# YUN@HASOC-Dravidian-CodeMix-FIRE2020: A Multi-component Sentiment Analysis Model for Offensive Language Identification

Kunjie Dong[a], Yao Wang[a]

[a]*School of Information, Yunnan University, Yunnan, Kunming, 650504, P.R. China*

## Abstract

The research of discerning the offensive language formatted with code-mixed in social media has a wide range of applications in mining the available information to provide powerful assistance for sentiment analysis. This paper describes all of our work on the HASOC-Offensive Language Identification Dravidian Code-Mix FIRE 2020 tasks, which includes a message-level classification task that classifying a YouTube comment in Code-mixed Malayalam into the offensive (OFF) or Not-offensive (NOT) language, and another message-level label classification task that classifying a Tweet or YouTube comment in Tanglish and Manglish (Tamil and Malayalam using Roman Characters) into the offensive or Not-offensive language. As far as we know, this is the first shared task on offensive language in Dravidian Code-Mixed text. To achieve this goal, in this paper, we propose an ensemble model which makes full use of the information of rich sequential patterns. More precisely, the proposed model contains a self-attention based on the BiLSTM and the sub-word representation learning. Experimental results of our model on the Malayalam-English of subtask 1, Tamil-English and Malayalam-English of subtask 2 have achieved the F1 values of 0.93, 0.85 and 0.67, respectively, and ranked 3rd, 5th, 9th, respectively.

## Keywords

Sentiment analysis, Offensive language identification, Code-mixed Text, Dravidian languages

## 1. Introduction

Social networks as a popular medium or platform play an indispensable role in our life, which can provide the convenient service for the users, i.e., transmit and receive message. Language is the powerful tool in communication of information, which not only can transmit the hot news and current events, but also with a wealth of emotional information. Mining the sentiment information affiliated to the text information can be help to fully understand the intention that the sender wants to express.

With the development of the those social platform, i.e., Tweet, Weibo and YouTube, users who have different native languages can communicate freely, and even multiple types of languages were used in a conversation. The language composed of the multiple types of languages named as the code-mixed language, also known as the code-mixing, which is a kind of common language in the multilingual societies. More precisely, residents who are living in the multilingual regions

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

kunjiedong@qq.com (K. Dong); wangyao$_9$1@163.*com* (Y. Wang)
https://github.com/DaDongjie/Test (K. Dong)
0000-0001-8340-7110 (K. Dong)

tend to use English-based speech types and insert English into their primary language to communicate. With the advantages of ease-of-communication, code-mixed language have been widely used and popularized in the multilingual regions. Specifically, the Malayalam-English and Tamil-English are two kinds of code-mixed languages in social medias. Therefore, sentiment analysis of the offensive language identification of the code-mixed language has attracted lots of researchers' interests.

The Malayalam-English and Tamil-English code-mixed languages collected from the social medias are shared by the organizers in the form of two sub-tasks. To be specific, sub-task 1 provided the Malayalam language in the Latin script and sub-task 2 contains both Malayalam and Tamil language in Roman script[1] [2].

Code-mixed generally has the following forms:

- `Mixed script` - its a combination of native script and Roman script;
- `Code-Mixed script` - its a type of script in which both native and English is written in Roman script;
- `Native script` - regional language is written in native script;

Acronyms, non-standard spellings, and non-grammatical structures are all challenges of this research task, and the scarcity of annotated data available for sentiment analysis also limit the development of discerning the offensive language from the code-mixed text. Because of the complexity of code-switching at different language levels in text, the monolingual data training system cannot process the complicated code-mixed data. More information about the code mixed can be found in [3] [4].

To address the aforementioned problems, we integrate the convolution neural network (CNN) and self-attention based LSTM into a unified framework, which have the powerful ability to classify any tweet into the offensive or Not-offensive language under the help of the self-attention mechanism. In the data processing stage, motivated by [5], we introduce the sub-word scheme and attention mechanism to learn the inherent features better and further to improve the classification accuracy. As a result, our model obtain the good performance of the F1 value on the test text. Specifically, based on the two sub-tasks released by the officials, our model achieved the F1 value of 0.93 (ranked 3rd) in sub-task 1 in Malayalam-English, and the F1 values of 0.85 (ranked 5th) and 0.67 (ranked 9th) in Tamil-English and Malayalam-English, respectively.

The rest of this article is organized as follows. Section 2 introduces the related work. Section 3 and Section 4 describe the data and architecture of out model, respectively. Section 5 presents the experimental results. Finally, conclusions and future work are shown in Section 6.

## 2. Related Works

In this section, we briefly summarize the development of sentiment analysis methods in items of the automated detection of the offensive, hateful, abusive, aggressive, and profane texts.

Utsab Barman et al. [6] proposed the sub-word level representations method with the LSTM (Sub-word LSTM) architecture, which works well in highly noisy text containing misspellings and obtains the better performance under the metric of F1 score on the manually annotated dataset. Sarkar Kamal [7] implemented a machine learning algorithm, called the Multinomial

**Table 1**
Table Data description

| Task | Language | Train Data |  | Validation Data |  | Test Data |
|---|---|---|---|---|---|---|
|  |  | not-offensive | offensive | not-offensive | offensive |  |
| Subtask1 | Malayalam | 567 | 2633 | 328 | 72 | 400 |
| Subtask2 | Malayalam | 2047 | 1953 | - | - | 1000 |
| Subtask2 | Tamil | 2020 | 1980 | - | - | 940 |

Naive Bayes, which first to use n-gram tokenizer, i.e., unigram and bigram, and words features to train the classifier, and then predicts the sentiment classification. Madan et al. [8] supposed a self-attention based on the BiLSTM model to achieve the sentiment analysis of code-mixed tweets. Batuhan Guler et al. [9] achieved a combination model consist of a bi-directional RNN using LSTM cells, a CNN, and a Feed Forward Neural Network (FFNN), in which adopts the Bayesian Optimization (BO) to search the hyper-parameter.

## 3. Data

With the rapid development of the social medias, code-mixed languages have been widely used in various social activities. There be a lot of code-mixed data between Malayalam and English among the YouTube comments [2]. Recently, the shared task was conducted by the HASOC-Offensive Language Identification Dravidian Code-Mixed FIRE 2020 for to detect the offensive words from the Dravidian languages formatted with the code-mixed text.

In this paper, we employ the datasets in two languages Malayalam-English and Tamil-English provided by the organizer, which are mainly come from YouTube video comments. The datasets contain all three types of code-mixed sentences: Inter-Sentential switch, Intra-Sentential switch and Tag switching. Specifically, Table 1 shows the distribution of the training sets, validation sets, and test sets for the two languages.

Malayalam-English provided in sub-task 1 be separated the training set and the validation set, in which Malayalam use the non-Roman script. However, Malayalam and Tamil provided in sub-task 2 only provide the training set denoted in Roman script, without the validation set. To make a persuasive assessment, we randomly extract 30% nodes from training set as the validation set for verifying the classifier and the remained nodes as the training set for training the classifier, respectively.

## 4. Architecture

Malayalam text written in latin script provided by the official organizer, for sub-task 1, so we uniformly translated it into the Roman script. Then, we remove all useless characters (Emotional symbols, @username and so on) from the text and convert to lowercase, and input them to the sub-word level representation model. Given a sentence, the 1-D convolution operation on input character shown as the figure 1.

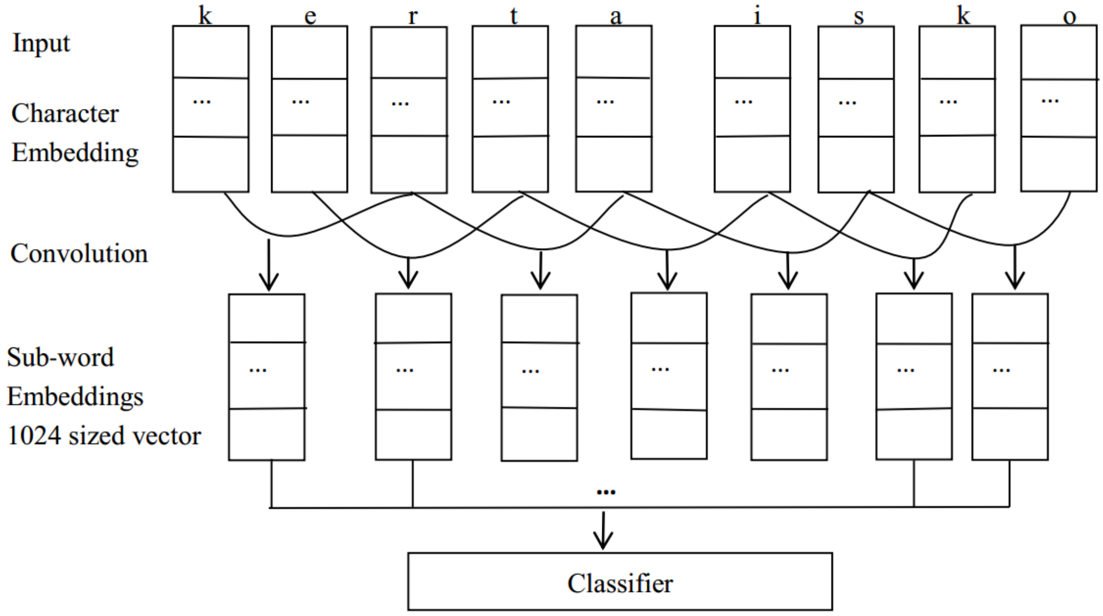

**Figure 1:** The sub-word level representations through 1-D convolutions on the character

Let be the matrix $Q \in R^{d \times l}$ represented the input sentences, we perform 1-$D$ convolution operation on the $Q$, and then apply a non-linear function to indicate the feature map $f \in R^{l-m+1}$ at the sub-word level. Specifically, the $i$-th element of $f$ annotated as:

$$f[i] = g((Q[:, i : i + m - 1] * H + b) \tag{1}$$

where the notations $l, d, m$ are the length of the input, the dimension of character embedding and the length of the filter, respectively. Then, we obtain the sub-word representations by pooling the maximal responses from $p$ feature representations, denoted as the following:

$$y_i = \max(f[p * (i : i + p - 1)]) \tag{2}$$

Based on the learned sub-word embeddings, we feed them into a classifier to conduct the classification task. In the classifier, more accurately, we feed the sub-word embeddings into the BiLSTM and add a weight on hidden states obtained from the BiLSTM.

$$a_i = \frac{\exp(k_i^T k_n)}{\sum_{j=1}^{n} \exp(k_i^T k_n)} \tag{3}$$

In brief, we first to obtain the latent representations and feed them into the softmax function to get the prediction through a full conneted layer (FC). The architecture of our model is displayed in figure 2.

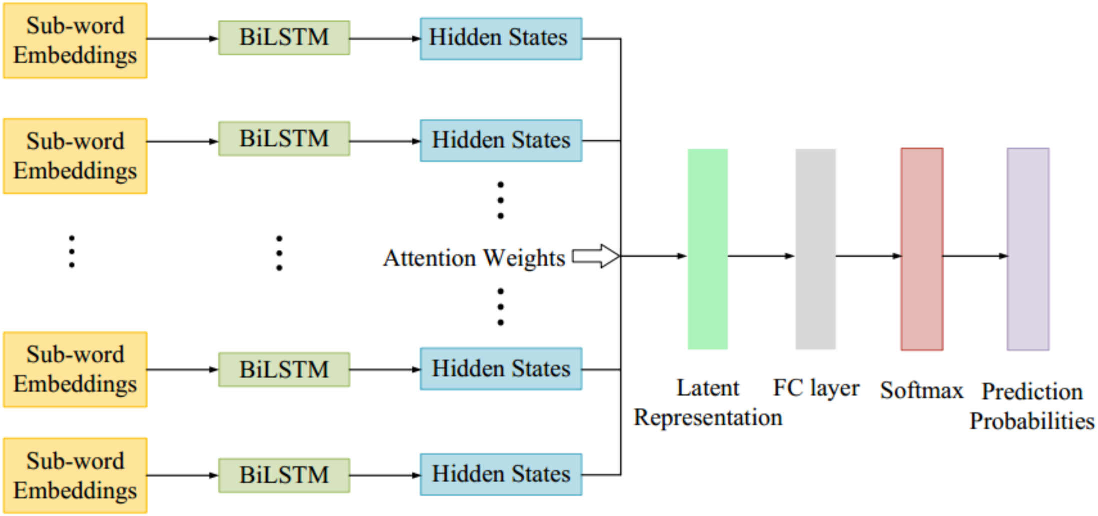

**Figure 2:** The architecture of our model

**Table 2**
The experimental results

| Task | Language | Precision | Recall | F-score |
|---|---|---|---|---|
| Task1 | Malayalam-English | 0.93 | 0.93 | 0.93 |
| Task2 | Tamil-English | 0.85 | 0.85 | 0.85 |
| Task3 | Malayalam-English | 0.67 | 0.67 | 0.67 |

## 5. The Experimental Results

In this section, we summarize the experimental results on classification test on two sub-tasks provided by the officials. More precisely, the F1 value of 0.93 (ranked 3rd) on the Malayalam-English was achieved in sub-task 1, and the F1 values of 0.85 (ranked 5th) and 0.67 (ranked 9th) on the Tamil-English and Malayalam-English were achieved in sub-task 2, respectively.

## 6. Conclusions and Future work

Recognizing offensive verbal comments plays an important role in social activities. In this study, we proposed a self-attention based on the BiLSTM model to identify the offensive language, which is the first shared task on Offensive language in Dravidian code-Mixed text, i.e., Malayalam-English and Tamil-English.

Besides word-level attention, semantic-level attention can provide a powerful assistance for identifying the offensive comments. Therefore, we plan to consider the hierarchical attention model to further improve the classification performance in our future work.

## Acknowledgments

Thanks to the developers of ACM consolidated LaTeX styles https://github.com/borisveytsman/acmart and to the developers of Elsevier updated LaTeX templates https://www.ctan.org/tex-archive/macros/latex/contrib/els-cas-templates.

## References

[1] 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.

[2] 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.

[3] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, S. Suryawanshi, N. Jose, J. P. Sherly, Elizabeth McCrae, Overview of the track on Sentiment Analysis for Davidian Languages in Code-Mixed Text, in: Proceedings of the 12th Forum for Information Retrieval Evaluation, FIRE ’20, 2020.

[4] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, S. Suryawanshi, N. Jose, J. P. Sherly, Elizabeth McCrae, Overview of the track on Sentiment Analysis for Davidian 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, 2020.

[5] D. S. Nair, J. P. Jayan, R. R. R, E. Sherly, Sentima - sentiment extraction for malayalam, in: International Conference on Advances in Computing, 2014, pp. 1719–1723.

[6] U. Barman, A. Das, J. Wagner, J. Foster, 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, 2014.

[7] K. Sarkar, Ju_ks@sail_codemixed-2017: Sentiment analysis for indian code mixed social media texts, CoRR abs/1802.05737 (2018). URL: http://arxiv.org/abs/1802.05737. arXiv:1802.05737.

[8] M. G. Jhanwar, A. Das, An ensemble model for sentiment analysis of hindi-english code-mixed data, CoRR abs/1806.04450 (2018). URL: http://arxiv.org/abs/1806.04450. arXiv:1806.04450.

[9] N. Frisiani, A. Laignelet, B. Güler, Combination of multiple deep learning architectures for offensive language detection in tweets, CoRR abs/1903.08734 (2019). URL: http://arxiv.org/abs/1903.08734. arXiv:1903.08734.