Offensive Language Classification of Code-Mixed Tamil with Keras

Suchismita Tripathy, Ameya Pathak and Yashvardhan Sharma

Department of Computer Science and Information Systems, Birla Institute of Technology and Science Pilani, Pilani Campus

Abstract

This paper presents the method adopted for completing Task 1 of Dravidian-CodeMix-HASOC (Hate Speech and Offensive Content Identification in English and Indo-European Languages) Shared Task proposed by the Forum of Information Retrieval Evaluation in 2021, for offensive language detection. For detecting offensive language, a custom model architecture using convolutional neural networks was created using Keras for supervised learning, and trained on a dataset of YouTube comments, written in code-mixed Tamil in both Roman and Tamil scripts. The 5 layer neural network was built only using Keras, and required simple tokenized data, padded to an appropriate length. Recurrent neural networks and transfer learning were not used, and an F-score of 0.835 was achieved with the created CNN model.

Keywords

offensive language detection, code-mixed text, Tamil, HASOC

1. Introduction

Offensive language detection is a classification task that uses supervised learning to identify offensive statements/texts in corpora [1, 2]. With the increasing usage of social media in today's hyper-connected world, being able to prevent the abuse of the freedom of speech by writers of hate comments is very important, and can pave the way to less hostile social environments that decrease the negative effects of social media usage on mental health and self-esteem [3, 4]. This offensive language detection needs to further be implemented on a variety of languages, including different scripts, while adapting to code-switching between 2 or more languages as well. Most efforts in offensive language detection have been limited to corpora in only one language and script, using usually pretrained models for classification, which do not work too well when code-switching is involved [5].

This paper focuses on the tasks under Dravidian-CodeMix-HASOC Shared Task [6], specifically Task 1, which involves classifying YouTube comments in Tamil as offensive/inoffensive. Tamil is a very widely spoken language, with its primary speakers residing in South India, Singapore, Malaysia, Canada, and Sri Lanka [7]. With the advent of social media, and the popularity of the Roman script in interfaces for the same, Tamil speakers often resort to writing social media

FIRE 2021 Forum for Information Retrieval Evaluation, December 13-17, 2021, India

✉ f20190554@pilani.bits-pilani.ac.in (S. Tripathy); f20180259@pilani.bits-pilani.ac.in (A. Pathak); yash@pilani.bits-pilani.ac.in (Y. Sharma)
🌐 https://www.bits-pilani.ac.in/pilani/yash/profile (Y. Sharma)

CEUR Workshop Proceedings (CEUR-WS.org)

Table 1
Training Data labels and distribution

Label	Category	Count
NOT	Inoffensive	4724
OFF	Offensive	1153
not-Tamil	Language other than Tamil/code-mixed Tamil and English	3

posts/comments in the Roman script itself, with the influence of the English language reaching the level of sentence structure and vocabulary as well, resulting in code-mixed sentences with both Tamil and English words, and grammar rules [5]. The task thus involved dealing with this code-mixing in the YouTube comments for classification. The approach used achieved a rank of 5 among all the submissions, with an F-score of 0.835.

2. Datasets

Labeled training dataset was provided for HASOC, along with an unlabeled test dataset. The training dataset included a total of 5880 YouTube comments, each with one of the three following labels, with the split-up as shown in Table 1.

The task involved labeling the 654 un-annotated comments of the test dataset appropriately, along with an ID for each comment. A validation dataset was not provided for this particular task, unlike for Task 2 of HASOC, and hence, the training dataset was appropriately divided as part of pre-processing, to generate a training dataset with fewer points.

3. Past Work on Offensive Language Detection

Offensive language datasets have been hard to find, given the specific nature of the problem [8]. Early approaches were based on a sentence level / user level scoring mechanism[9] where a score was assigned to each word. Basis this, the sentence/user was evaluated and maked as offensive or not based on a certain threshold. Liu et al [10] tried to use a novel augmentation scheme to improve the performance of the imbalanced and low resource data. Offensive language detection has proved to be an important field for websites to monitor content on their platform and for governments to tackle abuse. Risch et al[11] showed that more complex approaches like LSTM with an attention mechanism offer better accuracy and logical explanation while BERT[12] based approaches[13] were most favoured at the detection task in HASOC 2020. A team from Norway[14] used an ensemble of RNNs to not only detect hate speech, but also racism and sexism. Ranasinghe et al explored the effectiveness of cross lingual embeddings while a Microsoft study[15] showed that transfer learning for embeddings proves to be the most effective in offensive language detection. A study on Bahasa based offensive language[16] shows the effectiveness of the sigmoid activation function for detection while highlighting the challenges faced due to abbreviations. Recent studies like Saumya et al[17] showed that naive bayes, logistic regression, and vanilla neural network were better than transfer learning models at offensive language detection for Tamil/Malayalam based scripts.

4. Approach

We created a neural network with a custom architecture using Keras, and trained it on a section of the training dataset.

4.1. Preprocessing and Tokenisation

Since a validation dataset wasn't provided for the task, the training dataset was divided into training and validation sets. For this, the cutoff was set at 4000 i.e. 4000 data points in the training set and 1880 in the validation set. Further, the labels of the comments, were each converted to a 3 by 1 vector through one hot encoding.

The data was then passed through the keras[18] text pre-processing Tokenizer. A maximum of 5000 unique keys were saved by the tokenizer. Each comment was also post-padded with 0s, to achieve a length of 100 token indices. The tokenizer was used separately on the combination of validation and training sets, and finally the test dataset as well.

4.2. Custom Architecture

A Keras Sequential model was built from scratch for the task using a variety of layers to build a 5 layer network. An embedding layer was included first, running on the tokenized output, which turns the indices into dense vectors of a fixed size. Recurrent layers were not used, instead, a pooling and two dense layers were used. The last dense layer used the sigmoid function as its activation function (for final classification into the above-mentioned 3 classes : NOT, OFF and not-Tamil), while ReLU was used as the activation function of the former dense layer.

A variety of dropout layers (with different rates) were included in the model to correct overfitting. It was observed, however, that the model performed better with just one dropout layer before the penultimate layer of the model. This architecture gave the best performance out of all the combinations that were tried.

4.3. Training

Cross entropy loss was calculated and the Adam optimizer was used for training. The model was trained using the reduced training set, with the remaining data points used as part of the validation set. A range of batch and epoch sizes were tried to optimise performance. A final batch size of 512 was used, over 25 epochs to correct overfitting and ensure the validation loss kept decreasing with the corresponding increase of validation accuracy.

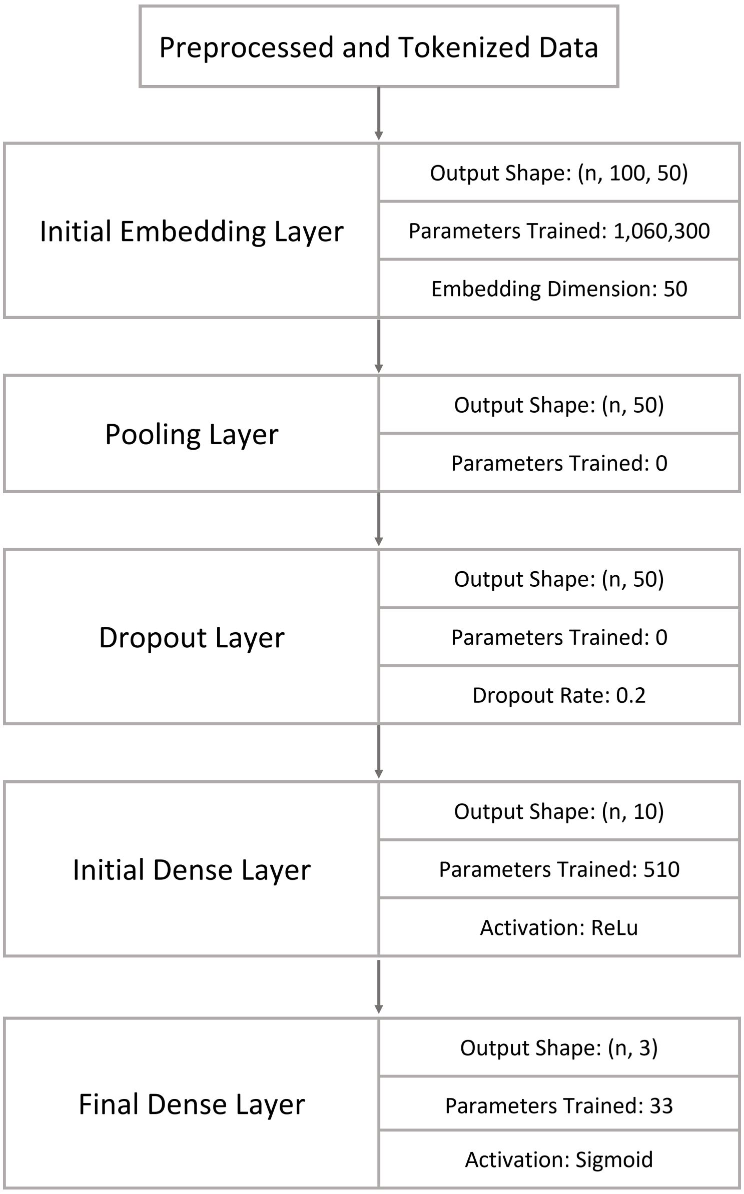

Figure 1: Custom Model Architecture

5. Evaluation

The submitted model achieved an overall rank of 5 among all the submitted runs based on its F-score. The model was evaluated on the basis of classic classification metrics - macro averaged recall, precision and F-score. The overall results were:

Table 2
Evaluation

Metric	Calculation	Value
Precision	$\frac{TruePositives}{TruePositives+FalsePositives}$	0.831
Recall	$\frac{TruePositives}{TruePositives+FalseNegatives}$	0.846
F-Score	$2\frac{Precision.Recall}{Precision+Recall}$	0.835

6. Conclusions and Future Work

We find that a deep learning based approach is highly effective at identifying offensive Tamil language texts. Further modifications can be done to fine tune the model to suit to the specificity of the Tamil language. This approach can be extended to other Indian languages which have a similar lexical pattern, thus creating a robust solution for flagging offensive content in news and social media websites.

References

[1] R. Kumar, B. Lahiri, A. K. Ojha, Aggressive and offensive language identification in hindi, bangla, and english: A comparative study, SN Computer Science 2 (2021) 1–20.

[2] A. Hande, S. U. Hegde, R. Priyadharshini, R. Ponnusamy, P. K. Kumaresan, S. Thavareesan, B. R. Chakravarthi, Benchmarking multi-task learning for sentiment analysis and offensive language identification in under-resourced Dravidian languages, arXiv preprint arXiv:2108.03867 (2021).

[3] R. Kumar, A. K. Ojha, S. Malmasi, M. Zampieri, Evaluating aggression identification in social media, in: Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying, 2020, pp. 1–5.

[4] B. R. Chakravarthi, HopeEDI: A multilingual hope speech detection dataset for equality, diversity, and inclusion, in: Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media, Association for Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 41–53. URL: https://aclanthology.org/2020.peoples-1.5.

[5] T. Mandla, S. Modha, G. K. Shahi, A. K. Jaiswal, D. Nandini, D. Patel, P. Majumder, J. Schäfer, Overview of the hasoc track at fire 2020: Hate speech and offensive content identification in indo-european languages, 2021. arXiv:2108.05927.

[6] B. R. Chakravarthi, P. K. Kumaresan, R. Sakuntharaj, A. K. Madasamy, S. Thavareesan, P. B, S. Chinnaudayar Navaneethakrishnan, J. P. McCrae, T. Mandl, Overview of the HASOC-DravidianCodeMix Shared Task on Offensive Language Detection in Tamil and Malayalam, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.

[7] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, N. Jose, S. Suryawanshi, E. Sherly, J. P. McCrae, DravidianCodeMix: Sentiment Analysis and Offensive Language Identification

Dataset for Dravidian Languages in Code-Mixed Text, arXiv preprint arXiv:2106.09460 (2021).

[8] J. J. Andrew, JudithJeyafreedaAndrew@DravidianLangTech-EACL2021:offensive language detection for Dravidian code-mixed YouTube comments, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv, 2021, pp. 169–174. URL: https://aclanthology.org/2021.dravidianlangtech-1.22.

[9] Y. Chen, Y. Zhou, S. Zhu, H. Xu, Detecting offensive language in social media to protect adolescent online safety (2012) 71–80. doi:10.1109/SocialCom-PASSAT.2012.55.

[10] R. Liu, G. Xu, S. Vosoughi, Enhanced offensive language detection through data augmentation, CoRR abs/2012.02954 (2020). URL: https://arxiv.org/abs/2012.02954. arXiv:2012.02954.

[11] J. Risch, R. Ruff, R. Krestel, Offensive language detection explained (2020) 137–143. URL: https://aclanthology.org/2020.trac-1.22.

[12] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[13] T. Mandl, S. Modha, G. K. Shahi, A. Jaiswal, D. Nandini, D. Patel, P. Majumder, J. Schäfer, Overview of the hasoc track at fire 2020: Hate speech and offensive content identification in indo-european languages (2020).

[14] G. K. Pitsilis, H. Ramampiaro, H. Langseth, Effective hate-speech detection in twitter data using recurrent neural networks, Applied Intelligence 48 (2018) 4730–4742. URL: https://doi.org/10.1007%2Fs10489-018-1242-y. doi:10.1007/s10489-018-1242-y.

[15] H. Rizwan, M. H. Shakeel, A. Karim, Hate-speech and offensive language detection in roman urdu (2020) 2512–2522.

[16] M. Susanty, Sahrul, A. F. Rahman, M. D. Normansyah, A. Irawan, Offensive language detection using artificial neural network (2019) 350–353. doi:10.1109/ICAIIT.2019.8834452.

[17] S. Saumya, A. Kumar, J. P. Singh, Offensive language identification in Dravidian code mixed social media text (2021) 36–45. URL: https://aclanthology.org/2021.dravidianlangtech-1.5.

[18] F. Chollet, et al., Keras, https://keras.io, 2015.