CFILT IIT Bombay@HASOC-Dravidian-CodeMix FIRE 2020: Assisting ensemble of transformers with random transliteration

Pankaj Singh[a], Pushpak Bhattacharyya[a]

[a]*Indian Institute of Technololgy, Bombay, India*

Abstract

This paper describes our system submitted to HASOC-Dravidian-CodeMix task at FIRE 2020. The goal of the task was to detect hate speech and offensive content for Dravidian Languages. With the advent of various social media platforms, a large number of people from various communities engage online, on a daily basis. Hence, it is essential to develop robust mechanisms to tackle hate speech and offensive contents posted online as it can potentially impact individuals and communities. In India, most people use both English, and their mother tongue interchangeably, which results in code-mixing and script-mixing of local Indian languages with English. This shared task aims to deal with challenges arising due to code-mixing and script mixing of Tamil and Malayalam language with English, for the hate speech detection task. We employ an ensemble of multilingual BERT models for this task and devise a novel training strategy involving data augmentation using random transliteration. We achieve an F-score of 0.95 for hate speech and offensive content detection on the Malayalam code-mixed YouTube comments test data in task 1. In task 2, we achieve F-scores of 0.86 and 0.72 respectively for hate speech and offensive content detection on Tamil and Malayalam code-mixed Twitter test data. We have made our system publicly available at GitHub.

Keywords

Deep Learning, Hate Speech and Offensive Content Detection, Multilingual BERT, Code-mixing,

1. Introduction

In recent years, the growth of the Internet has been exponential, which derived the boom of social media platforms. With more than half of the population of the earth having access to the Internet, it has become practically impossible to manually track and monitor the content posted online on various social media platforms. Most of the social media platforms have a very loose identity check system which makes it easy for someone to post hate speech and offensive content online without revealing their actual identity. Therefore the need for robust and automatic hate speech and offensive content detection systems is now stronger than ever. Hence this area of research has caught the attention of both industry and academia in recent years. HASOC-Dravidian-CodeMix [1, 2, 3] provides a shared task to detect hate speech and offensive content for Dravidian languages on Youtube comments and Twitter data.

FIRE 2020: Forum for Information Retrieval Evaluation, December 16-20, 2020, Hyderabad, India

pankajsingh7@iitb.ac.in (P. Singh); pb@cse.iitb.ac.in (P. Bhattacharyya)

https://www.cse.iitb.ac.in/~pb/ (P. Bhattacharyya)

© 2020 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

In the previous version of HASOC [4], a similar shared task competition was organized in FIRE 2019 for English, Hindi and German languages. Many systems were proposed for the detection of hate speech and offensive content, for the given three languages. These systems include classical machine learning algorithms like SVM, ensemble models and deep learning models. The use of CNN, LSTM and Transformer models was common among the best performing systems. Many other shared task such as GermEval 2018 [5], HatEval [6] and OffensEval [7] has been organized in this research area along some common lines. To tackle the challenges arising due to code-mixing and script-mixing, there have been efforts to create datasets for other tasks such as sentiment analysis in code-mixed languages [8, 9].

HASOC-Dravidian-CodeMix is a subt-rack of HASOC: FIRE 2020 which consist of two shared tasks of detecting hate and offensive text in Youtube and Twitter data. The use of script and code mixed language is very common for Indian users, on social media platforms. These users tend to mix local Indian languages with English. The dataset provided in the task also depicts this real-world scenario of code and script mixing, which is an interesting NLP problem. Two tasks proposed in HASOC-Dravidian-CodeMix are the following:

- **Task 1:** Hate speech and offensive content detection in Youtube comments in Malayalam language (Code-mixed, Script-mixed)
- **Task 2:** Hate speech and offensive content detection in Youtube or Twitter dataset in Malayalam languages (Code-mixed: Manglish and Tanglish)

We proposed a system based on an ensemble of multilingual transformers [10] and achieved very good results in both tasks. A novel training strategy involving data augmentation by randomly transliterating the text during training resulted in a robust system for task 1. In section 2 we explain our system and training strategy in detail. In section 3 we report our performance in both tasks of the competition.

2. Materials and method

This section details the dataset, pre-processing steps, network architecture and training strategy for both shared tasks.

2.1. Dataset

We use the datasets provided by the organizers, for both the tasks mentioned above. Task 1 has Youtube comments in the Malayalam language as text which are categorized into hate or offensive and normal. The text in this task is both code and script mixed. It has both Dravidian and Roman script along with a mixed grammatical structure. This type of mixed structure allowed us to devise a data augmentation strategy in which we randomly transliterate words in Roman script to Malayalam with 0.5 probability during training. Task 2 has text from Youtube or Twitter in Malayalam and Tamil. The text follows the grammatical structure of both Indian languages and English but the text is majorly in Roman script. Here also, each text instance is labelled as hate or offensive and normal. Table 1 shows the dataset sizes and the distribution of the positive and negative class samples. In task 1, the dataset is imbalanced, and only 16.50% of

Table 1
Details of the datasets provided for task 1 and task 2

Task	Language	Train and set size	Test set size	hate and offensive text
Task 1	Malayalam	3600	400	16.50%
Task 2	Malayalam	4000	951	50.26%
Task 3	Tamil	4000	940	50.53%

the text is labelled as hate or offensive text which depicts real word scenarios where most of the text on social media is not be hate or offensive text. In task 2, the datasets are fairly balanced between the two classes.

2.2. System description

The core of our system is multilingual BERT [10] which is used to obtain vector representation corresponding to each sentence. We use the final hidden state vector of special [CLS] token as an aggregate representation of the entire input sentence [11]. This vector is then passed through 2 fully connected layers and a softmax layer to obtain the final prediction. Finally, we use an ensemble of multilingual BERT models trained with different setups to obtain the final prediction. We explain these setups in more detail in section 3.

We also perform some basic pre-processing steps to clean the text. The following information was removed from the text, which is not much important for hate speech and offensive content detection:

- @mention from and RT from tweets
- website URLs
- digits
- replacing multiple spaces with a single space

2.3. Training strategy

The original BERT paper [11] suggested updating the weights of the entire model for fine-tuning it on new tasks. Since our dataset was relatively small, this strategy of updating all weight of the model at once resulted in instability in the convergence of the loss function. Therefore, we employed gradual unfreezing of network layers one by one. For the first ten epochs, we trained only the last fully connected and softmax layers using cross-entropy loss and froze the parameters of all BERT layers. In the next ten epochs, we unfreeze the last (12^{th}) BERT layer and update its weights during training. We kept repeating the process of unfreezing one layer at a time after every 10 epochs. This strategy of unfreezing BERT layers one by one resulted in smooth convergence of loss function. We trained our system for 150 epochs with appropriate values of the batch size, learning rate and weight decay which were obtained after extensive hyper-parameter tuning for each task individually.

We also employ a data augmentation strategy during the training of the multilingual BERT model. Analysis of the training and test data reveals that input text from Youtube and Twitter

contains grammatical structures of both English and Indian languages. In task 1 there are data samples that contain Dravidian and Roman scripts in either transliterated or original form. We exploited this peculiarity of the dataset to devise a data augmentation strategy by randomly transliterating the text during training with a 0.5 probability. For transliteration, the Indic NLP library [12] was used which converted the input text to either Roman or English script completely. This data augmentation strategy proved to be very useful and improved the overall performance of the system.

3. Experiments and results

In this section, we shed light on various experiments conducted by us for the given tasks and selection of our final systems for the competition. We also report the official results of our system on the test set.

3.1. Experiments

We ran multiple experiments with different setups and selected the best performing setups for creating a hard vote based ensemble of transformers. The best-performing setups selected for the ensemble were the following:

- Fine-tuning multilingual BERT with binary cross-entropy loss and assisted by data augmentation using random transliteration.
- Transliterating the entire dataset in the Dravidian script and fine-tuning the multilingual BERT on this common script dataset.
- Fine-tuning multilingual BERT with weighted binary cross-entropy loss, to overcome the class imbalance problem, for task 1.

3.2. Results

We participated in all categories of shared tasks. In Table 2, we report official results of our submitted systems published by the organizers of HASOC-Dravidian-CodeMix FIRE 2020. Precision, recall and f-score metrics were used to assess the performance of the submitted system. In validations sets, the performance of the system was similar to official results. On average, performance on the validation set was 0.02 absolute value more than the official results. Due to data augmentation using the transliteration step in task 1, the f1-score of the system was improved from 0.91 to 0.95 on the validation set. Given the complexity of the task and various challenges involved, these results are pretty encouraging and prove the effectiveness of our end-to-end system.

4. Conclusion

The need to have robust hate speech and offensive content detection systems which can also deal with the problems arising due to multilinguality has become very obvious in recent years. We presented an end-to-end system that effectively handled multilinguality and gave very

Table 2
Official results of the HASOC-Dravidian-CodeMix FIRE 2020

Task	Language	Precision	Recall	F-Score	Rank
Task 1	Malayalam	0.94	0.94	0.94	2
Task 2	Malayalam	0.74	0.70	0.72	6
Task 2	Tamil	0.86	0.86	0.86	4

competitive results in hate speech and offensive content detection tasks. Our approach also establishes the efficacy of data augmentation by randomly transliterating the input sentences during training.

In the future, we aim to develop a single system for hate speech and offensive content detection system for multiple languages and multiple tasks instead of one model per task and language. The multilingual BERT model is quite powerful in learning complex mappings from input to output, and sometimes it overfits when training data size is small. Joint training of multiple tasks across multiple languages may produce a regularization effect which will help the model to generalize better.

Acknowledgments

We thank all the organizers of HASOC-Dravidian-CodeMix FIRE 2020 for presenting us with the opportunity to tackle a very interesting and relevant problem and providing their timely support during the competition.

References

[1] B. R. Chakravarthi, M. A. Kumar, J. P. McCrae, P. B, S. KP, T. Mandl, Overview of the track on "hasoc-offensive language identification- dravidiancodemix", in: Proceedings of the 12th Forum for Information Retrieval Evaluation, FIRE '20, 2020.

[2] B. R. Chakravarthi, M. A. Kumar, J. P. McCrae, P. B, S. KP, T. Mandl, Overview of the track on "hasoc-offensive language identification- dravidiancodemix", in: Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2020). CEUR Workshop Proceedings. In: CEUR-WS. org, Hyderabad, India, 2020.

[3] B. r. Chakravarthi, Leveraging orthographic information to improve machine translation of under-resourced languages, Ph.D. thesis, NUI Galway, 2020. URL: http://hdl.handle.net/10379/16100.

[4] T. Mandl, S. Modha, P. Majumder, D. Patel, M. Dave, C. Mandlia, A. Patel, 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, New York, NY, USA, 2019, p. 14–17. URL: https://doi.org/10.1145/3368567.3368584. doi:10.1145/3368567.3368584.

[5] M. Wiegand, M. Siegel, J. Ruppenhofer, Overview of the germeval 2018 shared task on the identification of offensive language, Proceedings of GermEval 2018, 14th Conference on

Natural Language Processing (KONVENS 2018), Vienna, Austria – September 21, 2018, Austrian Academy of Sciences, Vienna, Austria, 2018, pp. 1 – 10. URL: http://nbn-resolving.de/urn:nbn:de:bsz:mh39-84935.

[6] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. Rangel Pardo, P. Rosso, M. Sanguinetti, SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter, in: Proceedings of the 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019, pp. 54–63. URL: https://www.aclweb.org/anthology/S19-2007. doi:10.18653/v1/S19-2007.

[7] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, R. Kumar, 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, Minneapolis, Minnesota, USA, 2019, pp. 75–86. URL: https://www.aclweb.org/anthology/S19-2010. doi:10.18653/v1/S19-2010.

[8] B. R. Chakravarthi, V. Muralidaran, R. Priyadharshini, J. P. McCrae, 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, Marseille, France, 2020, pp. 202–210. URL: https://www.aclweb.org/anthology/2020.sltu-1.28.

[9] B. R. Chakravarthi, N. Jose, S. Suryawanshi, E. Sherly, J. P. McCrae, 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, Marseille, France, 2020, pp. 177–184. URL: https://www.aclweb.org/anthology/2020.sltu-1.25.

[10] T. Pires, E. Schlinger, D. Garrette, How multilingual is multilingual BERT?, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 4996–5001. URL: https://www.aclweb.org/anthology/P19-1493. doi:10.18653/v1/P19-1493.

[11] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://www.aclweb.org/anthology/N19-1423. doi:10.18653/v1/N19-1423.

[12] A. Kunchukuttan, The IndicNLP Library, https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf, 2020.