SJ_AJ@DravidianLangTech-EACL2021: Task-Adaptive Pre-Training of Multilingual BERT models for Offensive Language Identification

Sai Muralidhar Jayanthi
Language Technologies Institute
Carnegie Mellon University
sjayanth@cs.cmu.edu

Akshat Gupta
Electrical and Computer Engineering
Carnegie Mellon University
akshatgu@andrew.cmu.edu

Abstract

In this paper we present our submission for the EACL 2021-Shared Task on Offensive Language Identification in Dravidian languages. Our final system is an ensemble of mBERT and XLM-RoBERTa models which leverage task-adaptive pre-training of multilingual BERT models with a masked language modeling objective. Our system was ranked 1st for Kannada, 2nd for Malayalam and 3rd for Tamil. [1]

1 Introduction

The task of offensive language identification aims to identify offensive language used in social media texts, which could be in the form of threats, abusive language, insults (Zampieri et al., 2020). Social media platforms like Twitter, Facebook, YouTube are now mediums where a large number of people interact and express their opinions. Conflicts and target abuses are common on social media platform. Offensive language detection is a form of social media monitoring which is best done by building computational models and automatic methods due to the large stream of data involved.

Code-Switching[2] is a phenomenon common in bilingual and multilingual communities where words from two (or more) languages are used in the same sentence (Sitaram et al., 2019) and is very commonly used for social media interactions. Thus it is vital to build models that are able to process code-switched data for the relevant social media tasks. Previous shared tasks in the domain of offensive language identification have focused on identifying offensive language in English (Zampieri et al., 2019), Arabic, Greek, Danish and Turkish (Zampieri et al., 2020). The datasets used for this shared task in Dravidian languages are characterized by code-mixing.

Pretrained BERT models (Devlin et al., 2018) and their multilingual versions– mBERT and XLM-RoBERTa (Conneau et al., 2019) have produced state of the art results in multiple tasks. These models have also been used to built best performing sentiment analysis systems in various shared tasks for code-mixed datasets (Patwa et al., 2020; Chakravarthi et al., 2020c). In this paper we present our submission for the Shared Task on Offensive Language Identification in Dravidian languages. We employ task-adaptive pre-training with a Masked Language Modeling objective and ensemble predictions from mBERT and XLM-RoBERTa models for our final submission.

2 Dataset

The dataset statistics for the task of Offensive Language Identification in Malayalam (Chakravarthi et al., 2020a), Tamil (Chakravarthi et al., 2020b) and Kannada (Hande et al., 2020) are shown in Table 1. The class-wise training dataset statistics are shown in Table 2. Hande et al. (2020) outlines a brief about the data collection procedure and some baselines. We see that the datasets are highly imbalanced.[3]

Language	Split		
	Train	Dev	Test
Kannada	6217	777	778
Tamil	35139	4388	4392
Malayalam	16010	1999	2001

Table 1: Dataset statistics

[1] Code and pretrained models are available at github.com/murali1996/eacl2021-OffensEval-Dravidian.

[2] Interchangeably used with the term code-mixing

[3] More details of the task can be found at https://competitions.codalab.org/competitions/27654

Class	Language		
	Kannada	Tamil	Malayalam
Not-{language}	1522	1454	1287
Not_offensive	3544	25425	14153
Offensive_Targeted_Insult_Individual	487	2343	239
Offensive_Targeted_Insult_Group	329	2557	140
Offensive_Targeted_Insult_Other	123	454	0
Offensive_Untargetede	212	2906	191
Total Training Data	6217	35139	16010

Table 2: Class-wise training data statistics

3 Modeling

In this section, we describe the different techniques and models that were used in developing offensive language classifiers. We use two popular transformer architecture based models- Multilingual BERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020) as our backbone models. Inspired by some of the recent works on Hindi-English codemixed datasets, such as Khanuja et al. (2020) and Aguilar et al. (2020), we develop solutions based on language model pretraining and transliteration. We additionally evaluate the usefulness of representation fusion on all the three task datasets.

3.1 Architectures

Motivated by the successes of BERT model and its underlying Masked Language Modeling (MLM) pretraining strategy (Devlin et al., 2019), several derivatives evolved over the last couple of years. In this task, since the datasets contain texts in English, Tamil, Kannada and Malayalam in their native scripts as well as in Romanized form, we identify two suitable BERT based models for creating offensive identification classifiers.

Multilingual-BERT[4], aka. mBERT, is based on BERT architecture but pretrained with Wikipedias of 104 different languages including English, Tamil, Kannada and Malayalam. In this work, we specifically use the *cased* version of mBERT[5]. mBERT is pretrained on textual data from different languages with a shared architecture and a shared vocabulary, due to which there is a possible representation sharing across native and transliterated text forms. Corroborating it, Pires et al. (2019) presented its superior zero-shot abilities for an NER task in Hinglish.

XLM-RoBERTa, on the other hand, is also a transformer based model but is pretrained on Common Crawl data[6]. The common crawl data consists of textual data both in native scripts as well as in romanized script for some of the languages. Among the three languages in this task, only Tamil has Romanized data in the Common Crawl.

Given a stream of text as input, mBERT splits the text into sub-words by using WordPiece tokenization, whereas XLM-RoBERTa uses SentencePiece tokenization strategy. The sub-tokens are then appended with terminal markers such as `[CLS]` and `[SEP]`, indicating the start and end of the input stream, as shown below.

$$[CLS], w_1, w_2, ..., w_n, [SEP]$$

Popularized by BERT, the representation corresponding to the `[CLS]` token in the output layer is utilized for multi-class classification via a softmax layer. We utilize the same methodology in this work. While there are alternative choices for obtaining sentence-level representations, such as *sentence-transformers*[7], we leave their evaluation for future work. We obtain results with this setup and consider them as our baselines.

In an attempt to understand the usefulness of character embeddings when augmented with representations from BERT-based models, we develop a fusion architecture. In this architecture, we first obtain sub-word level representations from mBERT or XLM-RoBERTa models. We then convert them back to word-level representations by averaging the representations of sub-tokens of a given word. If a word is not split into sub-tokens, we use its representation as the word's representations. For every batch of training data, we keep track of words and their corresponding sub-tokens for the purpose of averaging. In a parallel, we also pass character-level embedding sequences through a Bidirectional-LSTM for every word. The hope is that the character-level BiLSTM helps to capture different variations in word patterns due to Romanization, especially useful in the context of social media text.

Once we obtain word-level representations from both character-level BiLSTM as well as BERT-based models, we concatenate the representations at word-level and pass them through a word-level Bidirectional-LSTM. The representation of the last word at the output of BiLSTM is then used for classification.

[4] https://github.com/google-research/bert/multilingual.md

[5] https://huggingface.co/transformers/multilingual.html

[6] http://data.statmt.org/cc-100/

[7] https://github.com/UKPLab/sentence-transformers

Model	Input Text	F1 / Acc (Dev Split)		
		Kannada	Tamil	Malayalam
Baselines				
mBERT	as-is	67.30 / 70.79	76.79 / 79.01	95.77 / 96.10
XLM-RoBERTa		68.70 / 71.56	76.28 / 78.19	94.12 / 94.85
Task-adaptive pretraining				
mBERT	as-is	68.40 / 72.07	76.95 / 78.87	96.58 / 96.75
XLM-RoBERTa		69.66 / 72.72	77.10 / 78.83	96.28 / 96.55
Transliteration				
mBERT	romanized	68.84 / 72.07	75.16 / 78.49	95.72 / 95.80
XLM-RoBERTa		67.73 / 71.17	75.73 / 78.08	94.18 / 95.00
Fusion Architecture (w/ char-BiLSTM)				
XLM-RoBERTa	as-is	69.79 / 72.97	75.51 / 77.87	95.78 / 96.05
Ensemble (Task-adaptive pretraining))				
mBERT	as-is	70.30 / 74.00	76.94 / 79.67	96.76 / 97.00

Table 3: Results on all three languages of the task. *as-is* implies the text is inputted without any preprocessing.

3.2 Training Methodologies

In this section, we discuss some techniques to tailor mBERT and XLM-ROBERTa models for the task of offensiveness identification in Dravidian languages.

MLM Pretraining: Both the backbone models are pretrained with their respective corpora using Masked Language Modeling (MLM) objective– given a sentence, the model randomly masks 15% of the tokens in the input stream, runs the masked stream through several transformer layers, and then has to predict the masked tokens. Doing so helps the model to learn bi-directional contextual representations, unlike LSTM (Hochreiter and Schmidhuber, 1997) based language modeling.

Recent works such as Gururangan et al. (2020) showed that a second phase of pretraining (called domain-adaptive pretraining) with in-domain data can lead to performance gains. Relevant in the realm of codemixing NLP, Khanuja et al. (2020) and Aguilar et al. (2020) have shown that such a domain-adaptive pretraining can improve performances for classification tasks in Hinglish. They curated a corpora with millions of entries consisting of codemixed Hinglish in Romanized form. Once the multilingual BERT-based models are trained with such data, they utilized those models for downstream tasks. Their downstream tasks generally contained only texts in Romanized form. Gururangan et al. (2020) have further shown that adapting the models to the task's unlabeled data (called task-adaptive pretraining) improves performance even after domain-adaptive pretraining.

However, we identify few challenges in order to directly adopt ideologies from the related works to our task at hand. Firstly, the datasets provided for each language in this task consists of texts in English, native script of that language and its transliterated form. Thus, in order to convert everything into Romanized form, we need an accurate transliterator tool. Moreover, even if such a oracle transliterator exists, there could be challenges related to word normalization due to peculiarity of a given language's usage in social media platforms. Secondly, obtaining large amounts of Malayalam-English or Kannada-English code-mixed data for the purpose of pretraining is not readily feasible due to scarcity of relevant public datasets. Thus, we resort to only task-adaptive pretraining.

Some ideas towards curating corpora for domain-adaptive pretraining could be using machine translation datasets which have fair amounts of code-mixing or by adopting semi-supervised code-mixed data creation techniques (Gupta et al., 2020). Alternate approaches could be developing techniques for efficient representation-sharing between words from native script and their Romanized forms, similar to works such as Chaudhary et al. (2020). We leave these explorations to future work.

Transliteration: In order to evaluate the performance of mBERT and XLM-RoBERTa models when inputted with only Romanized script, we first transliterate the task datasets. To this end, we identify `indic-trans`[8] as the transliterator. While task-adaptive pretraining could also be con-

[8] https://github.com/libindic/indic-trans

ducted with transliterated texts, we leave this exploration to future work.

4 Experiments

Dataset and Metrics: The number of training and validation examples for each language are presented in Table 1. Table 2 presents the class-wise distribution of the training data. As observed, the distribution is skewed with the dominant class being *Not_offensive*. In this work, we experiment with all three languages presented in the shared task. We use F1 and Accuracy as evaluation metrics[9].

Implementation details: For the purpose of training and validation, the corresponding data splits are utilized. Since gold labels for test sets were not previously available, different techniques were compared using evaluations on the validation set itself. For the purpose of task-adaptive pretraining using MLM objective, we combine the train and validation splits of the task data for training BERT models, and utilize test split of the task data for testing MLM pretraining. Due to compute resource limitations, we trim longer sentences to 300 characters (including spaces) and always use a batch size of 8. We do not perform any preprocessing on the text so as to allow for a generalized evaluation across the different datasets in the task.

We use Huggingface library (Wolf et al., 2020) for implementation of our backbone models. Specifically, we use `bert-base-multilingual-cased` and `xlm-roberta-base` models. For experiments related to representation fusion, we implement the BiLSTM models in PyTorch (Paszke et al., 2019). During task specific training for offensive classification, we optimize using the BertAdam[10] optimizer for models with a BERT component and with Adam (Kingma and Ba, 2014) optimizer for the remainder. These optimizers are used with default parameter settings. All experiments are conducted for 5 epochs and we pick the best model using the dev set. We use a hidden size of 128 for character-level BiLSTM and a size of 256 for word-level BiLSTM. We add dropouts with 0.40 rate at the outputs of both BiLSTMs.

Ensemble: For the final submission, we create an ensemble of 6 models- 3 of mBERT and 3 of XLM-RoBERTa. Each of the six models are based on task-adaptive pretraining technique. Table 4 shows the results of various runs and the ensemble scores across all three languages.

Discussion: We observe that task-adaptive pretraining improves performance (1-2% absolute) in case of both the BERT-based models, with major gains obtained for Malayalam dataset (upto 2% absolute improvement). We also observe that the baseline performances are retained when the input text is fully Romanized, thereby opening up scope for future research along the direction of domain-adaptive pretraining. We also observe that the representation fusion helps in some datasets and degrades performance in others. However, we believe that more evaluation needs to be conducted to ascertain the usefulness of such fusion.

5 Conclusions

In this paper, we present our submission for EACL-2021 Shared Task on Offensive Language Identification in Dravidian languages. Our system was an ensemble of 6 different mBERT and XLM-RoBERTa models fine-tuned for the task of offensive language identification. The BERT models were pre-trained on given datasets in Malayalam, Tamil and Kannada. Our system ranked 1st for Kannada, 2nd for Malayalam and 3rd for Kannada. We also proposed a fusion-architecture to leverage character-level, subword-level and word-level embedding to improve performance.

References

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A Centralized Benchmark for Linguistic Code-switching Eva In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France. European Language Resources Association.

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 Priyad-

[9] sklearn.metrics.classification_report.html

[10] github.com/cedrickchee/pytorch-pretrained-BERT

Model	Input Text	F1 / Acc (Dev Split)		
		Kannada	Tamil	Malayalam
mBERT (run-1)	as-is	69.88 / 73.49	76.04 / 78.30	96.48 / 96.65
XLM-RoBERTa (run-1)		69.72 / 72.59	76.80 / 79.40	95.80 / 96.25
mBERT (run-2)	as-is	68.67 / 72.72	76.58 / 79.17	96.50 / 96.70
XLM-RoBERTa (run-2)		68.39 / 71.69	76.20 / 79.03	96.30 / 96.70
mBERT (run-3)	as-is	66.28 / 70.91	76.30 / 79.54	96.04 / 96.55
XLM-RoBERTa (run-3)		69.08 / 71.94	**77.86** / 78.71	96.17 / 96.30
Ensemble (Mode)	as-is	**70.30 / 74.00**	76.94 / **79.67**	**96.76 / 97.00**

Table 4: Ensemble results on all three languages. *as-is* implies the text is inputted without any preprocessing.

harshini, 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.

BR Chakravarthi, R Priyadharshini, V Muralidaran, S Suryawanshi, N Jose, E Sherly, and JP McCrae. 2020c. Overview of the track on sentiment analysis for dravidian languages in code-mixed text. In *Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2020). CEUR Workshop Proceedings. In: CEUR-WS. org, Hyderabad, India*.

Aditi Chaudhary, Karthik Raman, Krishna Srinivasan, and Jiecao Chen. 2020. Dict-mlm: Improved multilingual pre-training using bilingual dictionaries.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

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. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

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*.

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.

Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2020. A semi-supervised approach to generate the code-mixed text using pre In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2267–2280, Online. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Adeep Hande, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. 2020. KanCMD: Kannada CodeMixed dataset for sentiment analysis and of 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.

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

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. GLUECoS: An evaluation benchmark for code-switched NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3575–3585, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas PYKL, Björn Gambäck, Tanmoy

Chakraborty, Thamar Solorio, and Amitava Das. 2020. Semeval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. *arXiv e-prints*, pages arXiv–2008.

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.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W Black. 2019. A survey of code-switched speech and language processing. *arXiv preprint arXiv:1904.00784*.

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.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

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). *arXiv preprint arXiv:2006.07235*.