Automatic Detecting the Sentiment of Code-Mixed Text by Pre-training Model

Yang Bai, Bangyuan Zhang, Wanli Chen, Yongjie Gu, Tongfeng Guan and Qisong Shi

Data Intelligence Center, ZhongYuan Bank, Henan, P.R.China

Abstract

With the development of the Internet, more and more people can express their opinions on social media, which are positive or negative. They sometimes have a positive or negative impact on society or individuals. So how to detect sentiment automatically is very important, especially in code-mixed text. In this paper, we describe our system submitted to Dravidian-CodeMix-FIRE2021 which consists of three sub-tasks. We participate in Malayalam-English task and Tamil-English task. For the system we submitted, XLM-Roberta is used as the base model, which is fine-tuned in the Malayalam-English and Tamil-English data in 2020 and 2021. Finally, our system achieves 80.40% of the F1-score in the Malayalam-English task and 67.60% of the F1-score in the Tamil-English task.

Keywords

code-mixed, sentiment, XLM-Roberta

1. Introduction

With the rapid development of social networks such as Twitter and Facebook, people can express their views and opinions on the Internet anytime, anywhere. In social media, users can freely express their views which are negative or positive.[1]These views will have an impact on individuals and society.[2] So an automated method is needed to recognize the sentiment of these views. This involves sentiment analysis.

With the development of globalization, people from different countries communicate with each other on social media. So there are a lot of code-mixed texts on the Internet.[3] For example, there are many Malayalam-English[4], Tamil-English[5] and Kannada-English[6] code-mixed text on the Internet. Malayalam is one of the Dravidian languages in southern India. Tamil is a language with a history of more than 2000 years. It belongs to the Dravidian language family and is popular in southern India and northeast Sri Lanka. Kannada is the official language of Karnataka in India and belongs to the Dravidian language. In this paper, the used methods and the results obtained by our team, ZYBank-AI, on the Dravidian-CodeMix-FIRE 2021 are presented.[7] We participate in the Malayalam-English and Tamil-English.

2. Related Work

In recent years, more and more researchers have focused on the sentiment analysis of code-mixed texts.[8] Among the traditional methods, there are methods based on dictionaries and machine learning to solve sentiment analysis problems. But these methods are not effective on social media data. In recent years, with the development of transfer learning technology, language models such as word2vec[9], BERT[10], XLM-RoBERTa[11], etc. have emerged. At the same time, they have greatly promoted the development of multilingual sentiment analysis. In 2020, Bharathi et al.[12] organized Dravidian-CodeMix-FIRE 2020. This is the first shared task of sentiment analysis in Malayalam-English and Tamil-English. The SRJ team[13] combined XLM-RoBERTa and CNN method of fine-tuning through downstream tasks and ranked first in the two subtasks.

3. Methodology

3.1. Dataset

For the Malayalam and Tamil tasks, all the data in Dravidian-CodeMix-FIRE 2020 (last year's data) and the data in Dravidian-CodeMix-FIRE 2021 (this year's data) are used as the training data of the model.

In the Malayalam task. (1) For Dravidian-CodeMix-FIRE 2020, we combine the training dataset, validation dataset and test dataset. After combining, we scramble it and divide it into the new training dataset and validation dataset according to the ratio of 5:1. In the new training set, there are 5777 sentences, 2410 sentences for 'Positive' class, 632 sentences for 'Negative', 1631 sentences for 'unknown_state', 347 sentences for 'Mixed_feelings', and 757 sentences for 'not-malayalam'. In the new validation set, there are 963 sentences, 401 sentences for 'Positive' class, 109 sentences for 'Negative', 272 sentences for 'unknown_state', 57 sentences for 'Mixed_feelings', and 127 sentences for 'not-malayalam'. (2) For Dravidian-CodeMix-FIRE 2021, we combine the training dataset and validation dataset. After combining, we use stratified 5-fold cross-validation to divide the data.

In the Tamil task. (1) For Dravidian-CodeMix-FIRE 2020, we combine the training dataset, validation dataset and test dataset. After combining, we scramble it and divide it into the new training dataset and validation dataset according to the ratio of 5:1. In the new training set, there are 13494 sentences, 9050 sentences for 'Positive' class, 1746 sentences for 'Negative', 729 sentences for 'unknown_state', 1543 sentences for 'Mixed_feelings', and 426 sentences for 'not-Tamil'. In the new validation set, there are 2250 sentences, 1509 sentences for 'Positive' class, 291 sentences for 'Negative', 121 sentences for 'unknown_state', 258 sentences for 'Mixed_feelings', and 71 sentences for 'not-Tamil'. (2) For Dravidian-CodeMix-FIRE 2021, we combine the training dataset and validation dataset. After combining, we use stratified 5-fold cross-validation to divide the data.
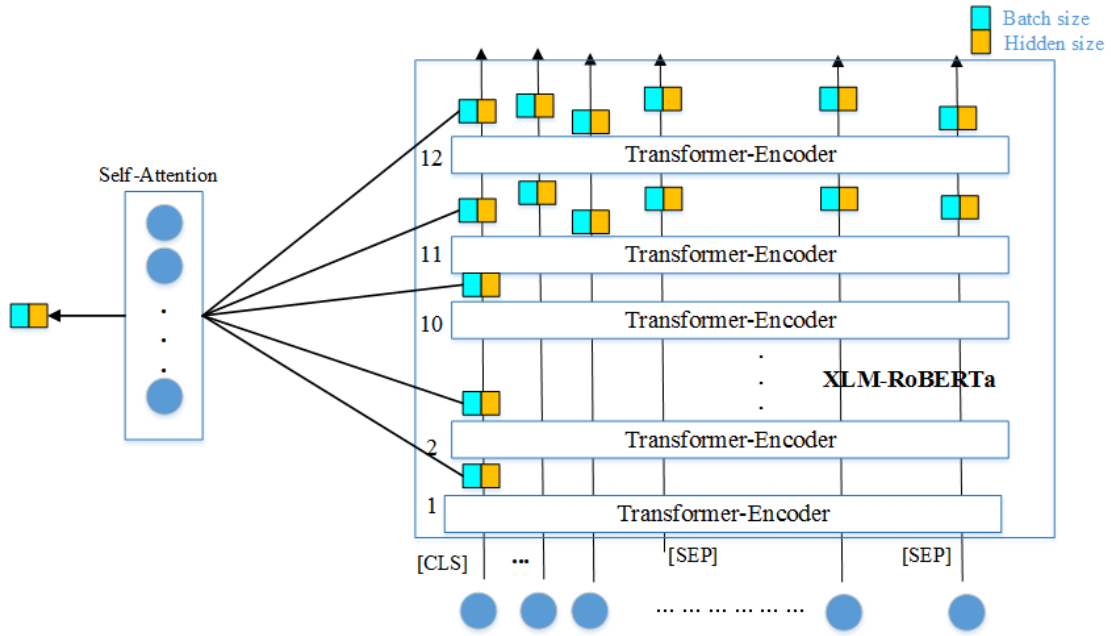
Figure 1: XLM-RoBERTa with Attention.

3.2. Method Description

In this task, our main work is carried out on the pre-training language model. For the pre-training language model, we choose the XLM-RoBERTa as the base model. The XML-RoBERTa model was proposed by the Facebook AI team. It can be understood as the combination of XLM and Roberta. It is trained in 100 languages and 2.5TB of text (crawled by the common Crawl corpus). It shows excellent performance in many multilingual tasks. So we choose XLM-RoBERTa model in this work. Recent research has pointed out that the 12 hidden layers of the BERT model contain different linguistic information. In order to obtain more semantic information, we apply the Self-Attention mechanism to the 12 hidden layers of XLM-RoBERTa. Figure 1 shows our model. The implementation flow of Attention mechanism is shown in Figure 2.

The size of the data often determines the quality of the final result. Due to the small number of official data sets, the performance of the model will be affected. Therefore, we fine-tune the model in the first stage of the Dravidian-CodeMix-FIRE 2020 Malayalam and Tamil language data, and select the checkpoint with the best result in the validation set as the first-stage fine-tuned model. Then, the fine-tuned model in the first stage is trained on this year's Malayalam language and Tamil language data, and the checkpoint with the best result in the validation set is selected as the second-stage fine-tuned model. The method we used is shown in Figure 3.
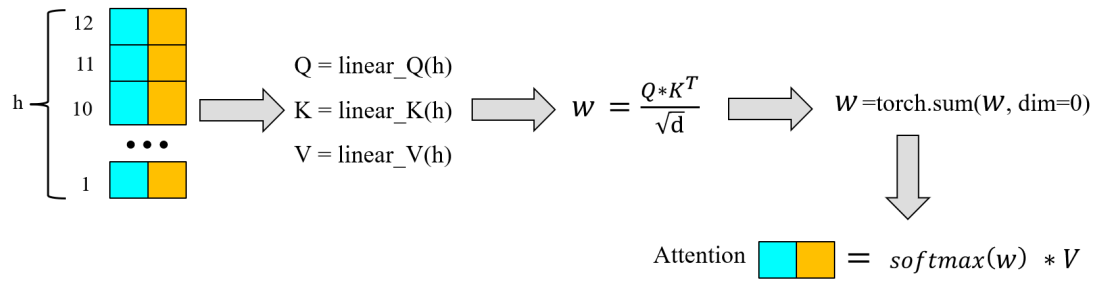
Figure 2: Attention mechanism.
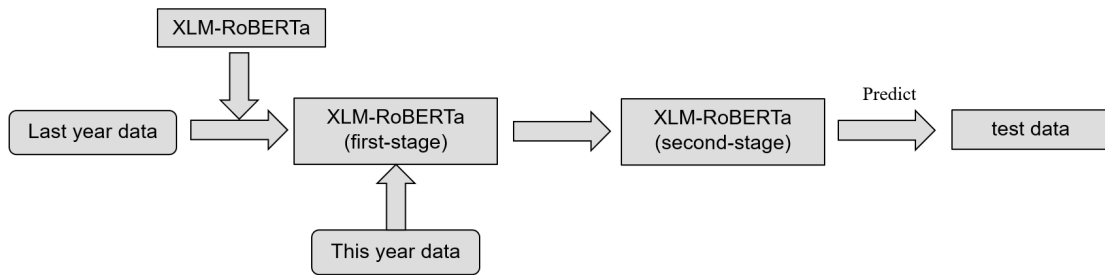
Figure 3: our Method.

3.3. R-Drop

Deep neural network (DNN) has achieved remarkable success in various fields recently. When training these large-scale DNN models, regularization techniques such as L2 normalization, batch normalization and dropout are indispensable modules. These techniques can prevent the model from over-fitting and at the same time improve the generalization ability of the model. Among them, dropout technology has become the most widely used regularization technology because it only needs to discard a part of neurons in the training process.

The researchers proposed a further regularization method based on Dropout: Regularized Dropout (R-Drop)[14]. In each mini-batch, each data sample has the same model with Dropout twice, and R-Drop uses KL-divergence to constrain the same output twice. The results show that R-Drop has achieved a good result improvement.

4. Experiments

4.1. Experiments Setup

These two subtasks are five classification tasks. The official evaluation metrics is weighted average F1-score. The F1-score formula is as follows:

Table 1
Hyperparameters of the first-stage and second-stage.(α is a parameter in R-Drop.)

Hyperparameters	first stage training	second stage training
bacth size	4	4
gradient accumulation step	4	4
learn rate	3e-5	6e-5
dropout	0.3	0.3
epoch	15	15
α	4	4

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = \frac{\text{Precision} * \text{Recall} * 2}{\text{Precesion} + \text{Recall}} \tag{3}$$

4.2. Results

In this work, our method is based on pytorch implementation. The xlm-roberta-base is used as the pre-training language model, Adamw is used as an optimizer, and R-Drop is used for regularization. There is no preprocessing for all data. For the first stage of training, the hyperparameters are the same in the two subtasks. For the second stage of training, the hyperparameters are also the same. Table 1 shows the hyperparameters of our experiment.Finally, our team achieve the 80.40% F1-score in Malayalam task and 67.60% F1-score in Tamil task.

5. Conclusion

This paper introduces the overall idea and specific scheme of ZYBank-AI Team in Dravidian-CodeMix-FIRE 2021. We participate in Malayalam-English task and Tamil-English task. Considering the problem of data size, we use the data set of Dravidian-CodeMix-FIRE2020. In order to be able to obtain more semantic information, we combine XLM-RoBERTa and self-attention mechanism. Our code is available at https://github.com/byew/codemix-2021-code.

References

[1] T. Mandl, S. Modha, A. Kumar M, B. R. Chakravarthi, Overview of the hasoc track at fire 2020: Hate speech and offensive language identification in Tamil, Malayalam, Hindi, English and German, in: Forum for Information Retrieval Evaluation, 2020, pp. 29–32.
[2] 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-Dravidian-CodeMix Shared Task on Offensive Language Detection in Tamil and

Malayalam, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.

[3] B. R. Chakravarthi, R. Priyadharshini, S. Thavareesan, D. Chinnappa, T. Durairaj, E. Sherly, J. P. McCrae, A. Hande, R. Ponnusamy, S. Banerjee, C. Vasantharajan, Findings of the Sentiment Analysis of Dravidian Languages in Code-Mixed Text 2021, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.

[4] 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.

[5] 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.

[6] A. Hande, R. Priyadharshini, B. R. Chakravarthi, KanCMD: Kannada CodeMixed dataset for sentiment analysis and offensive language detection, 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. 54–63. URL: https://www.aclweb.org/anthology/2020.peoples-1.6.

[7] R. Priyadharshini, B. R. Chakravarthi, S. Thavareesan, D. Chinnappa, T. Durairaj, E. Sherly, Overview of the Dravidiancodemix 2021 shared task on sentiment detection in Tamil, Malayalam, and Kannada, in: Forum for Information Retrieval Evaluation, FIRE 2021, Association for Computing Machinery, 2021.

[8] B. R. Chakravarthi, R. Priyadharshini, N. Jose, A. Kumar M, T. Mandl, P. K. Kumaresan, R. Ponnusamy, H. R L, J. P. McCrae, E. Sherly, Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada, in: Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, Association for Computational Linguistics, Kyiv, 2021, pp. 133–145. URL: https://aclanthology.org/2021.dravidianlangtech-1.17.

[9] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).

[10] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[11] 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).

[12] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, S. Suryawanshi, N. Jose, E. Sherly, J. P. McCrae, Overview of the track on sentiment analysis for Dravidian languages in code-mixed text, in: Forum for Information Retrieval Evaluation, 2020, pp. 21–24.

[13] R. Sun, X. Zhou, Srj@ Dravidian-codemix-fire2020: Automatic classification and identifi-

cation sentiment in code-mixed text., in: FIRE (Working Notes), 2020, pp. 548–553.

[14] X. Liang, L. Wu, J. Li, Y. Wang, Q. Meng, T. Qin, W. Chen, M. Zhang, T.-Y. Liu, R-drop: Regularized dropout for neural networks, arXiv preprint arXiv:2106.14448 (2021).