Bitions@DravidianLangTech-EACL2021: Ensemble of Multilingual Language Models with Pseudo Labeling for Offence Detection in Dravidian Languages

Debapriya Tula*, **Prathyush Potluri***, **Shreyas MS**,
Sumanth Doddapaneni, **Pranjal Sahu**, **Rohan Sukumaran**, **Parth Patwa**
IIIT Sri City, India
debapriya.t17@iiits.in

Abstract

With the advent of social media, we have seen a proliferation of data and public discourse. Unfortunately, this includes offensive content as well. The problem is exacerbated due to the sheer number of languages spoken in these platforms and the multiple other modalities used for sharing offensive content (images, gifs, videos and more). In this paper, we propose a multilingual ensemble based model that can identify offensive content targeted against an individual (or group) in low resource Dravidian language. Our model is able to handle code-mixed data as well as instances where the script used is mixed (for instance, Tamil and Latin). Our solution **ranked number one** for Malayalam dataset and ranked 4th and 5th for Tamil and Kannada, respectively. The code is available at github.com/Debapriya-Tula/EACL2021-DravidianTask-Bitions.

1 Introduction

Online communication has helped break a lot of barriers in terms of time, distance and ease of communication. The number of active Internet users has grown rapidly over the last few years. The ease of sharing content and the lack of automatic systems for monitoring them, has led to a great increase in the amount of offensive and hate speech in the open internet. Hate speech is often targeted towards a group of people or individuals hurting their identity, beliefs or sentiments. Owing to the ease of access and lack of monitoring, Individuals tend to misuse this freedom to hurl abuses and cause disharmony in the community. It is therefore important to address this issue. Social media is easily accessible by a larger domain of people and the scale of open internet restricts us from manually monitoring social media content, at scale. This calls for the need of automatic systems for identification of hate/offensive speech.

The style of data on open internet also plays a major role in the understanding of data. The language structure is often missing and people tend to make use of words from different languages, ultimately resulting in code-switched data (Barman et al., 2014; Patwa et al., 2020). The problem is exacerbated as people use words from different written scripts, mixing both latin script and native script (*Devanagri, Dravidian, Mandarin etc*) from the language. A unified model which can understand a multitude of these scripts can play a major role in understanding the discourse in open internet data and conducive to creating a safer virtual environment.

Majority of the research work in NLP has been predominantly in English (Bender, 2019; Hu et al., 2020). And the multilingual models currently available are trained on a multitude of languages making it hard to fine-tune for downstream tasks like Sentiment analysis, Text classification etc. on low-resource languages. Our work addresses this issue by employing pseudo labelling and ensemble based techniques.

The importance of the issue and the challenges posed, calls for novel ideas for offensive language detection. Owing to this many workshops (Waseem et al., 2017; Akiwowo et al., 2020) and shared tasks (Kumar et al., 2018, 2020; Chakravarthi et al., 2021), have been conducted to address the problem at hand.

In this paper, we present our system for the task of offensive language identification in Dravidian languages. We make use of multilingual BERT based models with pseudo labelling and

*Equal contribution.

Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pages 291–299

ensemble strategies to achieve 1st rank out out 30 participants on Malayalam data. Our models perform equally well on the Kannada (5th rank) and Tamil (4th rank) dataset as well.

2 Related Works

Ever since social media platforms started gaining popularity, the problem of detecting offensive language has existed. Many researchers have worked to develop different ways that automate the process to tackle the issue. Authors in (Fortuna and Nunes, 2018) have discussed the intricacy hate speech concept and its conclusive potential for societal impact, specifically in online communities and digital media platforms.

Detection of profanity and hate speech in tweets and comments has been a part of many shared tasks(Kumar et al., 2018, 2020; Chakravarthi et al., 2021; Patwa et al., 2021). The SemEval 2019 task 9 (Zampieri and Others, 2019b), aimed at identification of offensive and non-offensive comments in English tweets. It used the OLID dataset (Zampieri and Others, 2019a) which has 14000 tweets annotated using a hierarchical annotation model. OffensEval 2020 (Zampieri et al., 2020) was a profanity identification task presented in SemEval 2020. It was conducted for 5 languages (multilingual) language, namely English, Arabic, Danish, Greek, and Turkish.

Many researchers have tried to solve hate speech, offense and aggression detection using Deep Learning techniques like CNNs, LSTMs, etc. (Aroyehun and Gelbukh, 2018; Risch and Krestel, 2018; Mahata et al., 2019). Some researchers have also tried using Machine Learning algorithms for the same (Safi Samghabadi et al., 2018; Datta et al., 2020). Recently Language models like BERT (Devlin et al., 2018) have become very popular for this problem (Gupta et al., 2021; Safi Samghabadi et al., 2020; Risch and Krestel, 2020; Wiedemann et al., 2020).

There have been attempts at developing models for hate speech detection in English, Hindi-German (Mandl et al., 2019) and Italian (Corazza et al., 2020) have emerged, but not many works for Dravidian code-mix languages. Nevertheless, attempts are underway to accelerate advancements in NLP in Dravidian languages (Chakravarthi et al., 2021, 2020c), have emerged. Methods like LSTMs (Mahata et al., 2020), Transformer (Dowlagar and Mamidi, 2021) etc. have been previously tried to detect offense in dravidian languages.

3 Data

There are three datasets for the three languages that we consider: Kannada (Hande et al., 2020), Malayalam (Chakravarthi et al., 2020a), and Tamil (Chakravarthi et al., 2020b). The six class labels in the Kannada and Tamil data are:

- Not Offensive - (NO)
- Not Native - (NN)
- Offensive Individual - (OI)
- Offensive Group - (OG)
- Offensive Untargeted - (OU)
- Offensive Other - (OO)

All classes mentioned above except 'Offensive Other' are the classes in the Malayalam data set. The distribution of the data is described in Table 1 for all three classes. In total there are 5936 samples for Kannada, 11695 samples for Malayalam and 34898 samples for Tamil.

The majority class in all three languages is the 'Not Offensive' class. This accounts for 56.97% of the samples in Kannada, 88.77% of the samples in Malayalam and 72.25% of the samples in Tamil data.

Another important fact to note is the skewness in the data. The dataset is extremely skewed toward the non-offensive class and in order to overcome this challenge we make use of class weighting by penalising more for the under-represented classes. This is discussed in detail in the next section.

Table 2 shows a list of the most frequent words for each language for each class. We often see the same native word written in different ways in English. The word "your" in Kannada is written as ನಿನ್/ನಿನ್ನ/ ನಿಮ್/ ನಿಮ್ಮ/nin/ninna (singular) and nim/nimma (plural). From manual analysis it is clear that there are a lot of stop-words in the most frequent words.

Number of samples in training data

Class	Kannada	Malayalam	Tamil
NO	3382	10382	25215
NN	1407	882	1447
OI	486	171	2338
OG	327	106	2550
OU	212	154	2894
OO	122	-	454
Total	5936	11695	34898

Table 1: Data distribution of the three datasets.

4 Methods

This section describes our solution for the Offensive Language Detection Task. It is divided into 4 sub-sections viz, Models, Class-weighting, Pseudo-labelling and Ensemble. A look at the data provided for the task calls for a multi-lingual approach, as it has both Latin script and text in the native language.

4.1 Models

We leverage two transformer-based models viz, DistilmBERT (multilingual) and Indic-BERT, and a non-transformer based model, ULMFiT (Howard and Ruder, 2018) for the task.

4.1.1 DistilmBERT

DistilBERT (Sanh et al., 2019) has the same general architecture as BERT (Devlin et al., 2018), with the number of layers reduced by a factor of 2. A triple loss combines language modelling, distillation and cosine-distance losses to leverage the inductive biases of large models learnt during pre-training. With a 40% reduction in the size of the BERT, the DistilBERT retains 97% of its language understanding capabilities while being 60% faster. Inspired by the efficacy of the performance of DistilBERT[1], we use a distilled version of the BERT base multilingual model (mBERT-base) called the *DistilBERT-base-multilingual-model*. We use the cased model as the data is code-mixed with English (the only case sensitive language in the corpora). The model was pre-trained on the concatenation of Wikipedia in 104 different languages including Tamil, Malayalam and Kannada. DistilmBERT is twice as fast as mBERT-base based on the comparisons done by HuggingFace[1].

[1]https://huggingface.co/distilbert-base-multilingual-casedmodel-card

4.1.2 ULMFiT

The ULMFiT (Universal Language Model Fine-tuning for Text Classification) (Howard and Ruder, 2018) is an effective transfer learning method which achieves the state of the art results on various NLP tasks with the help of novel techniques like a) gradual unfreezing: beginning from the final most significant layer, one layer per epoch is unfrozen and is fine-tuned; b) discriminative fine-tuning: higher learning rate is used for the final layer and is lowered one by one to the first layer; c) slanted triangular learning rates: scheduler based learning rate approach which gradually propels the learning rate until it reaches it's maximum and then gradually reduces it. The ULMFiT is based on a 3-layer encoder and decoder based architecture of AWD-LSTM or the averaged stochastic gradient descent weight dropped LSTM. Training the ULMFiT can be broken down into three major tasks: Firstly, pre-training a language model on a Wikipedia-based corpus. Then, following an unsupervised approach, fine-tuning the language model to the target task and finally in a supervised approach, adding new classifier layers and fine-tuning the classifier to the actual task.

4.1.3 IndicBERT

IndicBERT (Kakwani et al., 2020) is an ALBERT based model trained exclusively on Indian languages. The model is pre-trained on 11 Indian languages and English using the standard Masked Language Modelling (MLM) objective. The model is pre-trained on news-articles, magazines and blog posts. Since the number of pre-training languages is much less compared to mBERT and includes only indic-languages, we explore IndicBERT with the intuition that it would better represent the three Indian languages at hand.

4.2 Class Weighting

Due to the class imbalance in the data, we use an inverse weighting strategy to penalize the under-represented classes more in the loss function. We also use **focal loss**(Lin et al., 2017) that can be considered as an improved version of the Cross-Entropy loss, that handles class

	Kannada	Malayalam	Tamil
Not Offensive	super, song, sir, guru, bro	oru, like, trailer, ഒരു, padam, ഈ	Like, Thala, Vera, Mass, படம்
Not Native	super, song, sir, guru, bro	like, fans, movie, fan, trailer	hai, movie, like, ki, me, ka
Offensive Individual	@nandi, parthasarathi, ನಿನ್ / nin, ನಿಮ್ಮ / nim, gowda, sule	oru, padam, ഈ, ee, ആണ്	da, ah, la, padam, trailer, like
Offensive Group	movie, fans, dislike, tiktok, like	dislike, fans, trailer, padam, പടം	da, padam, la, oru, ah, nu
Offensive untargeted	movie, ge, ಅಂತ, e, nam, song	dislike, trailer, oru, adicha, like	da, la, trailer, ah, dislike
Offensive Other	tiktok, guru, movie, e, na, madi	-	da, la, padam, trailer, like

Table 2: Most frequent words

imbalance by assigning more weights to hard examples and down-weighting easy examples.

4.3 Pseudo Labelling

Pseudo-labelling is a semi-supervised learning technique where the model is first trained over the small set of labelled examples available. This model is then used to approximate the labels on the test set and this newly labelled data is used together with the train set for further training the model. This results in a considerable increase in performance.

4.4 Ensemble

Ensembling of models have shown to have better performance in a multitude of tasks. The principle behind ensemble models is to leverage the various representations learnt by multiple (weak) learners/models to built a more robust model. Here, in this paper, we make use of a fairly simple, yet efficient form of ensembling. The output probability distributions from the models (DistilmBERT and ULMFiT) were added up, thereby making a new probability distribution. This was further converted to the required label using one-hot encoding.

5 Experiment

We describe the experiments that we perform. All the models are trained on google colab[2]. The code is available publicly[3].

5.1 DistilmBERT

We use DistilmBERT tokenizer which has a vocabulary size of 110k. A max sequence of 128 is used for truncating the input text and shorter sequences are padded with special tokens. The model uses a batch size of 8 for training. Adam optimizer with a learning rate of 1e-8 is used for optimizing the weights. The model is trained for 10 epochs.

5.2 ULMFiT

For the ULMFiT method, we follow the implementation provided by Arora (2020). The authors here pre-train ULMFiT on synthetically generated code mixed data which was created using a Markov model on preprocessed and transliterated versions of Wikipedia articles. Through transfer learning, the authors fine-tune the pre-trained ULMFiT model for the downstream task of hate speech detection[4]. We use the pretrained tokenizers (pre-trained using Google's Sentence Piece[5]) and language model provided in (Arora, 2020) for our task [6][7][8]. For each one of the 3 language sub-tasks, we split the training data into train-validation splits in the ratio of 80:20. For fine-tuning of the language model, the drop-out multiplicity is set to 0.3 with a batch size of 16. The model is trained for 1 epoch with the learning rate of 1e-2 with just the last layer unfreezed. After unfreezing all layers, The mode is fine-tuned for 5 epochs with a learning rate of 1e-3.

5.3 IndicBERT

For pre-processing we use the indicBERT tokenizer with a vocabulary size of 200k. The input sequences are truncated to a max length of 200 and padding is used for the shorter sequences. IndicBERT per-trained weights are used for fine-tuning the task. An additional fully connected layer with a dropout of 0.3 is used on top of the AlBERT model. We use a batch size of 32 for training and batch size 16 for validation. The optimization algorithm of choice was Adam with a learning rate of 1e-5.

In all the three models, to address the class

[2] https://colab.research.google.com/

[3] github.com/Debapriya-Tula/EACL2021-DravidianTask-Bitions

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

[5] https://github.com/google/sentencepiece

[6] https://github.com/goru001/nlp-for-manglish

[7] https://github.com/goru001/nlp-for-tanglish

[8] https://github.com/goru001/nlp-for-kannada

Language	Model	precision	Recall	f1-score
Tamil	ULMFiT	0.72	0.77	0.73
	ULMFiT (PL)	0.72	**0.78**	0.73
	DistilmBERT	0.75	0.77	0.76
	DistilmBERT (CW)	0.74	0.76	**0.76**
	DistilmBERT (FL)	0.75	0.77	0.75
	E [DistilmBERT (CW)+ULMFiT]	**0.75**	0.77	0.76
Kannada	ULMFiT	0.67	0.69	0.67
	ULMFiT(PL)	0.67	0.70	0.67
	DistilmBERT	0.68	0.69	0.68
	DistilmBERT (CW)	0.67	0.69	0.68
	DistilmBERT (FL)	0.68	0.69	0.69
	IndicBERT	0.59	0.59	0.59
	IndicBERT (CW)	0.65	0.66	0.65
	E [DistilmBERT (CW)+ULMFiT]	**0.692**	**0.705**	**0.697**
Malayalam	ULMFiT	0.95	0.95	0.95
	ULMFiT (PL)	0.95	0.95	0.95
	DistilmBERT	0.96	0.97	0.96
	DistilmBERT (CW)	0.96	0.96	0.96
	DistilmBERT (FL)	0.96	0.96	0.96
	IndicBERT	0.95	0.91	0.92
	IndicBERT (CW)	0.95	0.92	0.93
	E [DistilmBERT (CW)+ULMFiT]	**0.965**	**0.966**	**0.965**

Table 3: Weighted Precision, Recall, f1-scores using all methods for the 3 languages on the validation set. Abbreviations used: Pseudo-labelled (**PL**), Class-weighted (**CW**), Focal loss (**FL**), Ensemble of model X and Y(**E[X, Y]**)

Language	Model	precision	recall	f1-score
Tamil	E [DistilmBERT(CW+PL)+ULMFiT (PL)]	0.74	0.77	0.75
Kannada		0.69	0.72	0.70
Malayalam		0.97	0.97	0.97

Table 4: weighted Precision, Recall, f1-scores for the 3 languages using our best model on the test set

imbalance issue, we use class weighting where each class has been weighted inversely by the number of samples in the class. We also try to address the imbalance using focal loss (Lin et al., 2017), which however was not as effective as class-weighting w.r.t. improving performance for the under-represented classes.

For DistilmBERT and ULMFiT, we implement pseudo labelling. The trained model is used to make predictions on the unseen test set. These predictions along with the original train set was used to train a new model. Thus proving to be a good data augmentation strategy.

The outputs from DistilmBERT and ULMFiT are used for a soft voted ensemble strategy. We discard the indicBERT model from the ensemble due to poor results. The ensembling strategy of our best system is shown in figure 1.

6 Results

All the results for all experiments are reported in Table 3. Similar experiments were carried out in all the languages. The base model of ULMFiT on Malayalam gives an accuracy of 0.95 but the performance on under-represented classes was poor. DistilmBERT proved to be better on the data giving an f1-score of 0.965.

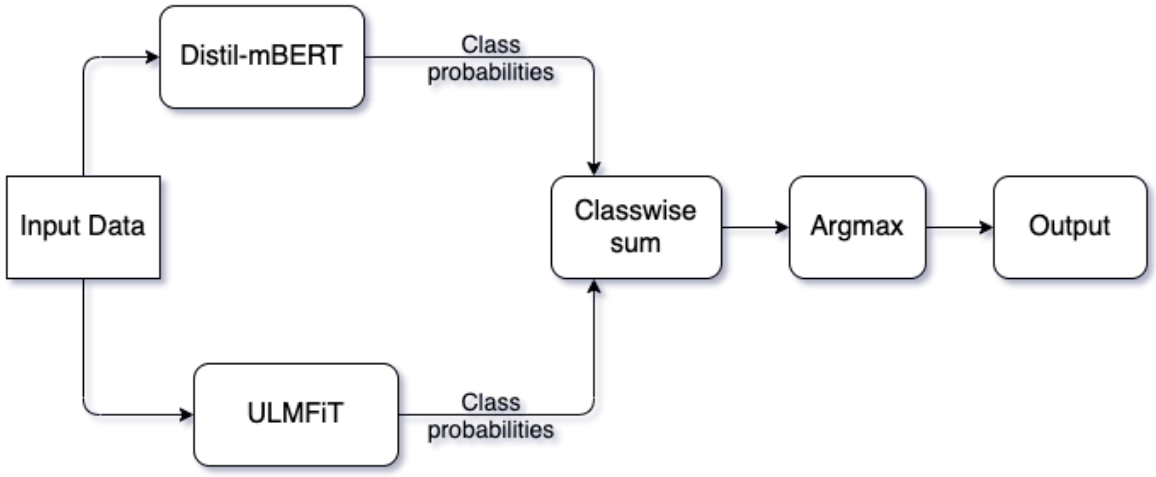

Figure 1: The input data is passed through an ensemble of DistilmBERT and ULMFiT. The aggregate of probability distributions of these models is the final prediction vector and the arg max of the probability vector is the final prediction.

Overall looking at all the results in table 3, class-weighting helps the model perform better on the under-represented classes thereby improving the overall performance of the model. Focal loss was also used to account for class imbalance the results with focal loss in DistilmBERT can be found in table 3.

Having different models which are pre-trained on different types of data, we ensembled our model predictions from ULMFit and DistilmBERT which boosted our model f1-score from 0.65 to 0.697 on Kannada data.

We observe that the DistilmBERT performs better than the ULMFiT (with and without pseudo-labelling). Its better performance can be attributed to the truly bidirectional nature of BERT (Sanh et al., 2019) based models. Secondly, BERT based models use transformers at their heart and hence do not suffer from long dependency issues. The use of the class-weighting scheme and focal loss (Lin et al., 2017) help to better represent the under-represented classes. It can be seen that the focal loss approach performs better than the naive class-weighting. But empirical results show that using focal loss led to much lower precision and recall for the minority classes than class-weighting. For example, for the Malayalam dataset, the precision and recall obtained using class-weighting were greater by 19% and 10% respectively than using focal loss. This is an interesting observation which we believe needs further experimentation for validation.

Overall our unified model which uses class-weighting, pseudo labelling and ensemble methods was the best performing model on Malayalam testset with an f1-score of 0.97. Our model was also in the top 5 best performing models for both Tamil and Kannada with f1-scores of 0.75 (4th place) and 0.70 (5th place).

7 Conclusion and Future Work

In this paper, we proposed an ensemble model utilising pseudo labelling to effectively detect offensive statements in Dravidian languages, namely Kannada, Malayalam and Tamil. We show competitive results on all three languages with first rank for Malayalam and within top-5 for Kannada and Tamil. Pre-trained multilingual model worked best for our use case as knowledge from similar language families was used across all the languages. In future research, we will consider synthetically creating new code-mixed data for each language and the usage of language specific tokenizers for the multi-lingual models.

References

Seyi Akiwowo, Bertie Vidgen, Vinodkumar Prabhakaran, and Zeerak Waseem, editors. 2020. *Proceedings of the Fourth Workshop on Online Abuse and Harms*. Association for Computational Linguistics.

Gaurav Arora. 2020. Gauravarora@ hasocdravidian-codemix-fire2020: Pre-training ulmfit on synthetically generated code-mixed data for hate speech detection. *arXiv preprint arXiv:2010.02094*.

Segun Taofeek Aroyehun and Alexander Gelbukh. 2018. Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Association for Computational Linguistics.

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code mixing: A challenge for language identification in the language of social media. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*. Association for Computational Linguistics.

Emily Bender. 2019. The benderrule: On naming the languages we study and why it matters. *The Gradient*.

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)*. 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)*. 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. Association for Computing Machinery.

Michele Corazza, Stefano Menini, Elena Cabrio, Sara Tonelli, and Serena Villata. 2020. A multilingual evaluation for online hate speech detection. *ACM Trans. Internet Technol.*, 20(2).

Anisha Datta, Shukrity Si, Urbi Chakraborty, and Sudip Kumar Naskar. 2020. Spyder: Aggression detection on multilingual tweets. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*.

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*.

Suman Dowlagar and Radhika Mamidi. 2021. Cmsaone@dravidian-codemix-fire2020: A meta embedding and transformer model for code-mixed sentiment analysis on social media text.

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

Ayush Gupta, Rohan Sukumaran, Kevin John, and Sundeep Teki. 2021. Hostility detection and covid-19 fake news detection in social media.

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*. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multitask benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics.

Ritesh Kumar, Guggilla Bhanodai, Rajendra Pamula, and Maheshwar Reddy Chennuru. 2018. TRAC-1 shared task on aggression identification: IIT(ISM)@COLING'18. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Association for Computational Linguistics.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2020. Evaluating aggression identification in social media. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*. European Language Resources Association (ELRA).

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Debanjan Mahata, Haimin Zhang, Karan Uppal, Yaman Kumar, Rajiv Ratn Shah, Simra Shahid, Laiba Mehnaz, and Sarthak Anand. 2019. MIDAS at SemEval-2019 task 6: Identifying offensive posts and targeted offense from Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Sainik Kumar Mahata, Dipankar Das, and Sivaji Bandyopadhyay. 2020. Junlp@dravidian-codemix-fire2020: Sentiment classification of code-mixed tweets using bi-directional rnn and language tags.

Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. 2019. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, FIRE '19. Association for Computing Machinery.

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. In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020)*. Association for Computational Linguistics.

Parth Patwa, Mohit Bhardwaj, Vineeth Guptha, Gitanjali Kumari, Shivam Sharma, Srinivas PYKL, Amitava Das, Asif Ekbal, Shad Akhtar, and Tanmoy Chakraborty. 2021. Overview of constraint 2021 shared tasks: Detecting english covid-19 fake news and hindi hostile posts. In *Proceedings of the First Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situation (CONSTRAINT)*. Springer.

Julian Risch and Ralf Krestel. 2018. Aggression identification using deep learning and data augmentation. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Association for Computational Linguistics.

Julian Risch and Ralf Krestel. 2020. Bagging BERT models for robust aggression identification. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*. European Language Resources Association (ELRA).

Niloofar Safi Samghabadi, Deepthi Mave, Sudipta Kar, and Thamar Solorio. 2018. RiTUAL-UH at TRAC 2018 shared task: Aggression identification. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Association for Computational Linguistics.

Niloofar Safi Samghabadi, Parth Patwa, Srinivas PYKL, Prerana Mukherjee, Amitava Das, and Thamar Solorio. 2020. Aggression and misogyny detection using BERT: A multi-task approach. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*. European Language Resources Association (ELRA).

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Zeerak Waseem, Wendy Hui Kyong Chung, Dirk Hovy, and Joel Tetreault, editors. 2017. *Proceedings of the First Workshop on Abusive Language Online*. 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*. International Committee for Computational Linguistics.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*. International Committee for Computational Linguistics.

Marcos Zampieri and Others. 2019a. Predicting the type and target of offensive posts in social media. 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)*. Association for Computational Linguistics.

Marcos Zampieri and Others. 2019b. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.