cs@DravidianLangTech-EACL2021: Offensive Language Identification Based On Multilingual BERT Model

Shi Chen
School of Information Science
and Engineering Yunnan University,
chen-s2020@139.com

Bing Kong
School of Information Science
and Engineering Yunnan University,
kongbing@ynu.edu.cn

Abstract

This paper introduces the related content of the task ”Offensive Language Identification in Dravidian LANGUAGES-EACL 2021”. The task requires us to classify Dravidian languages collected from social media into Not-Offensive, Off-Untargeted, Off-Target-Individual, etc. This data set contains actual annotations in code-mixed text posted by users on Youtube, not from the monolingual text in textbooks. Based on the features of the data set code mixture, we use multilingual BERT and TextCNN for semantic extraction and text classification. In this article, we will show the experiment and result analysis of this task.

1 Introduction

One of the manifestations of the rapid development of the Internet is the increasing number of users, which allows more and more people from different languages, different regions, and different cultures to communicate, but the frequent appearance of offensive speech will affect this harmonious atmosphere (Thavareesan and Mahesan, 2019, 2020a,b). This has become a serious problem for users of online communities and social media platforms. In such a multilingual social environment, it has become a serious problem for users of online communities and social media platforms (Jose et al., 2020; Priyadharshini et al., 2020; Chakravarthi et al., 2020c; Mandl et al., 2020).

This task is to identify offensive language content from Dravidian languages (Tamil-English, Malayalam-English, and Kannada-English) collected from social media. The Dravidian civilization of the Indus Valley civilisation (3,300–1,900 BCE) is believed to have flourished in the Northwestern Indian subcontinent (Tamil). The Dravidian languages were first documented in Tamil-Brahmi script engraved on cave walls in Tamil Nadu’s Madurai and Tirunelveli districts in the 6th century BCE. Tamil is India’s oldest language. Agglutinative languages are Dravidian languages. Subject–object–verb is the word order (SOV). There is a clusivity distinction in most Dravidian languages. All we need to do is to classify it into not-offensive, offensive-untargeted, offensive-targeted-individual, offensive-targeted-group, offensive-targeted-other, or not-in-indented-language.

To further extract semantic information, we adopt a text classification method based on the hierarchical connection between Bert and TextCNN, which is a combined model of Bert and TextCNN. Use multilingual BERT to vectorize each word, obtain the semantic features of the text, and construct the text mapping matrix. Use TextCNN convolutional neural network to perform convolution operation on text mapping matrix, get the output of all or part of the hidden layer, get the semantic feature matrix of the text, then use the pooling algorithm to reduce the dimension of the semantic feature matrix of the text to obtain the semantic feature vector of the text.

2 Related Work

Nowadays, offensive, false, and other remarks on social media have become issues that we have to pay attention to. To solve this problem, many scholars have done a lot of research activities.

(Sohn and Lee, 2019) developed multi-channel BERT models for different languages, integrated the hiding function of separate BERT models trained in different languages, and using transfer learning in NLP, the problem of the shortage of labeled data sets can be solved by pre-training the language model. (Shushkevich et al., 2020) proposed a method to solve multiple classification problems within the framework of active language recognition in Twitter. Created a collection of classic

Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pages 230–235
April 20, 2021 ©2021 Association for Computational Linguistics

machine learning models including Logistic regression, support vector machines, naive Bayes models, and a combination of Logistic regression and naive Bayes.

It proposed a CNN-gram deep learning architecture (Rizwan et al., 2020) for hate and offensive language detection in social media and compared its performance with the current baseline method, the model shows higher robustness. (Wanner et al., 2003) used the functional template described by Yarowsky to model hate speech as a classification problem to detect hate speech on the Internet. It trained the classifier through semi-supervised machine learning technology (Epstein and Mengibar, 2015), training the classifier to determine if there are potentially offensive terms in the text. (Pitsilis et al., 2018) proposed a detection scheme when distinguishing hateful content on social media. It is a combination of Recurrent Neural Network (RNN) classifiers. It contains various functions related to user-related information and achieves relatively high classification quality.

(Nugroho et al., 2019) used random forest methods to identify Twitter hate speech datasets and compared them with the accuracy results of neural networks and AdaBoost. (Hande et al., 2020) found that in sentiment classification, Logistic regression, random forest classifier, and decision tree performed relatively well, SVM performed poorly, and its heterogeneity was also poor. It used a HateBERT model retrained (Caselli et al., 2020) on RAL-E (offensive and hateful Reddit English data set) for the detection of abusive language and found that HateBERT is r tosuperio the corresponding conventional BERT model. (Paul and Saha, 2020) fine-tuned the BERT and modeled the BERT as a basic neural network. This model can enhance its detection performance and achieve good accuracy at a low computational cost.

(Xi et al., 2018) proposed a deep convolution model that uses unsupervised pre-trained word embedding to classify objectionable text. It used the pre-trained Arabic language model AraBERT in the task of offensive language detection, which (Djandji et al., 2020) showed good performance in classification tasks. (Kokatnoor and Krishnan, 2020) proposed a stack weighted ensemble (SWE) model with five independent classifiers to detect hate speech. (Gambäck and Sikdar, 2017) tried to use deep learning convolutional neural network models and creating CNN models to classify Twitter's hate speech text.

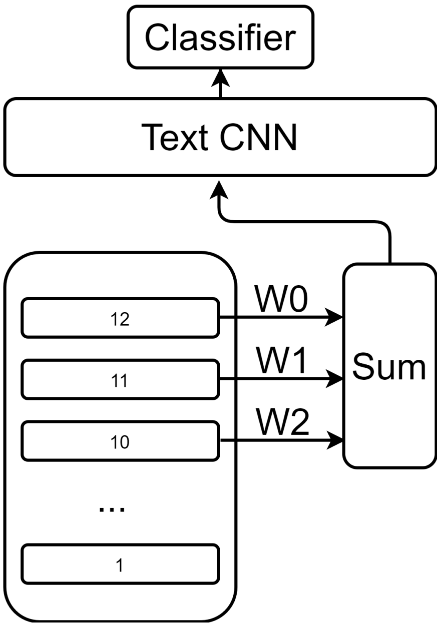

Figure 1: The overall architecture and data flow of our system.

(Mubarak et al., 2020) proposed a systematic method to construct a data set of tweets that do not support specific dialects and topics and use cross-validation to establish offensive language system detection. (Chakravarthi et al., 2020a, 2021) proposed a new gold standard corpus for sentiment analysis of annotated English-language mixed text. (Chakravarthi et al., 2020b) used logistic regression, naive Bayes, decision tree, etc. to code mixed data to classify emotions. Among them, logistic regression and random forest are used in this experiment to get the best results.

3 Data

The common feature of the data sets in three different languages is code-mixed. Code-mixed refers to words in multiple different languages that may appear in the same sentence. The data we used in this task is mainly text mixed with English ((Tamil-English, Malayalam-English, and Kannada-English)).

Labels distribution of Malayalam training set and validation set. In the training set, Not offensive: 88.4%, Not Malayalam: 8.04%, Offensive Targeted Insult Individual: 1.49%, Offensive Untargetede: 1.19%, Offensive Targeted Insult Group: 0.88%. In the validation set, Not offensive: 89%, Not Malayalam: 8.15%, Offensive Targeted Insult Individual: 1.2%, Offensive Untargetede: 1%, Offensive Targeted Insult Group: 0.65%.

Labels distribution of Kannada training set and validation set. In the training set, Not offensive: 57.01%, Not Kannada: 24.48%, Offensive Targeted Insult Individual: 7.83%, Offensive Targeted Insult Group: 5.29%, Offensive Untargetede: 3.41%, Offensive Targeted Insult Other: 1.98%. In the validation set, Not offensive: 54.83%, Not Kannada: 24.58%, Offensive Targeted Insult Individual: 8.49%, Offensive Targeted Insult Group: 5.79%, Offensive Untargetede: 4.25%, Offensive Targeted Insult Other: 2.06%.

Labels distribution of Tamil training set and validation set.
In the training set, Not offensive: 72.23%, Offensive Untargetede: 8.27%, Offensive Targeted Insult Group: 7.28%, Offensive Targeted Insult Individual: 6.67%, Not-Tamil: 4.14%, Offensive Targeted Insult Other: 1.29%.
In the validation set, Not offensive: 72.77%, Offensive Untargetede: 8.11%, Offensive Targeted Insult Group: 6.72%, Offensive Targeted Insult Individual: 7%, Not Tamil: 3.92%, Offensive Targeted Insult Other: 1.48%.

The data sets we can use are the training set and validation set in three languages(Tamil, Malayalam, and Kannada) provided by the task organizer team. Code mixing is the main feature of the data set provided by the task organizer. There are five different categories in the Malayalam dataset. Each dataset of Kannada and Tamil has six different categories. There is an imbalance of category labels in the data sets of the three languages.

4 Methods

Because the deep learning model can learn the complex distribution characteristics of data through deep artificial neural networks and nonlinearity. Especially the use of deep learning in tasks related to text data has attracted more and more attention(Zhang et al., 2018).

4.1 Multilingual BERT

BERT(Bidirectional Encoder Representations from Transformers) is a pre-trained language model method. It uses a plain text corpus to train the artificial neural network in the model. The BERT model can finally solve the most common tasks in NLP (natural language processing) and can achieve state-of-the-art results on many tasks. For example, text sequence labeling, text classification tasks, sentence relationship judgment, text generation tasks.

Lang	Epoch	Batch	lr	Sent len
Kannada	5	32	3e-5	55
Malayalam	5	32	3e-5	60
Tamil	4	32	4e-5	70

Table 1: The parameter settings of our system on the training sets of three different languages: Kannada, Malayalam, and Tamil.

It is a language representation model based on Transformer architecture (Vaswani et al., 2017). In the training phase of the model, the model BERT needs to complete the MLM (Masked Language Model) task and the NSP (Next Sentence Prediction) task. The combination of BERT's model structure and its pre-training process makes BERT capable of most NLP tasks. The advantage of BERT is that it is a deep two-way, unsupervised NLP pre-training system. Compared with the previous pre-training model, it learns bidirectional context information in the true sense Its use includes two stages: pre-training and fine-tuning. Generally, we only need to fine-tune the BERT to get good results when completing NLP-related downstream tasks.

The difference between multilingual BERT and BERT that uses other single language pre-training is that it uses a corpus composed of more than 100 different languages in the pre-training phase. The languages supported by the multilingual model are mainly from the top 100 most used languages on Wikipedia. Of course, multi-language models can also complete some single-language tasks, but there may be some gaps in the scores of the BERT pre-trained in a single language. In the model architecture, the multilingual BERT has 12 coding layers and uses a multi-head attention mechanism (a total of 12 heads). These parameters are the same as BERT-base. Therefore, in addition to the advantages of BERT mentioned above, the advantage of multilingual BERT is that it has good cross-language.

4.2 TextCNN

The TextCNN artificial neural network was first proposed by Kim et al. in 2014 (Kim, 2014). Compared with the CNN network in the image, the biggest difference of TextCNN is the difference in the input data. It is a text classification model that applies the CNN network. The biggest advantage of TextCNN is that the network structure is simple, which in turn leads to a small number of parameters, a small amount of calculation, and a fast training speed.

The general process of data passing through TextCNN is: first vectorize text data through word embedding, then perform convolution operation on the text converted into vectors, then vector data through the maximum pooling layer, and finally connect the output result to the linear classifier, the layer uses softmax for n classification. Because the convolutional layer and the maximum pooling layer do not activate the vector data, the activation function is usually used after the convolutional layer, such as the Relu function or the Tanh function. Besides, some regularization items are also used, commonly used are dropout, L2, etc.

4.3 Our System

TextCNN can use different sizes of convolution kernels to obtain local features of different window sizes in the text vectorized space. The BERT model can obtain true two-way contextual semantic information. We want to obtain both contextual semantic information and local features, so we choose to Combine these two models.

To obtain the top-level semantic information of BERT, we take out the last three layers of BERT output (layer12_output, layer11_output, layer10_output). Then apply weighting effects (W0, W1, W2) on these three output results. Three Different weight values are respectively weighted and sum layer12_output, layer11_output, layer10_output to get weighted_sum_output. Then, input the result of weighted_sum_output into the textCNN network to get a new result. Finally, Input this result into the linear classifier, and the result after linear classification is the final output result of our system.

5 Experiment and Results

5.1 Experimental details

We use the training set and validation set provided by the task organizer as the input data of our system. As we described in the method introduction section, we take the last three layers of the output of the multilingual BERT model as the input of TextCNN. In TextCNN, the size of the convolution kernel we choose is 2, 3, 4, each of different sizes The number of convolution kernels is 256. Both convolution and pooling are 1-dimensional. The activation function is ReLu. The regularization item is dropout, and the parameter is set to 0.3. Choose different hyperparameters for different language data. The loss function chooses the CrossEntropyLoss function. The parameter setting information can be obtained in Table 1.

Team/Lang	**Precision**	**Recall**	**F1**
Top1 Malayalam	0.97	0.97	0.97
Our Malayalam	0.92	0.94	0.93
Top1 Tamil	0.78	0.78	0.78
Our Tamil	0.74	0.75	0.74
Top1 Kannada	0.73	0.78	0.75
Our Kannada	0.64	0.67	0.64

Table 2: Our model and the Top1 team on each language data set score on the test set.

5.2 Result analysis

The weighted average F1-score is a reference indicator used by task organizers to rank. The scores of the participating teams in Malayalam are generally high. In comparison, the scores of the other two languages are relatively low. Our results are somewhat different from the first place results, especially the scores on the Kannada test set Overall, our system has a certain effect on the text classification task of recognizing code mixture, but there is still a lot of room for improvement.

6 Conclusion

For this task, we use multilingual BERT and TextCNN to complete the detection of offensive speech. The above is our description of this task. The result of this task may not be ideal. Therefore, in future work, we try to use new models and improve methods to obtain better results. This task has given us a better understanding of the detection of speech on social media. It not only has an understanding of the importance of the detection of offensive speech but also improved our ability to solve such problems. In the future, we will continue to learn about social media speech detection.

References

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2020. HateBERT: Retraining BERT for Abusive Language Detection in English. *arXiv preprint arXiv:2010.12472*.

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, 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.

Marc Djandji, Fady Baly, Hazem Hajj, et al. 2020. Multi-task learning using arabert for offensive language detection. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 97–101.

Mark Edward Epstein and Pedro J Moreno Mengibar. 2015. Classification of offensive words. US Patent App. 14/264,617.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using convolutional neural networks to classify hate-speech. In *Proceedings of the first workshop on abusive language online*, pages 85–90.

Adeep Hande, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. 2020. 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*, pages 54–63, Barcelona, Spain (Online). Association for Computational Linguistics.

Navya Jose, Bharathi Raja Chakravarthi, Shardul Suryawanshi, Elizabeth Sherly, and John P. McCrae. 2020. A Survey of Current Datasets for Code-Switching Research. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 136–141.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

S. A. Kokatnoor and B. Krishnan. 2020. Twitter hate speech detection using stacked weighted ensemble (swe) model. In *2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pages 87–92.

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.

Hamdy Mubarak, Ammar Rashed, Kareem Darwish, Younes Samih, and Ahmed Abdelali. 2020. Arabic offensive language on twitter: Analysis and experiments. *arXiv preprint arXiv:2004.02192*.

Kristiawan Nugroho, Edy Noersasongko, Ahmad Zainul Fanani, Ruri Suko Basuki, et al. 2019. Improving random forest method to detect hate-speech and offensive word. In *2019 International Conference on Information and Communications Technology (ICOIACT)*, pages 514–518. IEEE.

Sayanta Paul and Sriparna Saha. 2020. Cyberbert: Bert for cyberbullying identification. *Multimedia Systems*, pages 1–8.

Georgios K Pitsilis, Heri Ramampiaro, and Helge Langseth. 2018. Detecting offensive language in tweets using deep learning. *arXiv preprint arXiv:1801.04433*.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, Mani Vegupatti, and John P. McCrae. 2020. Named Entity Recognition for Code-Mixed Indian Corpus using Meta Embedding. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 68–72.

Hammad Rizwan, Muhammad Haroon Shakeel, and Asim Karim. 2020. Hate-speech and offensive language detection in Roman Urdu. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2512–2522.

Elena Shushkevich, John Cardiff, Paolo Rosso, and Liliya Akhtyamova. 2020. Offensive language recognition in social media. *Computación y Sistemas*, 24(2).

H. Sohn and H. Lee. 2019. Mc-bert4hate: Hate speech detection using multi-channel bert for different languages and translations. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 551–559.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Leo Wanner, Bernd Bohnet, and Mark Giereth. 2003. Deriving the communicative structure in applied NLG. In *Proceedings of the 9th European Workshop on Natural Language Generation (ENLG-2003) at EACL 2003*.

Jian Xi, Michael Spranger, and Dirk Labudde. 2018. Cnn-based offensive language detection. In *14th Conference on Natural Language Processing KONVENS 2018*, page 125.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253.