LucasHub@Dravidian-CodeMix-FIRE2020: Sentiment Analysis on Multilingual Code Mixing Text with M-BERT and XLM-RoBERTa

Bo Huang[a], Yang Bai[b]

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

Abstract

This paper presents LucasHub's system description which was submitted to the Dravidian-CodeMix-FIRE 2020 on Sentiment Analysis on Multilingual data. The goal of this shared task is to perform sentiment analysis on code-mixed text. The code-mixed text comes from a new gold standard corpus composed of Dravidian (Malayalam-English and Tamil-English). The tasks for the two languages mentioned above can be seen as two quinary classification tasks. Through our analysis of the data set, we provide a deep learning model that combines the fine-tuned Multilingual BERT (M-BERT) and the fine-tuned XLM-RoBERTa multi-step integration. Our weighted average F1-Scores for Malayalam-English and Tamil-English are 0.73 and 0.63, which rank 2nd and 3rd in the official rankings, respectively. We provide the codes of the two models described in the paper for the convenience of understanding the details of the models (https://github.com/Hub-Lucas/hasoc_codemix).

Keywords

Sentiment Analysis on Multilingual, Dravidian languages, deep learning, Multilingual BERT, XLM-RoBERTa

1. Introduction

Today, with the popularization of mobile Internet, social media has become one of the world's major industries, and nearly 75% of the world's population uses social media. In the past 20 years, sentiment analysis on social media data is a very valuable research task, which has always been highly valued by academia and industry[1]. The Dravidian-CodeMix-FIRE 2020 task organization team gives some comment data from two code-mixed texts in Dravidian languages (Malayalam-English and Tamil-English) from YouTube, and uses this data to carry out the message-level polarity classification task. The levels are Positive, Negative, Neutral (or Unknown state), Mixed motions (or Mixed-feeling), or not in the intended languages (not-Tamil or not-Malayalam)[2].

Many methods had been applied to similar data sets provided by task organizers. These methods tried to use a variety of traditional machine learning algorithms, such as Logistic regression (LR), Support vector machine (SVM), K-nearest neighbors (KNN), etc. as well as representative deep learning (DL) models such as Bidirectional Encoder Representations for

FIRE 2020: Forum for Information Retrieval Evaluation, December 16-20, 2020, Hyderabad, India
hublucashb@gmail.com (B. Huang); baiyang.top@gmail.com (Y. Bai)
0000-0002-4203-1935 (B. Huang); 0000-0002-7175-2387 (Y. Bai)

Transformers (BERT) and 1DConv-LSTM. The use of machine learning methods and deep learning models had provided us the great reference value in completing this task. In particular, what caught our attention is the performance of BERT in the two languages of Malayalam-English and Tamil-English. Compared with other models and methods, BERT has a better score on the Non-Malayalam label[3] and Non-Tamil label (Other languages)[4], but BERT's performance on the Mixed (or Mixed-feeling) label is too bad. The Precision-score, Recall-score, and F-score showed by the Bert model in both languages are 0 points. Of course, the other classification algorithms mentioned above perform poorly too on the code-mixed dataset. We think that this result may be caused by the characteristics of the data set and the impact of data imbalance. These factors are exactly the challenges we have to face to complete this task.

According to the characteristics of the dataset, we propose a multi-step integration method based on M-BERT[5] and XLM-RoBERTa[6]. In terms of method, we split a single quinary task into two subtasks, a coarse-grained binary classification task, and a fine-grained quaternary classification task. For the model, we use the combination of fine-tuned M-BERT and fine-tuned XLM-RoBERTa to complete the two split subtasks. According to the ranking results published by the task organization team and the scores on the published labeled test dataset, our method has proved to be effective.

2. Related Work

In recent years, the popularity of social media has lowered the threshold for the news release, and various issues have attracted widespread attention. Sentiment analysis in social media is worthy of our attention[7].

Mohammad etc.[8] used SVM to obtain the results of state-of-the-art in sentiment analysis of Tweets for message-level tasks. A machine learning method that replaces text with vectors and requires less computational resources was proposed by Giatsoglou etc.[9]. Sharma et al.[10] first proposed a method to solve the shallow analysis problem of Hindi-English code-mixed social media text (CSMT). Research on the word-level language recognition system was performed by Hittaranjan et al.[11]. The features obtained by these methods have good results on coarse-grained sentiment classification tasks. However, for more fine-grained sentiment classification tasks, it is necessary to obtain the semantic information of the entire sentence or the entire paragraph. Therefore, the use of supervised deep learning methods in sentiment analysis tasks has become a new solution. Deep learning can use deeper artificial neural networks to learn richer semantic information. Joshi et al.[12] introduced the learning sub-word level representation in the LSTM (Subword-LSTM) architecture to capture information about the emotional value of important morphemes. Related work using CNN and BiLSTM was reported to separate emotions from text with code-mixed[13]. Chakravarthi[14] used orthography to reduce the impact of code-mixing on results.

From the results of multiple experimental attempts mentioned above and the work of Bharathi et al. [4][15], we know their attempts are less effective on the Mixed-feeling label. The main reason is that fine-grained emotion classification models need to obtain rich contextual semantic information to have good results. For this task, we have to solve the difficulties caused by Multilingual Code MixingText. Our method combines the multi-language pre-training model M-

BERT and XLM-RoBERTa based on the Transformer architecture. M-BERRT and XLM-RoBERTa not only perform well in obtaining contextual semantic information, but also can deal with the difficulties caused by mixed language problems. In terms of method, we are also different from previous work. We split the fine-grained quinary classification task into two related subtasks. Convert the difficult problem into two relatively simple sub-problems.

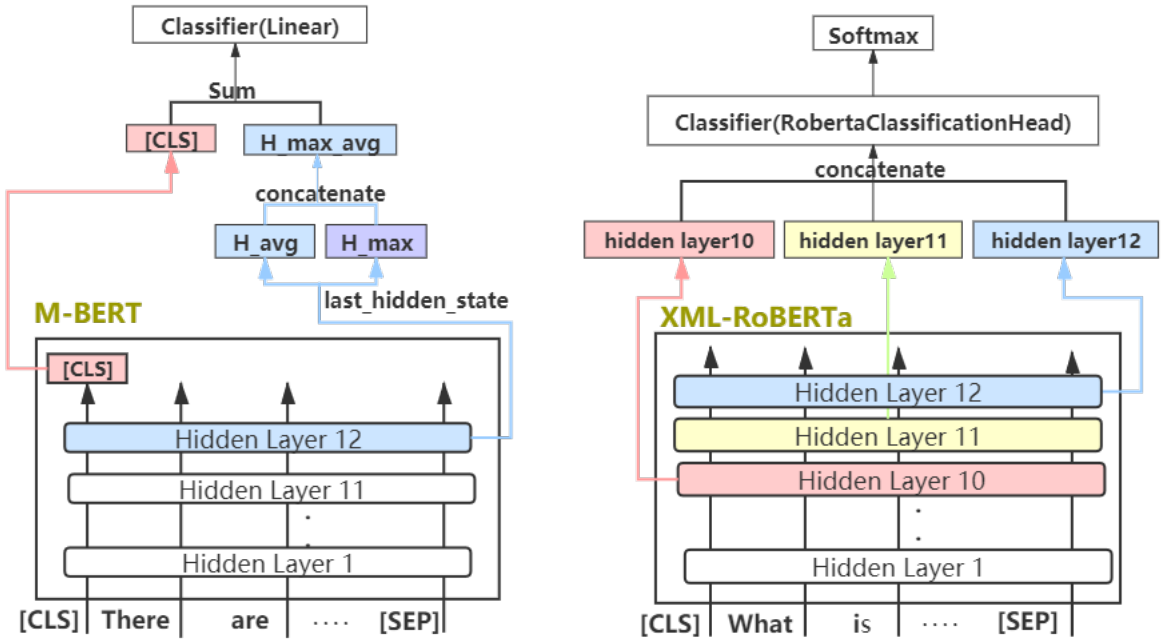

Figure 1: Fine-tuned Multilingual BERT (M-BERT) and the fine-tuned XLM-RoBERTa. *Linear* is a linear classifier function. *RobertaClassificationHead* is a classification function of Roberta. The *RobertaClassificationHead* classifier is a combination of the dropout layer, dense layer, tanh activation function layer, dropout layer, and linear layer in order.

3. Data and Methods

3.1. Data Description

The official training set and validation set of Malayalam-English and Tamil-English announced during the task are all from YouTube comments. As described in Introduction part, both the Malayalam language and the Tamil language use five types of tags to label each piece of text data. The training set and validation set of the Malayalam language in the data set are 4,851 and 540, and the Tamil language is 11,335 and 3,149 respectively. These data are very unbalanced in the distribution of the five categories, and the text in the data contains many special symbols, emoticons, and some unknown letter combinations. Some data examples are given in the Data Preprocessing part.

3.2. Fine-tuned of M-BERT and XLM-RoBERTa

We briefly analyze the performance of BERT on the data set in the second paragraph of the **Introduction**. Therefore, in this task, we choose the M-BERT (multi-language BERT) and XLM-RoBERTa models for fine-tuning. The difference between M-BERT and BERT is that M-BERT is not trained in a single language. M-BERT's corpus comes from Wikipedia's 104 language pages, which share a vocabulary of 119,547 words. Compared with M-BERT, XLM-RoBERTa uses larger and updated multilingual training data.

The models are built using HuggingFaces Transformers[16]. For the fine-tuning of M-BERT, set *kernel_size* to 2 for pooling, and use *max_pool1d* to process the last hidden layer output to get *H_max*. In a similar process, the *avg_pool1d* is used to process the output of the last hidden layer to obtain *H_avg*. Then, concatenate the two results according to the 2-dimensional position to obtain *H_max_avg*. Next, the output result of *[CLS]* is obtained, and the 0-dimensional and 2-dimensional results of the *H_max_avg* matrix are obtained. Finally, the two results are added and sent to the linear classifier[17].

For the fine-tuning of XLM-RoBERTa, the output results of the last three hidden layers (hidden layer 10,11,12) of XLM-RoBERTa are obtained, and then the three results are connected according to the 2-dimensional position to obtain a new matrix. Next, input this new matrix into the classifier(*RobertaClassificationHead* classifier [1]) to get the result. Finally, the result of the classifier processing obtained in the previous step is input to the softmax layer. The detailed structure of the two models is shown in **Figure 1**.

3.3. Model Training/Prediction and Result Processing Flow

The idea we explore in our work is to combine the binary classification of M-BERT with the quaternary classification of XLM-RoBERTa. For the preprocessed data(refer to **Data Preprocessing**), we use M-BERT for binary classification operations and XLM-RoBERTa quaternary classification operations. Then two trained models are used to predict the preprocessed test set, and the two predicted results are spliced to obtain the final prediction result. When predicting the test set, the 0-label data is removed from the M-BERT binary classification prediction result, and then XLM-RoBERTa is used to predict the quaternary classification result. Model training, prediction process, and result processing are shown in **Figure 2**.

4. Experiment and Results

4.1. Data Preprocessing

According to our previous data analysis of **Data Description**, we have preprocessed the special symbol data, number data, and continuously repeated character data. We do not directly delete these special characters, we think they contain certain emotional information, especially emoticons. Our approach is to replace these special symbols with corresponding text. As is shown in the comparison case below.

[1]https://github.com/huggingface/transformers/blob/466115b2797b6e01cce5c979bd8e20b3a1e04746/src/transformers/modeling_roberta.py#L1205

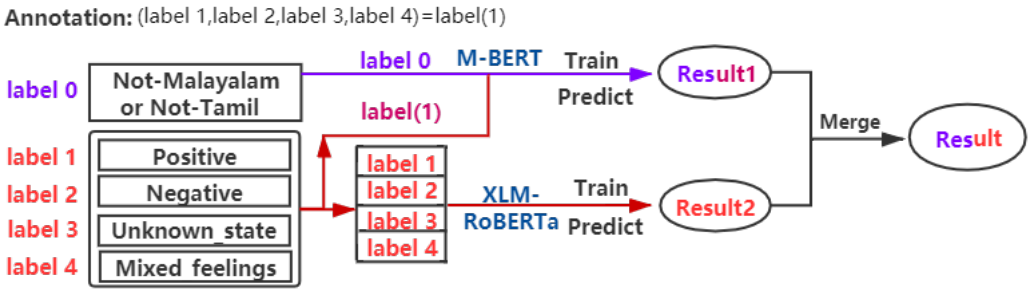

Figure 2: Flow chart of model training/prediction and result processing

- **Before**: Point black vs Kadaram Kondan .....!!!!!!!!!!!!!1;
  After : point black vs kadaram kondan . ! <repeated> <number>
- **Before**: Petta #1 Trending on Singapore :) superrrrrrrr ♡;
  After : petta <number> trending on singapore <happy> super <elongated> <red heart>
- **Before**: #1 trending in #Srilanka #3 trending;
  After : <number> trending in <hashtag> srilanka </hashtag> <number> trending [2] [3]

4.2. Experiment setting

In our experiment, we use the official validation set as the test set to test the results of the model prediction. To improve the generalization ability of the model, we perform k-fold cross-validation on the preprocessed data. The training set is used to perform 5-fold cross-validation processing. The model with the highest F1-Score is saved in model training. The *epoch, batch size, maximum sequence length*, and *learning rate* for M-BERT are 5, 32, 50, and 4e-5, respectively. The *epoch, batch size, maximum sequence length*, and *learning rate* for XLM-RoBERTa are 10, 32, 50, and 5e-5, respectively.

4.3. Results

In this competition, the evaluation index given by the task organizer is weighted average F1-Score. Among the three results we submitted, one of them is the result of XLM-RoBERTa's binary classification and M-BERT's quaternary classification. But the result is not as good as the method in this paper. In the official ranking results, the weighted average F1-Score of our Malayalam and Tamil are 0.01 and 0.02 lower than the first place. Compared with the result of BERT[3][4], the score of our method on the Mixed-feeling label has also improved. We analyze the results from two aspects. In terms of method, we decompose the complex quinary classification problem into two relatively simple sub-problems. The advantage of this is that it allows us to choose appropriate methods and models for different sub-problems. In terms

[2] https://emojipedia.org/
[3] https://github.com/huggingface/transformers

of models, after we split the task of quinary classification into two different subtasks, we use different models for different subtasks. We process the output of the last hidden layer of M-BERT, and the output of the last three hidden layers of XLM-RoBERTa. Therefore, there are some gaps between the results of the two models in the coarse-grained binary classification sub-problem and the fine-grained quaternary classification sub-problem. The comparison results can be found in **Table 1** and **Table 2**.

Table 1
Comparison of classification results of different combinations of M-BERT and XLM-RoBERTa. (2) (4) are binary classification and quaternary classification, M=Malayalam, T=Tamil

Type	Precision_M	Recall_M	F1-Score_M	Precision_T	Recall_T	F1-Score_T
M-BERT (2) and XLM-RoBERTa (4)	0.73	0.73	0.73	0.62	0.67	0.64
XLM-RoBERTa (2) and M-BERT (4)	0.71	0.71	0.71	0.61	0.66	0.63

Table 2
Comparison of the scores of the Mixed-feeling label in BERT and our method. M=Malayalam, T=Tamil

Type	Precision_M	Recall_M	F1-Score_M	Precision_T	Recall_T	F1-Score_T
Our method	0.38	0.47	0.42	0.23	0.09	0.13
BERT/M-BERT	0.00	0.00	0.00	0.00	0.00	0.00

5. Conclusion

In this paper, we propose a method combining M-BERT and XLM-RoBERTa to complete the sentiment analysis of multilingual Code-Mixed Texts. We make several contributions to similar issues in this task. The first part is the replacement scheme we use in data preprocessing. The second part is to convert the multi-label classification problem into multiple sub-problems, and then solve the problem step by step. The third part is about the fine-tuning scheme of XLM-RoBERTa and M-BERT. Good results have been achieved in both the Malayalam language and the Tamil language. In future research, we will consider how to better improve the recognition rate of the Mixed-feeling label.

References

[1] V. Subramaniyaswamy, R. Logesh, M. Abejith, S. Umasankar, A. Umamakeswari, Sentiment analysis of tweets for estimating criticality and security of events, in: Improving the Safety and Efficiency of Emergency Services: Emerging Tools and Technologies for First Responders, IGI Global, 2020, pp. 293–319.

[2] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, S. Suryawanshi, N. Jose, J. P. Sherly, Elizabeth McCrae, 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, 2020.
[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, 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.
[5] 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.
[6] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, arXiv preprint arXiv:1911.02116 (2019).
[7] L. Yue, W. Chen, X. Li, W. Zuo, M. Yin, A survey of sentiment analysis in social media, Knowledge and Information Systems (2019) 1–47.
[8] H. T. Madabushi, E. Kochkina, M. Castelle, Cost-sensitive bert for generalisable sentence classification with imbalanced data, arXiv preprint arXiv:2003.11563 (2020).
[9] M. Giatsoglou, M. G. Vozalis, K. Diamantaras, A. Vakali, G. Sarigiannidis, K. C. Chatzisavvas, Sentiment analysis leveraging emotions and word embeddings, Expert Systems with Applications 69 (2017) 214–224.
[10] A. Sharma, S. Gupta, R. Motlani, P. Bansal, M. Srivastava, R. Mamidi, D. M. Sharma, Shallow parsing pipeline for Hindi-English code-mixed social media text, arXiv preprint arXiv:1604.03136 (2016).
[11] G. Chittaranjan, Y. Vyas, K. Bali, M. Choudhury, Word-level language identification using crf: Code-switching shared task report of msr India system, in: Proceedings of The First Workshop on Computational Approaches to Code Switching, 2014, pp. 73–79.
[12] A. Joshi, A. Prabhu, M. Shrivastava, V. Varma, Towards sub-word level compositions for sentiment analysis of Hindi-English code mixed text, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 2482–2491.
[13] Y. K. Lal, V. Kumar, M. Dhar, M. Shrivastava, P. Koehn, De-mixing sentiment from code-mixed text, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, 2019, pp. 371–377.
[14] B. R. Chakravarthi, Leveraging orthographic information to improve machine translation of under-resourced languages, Ph.D. thesis, NUI Galway, 2020.
[15] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, S. Suryawanshi, N. Jose, J. P. Sherly, Elizabeth McCrae, Overview of the track on Sentiment Analysis for Dravidian Languages in

Code-Mixed Text, in: Proceedings of the 12th Forum for Information Retrieval Evaluation, FIRE '20, 2020.

[16] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Huggingface's transformers: State-of-the-art natural language processing, ArXiv abs/1910.03771 (2019).

[17] C. Sun, X. Qiu, Y. Xu, X. Huang, How to fine-tune bert for text classification?, in: China National Conference on Chinese Computational Linguistics, Springer, 2019, pp. 194–206.