Zyy1510@HASOC-Dravidian-CodeMix-FIRE2020: An Ensemble Model for Offensive Language Identification

Yueying Zhu, Xiaobing Zhou*

School of Information Science and Engineering, Yunnan University, Yunnan, P.R. China

Abstract

This paper reports the zyy1510 team's work in the HASOC-Offensive Language Identification-Dravidian Code-Mixed FIRE 2020 shared task, whose goal is to identify the offensive language of the code-mixed text of comments/posts in Dravidian Languages (Malayalam-English and Tamil-English) collected from social media. This task is a message-level label classification task. Given a tweet or YouTube comments code-mixed text, and systems accurately classify it into offensive or not-offensive. We propose an ensemble model combines with different models to improve the F-1 value of the framework. The ensemble model is a combination of a BiLSTM (Bidirectional LSTM), an LSTM+Convolution, and a CNN (Convolution Neural Network) model. The proposed model have achieved an F-1 of 0.93 (ranked 3^{rd}) in Malayalam-English of task1, and F-1 of 0.87 (ranked 3^{rd}) and 0.67 (ranked 9^{th}) in Tamil-English and Malayalam-English of task2, respectively.

1. Introduction

With the advent of social media, the Internet provides a platform for users who can comment on any topic in the code-mixed format. These comments carry a rich of sentiment information, it can provide a better service for users by mining and making full use of the available sentiment information. This type of language occurs mainly in multilingual societies, such as Europe and India, usually with informal or casual conversation, such as social media, chat or face-to-face conversation. Nowadays millions of Internet users, especially in India, are communicating with code-mixed that embed their regional languages into English, which has provided resources for the code-mixed study. So there is an increasing demand for offensive language detection on social media texts [1]. Code-mixing is the act of interchanging between two or more types of languages in a conversation. The most common language is Hindi-English but this task provided the Malayalam-English and Tamil-English code-mixed text. Malayalam belongs to the Dravidian family, a large family of languages of South and Central India, and Sri Lanka [2]. And Tamil is up to the Dravidian southern language and is the most important of the Dravidian language. Code-mixed can be mixed in many ways, such as word aspect, sentence aspect, etc. For example, Malayalam-English: *Innaleyaaane kandath super Padam.....ellarum familyaaayi*

https://competitions.codalab.org/competitions/25295

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

EMAIL: zhouxb@ynu.edu.cn (X. Zhou*)

© 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)

poyi kananam super abinayam. The English words 'super' and 'family' intra-sententially code-mixed and the word 'familyaayi' is a neologism that combines English and Malayalam and is another encoding mixed, called Intra-word conversion, that occurs at the word level [3]. And Malayalam-English: *Enthu oola trailer aanu ithu. poor dialogue delivery.* This is an example of inter-sentential code-mixing.

This task consists of two subtasks, which is a message-level label classification task. Given a tweet or Youtube comments in Manglish (Malayalam not written using Roman Characters in task1), or Tanglish and Manglish (Tamil and Malayalam written using Roman Characters in task2), systems have to classify it into offensive or not-offensive [4]. As we all known systems that train on monolingual data, like English, fail on code-mixed data because of the complexity of switching code between different language levels in text.

We propose an ensemble model that combined with different models by a BiLSTM (Bidirectional LSTM), an LSTM+Convolution, and a CNN (Convolution Neural Network) model, which can improve the F-1 values from different aspects. We'll discuss this model more detail in the system description section. We have tested our system on the test data in Dravidian languages released for the task. The model have achieved an F-1 of 0.93 (ranked 3^{rd}) in Malayalam-English of task1 and F-1 of 0.87 (ranked 3^{rd}) and 0.67 (ranked 9^{th}) in Tamil-English and Malayalam-English of task2, respectively. Our code is available on GitHub[1]

2. Related Work

As far as we know, this is the first shared task on offensive language in Dravidian code-mixed text. The goal of this task is to identify offenslve language of the code-mixed dataset of comments/posts in Dravidian Languages (Malayalam-English and Tamil-English) collected from social media[2]. The corpus available for code-mixed is small in itself, Tamil and Malayalam languages are even less common. There are some work of other languages of the code-mixed as reference.

Gupta et al.[5] developed a supervised system based on conditional random field classifier which assigned coarse-grained and fine-grained PoS tags for the English-Hindi. Zhang et al. [6] demonstrated that a feed-forward network with a simple globally constrained decoder can accurately and quickly annotate 100 languages and 100 pairs of code-mixed and single-language texts on the English-Bengali and English-Telugu. Dahiya et al. [7] introduced curricu-lum learning strategies for semantic tasks in code-mixed Hindi-English texts. Vyas et al. [8] described their initial efforts to create a multi-level annotated corpus of Hindi-English code-mixed text and explored language identification, back-transliteration, normalization and POS tagging of this data. Thamar et al. [9] described Language identification in the first shared task of the code-switched data held at EMNLP 2014. Prabhu et al. [10] introduced learning sub-word level representations and they also provided a usable data set of Hindi-English code-mixed text. Choudhary et al. [11] proposed a new approach, called mixed discourse emotion analysis (SACMT), which uses comparative learning to categorize sentences into corresponding emotions – positive, negative, or neutral.

[1]https://github.com/TroubleGilr/HASOC-Dravidian-CodeMix—FIRE-2020
[2]https://sites.google.com/view/dravidian-codemix-fire2020/overview

Table 1
Description of the dataset on the test set provided by organizers

Task	Language	Train	Dev	Test
task1	Malayalam-English	3200	400	400
task2	Malayalam-English	4000	–	1000
task2	Tamil-English	4000	–	940

3. Dataset

The organizer provide YouTube comments in code-mixed Malayalam-English where Malayalam is the non-Roman script of task1, and task2 contains Tamil-English and Malayalam-English (Tamil and Malayalam written using Roman Characters) which are two kinds of labels of offensive or not-offensive. No labels are provided for all test text and no external data is used. We can get detailed data from Table 1.

The organizer provide two subtasks, in which task1 only contains Malayalam-English code-mixed text, but task2 includes Tamil and Malayalam code-mixed text. The NOT/OFF of training set and verification set in task1 are 2633/567 and 328/72, respectively. And task2 doesn't distinguish between the training set and validation set. The NOT/OFF of Tamil and Malayalam languages training set are 2020/1980 and 2047/1953, respectively, we automatically separate the 0.2 training set as the verification set. More data details can be seen in this paper [3] [12] and some of the processing of code mixed text can be seen in [13].

4. System Description

4.1. Pre-processing

The tweet or YouTube comments have been originally Malayalam using not-Roman script in task1 and Malayalam written using Roman Characters in task2. The tweets or comments are preprocessed using the following ways before feeding it to the training stage:

1. **Transliteration**: Non-English words in task1 are converted into Roman script by phonetic transliteration. The transliteration API[3] for Google is used for this. While English words are not changed, and all the words in task2 remain the same.

2. **Out of order**: We randomly scramble the order of all the datasets to improve the accuracy of the prediction.

3. **Noise removal**: Usernames (annotated as @username), and emoticons present in the tweets are removed altogether, while hashtags are left as it is and then fed the model.

4. **Label Encoding**: Categorical sentiment values were label encoded as 0,1 to offensive or not-offensive, respectively. This was done to give a numeric representation to the categorical data.

[3]http://google.ifanyi.com.cn/

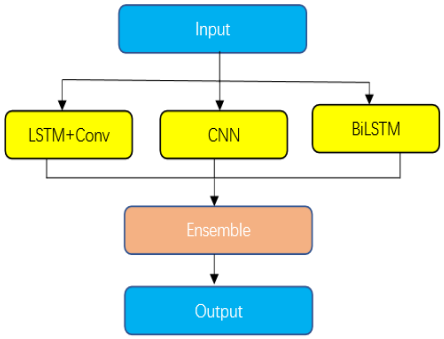

Figure 1: Ensemble architecture for HASOC-Offensive language

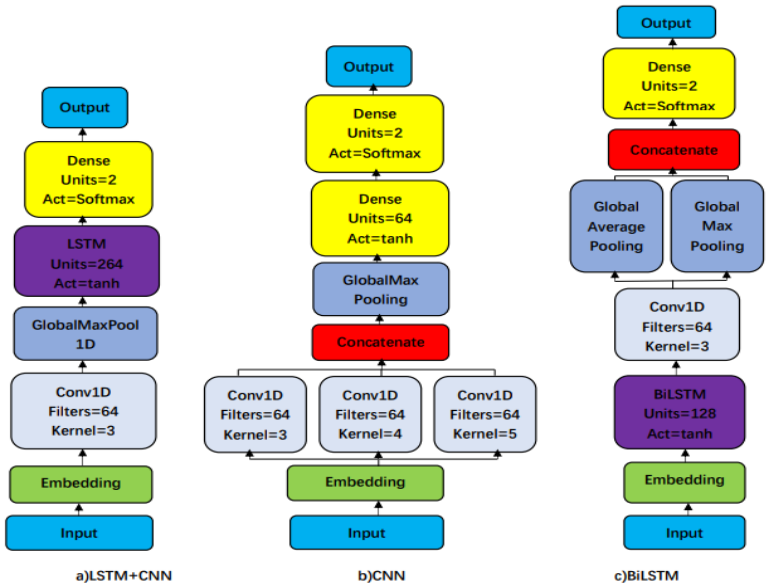

Figure 2: Individual model of CNN, LSTM+Conv and BiLSTMs

4.2. Model Architecture

The model consists of three parts, a basic CNN (Convolution Neural Network), an LSTM + Convolution, and a BiLSTM (Bidirectional LSTM). These three modules are ensemble as our classifier, as shown in Figure 1.

1. **LSTM+Conv**: The module consists of a convolutional layer with a kernel size of 3, followed by a global maximum pool layer, an LSTM layer and a dense layer[14], the details of which are shown in Figure 2(a). CNN, to some extent, takes into account the ordering of the words and the context in which each word appears.
2. **CNN**: This particular module uses 3 different convolutional layers, with the kernel of 3,4,5,

connected to the embedding layer. The output of each layer is connected and then passed to a global maximum pool layer, followed by two dense layers, as shown in Figure 2(b). The idea behind using several filter sizes is to capture contexts of varying lengths. The convolution layer is used to extract local features around each word window, while the global maximum pool layer is used to extract the essential features in the feature map.

3. **BiLSTMs**: In this module, a BiLSTM [15] layer is used, followed by a convolutional layer with a kernel size of 3. The output of this layer goes through two different layers, the global average pool and the global maximum pool. The output is connected and then passed to dense layer 2. Figure 2(c) shows the details of the model.

To achieve better F-1 accuracy, we build an ensemble model that utilizes the advantages of these individual model. Inputting the text processed in the pre-processing stage to all models, and the output after training is denoted as:

$$O_n = \sum_{i=1} x_n^i \tag{1}$$

Where i=number of sentences.

The final output matrix was calculated using the following formula:

$$O_{final} = max(O_{10}, O_{20}, O_{30}), max(11, O_{21}, O_{31}) \tag{2}$$

O_{nj} represents the probability of the class j for the n^{th} model (here n was the no of the model stated above). Where n=1, 2, 3 denotes model and j=0, 1 denotes thecategory (0-offensive, 1-not-offensive) in O_{nj}. After the calculation, the maximum probability of each sentence was assigned.

5. Experiments Detail

The officially provided dataset in task1 is divided into three parts - training, validation, and testing set but task2 has no validation set. We randomly divide the training data into 80-20 split to get the final training and validation data in task2. In this paper, we propose an ensemble model and train it on the training set. Then we have tested our system on the test data. Our model achieve an F-1 of 0.93 (ranked 3^{rd}) in Malayalam-English of task1 and F-1 of 0.87 (ranked 3^{rd}) and 0.67 (ranked 9^{th}) in Tamil-English and Malayalam-English of task2, respectively. Details are shown in table 2.

Through experimental comparison, we find that the epochs are 7,5,4 in the BiLSTM, the LSTM+Convolution and the CNN model, respectively, which have better accuracy with a batch size of 128, vocabulary size of 20000, the text sequence length of 50 with sparse categorical loss and learning rate of 0.01.

6. Conclusion and Future Work

In this paper, the detailed approach of us for the offensive language detection in Dravidian languages is described. We propose an ensemble model over three distinct modules that on their

Table 2
Description of the results

Task	Language	Precision	Recall	F-Score
task1	Malayalam-English	0.93	0.93	0.93
task2	Tamil-English	0.88	0.87	0.87
task2	Malayalam-English	0.68	0.67	0.67

own do perform well with the task. However, the ensemble model is able to catch a particular sentiment exceptionally well. We achieve a score of 0.93, just 0.02 below the first rank. In the future, we're going to put emotional information into the system and a voted ensemble may be attempted to improve the score. Bert is also one of the ways we think about.

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] D. S Nair, R. R R, J. Jayan, S. Elizabeth, Sentima- sentiment extraction for malayalam, 2014. doi:10.1109/ICACCI.2014.6968548.

[3] 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.

[4] 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.

[5] D. Gupta, S. Tripathi, A. Ekbal, P. Bhattacharyya, Smpost: Parts of speech tagger for code-mixed indic social media text (2017).

[6] Y. Zhang, J. Riesa, D. Gillick, A. Bakalov, J. Baldridge, D. Weiss, A fast, compact, accurate model for language identification of codemixed text, 2018, pp. 328–337. doi:10.18653/v1/D18-1030.

[7] A. Dahiya, N. Battan, M. Shrivastava, D. Sharma, Curriculum learning strategies for hindi-english codemixed sentiment analysis, 2019.

[8] Y. Vyas, S. Gella, J. Sharma, K. Bali, M. Choudhury, Pos tagging of english-hindi code-mixed social media content, 2014, pp. 974–979. doi:10.3115/v1/D14-1105.

[9] T. Solorio, E. Blair, S. Maharjan, S. Bethard, M. Diab, M. Ghoneim, A. Hawwari, F. Alghamdi, J. Hirschberg, A. Chang, P. Fung, Overview for the first shared task on language identification in code-switched data, 2014. doi:10.3115/v1/W14-3907.

[10] A. Prabhu, A. Joshi, M. Shrivastava, V. Varma, Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text (2016).

[11] N. Choudhary, R. Singh, I. Bindlish, M. Shrivastava, Sentiment analysis of code-mixed languages leveraging resource rich languages (2018).

[12] 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.

[13] 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.

[14] R. Sawhney, M. Ayyar, R. R. Shah, Did you offend me? classification of offensive tweets in hinglish language, 2018, pp. 138–148. doi:10.18653/v1/W18-5118.

[15] G. Xu, Y. Meng, X. Qiu, Z. Yu, X. Wu, Sentiment analysis of comment texts based on bilstm, IEEE Access 7 (2019) 51522–51532. doi:10.1109/ACCESS.2019.2909919.