeth Sherly, and John P. McCrae. 2020d. Overview of the Track on Sentiment Analysis for Dravidian Languages in Code-Mixed Text. In *Forum for Information Retrieval Evaluation*, FIRE 2020, page 21–24, New York, NY, USA. Association for Computing Machinery.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9268–9277.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv: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.

Navya Jose, Bharathi Raja Chakravarthi, Shardul Suryawanshi, Elizabeth Sherly, and John P. McCrae. 2020. A Survey of Current Datasets for Code-Switching Research. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 136–141.

Ritesh Kumar, Aishwarya N Reganti, Akshit Bhatia, and Tushar Maheshwari. 2018. Aggression-annotated corpus of Hindi-English code-mixed data. *arXiv preprint arXiv:1803.09402*.

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*, pages 14–17.

Pieter Muysken, Pieter Cornelis Muysken, et al. 2000. *Bilingual speech: A typology of code-mixing*. Cambridge University Press.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas PYKL, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. Semeval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. *arXiv e-prints*, pages arXiv–2008.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, Mani Vegupatti, and John P. McCrae. 2020. Named Entity Recognition for Code-Mixed Indian Corpus using Meta Embedding. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 68–72.

Priya Rani, Shardul Suryawanshi, Koustava Goswami, Bharathi Raja Chakravarthi, Theodorus Fransen, and John Philip McCrae. 2020. A comparative study of different state-of-the-art hate speech detection methods in hindi-english code-mixed data. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 42–48.

Jürgen Schmidhuber and Sepp Hochreiter. 1997. Long short-term memory. *Neural Comput*, 9(8):1735–1780.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2019. Sentiment Analysis in Tamil Texts: A Study on Machine Learning Techniques and Feature Representation. In *2019 14th Conference on Industrial and Information Systems (ICIIS)*, pages 320–325.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2020a. Sentiment Lexicon Expansion using Word2vec and fastText for Sentiment Prediction in Tamil texts. In *2020 Moratuwa Engineering Research Conference (MERCon)*, pages 272–276.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2020b. Word embedding-based Part of Speech tagging in Tamil texts. In *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, pages 478–482.

Salah G Thomason. 2001. *Language contact*. Citeseer.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910.