indicnlp@kgp at DravidianLangTech-EACL2021: Offensive Language Identification in Dravidian Languages

Kushal Kedia
IIT Kharagpur
kushal.k@iitkgp.ac.in

Abhilash Nandy
IIT Kharagpur
nandyabhilash@gmail.com

Abstract

The paper presents the submission of the team indicnlp@kgp to the EACL 2021 shared task "Offensive Language Identification in Dravidian Languages". The task aimed to classify different offensive content types in 3 code-mixed Dravidian language datasets. The work leverages existing state of the art approaches in text classification by incorporating additional data and transfer learning on pre-trained models. Our final submission is an ensemble of an AWD-LSTM based model along with 2 different transformer model architectures based on BERT and RoBERTa. We achieved weighted-average F1 scores of 0.97, 0.77, and 0.72 in the Malayalam-English, Tamil-English, and Kannada-English datasets ranking 1^{st}, 2^{nd}, and 3^{rd} on the respective tasks.

1 Introduction

Offensive language identification is a natural language processing (NLP) text classification task where the goal is to moderate and reduce objectionable social media content. There has been a rapid growth in offensive content on social media and users from different ethnicities and cultures worldwide. A significant portion of offensive content is specifically targeted at various individuals and minority & ethnic groups. Consequently, the identification and classification of these different kinds of foul language are receiving increased importance. Dravidian languages like Kannada, Malayalam, and Tamil (Chakravarthi, 2020) are low-resourced, making this task challenging. Training embeddings of words has previously been a common approach employed in text classification tasks. However, transfer learning approaches in deep learning (Mou et al., 2016) have been shown unsuccessful or requiring extensive collections of in-domain documents to produce strong results (Dai and Le, 2015).

Code-mixing is a prevalent practice in a multilingual culture, and code-mixed texts are often written in native scripts. Due to code-switching complexity at multiple linguistic levels, systems trained on monolingual data can fail on code-mixed data. While multilingual versions of transformer models have been shown to perform remarkably well, even in zero-shot settings (Pires et al., 2019), a zero-shot transfer may perform poorly or fail altogether (Søgaard et al., 2018). This is when the target language, here code-mixed Dravidian data, is different from the source language, mainly monolingual in English. In our work, we tackle these problems by exploiting additional datasets for fine-tuning our models and using effective transfer learning techniques. Our code and experiments are available on GitHub[1] for reproducing our models.

2 Task Description and Datasets

This task aims to classify offensive language material gathered from social media from a set of code-mixed posts in Dravidian Languages. The systems have to classify each post into one of the six labels:

- not offensive
- untargeted offense
- offense targeted at an individual
- offense targeted at a group
- offense targeted at someone else
- not in intended language

There is also a significant class imbalance in all the datasets representing a real-world situation. This shared task presents a new gold standard corpus for offensive language identification of code-mixed

[1] https://github.com/kushal2000/Dravidian-Offensive

Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pages 330–335

text in three Dravidian languages: Tamil-English (Chakravarthi et al., 2020b), Malayalam-English (Chakravarthi et al., 2020a), and Kannada-English (Hande et al., 2020). The Malayam dataset does not contain the *offense targeted at someone else* tag. The posts can contain more than one sentence, but the average number of sentences is 1. The Tamil and Malayalam datasets are considerably large, containing over $30k$ and $20k$ annotated comments, while the Kannada dataset is relatively smaller with almost $8k$ annotations. Apart from the dataset supplied by the organizers, we also use a monolingual English Offensive Language Identification Dataset (OLID) (Zampieri et al., 2019a) used in the SemEval-2019 Task 6 (OffensEval) (Zampieri et al., 2019b). The dataset contains the same labels as our task datasets except for the *not in intended language* label. The one-to-one mapping between the labels in OLID and its large size of $14k$ tweets makes it suitable for aiding the transfer learning detailed in Section 3.3.

3 Methods

A variety of methods are experimented on our datasets to provide a complete baseline. In Section 3.1, we describe our implementation of three traditional machine learning classifiers; Multinomial Naive Bayes, Linear Support Vector Machines (SVM), and Random Forests. These approaches work well on small datasets and are more computationally efficient than deep neural networks. Their performance is similar to the later sections described in the absence of pretraining and additional data. In Section 3.2, our Recurrent Neural Network (RNN) models are explained. We have compared an LSTM model, using word-level embeddings trained from scratch, to an ULMFiT model, an effective transfer learning approach for language models. Finally, in Section 3.3, we discuss transformer architectures using their cross-lingual pretrained models, which can be data-intensive during fine-tuning but provide the strongest results.

3.1 Machine Learning Classifiers

Dataset Preprocessing We preprocess the datasets by removing punctuation, removing English stop words, removing emojis, and lemmatizing the English Words. The Natural Language Toolkit library (Bird and Loper, 2004) was used for lemmatization and removing stop words. The word vocabulary is constructed, and vocabulary-length vectors containing each word's counts are used to represent each input. Based on each word's Mutual Information scores, feature selection is done to reduce the vocabulary size.

Hyperparameters For all three models, the number of words selected using the top Mutual Information scores was varied from 1000 to the length of the vocabulary. Further hyperparameters were specific to the SVM and Random Forest. The random state, the regularisation parameters, and max iterations were tuned for the SVM, and the number of decision trees used was the only hyperparameter in the case of random forests.

3.2 RNN Models

Vanilla LSTM To set a baseline for an RNN approach, we build word embeddings from scratch using just the individual datasets. For this, we selected the top $32,000$ occurring words in each dataset for one-hot encoding, which is passed through an embedding layer to form 100-dimension word vectors. A spatial dropout of 0.2 followed by a single LSTM cell and a final softmax activation forms the rest of the model. While the results for larger datasets are marginally better than the previous section, they are worse than the transfer learning approach.

ULMFiT Transfer learning has been shown to perform well in text classification tasks. Usually, language models are trained on large corpora, and their first layer, i.e., the word embeddings, are fine-tuned on specific tasks. This approach has been a very successful deep learning approach in many state of the art models. (Mikolov et al., 2013) However, Howard and Ruder, 2018 argue that we should be able to do better than randomly initializing the remaining parameters of our models and propose *ULMFiT: Universal Language Model Fine-tuning for Text Classification*. For the Dravidian languages in this task, the problem of in-domain data collection for effective transfer is also significant, especially in hate speech domains. ULMFiT provides a robust framework for building language models from moderate corpora and fine-tunes them on our specific tasks.

Language Models & Corpora We make use of language models open-sourced by the team *gauravarora* (Arora, 2020) in the shared task at HASOC-Dravidian-CodeMix FIRE-2020 (Mandl et al., 2020). They build their corpora for language modeling from large sets of Wikipedia articles. For

Tamil & Malayalam languages, they also generate code-mixed corpora by obtaining parallel sets of native, transliterated, and translated articles and sampling sentences using a Markov process, which has transition probabilities to 3 states; native, translated, and transliterated. For Kannada, only a native script corpus is available, and we had to transliterate our code-mixed dataset to Kannada to match their language model. The models are based on the Fastai (Howard and Gugger, 2020) implementation of ULMFiT. Pre-trained tokenizers and language models are available on Github. [2] [3] [4]

Preprocessing & Model Details Basic preprocessing steps included lower-casing, removing punctuations and mentions. Subword tokenization using unigram segmentation is implemented, which is reasonably resilient to variations in script and spelling. The tokenization model used is SentencePiece[5]. The language model is based on an AWD-LSTM (Merity et al., 2018), a regular LSTM cell with additional parameters related to dropout within the cell. The text classification model additionally uses two linear layers followed by a softmax on top of the language model. To tackle the difference in distributions of the target datasets and the pretraining corpora, ULMFiT proposes using 1) *discriminative fine-tuning*, i.e, layers closer to the last layer have higher learning rates, 2) *slanted triangular learning rates* which increase aggressively during the start of training and then decay gradually and 3) *gradual unfreezing*, i.e, instead of learning all layers of the model at once, they are gradually unfrozen starting from the last layer. The combination of these techniques leads to robust transfer learning on our datasets.

3.3 Transformer Models

In recent years, transformer networks like the Bidirectional Encoder Representation from Transformer (BERT) (Devlin et al., 2019) and its variant RoBERTa (Liu et al., 2019) have been used successfully in many offensive language identification tasks. For our work, we use the already pre-trained cross-lingual versions of these models available in the HuggingFace[6] library. Specifically, we use the *bert-base-multilingual-cased* model, mBERT trained on cased text in 104 languages from large

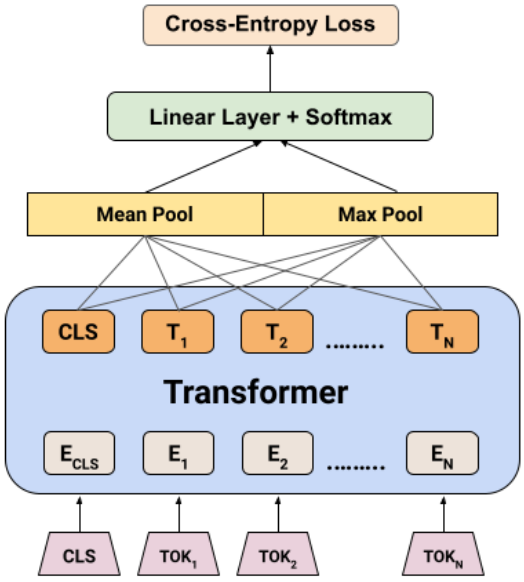

Figure 1: Transformer Model Architecture

Wikipedia articles, and the *xlm-roberta-base* model, XLM-R (Conneau et al., 2020) trained on 100 languages, using more than two terabytes of filtered CommonCrawl data. Both of these models were originally trained on a masked language modeling objective, and we fine-tune them on our specific downstream text classification tasks.

Transfer Learning The transfer learning approach's core principle is to use a pre-trained transformer model for training a classification model on a resource-rich language first, usually English, and transfer the model parameters to a less resource-rich language. For this approach, we concatenate all three code-mixed datasets as well as the OLID dataset. Our results do not change significantly on the transliteration of all datasets to Roman script. The *not in intended language* label is also removed for fine-tuning on the combined dataset since this label does not represent the same meaning across the datasets. We then use these learned model weights replacing the final linear layer to include the additional removed label *not in intended language*. This fine-tuning approach is shown to increase the performance of various scarce-resourced languages such as Hindi and Bengali.

Model Architecture We restrict the maximum length of the input sentences to be 256 by truncation and zero-padding. As shown in Fig 1, using the contextual embeddings from the last hidden states of all tokens in the sentence, we build a vector representation by concatenating the max pooling and mean pooling of these hidden states. Corre-

[2] github.com/goru001/nlp-for-tanglish
[3] github.com/goru001/nlp-for-manglish
[4] github.com/goru001/nlp-for-kannada
[5] github.com/google/sentencepiece
[6] https://huggingface.co/

spondingly, the dimension of the final sentence representation is 1536 x 1 . This is passed through a linear layer with a dropout of 0.3. The learning rate for all fine-tuning was fixed as $2e^{-5}$ and batch size was 32. The only preprocessing step before feeding the input to the transformer tokenizers was replacing emojis with their English language description. This is done because the tokenizers might not recognize the emojis, but they contain useful information about the sentence's sentiment.

4 Experiments and Results

This task's assessment metric is Weighted-F1, which is the F1 score weighted by the number of samples in all the classes. The datasets are divided into train, validation, and test sets in an approximately 8:1:1 ratio. The test labels are hidden and only available to us after the evaluation phase is over. We strictly train our models on the train set using the validation set scores for hyperparameter tuning. We have reported the results of our various models on the validation set. The average-ensemble of our top three performing models is submitted finally, and we also report its scores on the validation and test set using the scores we are ranked with on the task leader board.

Model	T	M	K
Random Forest	0.69	0.94	0.62
Naive Bayes	0.74	0.94	0.64
Linear SVM	0.74	0.95	0.65

Table 1: Weighted-F1 scores for ML models on Tamil (T), Malayalam (M) and Kannada (K) datasets.

Table 1 showcases the scores we have obtained for standard machine learning algorithms. Out of the three traditional machine learning algorithms, the Linear SVM model is best across all three datasets. The results in Table 2 summarize our RNN approaches where ULMFiT is markedly superior. The performance of our transformer models detailed in Table 3 considers two settings, one without transfer learning and one with transfer learning using the OLID and other Dravidian code-mixed datasets in conjunction.

The results on the validation set of our transfer-learned XLM-R model are the best across all 3 datasets and are followed closely by the transfer learned multilingual BERT model and the ULMFiT model. We finally submit an average ensemble of these three models, and our results on the validation set and the test set used in the final task evaluation are also enlisted in Table 4 below.

Model	T	M	K
Vanilla LSTM	0.74	0.95	0.64
ULMFiT	0.76	0.96	0.71

Table 2: Weighted-F1 scores for RNN models on Tamil (T), Malayalam (M) and Kannada (K) datasets.

Model	T	M	K
mBERT	0.74	0.95	0.66
XLM-R	0.76	0.96	0.67
mBERT (TL)	0.75	0.97	0.71
XLM-R (TL)	0.78	0.97	0.72

Table 3: Weighted-F1 scores for transformers on Tamil (T), Malayalam (M) and Kannada (K) datasets. TL indicates transfer learning using OLID and other datasets.

Model	T	M	K
avg-Ensemble (V)	0.78	0.97	0.73
avg-Ensemble (T)	0.77	0.97	0.72

Table 4: Final weighted-F1 scores using average ensembling on Tamil (T), Malayalam (M) and Kannada (K) validation (V) and test (T) datasets.

5 Conclusion

This paper describes various approaches for offensive language identification in three code-mixed English-Dravidian language datasets. We also discuss the final system submitted by the indic-nlp@kgp team, which ranks first, second, and third on the competition's three tasks. The benefit of pre-trained language models was shown by the significant improvement in results using a robust transfer learning framework (ULMFiT) compared to a vanilla LSTM model trained from scratch. Transformer networks' performance also improved when all the Dravidian language datasets were combined. This suggests that learning from one Dravidian language may help in zero-shot or few-shot transfer to other new Dravidian languages. In future works, we wish to explore these effects in more detail.

References

Gaurav Arora. 2020. Gauravarora@HASOC-Dravidian-CodeMix-FIRE2020:Pre-training ULMFiT on Synthetically Generated Code-Mixed Data for Hate Speech Detection. In *FIRE-2020 (Working Notes)*. CEUR, Hyderabad, India.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Bharathi Raja Chakravarthi. 2020. *Leveraging orthographic information to improve machine translation of under-resourced languages*. NUI Galway.

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.

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*. Association for Computational Linguistics.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, volume 28, pages 3079–3087. Curran Associates, Inc.

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.

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.

Jeremy Howard and Sylvain Gugger. 2020. Fastai: A layered api for deep learning. *Information*, 11(2):108.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

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.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. Regularizing and optimizing LSTM language models. In *International Conference on Learning Representations*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26, pages 3111–3119. Curran Associates, Inc.

Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in NLP applications? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 479–489, Austin, Texas. Association for Computational Linguistics.

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.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 778–788, Melbourne, Australia. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 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)*, pages 1415–1420, Minneapolis, Minnesota. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.