Hypers@DravidianLangTech-EACL2021: Offensive language identification in Dravidian code-mixed YouTube Comments and Posts

Charangan Vasantharajan
Department of Computer Science and Engineering University of Moratuwa
Colombo, Srilanka
Charangan.18@cse.mrt.ac.lk

Uthayasanker Thayasivam
Department of Computer Science and Engineering University of Moratuwa
Colombo, Srilanka
rtuthaya@cse.mrt.ac.lk

Abstract

According to the use of Code-Mixed Offensive contents in social media posts, this study has the focus on such contents in low-resourced Dravidian languages such as Tamil, Kannada, and Malayalam using the bidirectional approach and fine-tuning strategies. Our proposed model got a 0.96 F1-score for Malayalam, 0.73 F1-score for Tamil, and 0.70 F1-score for Kannada based on the benchmark leader-board. Moreover, in the view of multilingual models, our modal ranked $3rd$ and achieved favorable results and confirmed the model as the best among all systems submitted to these shared tasks in these three languages.

1 Introduction

The use of social media platforms (e.g., Facebook, Twitter, and YouTube) has become an important activity in the everyday lives of most internet users. The users share their thoughts with multilingual societies in the form of comments and posts on social media. In the last few years, with a proliferation of social media users, posts and comments have become an information form that is simply called **code-mixed** text (Chakravarthi et al., 2020; Androutsopoulos, 2013; Chakravarthi et al., 2019) due to the lack of impressions in a single language when conveying their thoughts online with the multi-linguistic community (Suryawanshi et al., 2020).

Most of the social media platforms encourage users to convey and express their opinions in their preference with a small number of restrictions and intend to collect user comments and posts to provide a customized feed. In addition to this, these collected data are also used for advertising, marketing, and detecting the occurrence of bad activities. It is rather difficult to identify offensive code-mixed content written in a non-native (Roman) script (Rosowsky, 2010). This complexity in such execution of the natural language processing tasks cannot be solved by the technologies developed for a monolingual text (Solorio et al., 2014; Diab et al., 2014) but multilingual models can.

Offensive language identification of code-mixed data on social media platforms is important to identify the movie reviews, product reviews, enable advertisements, monitor political activities, and identify social trends. The code-mixed data always adopts the vocabulary and grammar of multiple languages and generates new user-based content structures (Choudhary et al., 2018). This is challenging for offensive language identification tasks as traditional and unidirectional approaches and lack of dataset result in not capturing the meaning and limits the accuracy of the task.

According to the shared task, this study executed a comment or post level offensive language identification of the code-mixed contents which was collected from YouTube comments and posts on **Dravidian language** pairs (e.g. Tamil-English, Kannada-English, and Malayalam-English). Each of given sentence is annotated with labels as follows: Not-offensive, Offensive Targeted Insult Other, Offensive Targeted Insult Individual, Offensive Targeted Insult Group, not Tamil or not Malayalam or not Kannada, and Offensive Untargeted. This research work proposed a system that uses **M-BERT** which stands for Multilingual-Bidirectional Encoder Representations with Transformers (Devlin et al., 2019), as a pre-trained model and applied fine-tuning strategies as a transfer learning approach to solve the given task.

The rest of the sections in the paper are as follows. Section 2, reviews related experiment works in offensive language identification in three views. Section 3 describes the methodology with system architecture which includes the classification features and transfers learning strategies. The fourth section presents the conducted experiments using

Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pages 195–202

different approaches and features. Section 5 analyses the results of the proposed system in the benchmark. Benchmark results are discussed in section 6 and finally, the conclusion followed by future research directions.

2 Related Work

Due to the code-mixed dataset limitation on the low-resourced languages such as Tamil, Sinhala, Malayalam, and Kannada, there were a few numbers experiments have been done in this area (Jose et al., 2020; Hande et al., 2020; Chakravarthi et al., 2020; Ghanghor et al., 2021b,a; Puranik et al., 2021; Hegde et al., 2021; Yasaswini et al., 2021). Arora (2020) developed a model to identify Hate Speech and Offensive Contents in Dravidian languages (e.g. Tamil-English and Malayalam-English) from the code-mixed comments and posts collected from social media using pre-trained ULMFiT on synthetically generated code-mixed data. Their proposed method achieved 0.88 and 0.91 weight F1-score for code-mixed Tamil-English and Malayalam-English, respectively. On the other hand, some tasks have been done based on dataset creations to provide support for low-resource language identification tasks. To encourage sentiment analysis tasks and overcome the non-availability of the code-mixed dataset in Tamil-English Chakravarthi et al. (2020) created a sentiment-annotated corpus containing 15744 YouTube comments and posts. In addition to this, they examined the benchmark system with various machine learning algorithms such as Logistic Regression (LR), Support Vector Machine (SVM), K-Nearest Neighbour (K-NN), Decision Tree (DT), Random Forest (RF), Multinominal Naive Bayes (MNB), 1DConv-LSTM, BERT Multilingual, DME and CDME. In this task, the Logistic regression, random forest classifiers, and decision trees were provided better results compared to other algorithms. Jose et al. (2020) build a standard corpus for Malayalam-English to increase the sentiment analysis tasks in the code-mixed contents. Thayasivam and Smith (2019) created a newly code-mixed dataset annotated in the sentence and word level collected from Facebook comments, chat history, and from public posts for Sinhala-English language pairs. Hande et al. (2020) explored in Kannada-English by the traditional learning approaches such as Logistic Regression (LR), Support Vector Machine (SVM), Multinomial Naive Bayes (MNB), K-Nearest Neighbors (KNN), Decision Trees (DT), and Random Forest (RF).

In high-resource languages, Priyadharshini et al. (2020) examined methods using Transformer models to predict the named entities in corpus written both in the native script as well as in the Roman script for the English-Hindi language pairs. On the other hand, the explorations for English-Spanish (Solorio et al., 2014), Chinese-English (Lee and Wang, 2015), and English-Bengali (Patra et al., 2018) language pairs were used multiple-classifier based automatic detection approach, unidirectional LSTM, and common generic forms of Deep Learning and Neural Network approaches to perform their identification tasks.

According to the social media explosion, offensive language identification has become an increasingly important task and a few studies have been published on single languages (Thavareesan and Mahesan, 2019, 2020a,b). Malmasi and Zampieri (2017) performed an analysis to separate the general profanity from the hate speech in social media. They presented a supervised classification system that used character n-grams, word n-grams, and word skip grams and their model was able to achieve 80% accuracy on a huge dataset which contained English tweets annotated with three labels, namely, "hate speech (HATE)", "offensive language but no hate speech (OFFENSIVE)", and "no offensive content (OK)". Pitenis et al. (2020) addressed the problem of offensive language identification for the Greek language by created their annotated corpus using comments retrieved from Twitter Posts and experimented with seven different classification models: Pooled GRU (Plum et al., 2019), Stacked LSTM with Attention (Plum et al., 2019), LSTM and GRU with Attention (Plum et al., 2019), 2D Convolution with Pooling, GRU with Capsule, LSTM with Capsule and Attention and BERT (Devlin et al., 2019). In this task, LSTM and GRU with Attention performed well compared to all other models in-terms of macro-f1. However, the fine-tuning approach with the BERT-Base Multilingual Cased model did not provide good results.

Based on the above analysis, most of the approaches were unidirectional-based except the works done by (Chakravarthi et al., 2020) and (Pitenis et al., 2020). They used BERT (Devlin et al., 2019) as their bidirectional algorithm but it did not

reach the state-of-the-art on code-mixed contents at that time. After a lot of studies and observations, BERT is chosen for this low-resource language experiment as a challenge.

3 Methodology

3.1 Data Preprocessing

Two data pre-processing techniques followed and kept them as minimal methods to suit all other languages. Emojis play a key role in expressing emotions in the posts and comments of social media (Hettiarachchi and Ranasinghe, 2019). The better approach to deal with emojis is to convert emojis to words so that it is being helpful to preserve information. So converting Emojis into text is the first preprocessing technique. For this purpose, it is used a dictionary of emojis[1] to do this conversation due to the unconfirmed existence of embeddings for emojis in the pre-trained model. After that, it is discarded a list of punctuation that is available in Python's string module from the input text, based on the use case.

3.2 BERT Transformer

BERT is a transformer-based, multi-layer, and bidirectional model with an attention mechanism that has the learning ability to contextual relations between words and sub-words in a sequence or text. The default form of the transformer contains two separate objects such as an encoder and a decoder. The encoder reads the input text and the decoder makes predictions for the given classification task (Devlin et al., 2019). BERT overcome the limitation which is in the previous language models (e.g. **word2vec** and **GloVe**) when interpreting context and polysemous words and it effectively performs in monolingual as well as multilingual classifications and leads to the greatest performance increase in the natural language processing task, including Question Answering (SQuAD v1.1), Natural Language Inference (MNLI), and others (Devlin et al., 2019). Apart from the unidirectional language models, transformer encoders read the whole input sequence of words at once. Therefore, BERT is the best bidirectional transformer rather than bidirectional LSTMs and would perform with high accuracy. This behavior leads the transformer model to learn the context of a word from left to right and right to left of the word at the same time (Devlin et al., 2019).

Primarily, the task of this research is multilingual offensive language identification, therefore it is used a pre-trained transformer-based multilingual BERT model (Devlin et al., 2019) and implemented using **HuggingFace**. The pre-trained models are available in the HuggingFace model repository [2]. BERT supports 104 languages including Tamil, Kannada, and Malayalam. It includes approximately 110M parameters with 12-layers, 768 hidden-states, and 12-heads (Gopalan and Hopkins, 2020). Normally, BERT fetches input data in a specific format, with special tokens to mark the beginning ([CLS]) and separation/end of sentences ([SEP]) to mark the end. Furthermore, it is necessary to tokenize the text into tokens that correspond to BERT's vocabulary. For each tokenized sentence, BERT requires input ids, a sequence of integers identifying each input token to its index number in the BERT tokenizer vocabulary. For a given token, its input representation is built by adding the related token, segment, and position embedding (Devlin et al., 2019). A visual representation of the BERT model can be seen in the Figure 1.

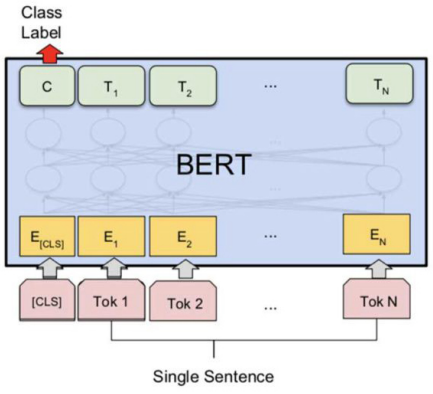

Figure 1: BERT model designed for Text classification tasks for a single sentence input. [CLS] and [SEP] are special and separate tokens.

3.3 Fine-tuning

The main idea of the fine-tuning strategies is to load a pre-trained (**bert-base-multilingual-cased**) classification model which is trained on more than the top 100 languages in the largest Wikipedia (Devlin

[1] Bhavika Kanani, 'Text Preprocessing: Handle Emoji & Emoticon', *Machine Learning Tutorials*, 2020, https://studymachinelearning.com/text-preprocessing-handle-emoji-emoticon/, (accessed 14 January 2021).

[2] https://github.com/huggingface/transformers

et al., 2019), add or remove layers, prepare multiple inputs, and adjust parameters such as a learning rate, the number of epochs, the optimizer, and regularization parameters. According to the recent studies, some fine-tuning strategies have ensured results such as self-ensemble, and language modeling and Hindi, and Bengali. Therefore, this research work examined this strategy to see whether it improves the results for low-resource languages too.

BERT is adopted for the given multilingual offensive language identification task and performs fine-tuning from the pre-trained **bert-base-multilingual-cased** classification model which is trained on more than the top 100 languages with the largest Wikipedia. TFBertModel, the mother of all different BERT classification models[3] is chosen since the task is to classify posts and comments into six distinct classes as above indicated. **BertTokenizerFast** was loaded from the same transformer library[4] to tokenize the input texts and prepare some additional inputs than *input_ids* such as *attention_mask* which leads to performance increase and *token_type_ids*. Dropouts were applied to the BERT layers and passed *TruncatedNormal* distribution as *kernel_initializer* to the output dense layer. Finally, the system was trained on the Tamil, Kannada, and Malayalam languages separately using the training datasets provided by the organizers (Chakravarthi et al., 2021; Jose et al., 2020; Chakravarthi et al., 2020; Hande et al., 2020). The system architecture is shown in Figure 2.

4 Experiments

4.1 Data

For this study, it is used separate larger datasets with manually annotated labels for training, development, and testing for each language introduced by (Chakravarthi et al., 2021; Jose et al., 2020; Chakravarthi et al., 2020; Hande et al., 2020) as described in section 1. The given dataset was divided into training and validation sets using a 0.8:0.2 split during the training. Dataset statistics and the data visualization according to the labels are shown in the table 1 and Figure 3.

[3] https://huggingface.co/transformers/model_doc/bert.html

[4] https://huggingface.co/transformers/model_doc/bert.html#berttokenizerfast

Languages	Train	Dev	Test
Tamil	35139	4388	4392
Kannada	6217	777	778
Malayalam	16011	2000	2002

Table 1: Dataset statistics according to the languages. As you can see, there is a huge difference in the number of comments and posts between Tamil and the other two languages.

4.2 Implementation

Fine-tune was mainly done on the learning rate and the number of epochs of the classification model manually to obtain the best results for the validation set. It is obtained that $5e-05$ as the best value for the learning rate for the **Adam** optimizer and 50 as the best value for the epochs for all the languages. It took around 1 hour to train the Tamil model due to a large amount of training dataset and other language models took only around 30 minutes. The research model of this study achieved 0.7645, 0.7154, and 0.9634 validation accuracy in Tamil, Kannada, and Malayalam respectively during the training period. After evaluating it by using the development dataset, the testing dataset was used to make predictions.

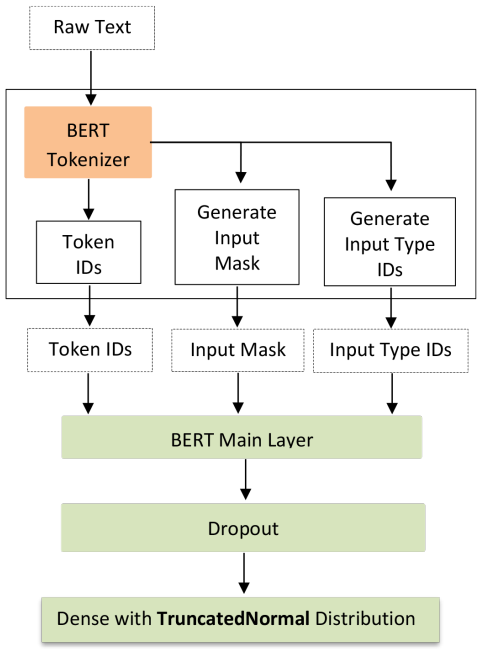

Figure 2: Architecture of our proposed model. BERT Tokenizer converts the word into tokens and generates token ids, input masks, and input type ids according to the input word. The main goal of using an input mask is to gain some performance increase. In addition to the best layer, we added a dropout and pooled output layer.

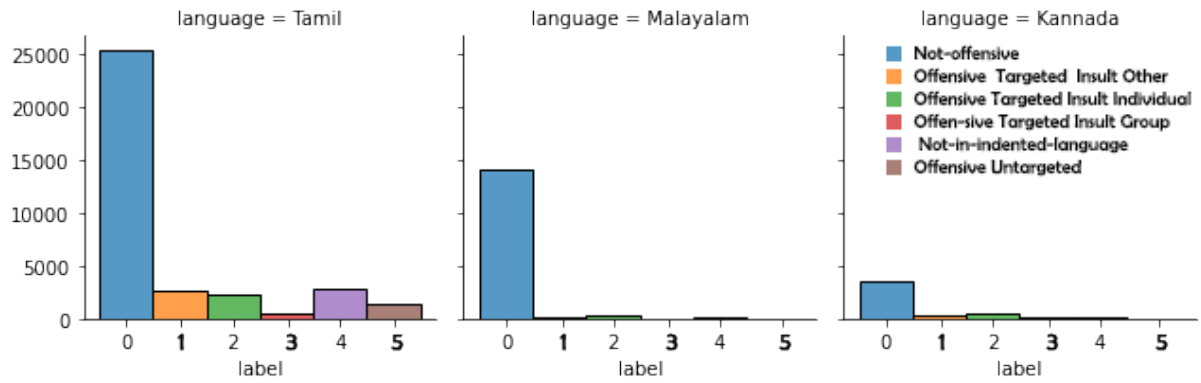

Figure 3: Dataset labels distribution on Tamil, Malayalam, and Kannada code-mixed content. the x-axis 0,1,2,3,4, and 5 indicates Not-offensive, Offensive Targeted Insult Other, Offensive Targeted Insult Individual, Offensive Targeted Insult Group, Not-in-indented-language, and Offensive Untargetede respectively.

5 Results and Analysis

Teams are ranked by the weighted average F1 score of their classification system. According to the models which were submitted to all three languages, we have found ourselves as 3^{rd} in the order of models. This shows that the research model of this study achieved state-of-the-art results in all three languages. Moreover, table 2 illustrate our ranks in the competition with their weighted Precision, weighted Recall, and weighted F1-score. According to the results published in the benchmark leader-board, the Malayalam, Kannada, and Tamil language models of this study have scored 0.96, 0.70, and 0.73 weighted average F1 scores on the test set and ranked 2^{nd} out of 30 participants, 5^{th} out of 29 participants, 6^{th} out of 31 participants respectively.

Even though the research model of this study got the above rank, the F1-score difference between the first team and **Hypers** is only 1%, 7%, and 6% respectively. Therefore, it is assumed that the proposed model of this study provided better results compared to other models for monolingual languages and also multilingual languages.

Language	P	R	F1	Rank
Malayalam	0.96	0.96	0.96	2
Kannada	0.69	0.72	0.70	5
Tamil	0.71	0.76	0.73	6

Table 2: Leaderboard - Offensive Language Identification in Dravidian Languages-EACL 2021. This shows our ranks in all three languages in the benchmark.

6 Discussions

Even with the better F1-scores for all three models, there was some confusion according to the results. F1-score for Malayalam is comparatively higher than the other two models. According to the dataset, the Tamil training set has approximately 35139 posts and comments, in the meantime, Kannada and Malayalam are 6214, and 16011 respectively but given results behaves as opposed to the number of posts and comments in the training sets. It may have happened due to the lack of training of the BERT multilingual based model in the Tamil and Kannada language. Since BERT multilingual model was trained on Wikipedia and analyzed wiki contents[5] and got the total contents in Kannada, Malayalam, and Tamil are 26,769, 71,762, and 134,011 respectively. The lower f1 score for Kannada could be attributed to the less training data. On the contrary, despite having more training data, BERT trained on Tamil and Malayalam performs poorly.

In addition to the above Wikipedia analysis, it was carried out a qualitative analysis to find patterns and observations on the given dataset. Commonly, in the training dataset, more than 80% of the posts and comments were labeled as Not-offensive and the remaining were labeled as other types of labels. The use of English words was minimal although there are many comments and posts which are in the native language but written in Roman script. Therefore, it is assumed that the lack of training in those informal forms of languages led to such results in the benchmark evaluation.

[5] https://meta.wikimedia.org/wiki/List_of_Wikipedias

7 Conclusion

Offensive Language Identification in Dravidian Language code-mixed text is one of the most challenging tasks in natural language processing. We have done such a task in this study using BERT with a fine-tuning strategy in the low-resourced languages. In this benchmark, our (**Hypers** team) proposed system got 2^{nd}, 5^{th}, 6^{th} rankings for the Malayalam, Kannada, and Tamil test set respectively. Finally, it was proved that the BERT system can be used on low-resource languages to the offensive language identification tasks with the best F1- score.

8 Future Work

In the future, we are planning to use character-level embedding along with the word embeddings to get a better representation of an input sentence and words. On the other hand, misspelled or other words not appearing in the vocabulary are usually detected and treated as a special unknown token, so this will also help in getting a representation for out-of-vocabulary (OOV) words. Moreover, due to the size of vocabularies, the word embeddings requires a lot of memory to store embeddings, and including word embeddings in a model adds too many parameters to the model, so the computational cost is much higher on it than the character-level model.

Also, it is decided to deal with imbalanced data in the training and development sets by applying oversampling to increase the number of samples for minority classes. It is assumed that the above approaches will increase the model performance.

9 Acknowledgments

We would like to thank the EACL organizers for running this interesting shared task and for replying promptly to all our inquiries. We further thank the three anonymous reviewers for their insightful suggestions and feedback.

References

Jannis Androutsopoulos. 2013. Code-switching in computer-mediated communication. *Pragmatics of Computer-mediated Communication*, pages 667–694.

Gaurav Arora. 2020. Gauravarora@HASOC-Dravidian-CodeMix-FIRE2020: Pre-training ULMFiT on Synthetically Generated Code-Mixed Data for Hate Speech Detection.

Bharathi Raja Chakravarthi, Mihael Arcan, and John P. McCrae. 2019. Comparison of Different Orthographies for Machine Translation of Under-Resourced Dravidian Languages. In *2nd Conference on Language, Data and Knowledge (LDK 2019)*, volume 70 of *OpenAccess Series in Informatics (OASIcs)*, pages 6:1–6:14, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

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

Nurendra Choudhary, Rajat Singh, Ishita Bindlish, and Manish Shrivastava. 2018. Sentiment analysis of code-mixed languages leveraging resource rich languages. *CoRR*, abs/1804.00806.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mona Diab, Julia Hirschberg, Pascale Fung, and Thamar Solorio, editors. 2014. *Proceedings of the First Workshop on Computational Approaches to Code Switching*. Association for Computational Linguistics, Doha, Qatar.

Nikhil Kumar Ghanghor, Parameswari Krishnamurthy, Sajeetha Thavareesan, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. 2021a. IIITK@DravidianLangTech-EACL2021: Offensive Language Identification and Meme Classification in Tamil, Malayalam and Kannada. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, Online. Association for Computational Linguistics.

Nikhil Kumar Ghanghor, Rahul Ponnusamy, Prasanna Kumar Kumaresan, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja

Chakravarthi. 2021b. IIITK@LT-EDI-EACL2021: Hope Speech Detection for Equality, Diversity, and Inclusion in Tamil, Malayalam and English. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, Online.

Vinay Gopalan and Mark Hopkins. 2020. Reed at SemEval-2020 task 9: Fine-tuning and bag-of-words approaches to code-mixed sentiment analysis. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1304–1309, Barcelona (online). International Committee for Computational Linguistics.

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.

Siddhanth U Hegde, Adeep Hande, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. 2021. UVCE-IIITT@DravidianLangTech-EACL2021: Tamil Troll Meme Classification: You need to Pay more Attention. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Hansi Hettiarachchi and Tharindu Ranasinghe. 2019. Emoji powered capsule network to detect type and target of offensive posts in social media. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 474–480, Varna, Bulgaria. INCOMA Ltd.

Navya Jose, Bharathi Raja Chakravarthi, Shardul Suryawanshi, Elizabeth Sherly, and John Philip McCrae. 2020. 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.

Sophia Lee and Zhongqing Wang. 2015. Emotion in code-switching texts: Corpus construction and analysis. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 91–99, Beijing, China. Association for Computational Linguistics.

Shervin Malmasi and Marcos Zampieri. 2017. Detecting hate speech in social media. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 467–472, Varna, Bulgaria. INCOMA Ltd.

Braja Gopal Patra, Dipankar Das, and Amitava Das. 2018. Sentiment Analysis of Code-Mixed Indian Languages: An Overview of SAIL_Code-Mixed Shared Task @ICON-2017. *CoRR*, abs/1803.06745.

Zesis Pitenis, Marcos Zampieri, and Tharindu Ranasinghe. 2020. Offensive language identification in Greek. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5113–5119, Marseille, France. European Language Resources Association.

A Plum, Tharindu Ranasinghe, Constantin Orasan, and R Mitkov. 2019. RGCL at GermEval 2019: offensive language detection with deep learning.

Ruba. Priyadharshini, Bharathi. R. 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.

Karthik Puranik, Adeep Hande, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. 2021. IIITT@LT-EDI-EACL2021-Hope Speech Detection: There is always hope in Transformers. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*. Association for Computational Linguistics.

Andrey Rosowsky. 2010. 'writing it in english': Script choices among young multilingual muslims in the uk. *Journal of Multilingual and Multicultural Development*, 31:163–179.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72, Doha, Qatar. Association for Computational Linguistics.

Shardul Suryawanshi, Bharathi Raja Chakravarthi, Mihael Arcan, and Paul Buitelaar. 2020. Multimodal meme dataset (MultiOFF) for identifying offensive content in image and text. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 32–41, Marseille, France. European Language Resources Association (ELRA).

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.

Uthayasanker Thayasivam and Ian Smith. 2019. Sinhala-english code-mixed data analysis: A review on data collection process. In *2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer)*, volume 250, pages 1–6.

Konthala Yasaswini, Karthik Puranik, Adeep Hande, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. 2021. IIITT@DravidianLangTech-EACL2021: Transfer Learning for Offensive Language Detection in Dravidian Languages. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.