Maoqin @ DravidianLangTech-EACL2021: The Application of Transformer-Based Model

Maoqin Yang
School of Information Science and Engineering
Yunnan University, Yunnan, P.R. China
1695157605@qq.com

Abstract

This paper describes the result of team-Maoqin at DravidianLangTech-EACL2021. The provided task consists of three languages (Tamil, Malayalam, and Kannada), I only participate in one of the language task-Malayalam. The goal of this task is to identify offensive language content of the code-mixed dataset of comments/posts in Dravidian Languages (Tamil-English, Malayalam-English, and Kannada-English) collected from social media. This is a classification task at the comment/post level. Given a Youtube comment, systems have to classify it into Not-offensive, Offensive-untargeted, Offensive-targeted-individual, Offensive-targeted-group, Offensive-targeted-other, or Not-in-indented-language. I use the transformer-based language model with BiGRU-Attention to complete this task. To prove the validity of the model, I also use some other neural network models for comparison. And finally, the team ranks 5th in this task with a weighted average F1 score of 0.93 on the private leader board.

1 Introduction

Offensive language refers to direct or indirect use of verbal abuse, slander, contempt, ridicule, and other means to infringe or damage the dignity, spiritual world, and mental health of others. It will seriously affect the mental state of others, disrupt work, the life and learning order of others, and seriously pollute the public opinion environment of the entire network(Schmidt and Wiegand, 2017).

Due to the development of the Internet and the popularity of anonymous comments, many offensive languages have spread on the Internet and caused trouble to relevant personnel (Thavareesan and Mahesan, 2019, 2020a,b). Relevant organizations should take measures to prevent this from happening. It is unrealistic to judge whether online sentences are completely offended by humans. Therefore, mechanical methods must be used to distinguish whether the language is offensive. The task is to directly test whether the system can distinguish offensive language in Dravidian languages. Dravidian languages are a group of languages spoken by 220 million people, predominantly in southern India and northern Sri Lanka, but also in other areas of South Asia. The Dravidian languages were first recorded in Tamili script inscribed on cave walls in Tamil Nadu's Madurai and Tirunelveli districts in the 6th century BCE. The Dravidian languages are closely related languages the are under-resourced (Chakravarthi, 2020).

Existing deep learning and pre-training models have achieved good results on other tasks(Zampieri et al., 2019), so I use the deep learning method to deal with the related task. According to the latest related research progress, the transformer-based language model has become my preferred model. Because the pre-trained and fine-tuned transformers-based models have shown excellent performance in many NLP problems, such as sentiment classification and automatic extraction of text summaries. So I choose ALBERT(Lan et al., 2019) as my basic model in this task. To get a more effective and higher accuracy model, BiGRU combined with attention. To prove the effectiveness of this model, I have also done comparative experiments with other neural networks. In this task, my model is an effective way to perform well. To obtain as much effective information as possible from the limited data, I also use the 5-fold cross-validation method. my model achieves the desired result.

The rest of this article is structured as follows. Section 2 introduces related work. Model and data preparation are described in Section 3. Experiments and evaluation are described in Section 4. Section 5 describes the results of my work. The conclusions and future work are drawn in Section 6.

Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pages 281–286
April 20, 2021 ©2021 Association for Computational Linguistics

2 Related Work

There are many competitions about offensive language detection(such as HASOC (Chakravarthi et al., 2020c; Mandl et al., 2020) and TRAC (Kumar et al., 2018)), and many corresponding methods have been produced. People often tend to abstract this task into a text classification task (Howard and Ruder, 2018).

Text classification is called extracting features from original text data and predicting the category of text data based on these features. In the past few decades, many models for text classification have been proposed (Qian, 2020).

From the 1960s to the 2010s, text classification models based on shallow learning dominated. Shallow learning means statistical-based models such as Naive Bayes (NB), K Nearest Neighbors (KNN)(Cover and Hart, 1967) and Support Vector Machines (SVM). Compared with earlier rule-based methods, this method has obvious advantages in accuracy and stability. However, these methods still require functional design, which is time-consuming and expensive. In addition, they usually ignore the natural order structure or context information in the text data, which makes learning the semantic information of words difficult. Since the 2010s, text classification has gradually changed from a shallow learning model to a deep learning model. Compared with methods based on shallow learning, deep learning methods avoid the manual design of rules and functions and automatically provide semantically meaningful representations for text mining. Therefore, most of the text classification research work is based on DNN(Yu et al., 2013), which is a data-driven method with high computational complexity. Few studies have focused on shallow learning models to solve the limitations of computation and data.

The shallow learning model speeds up the text classification speed, improves the accuracy, and expands the application range of shallow learning.

The shallow learning method is a type of machine learning. It learns from data, which is a predefined function that is important to the performance of the predicted value. However, element engineering is an arduous and giant job. Before training the classifier, we need to collect knowledge or experience to extract features from the original text. The shallow learning method trains the initial classifier based on various text features extracted from the original text. For small data sets, under the limitation of computational complexity, shallow learning models generally show better performance than deep learning models. Therefore, some researchers have studied the design of shallow models in specific areas of data replacement.

Set	Total number
train	16010
development	1999
test	2001

Table 1: The number of sentences in each set.

Deep learning consists of multiple hidden layers in a neural network(Aroyehun and Gelbukh, 2018), has higher complexity, and can be trained on unstructured data. The deep learning architecture can directly learn feature representations from the input without excessive manual intervention and prior knowledge. However, deep learning technology is a data-driven method that usually requires a lot of data to achieve high performance. And the self-attention-based model can bring some interword interpretability to DNN, but the comparison with the shallow model does not explain why and how it works.

3 Methodology and Data

An overall framework and processing pipeline of my solution are shown in Figure 1.

In my job, I use the ALBERT model as my base model and take BiGRU-Attention behind it. My model is shown in Figure 2.

3.1 Data Preparation

This is a comment/post level classification task. Given a Youtube comment (Chakravarthi et al., 2020b,a, 2021; Chakravarthi and Muralidaran, 2021), the system has to classify it into one of the five categories mentioned in the $Abstract$ section. For this task, the available sentences including 16010 training sentences, 1999 development sentences, and 2001 testing sentences. The label distribution is very uneven(Not-offensive label accounts 88.4%. The label with the second largest number is not-malayalam, which accounts for only 0.08% of the total. And there are relatively fewer labels in other categories.)The number of sentences for each domain is listed in Table 1.

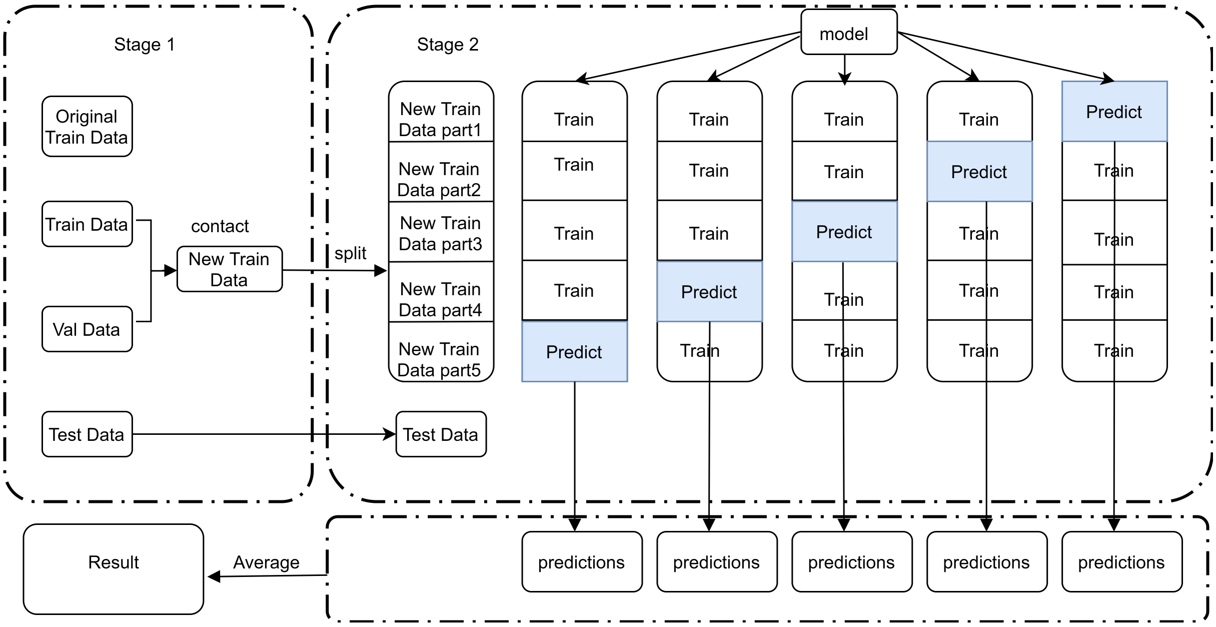

Figure 1: An overall framework

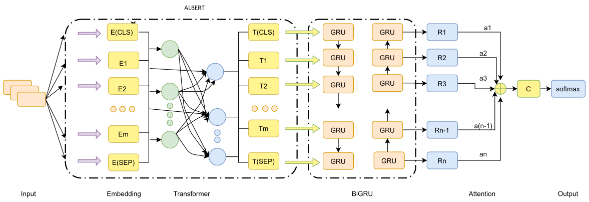

Figure 2: The architecture of the model, where the $E[CLS]$ and $E[SEP]$ are added at the beginning and end of each instance respectively, which can separate different sentences. The format is as follows: $[CLS]$+sentence+$[SEP]$.

3.2 ALBERT

The ALBERT model belongs to transformer-based language models. The ALBERT model is improved on the basis of Bidirectional Encoder Representations for Transformers(BERT)(Devlin et al., 2018) model. It has designed a parameter reduction method to reduce memory consumption by changing the result of the original embedding parameter P (the product of the vocabulary size V and the hidden layer size H).

$$V * H = P \rightarrow V * E + E * H = P \quad (1)$$

E represents the size of the low-dimensional embedding space. In BERT, $E = H$. While in ALBERT, $H >> E$, so the number of parameters will be greatly reduced. At the same time, the self-supervised loss is used to focus on the internal coherence in the construction of sentences. The ALBERT model implements three embedding layers: word embedding, position embedding, and segment embedding. The token embedding layer predicts each word as a fixed-size vector. Position embedding is used to retain position information, use a vector to randomly initialize each position, add model training, and finally obtain an embedding containing position information. Segment embedding helps BERT distinguish between paired input sequences.

3.3 BiGRU-Attention

The BiGRU-Attention model(Cover and Hart, 1967) is divided into three parts: text vector input layer, hidden layer, and output layer. Among them, the hidden layer consists of three layers: the BiGRU layer, the attention layer, and the Dense layer (fully connected layer). I set the output of the ALBERT model as the input. After receiving the input, it uses the BiGRU neural network layer to extract features of the deep-level information of the text firstly. Secondly, it uses the attention layer to assign corresponding weights to the deep-level information of the extracted text. Finally, the text feature information with different weights is put into the softmax function layer for classification. The structure of the BiGRU-Attention model is shown in Figure 3.

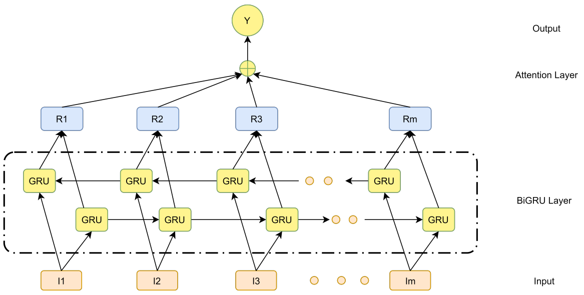

Figure 3: The structure of the BiGRU-Attention model. The $I1, I2...Im$ represent the output of the ALBERT layer and the $R1, R2...Rm$ represent the output of the BiGRU layer and will be input to the Attention layer.

Model	ALBERT(Base)
train step	2501
learning rate	2e-5
batch size	32
epoch	5

Table 2: The parameter configuration of ALBERT.

4 Experiment

In this task, I use the ALBERT model to pre-train the task. For the ALBERT model, the main hyper-parameters I pay attention to are the training step size, batch size and learning rate. The parameters of my model are shown in Table 2.

I have obtained good performance using the ALBERT-BASE.[1] model. Considering that BiGRU-Attention can capture contextual information well and extract text information features more accurately(Radford et al., 2018), I add it after ALBERT. I use the development data set to verify the performance of the models. The standard of judgment is a weighted F1-score, and this standard is the judgment standard used for my task. Table3 lists the results of various models described previously. The best performance is in bold. My model gets the best performance of 0.93. As shown in the table my model can greatly improve the performance and my overall approach achieved 5th place on the final leader board.

5 Results

The output of the classification result is shown in Figure 4. We can see that the label of $Offensive - Targeted - Insult - Other$, $Offensive - Targeted - Insult - Individual$, and $Offensive - Targeted - Insult - Group$ is zero. $Not - Offensive$ labels account for the majority, accounting for 91.15% of the total number of labels. The $Not - Malayalam$ labels account for the second most significant 7.5% of the total. Offensive-Untargeted labels are the least, only about 1%. This may be due to data imbalance ($Not - Offensive$ labels in the training set account for about 88% of the total) resulting in only three categories being identified.

Model	**F1**
ALBERT(Base)	0.919
BERT(Base)	0.912
RoBERTa(Base)	0.920
BERT(Base)+BiGRU-Attention	0.928
Mine(ALBERT+BiGRU-Attention)	**0.930**

Table 3: Results of comparative experiments.

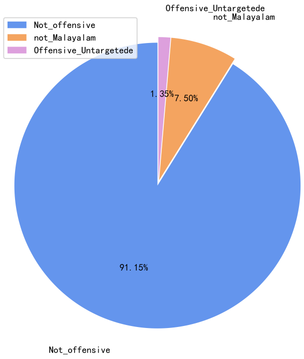

Figure 4: The classification result

6 Conclusion and Future Work

In this paper, I present my result on Offensive Language Identification in Dravidian Languages-EACL 2021 which includes three tasks of different languages. For this task, I regard it as a multiple classification task, I use the BiGRU-Attention based on the ALBERT model to complete, and my model works very well. I also summarized the possible reasons for classifying only three types of labels. At the same time, I also use some other neural networks for comparative experiments to prove that my model can obtain excellent performance. The result shows that my model ranks 5th in the Malayalam task.

Due to the continuous development of the definition of offensive information on the Internet, it is difficult to accurately describe the nature of

[1] https://huggingface.co/albert-base-v2

this information only from the perspective of data mining, which makes it impossible to model this information effectively. In the future, I will use methods based on multidisciplinary discovery to guide model learning. These models are more likely to use limited data to learn more effective models. At the same time, I will also consider whether I can use other transfer learning models to perform better on multi-classification tasks.

References

S. T. Aroyehun and A. Gelbukh. 2018. Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 90–97.

Bharathi Raja Chakravarthi. 2020. *Leveraging orthographic information to improve machine translation of under-resourced languages*. Ph.D. thesis, NUI Galway.

Bharathi Raja Chakravarthi, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John Philip McCrae. 2020a. 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)*, pages 177–184, Marseille, France. European Language Resources association.

Bharathi Raja Chakravarthi and Vigneshwaran Muralidaran. 2021. Findings of the shared task on Hope Speech Detection for Equality, Diversity, and Inclusion. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John Philip McCrae. 2020b. 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)*, pages 202–210, Marseille, France. European Language Resources association.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Navya Jose, Anand Kumar M, Thomas Mandl, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Hariharan V, Elizabeth Sherly, and John Philip McCrae. 2021. 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.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Shardul Suryawanshi, Navya Jose, Elizabeth Sherly, and John P. McCrae. 2020c. Overview of the Track on Sentiment Analysis for Dravidian Languages in Code-Mixed Text. In *Forum for Information Retrieval Evaluation*, FIRE 2020, page 21–24, New York, NY, USA. Association for Computing Machinery.

Cover and Hart. 1967. Nearest neighbor pattern classification.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. arXiv:1801.06146.

R. Kumar, A. K. Ojha, S. Malmasi, and M. Zampieri. 2018. Benchmarking aggression identification in social media. *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, and Piyush Sharma. 2019. Albert: A lite bert for self-supervised learning of language representations. arXiv:1909.11942. Version 6.

Thomas Mandl, Sandip Modha, Anand Kumar M, and Bharathi Raja Chakravarthi. 2020. 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*, FIRE 2020, page 29–32, New York, NY, USA. Association for Computing Machinery.

Qu Qian. 2020. A review of the latest text classification 2020-the development of text classification from shallow to deep from 1961 to 2020.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. *Technical report, OpenAI.*

A. Schmidt and M. Wiegand. 2017. A survey on hate speech detection using natural language processing. *Proceedings of the Fifth International workshop on natural language processing for social media*, (1):1–10.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2019. Sentiment Analysis in Tamil Texts: A Study on Machine Learning Techniques and Feature Representation. In *2019 14th Conference on Industrial and Information Systems (ICIIS)*, pages 320–325.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2020a. Sentiment Lexicon Expansion using Word2vec and fastText for Sentiment Prediction in Tamil texts. In *2020 Moratuwa Engineering Research Conference (MERCon)*, pages 272–276.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2020b. Word embedding-based Part of Speech tagging in Tamil texts. In *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, pages 478–482.

Dong Yu, Michael L. Seltzer, Jinyu Li, Jui-Ting Huang, and Frank Seide. 2013. Feature learning in deep neural networks - studies on speech recognition tasks. arXiv:1301.3605.

M. Zampieri, S. Malmasi, P. Nakov, and S. Rosenthal. 2019. Predicting the type and target of offensive posts in social media. arXiv:1902.09666.