IIITT@DravidianLangTech-EACL2021: Transfer Learning for Offensive Language Detection in Dravidian Languages

Konthala Yasaswini[1], Karthik Puranik[1], Adeep Hande[1],
Ruba Priyadarshini[2], Sajeetha Thavareesan[3], Bharathi Raja Chakravarthi[4]
[1]Indian Institute of Information Technology Tiruchirappalli, Tamil Nadu,
[2]ULTRA Arts and Science College, Tamil Nadu, India,
[3]Eastern University, Sri Lanka, [4]National University of Ireland Galway
konthalay18c@iiitt.ac.in

Abstract

This paper demonstrates our work for the shared task on Offensive Language Identification in Dravidian Languages-EACL 2021. Offensive language detection in the various social media platforms was identified previously. However, with the increase in the diversity of users, there is a need to identify the offensive language in multilingual posts which are largely code-mixed or written in a non-native script. We approach this challenge with various transfer learning-based models to classify a given post or comment in Dravidian languages (Malayalam, Tamil and Kannada) into 6 categories. The source codes for our systems are published [1].

1 Introduction

Over the past decade, there has been a tremendous increase in the user-generated content on social media platforms such as Twitter, YouTube, and Instagram (Wiedemann et al., 2020). They provide a common space for discussion and interactions, for users to connect with each other, express their opinions, and share their knowledge. Users may use offensive posts/comments which may be directed towards an individual or community(Chowdhury et al., 2020) which is one of the common problems in the online social media platforms (Nogueira dos Santos et al., 2018). They act as catalysts for leaving offensive content which could have a harmful and detrimental effect on users' mental health. The automatic detection of such malevolent comments/posts has become a crucial field of research in natural language processing in recent years(Wiedemann et al., 2019).

Tamil (ISO 639-1: ta), Malayalam (ISO 639-1: ml), and Kannada (ISO 639-3:kan) belong to the Dravidian languages, spoken mainly in India (Chakravarthi et al., 2019). The earliest inscription in India dated to 580 BCE was the Tamil inscription in pottery. A Tamil prayer book in ancient Tamil script called *Thambiran Vanakkam*, was written by Portuguese Christian missionaries in 1578, thereby rendering Tamil the first Indian language to be printed and published. One of the first dictionaries written in the Indian language was the Tamil Lexicon, published by the University of Madras. Tamil, Malayalam, and Kannada has its own script however users in the social media use the Latin script generating code-mixing (Chakravarthi et al., 2020c; Mandl et al., 2020). Code-mixing refers to the coupling of two or more languages in a single sentence (Priyadharshini et al., 2020). It is a quite common phenomenon observed in multilingual societies throughout the world (Chakravarthi, 2020; Bali et al., 2014; Jose et al., 2020). It is widely considered as a default mode of communication in countries like India and Mexico (Parshad et al., 2014; Pratapa et al., 2018; Chakravarthi et al., 2018). Code-mixed sentence maintains the fundamental grammar and script of the languages it is comprised of (Lal et al., 2019).

This paper is a description of our submission to the shared task for Offensive Language Detection (Chakravarthi et al., 2021). The task is to identify offensive content in the code-mixed comments/posts in the Dravidian languages collected from social media and classify it into Not Offensive, Offensive Untargeted, Offensive Targeted Insult Individual, Offensive Targeted Insult Group, Offensive Targeted Insult Other and Not in-indented-language.

The rest of the paper is organized as follows, Section 2 represents previous work on Offensive Language Detection in Dravidian Languages. Section 3 entails a detailed analysis of the datasets for Tamil, Malayalam, and Kannada. Section 4 presents a description of the models used for our purpose, while Section 5 explains the experiment setup for the models. Section 6 analyzes our re-

[1] https://github.com/adeepH/DravidianLangTech-OLD

Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pages 187–194
April 20, 2021 ©2021 Association for Computational Linguistics

sults achieved, and Section 7 presents the future direction for our work.

2 Related Work

The extensive use of offensive content on social media platforms is disastrous to an advancing society as it serves to promote violence, chaos, abuse, and verbal hostility and extremely affects individuals at distinct levels. Research in offensive language detection has been evolving rapidly over the past few years. Fortuna and Nunes (2018) gives an outline of the current state-of-the-art in offensive language detection and related tasks like hate speech detection. Davidson et al. (2017) introduced a publicly available dataset, notably for offensive language detection, by classifying tweets into hate speech, offensive but not hate speech, and neither. Several attributes like TF-IDF, n-grams, readability scores, and sentiment were used to build machine learning models such as logistic regression and Support Vector Machine in their work. A system combination of SVM and deep neural networks were developed by Hassan et al. (2020) for detecting abusive language which achieved F1-score of 90.51% on the test set.

Various experiments have been performed on code-mixed data. Kumar et al. (2018) developed numerous systems for detecting offensive language in Hindi and English which used data from Twitter and Facebook. Hindi-English Offensive Tweet (HEOT) dataset comprising of tweets in Hindi-English code mixed language classified into three classes; non-offensive, abusive, and hate-speech was introduced by Mathur et al. (2018). Their work utilized transfer learning wherein the model used Convolutional Neural Networks which was pre-trained on tweets in English followed by retraining on Hinglish tweets. Bohra et al. (2018) examined the problem of hate speech detection in code-mixed texts and presented a dataset of code-mixed Hindi-English comprising of tweets posted on Twitter. Hussein et al. (2020) presented a system, C-BiGRU, comprised of a convolutional neural network(CNN) along with a bidirectional recurrent neural network(RNN) to identify offensive speech on social media. An embedding model-based classifier to identify offensive language from Manglish dataset was developed in Renjit and Idicula (2020). Multimodal systems of Tamil troll memes were developed to classify memes that were deemed offensive towards other people (Suryawanshi et al., 2020; Hegde et al., 2021).

3 Dataset

The organizers provided us with Tamil-English (Chakravarthi et al., 2020b), Malayalam-English (Chakravarthi et al., 2020a) and Kannada-English (Hande et al., 2020) code-mixed text data derived from social media. The datasets comprised of all six types of code-mixed sentences : No-code-mixing, Inter-sentential Code-Mixing, Only Tamil/Kannada/Malayalam (written in Latin script), Code-switching at morphological level (written in both Latin and Tamil/Kannada/Malayalam script), Intra-sentential mix of English and Tamil/Kannada/Malayalam (written in Latin script only) and Inter-sentential and Intra-sentential mix (Hande et al., 2020). The training dataset consists of comments in six different classes:

- **Not-Offensive**: Comments which are not offensive, impolite, rude, or profane.
- **Offensive-Targeted-Insult-Individual**: offensive comments targeting an individual.
- **Offensive-Targeted-Insult-Group**: offensive comments targeting a group.
- **Offensive-Targeted-Insult-Other**: offensive comments targeting an issue, an organization, or an event other than the previous two categories.
- **Offensive-Untargeted**: offensive comments targeting no one.
- **Not-in-intended-language**: comments not in Tamil/Malayalam/Kannada.

Label	Tamil	Malayalam	Kannada
NO	25,425	14,153	3,544
NIL	1,454	1,287	1,522
OTI	2,343	239	487
OTG	2,557	140	329
OTO	454	-	123
OU	2,906	191	212
Total	**35,139**	**16,010**	**6,217**

Table 1: Class distribution for Training set in Tamil, Malayalam and Kannada. **NO**-Not offensive, **NIL**-Not in indented language, **OTI**-Offensive-Targeted-Insult-Individual, **OTG** - Offensive Targeted Insult Group, **OTO**- Offensive Targeted Insult Other, **OU**- Offensive Untargeted

Table 1 shows the class distribution in Tamil, Malayalam, and Kannada training datasets. The imbalance of the dataset depicts a realistic picture observed on social media platforms.

4 System Description

We use pre-trained transformer models for classifying offensive speech in Tamil, Kannada, and Malayalam. We do not perform text preprocessing techniques such as lemmatization, stemming, removing sinp words, etc, to preserve context to the users' intent. Since we use transformer models, it is observed that stop words receive a similar amount of attention as non-stop words, as transformer models are contextual models (BERT, XLM-RoBERTa, etc).

4.1 CNN-BiLSTM

This is a hybrid of bidirectional LSTM and CNN architectures (Chiu and Nichols, 2016). The convolutional neural network extracts character features from each word. The Convolutional neural network extracts feature vector from character-level feature. For each word, these vectors are concatenated and fed to the BiLSTM network and then to the output layers. CNN-BiLSTM, along with Doc2Vec embedding achieved very high results for sequence classification tasks (Rhanoui et al., 2019), thus we use GLoVE embedding along with CNN BiLSTM.

4.2 mBERT

Multilingual models of BERT (mBERT) (Pires et al., 2019) are largely based on the architecture of BERT (Devlin et al., 2019). This model was pretrained using the same pretraining strategy that was employed to BERT, i.e, Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). It was pretrained on the Wikipedia dump of top 104 languages. To account for the data imbalance due to the size of Wikipedia for a given language, exponentially smoothed weighting of data was performed during data creation and wordpiece vocabulary creation. This results in high resource languages being under-sampled, while low resourced languages being over-sampled.

4.3 XLM-RoBERTa

XLM-RoBERTa (Conneau et al., 2020) is a large multi-lingual language model, trained on 2.5TB of cleaned CommonCrawl data in 100 languages. It can be recognized as a union of XLM (Lample and Conneau, 2019) and RoBERTa (Liu et al., 2019). The training process involves sampling streams of text from different languages and masking some tokens, such that the model predicts the missing tokens. Using SentencePiece (Kudo and Richardson, 2018) with a unigram language model (Kudo, 2018) subword tokenization is directly applied on raw text data. Since there are no language embeddings used, this allows the model to better deal with code-switching. XLM-RoBERTa manifested remarkable performance in various multilingual NLP tasks.

4.4 DistilmBERT

DistilBERT (Sanh et al., 2020) follows the same architecture of that of BERT (Devlin et al., 2019), while reducing the number of layers by a factor of 2. DistilBERT follows a triple loss language modeling, which combines cosine distance loss with knowledge distillation for it (student) to learn from the larger pretrained natural language model (teacher) during pretraining. In spite being a 40% smaller model than BERT in terms of the number of parameters, DistilBERT is 60% faster than the latter, and retains 97% of language understanding capabilities to that of BERT. The main reason we use a cased pretrained multilingual DistilBERT model is due to the presence of code-mixed data in our corpus (These tend to be case sensitive language in the corpus).

4.5 ALBERT

Training models with hundreds of millions, if not billions of parameters is becoming increasingly difficult, mainly owing to GPU/TPU limitations. ALBERT (Lan et al., 2020) aimed to reproduce the natural language understanding capabilities of BERT (Devlin et al., 2019) by opting several parameter reduction techniques. ALBERT (A Lite BERT) achieves State of The Art (SoTA) results on GLUE, RACE and SQUAD datasets. ALBERT uses cross-layer parameter sharing and Sentence Order Prediction objective (SoP), while disregarding Next Sentence Prediction Loss (NSP) which was previously used in BERT.

4.6 ULMFiT

ULMFiT (Howard and Ruder, 2018) effectively presented a method to fine-tune neural networks for inductive transfer learning for performing NLP tasks. Language models are trained to adapt to various features of the target task. The quality of the base model determines the final performance after fine-tuning. The language model is pre-trained on a large corpus of language to adapt and capture the important aspects and features of the language. Fine-tuning is essential for small and medium-sized datasets.

The target task LM is then fine-tuned to fit the particular task well. Discriminative fine-tuning and slanted triangular learning rates are used for this process. Different layers are found to capture different information, thus, they require different learning rates.

$$\theta_t^l = \theta_{t-1}^l - \eta^l . \nabla_{\theta^l} J(\theta) \qquad (1)$$

The weights for each layer l=1, 2, . . . , L is the layer number, η^1 is the learning rate for the lth layer, L is the number of layers, θ_t^i is the weights of the lth layer at iteration t and $\Delta(\theta^1)[J(\theta)]$ is the gradient regarding the model's objective function

5 Experiment Setup

We describe the experiment setup for our experiments performed. All of our systems were trained on Google Colab (Bisong, 2019). All of our models' parameters are as stated in Table 2. The results on the test set are tabulated in Table 3. For developing systems with pretrained transformer-based models, we use huggingface's transformer library for easier implementation (Wolf et al., 2020).

Parameter	Value
Number of LSTM units	256
Dropout	0.3
Activation Function	Softmax
Max Len	128
Batch Size	32
Optimizer	AdamW
Learning Rate	2e-5
Loss Function	cross-entropy
n(Epochs)	5

Table 2: parameters for the models

5.1 CNN-BiLSTM

We implemented a CNN (Kim, 2014) followed by a Bidirectional LSTM layer. GloVe[2] embeddings of dimensions = 100 were used. The architecture of the model has a 1D convolutional layer followed by a dropout layer and then bidirectional LSTM layer. The embedding texts are then fed into the convolution layer. The dropout layer is used for regularization. The output of the convolutional layer is then passed into the bidirectional LSTM layer. Finally, it consists of a dense layer followed by the output layer. Stochastic Gradient Descent (SGD) was used as the optimizer with a learning rate = 0.01. Kullback leibler divergence (Kullback and Leibler, 1951) was used as the loss function.

5.2 mBERT

The pretrained BERT Multilingual model ***bert-base-multilingual-uncased*** having 12 layers, 768 hidden, 12 attention heads with 110M parameters[3] was used. The model was implemented using PyTorch. During the fine-tuning of the model, bidirectional LSTM layers were integrated into the model. From the transformer encoder, the BiLSTM layer can take the embeddings as the input which leads to the increase in the information being fed which results in the improvement of the context and precision (Fang et al., 2019; Puranik et al., 2021).

5.3 XLM-R

We use **XLM-RoBERTa-base**, a pretrained multilingual language model that has been trained on over 100 languages. This model has 12 Layers, 768 Hidden, 12 attention heads and 270M parameters. We fine-tune this model for sequence classification on Malayalam and Kannada. It is trained on 3.3 GB, 7.6 GB, and 12.2 GB of monolingual Kannada, Malayalam, and Tamil corpus, respectively(Conneau et al., 2020). This model is also pretrained on 300.8 GB of English corpus. This allows the model for effective cross-lingual transfer. As we are primarily dealing with code-mixed data, it is effective as it has been pretrained on other languages before hand.

5.4 DistilBERT

The DistilBERT (Sanh et al., 2020) is a transformer model trained by distilling BERT base. A pretrained DistilBERT, ***distilbert-base-multilingual-cased*** comprised of 6-layers, 768-hidden, 12-heads, and 134M parameters, was fine-tuned by implementing in PyTorch.

[2] `http://nlp.stanford.edu/data/glove.6B.zip`

[3] `https://github.com/google-research/bert`

Model	Weighted F1-Score		
	Malayalam	Tamil	Kannada
CNN-BiLSTM	0.8367	0.6102	0.4857
mBERT-cased + BiLSTM	0.9282	0.7149	0.7029
mBERT-uncased	0.8338	0.6189	0.3936
mBERT-cased	0.8296	0.6078	0.3882
XLM-R-base	0.8645	0.6173	0.4748
DistilmBERT-cased	0.9432	0.7569	**0.7277**
albert-base-v2	0.8268	0.6112	0.3890
ULMFiT	**0.9603**	**0.7895**	0.7000

Table 3: Weighted F1-scores of offensive language detection models on the datasets

5.5 ALBERT

The architecture of ALBERT is very similar to BERT and has a much smaller parameter size compared to BERT. We fine-tuned the pretrained ALBERT model, ***albert-base-v2*** which has 12 repeating layers, 128 embedding, 768-hidden, 12-heads and 12M parameters. The training environment is same as that of BERT.

5.6 ULMFiT

After preprocessing the tweets, a pretrained language model AWD-LSTM is fed to the data. AWD-LSTM language model has an embeddings size of 400 and 3 layers which consists of 1150 hidden activations per layer. It also has a BPTT batch size of 70. Adam optimizer with default values, $\beta_1 = 0.9$ and $\beta_2 = 0.99$ is employed. The start and end learning rates are set to *1e-8 and 1e-2* respectively, and it's then fine-tuned by adhering to the slanted triangular learning rates by freezing few of the layers and dropouts with a multiplier of 0.5 were applied.

6 Results and Analysis

We have experimented with various classifiers like Multilingual BERT, XLM-RoBERTa, distilBERT, ULMFiT, CNN. The evaluation metric of this task is weighted average F1-score . This is done to account for the class imbalance in the dataset. The results of the experiments performed using different models on the test datasets of Malayalam, Tamil and Kannada are shown in Table 3.

We have trained BERT-BiLSTM, XLM-RoBERTa, CNN-BiLSTM and ULMFiT models on the training datasets of Malayalam, Tamil and Kannada. Among the mentioned models, CNN-BiLSTM gave a good F1-score of 0.8444 on Malayalam development set. For Tamil and Kannada, this model showed rather poor performance with F1-scores of 0.6128 and 0.4827, respectively. ULMFiT and XLM-RoBERTa models gave almost similar F1-scores of 0.7034 and 0.7083 respectively on Tamil. We submitted BERT-BiLSTM model as it has obtained an F1-score of 0.7285 on Tamil development set. ULMFiT gave F1-scores of 0.9048 and 0.7077 on Malayalam and Kannada development set. For Malayalam and Kannada, XLM-RoBERTa model was submitted with F1-scores of 0.9113 and 0.7156 as the model has marginally outclassed ULMFiT and BERT-BiLSTM models.

Models like multilingual BERT, ALBERT, and XLM-RoBERTa gave similar and poor results on the three test datasets. One of the reasons for the poor performance of these models is the imbalance in the distribution of the classes. In the dataset, the majority of the texts belong to not-offensive while the other classes like not-in-indented language, offensive-targeted-insult-group, offensive-targeted-insult-other, offensive-untargeted have a small classification of texts. These models performed better on the majority class and poorly on the minority classes. XLM-RoBERTa gave better results on the validation set, but due to the class imbalances and the use of code-mixed and writing in non-native languages, it could have underperformed on the test set. It is observed that the CNN-BiLSTM model also performed poorly. In the CNN-BiLSTM model, the convolution layer was not capturing the correlations and patterns within the input. Moreover, the BiLSTM layer did not apprehend the dependencies within the attributes extracted by the CNN layer, which has led to the poor performance of the model. For the word embeddings, we used GloVe embedding which did not perform well on the CNN. Multilingual BERT-BiLSTM performed well on the test set, but did not perform well on the development set. Fine-tuning the transformer model DistilBERT has resulted in a good performance. ULMFiT model attained a better performance in predicting the minority classes as well. The major reasons for the better performance of ULMFiT over other models are due to its superior fine-tuning methods and learning rate scheduler.

7 Conclusion

In this paper, we have explored various transformer models for detecting offensive language in social media posts in Malayalam, Tamil and Kannada. We observed a class imbalance problem in the provided datasets of the task, which has a consequential impact on system performance. Different network architectures can show different results. Our work manifests that fine-tuning transformer models result in better performance. The relatively high F1-scores of 0.9603, 0.7895 on Malayalam, Tamil were achieved by ULMFiT and 0.7277 on Kannada was achieved by DistilmBERT model. For future work, we intend to explore pseudo-Labelling and class weighting for better performance of our models.

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, Doha, Qatar. Association for Computational Linguistics.

Ekaba Bisong. 2019. *Google Colaboratory*, pages 59–64. Apress, Berkeley, CA.

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, New Orleans, Louisiana, USA. Association for Computational Linguistics.

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, Mihael Arcan, and John P. McCrae. 2018. Improving wordnets for under-resourced languages using machine translation. In *Proceedings of the 9th Global Wordnet Conference*, pages 77–86, Nanyang Technological University (NTU), Singapore. Global Wordnet Association.

Bharathi Raja Chakravarthi, Mihael Arcan, and John P. McCrae. 2019. WordNet gloss translation for under-resourced languages using multilingual neural machine translation. In *Proceedings of the Second Workshop on Multilingualism at the Intersection of Knowledge Bases and Machine Translation*, pages 1–7, Dublin, Ireland. European Association for Machine Translation.

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.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Shardul Suryawanshi, Navya Jose, Elizabeth Sherly, and John P. McCrae. 2020c. 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.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Shammur Absar Chowdhury, Hamdy Mubarak, Ahmed Abdelali, Soon-gyo Jung, Bernard J. Jansen, and Joni Salminen. 2020. A multi-platform Arabic news comment dataset for offensive language detection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6203–6212, Marseille, France. European Language Resources Association.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language.

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.

M. Fang, H. Zhao, X. Song, X. Wang, and S. Huang. 2019. Using bidirectional LSTM with BERT for Chinese punctuation prediction. In *2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*, pages 1–5.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Comput. Surv.*, 51(4).

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.

Sabit Hassan, Younes Samih, Hamdy Mubarak, Ahmed Abdelali, Ammar Rashed, and Shammur Absar Chowdhury. 2020. ALT submission for OSACT shared task on offensive language detection. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 61–65, Marseille, France. European Language Resource Association.

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.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification.

Omar Hussein, Hachem Sfar, Jelena Mitrović, and Michael Granitzer. 2020. NLP_Passau at SemEval-2020 task 12: Multilingual neural network for offensive language detection in English, Danish and Turkish. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2090–2097, Barcelona (online). International Committee 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.

Yoon Kim. 2014. Convolutional neural networks for sentence classification.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.

S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yash Kumar Lal, Vaibhav Kumar, Mrinal Dhar, Manish Shrivastava, and Philipp Koehn. 2019. De-mixing sentiment from code-mixed text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 371–377, Florence, Italy. Association for Computational Linguistics.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach.

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.

Puneet Mathur, Rajiv Shah, Ramit Sawhney, and Debanjan Mahata. 2018. Detecting offensive tweets in Hindi-English code-switched language. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 18–26, Melbourne, Australia. Association for Computational Linguistics.

Rana D. Parshad, V. Chand, Neha Sinha, and Nitu Kumari. 2014. What is india speaking: The ”hinglish” invasion. *ArXiv*, abs/1406.4824.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553, Melbourne, Australia. Association for Computational Linguistics.

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.

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.

Sara Renjit and Sumam Mary Idicula. 2020. CUSATNLP@HASOC-Dravidian-CodeMix-FIRE2020:Identifying Offensive Language from ManglishTweets.

Maryem Rhanoui, Mounia Mikram, Siham Yousfi, and Soukaina Barzali. 2019. A CNN-BiLSTM Model for Document-Level Sentiment Analysis. *Machine Learning and Knowledge Extraction*, 1(3):832–847.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.

Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 189–194, Melbourne, Australia. Association for Computational Linguistics.

Shardul Suryawanshi, Bharathi Raja Chakravarthi, Pranav Verma, Mihael Arcan, John Philip McCrae, and Paul Buitelaar. 2020. A dataset for troll classification of TamilMemes. In *Proceedings of the WILDRE5– 5th Workshop on Indian Language Data: Resources and Evaluation*, pages 7–13, Marseille, France. European Language Resources Association (ELRA).

Gregor Wiedemann, Eugen Ruppert, and Chris Biemann. 2019. UHH-LT at SemEval-2019 task 6: Supervised vs. unsupervised transfer learning for offensive language detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 782–787, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Gregor Wiedemann, Seid Muhie Yimam, and Chris Biemann. 2020. UHH-LT at SemEval-2020 task 12: Fine-tuning of pre-trained transformer networks for offensive language detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1638–1644, Barcelona (online). International Committee for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.