ZYJ123@DravidianLangTech-EACL2021: Offensive Language Identification based on XLM-RoBERTa with DPCNN

Yingjia Zhao
Yunnan University/Yunnan, P.R. China
zyj1309700118@gmail.com

Xin Tao
Yunnan University / Yunnan, P.R. China
taoxinwy@126.com

Abstract

The development of online media platforms has given users more opportunities to post and comment freely, but the negative impact of offensive language has become increasingly apparent. It is very necessary for the automatic identification system of offensive language. This paper describes our work on the task of Offensive Language Identification in Dravidian language-EACL 2021. To complete this task, we propose a system based on the multilingual model XLM-Roberta and DPCNN. The test results on the official test data set confirm the effectiveness of our system. The weighted average F1-score of Kannada, Malayalam, and Tamil language are 0.69, 0.92, and 0.76 respectively, ranked 6th, 6th, and 3rd.

1 Introduction

With the development of the information society, people have become accustomed to uploading content on social media platforms in the form of text, pictures, or videos. At the same time, they also comment on the content uploaded by other users and interact with each other, thus increasing the activity of social media platforms (Thavareesan and Mahesan, 2019, 2020a,b). Inevitably, however, some users will post offensive posts or comments. The use of offensive discourse is a kind of impolite phenomenon which has negative effects on the civilization of the network community (Chakravarthi, 2020). It usually has the characteristics of causing conflicts and the purpose of publishing intentionally. The publisher of offensive language may use reproach, sarcasm, swear and other language means to achieve intentional offense, and express a variety of intentions, such as disturbing, provoking, and expressing negative emotions (Chakravarthi and Muralidaran, 2021; Suryawanshi and Chakravarthi, 2021). Most people will take measures to respond to offensive words. The way to respond to the direct conflict of offensive words is mainly rhetorical questions, swear, sarcasm and threat, so as to express dissatisfaction, deny and satirize the other party and provoke the other party. This will further cause conflicts and destroy the harmony of the network environment.

Many social media platforms use a content review process, in which human reviewers check users' comments for offensive language and other infractions, and which comments have been removed from the platform because of the violation (Mandl et al., 2020). It is up to the moderator to decide which comments will be removed from the platform due to violations and which ones will be kept. As the number of network users increases and user activity increases, the manual approach is undoubtedly inefficient. Therefore, the automatic detection and identification of offensive content are very necessary. However, offensive words often depend on the emotions and psychology of the listener, and some seemingly innocuous words can be potentially offensive, and words that often seem offensive are watered down by the emotions of the listener. This kind of language phenomenon is not uncommon in real life, either unintentionally or deliberately used to achieve the speaker's expected purpose, which is a challenging work for the current detection system.

Our team takes part in the shared task of Offensive Language Identification in Dravidian Languages-EACL 2021 (Chakravarthi et al., 2021, 2020a,b; Hande et al., 2020). This is a classification task at the comment/post level. 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. Tamil language is the oldest language in Indian languages, Malayalam and Kannada evolved

Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pages 216–221

from Tamil language. For a comment on Youtube, the system must classify it into not-offensive, offensive-untargeted, offensive-targeted-individual, offensive-targeted-group, offensive-targeted-other, or not-in-indented-language.

In our approach, the multilingual model XLM-RoBERTa and DPCNN are combined to carry out the classification task. This method can combine the advantages of the two models to achieve a better classification effect. The rest of the paper is divided into the following parts. In the second part, we introduce the relevant work in this field, which involves offensive language detection and text classification methods. In the third part, we introduce the model structure and the composition of our training data. The fourth part introduces our experimental setup and results. The fifth part is the conclusion.

2 Related Work

Due to the harm of offensive language to the network environment, the identification of offensive language has been carried out for a long time. Research so far has focused on automating the decision-making process in the form of supervised machine learning for classification tasks (Sun et al., 2019). As far back as 2012, Chen et al. (2012) proposed a lexical syntactic feature (LSF) framework to detect offensive content in social media, distinguished the roles of derogatory/profane and obscenity in identifying offensive content, and introduced handwritten syntax rules to identify abusive harassment. In contrast to the start-to-end training model, Howard and Ruder (2018)proposed an effective transfer learning method, Universal Language Model Tuning (ULMFIT), which can be applied to any task in natural language processing, and has shown significant results on six text classification tasks. Subsequently, Abdellatif and Elgammal (2020) used the ULMFiT transfer learning method to train forward and backward models on Arabic datasets and ensemble the results to perform an offensive language detection task.

Although English is currently one of the most commonly spoken languages in the world, work is ongoing to identify the offensive language in other languages that are less widely spoken. Pitenis et al. (2020) tested the performance of several traditional machine learning models and deep learning models on an offensive language dataset of Greek, and the best results were achieved with the attention model of LSTM and GRU. Ozdemir and Yeniterzi (2020) ensembled CNN-LSTM, BILSTM-Attention, and BERT three models, combined with pre-trained word embedding on Twitter to complete the identification task of offensive Turkish language, and achieved a good result.

A key challenge in automatically detecting hate speech on social media is to separate hate speech from other offensive languages. Davidson et al. (2017) used the crowd-sourced hate speech lexicon to collect tweets containing hate speech keywords. They trained a multi-class classifier to reliably distinguish hate speech from other offensive languages, and found that racist and homophobic tweets were more likely to be classified as hate speech, but sexist tweets were generally classified as offensive. Razavi et al. (2010) proposed to extract features at different conceptual levels and apply multilevel classification for offensive language detection. The system leverages a variety of statistical models and rule-based patterns, combined with an auxiliary weighted pattern library, to improve accuracy by matching text with its graded entries. Pitsilis et al. (2018) proposed the ensemble of a recursive neural network (RNN) classifier, which combines various characteristics related to user-related information, such as the user's sexist or racist tendencies, and was then fed to the classifier as input along with a word frequency vector derived from the text content.

When there is a large amount of labeled data, increasing the size and parameters of the model will definitely improve the performance of the model.However, when the amount of training is relatively small, the large-scale model may not be able to achieve good results, so solving the problem of model training under the condition of a small amount of target data has become a research hotspot. Sun et al. (2019) proposed a Hierarchical Attention Prototype Network (HAPN) for few-shot text classification, which designed multiple cross-concerns of a feature layer, word layer, and instance layer for the model to enhance the expressive power of semantic space. The model was validated on two standard reference text classification datasets, Fewrel and CSID. Prettenhofer and Stein (2010) built on structural correspondence learning, using untagged documents and simple word translation to induce task-specific, cross-language word correspondence. English was used as the source language and German, French, and Japanese

were used as the target language to conduct the experiment in the field of cross-language sentiment classification. Using English data, Ranasinghe and Zampieri (2020) trained the model by applying cross-language contextual word embedding and transfer learning methods, and then predicted the effect of cross-language contextual embedding and transfer learning on this task in less resource-intensive languages such as Bengali, Hindi, and Spanish.

3 Data and Methodology

3.1 Data description

We count the number of each type of tag in the training set and the validation set, and obtain the data distribution of Not-offensive, offensive-untargeted, offensive-targeted-individual, offensive-targeted-group, offensive-targeted-other, and Not-in-indented-language in Tamil, Malayalam, and Kannada. as shown in Table 1.

3.2 Why XLM-RoBERTa

Compared with the original BERT model, XLM-RoBERTa increases the number of languages and the number of training data sets. Specifically, a preprocessed CommonCrawl dataset of more than 2TB based on 100 languages is used to train cross-language representations in a self-supervised manner. This includes generating new unlabeled corpora for low-resource languages and expanding the amount of training data available for these languages by two orders of magnitude. In the fine-tuning period, the multi-language tagging data is used based on the ability of the multi-language model to improve the performance of the downstream tasks. This enables XLM-RoBERTa to achieve state-of-the-art results in cross-language benchmarks while exceeding the performance of the single-language BERT model for each language. Tune the parameters of the model to address cases where extending the model to more languages using cross-language migration limits the ability of the model to understand each language. The XLM-RoBERTa parameter changes include up-sampling of low-resource languages during training and vocabulary building, generating a larger shared vocabulary, and increasing the overall model to 550 million parameters.

3.3 XLM-RoBERTa with DPCNN

In this task, we combined XLM-RoBERTa with DPCNN (Johnson and Zhang, 2017) to make the whole model more suitable for the downstream classification task. DPCNN(Deep Pyramid Convolutional Neural Networks) is a kind of deep word level CNN structure, the calculation amount of each layer of the structure decreases exponentially. DPCNN simply stacks the convolution module and negative sampling layer. The computation volume of the whole model is limited to less than two times the number of convolution blocks. At the same time, the pyramid structure also enables the model to discover long-term dependencies in the text. In a common classification task, the last hidden state of the first token of the sequence (CLS token), namely the original output of XLM-Roberta (Pooler output), is further processed through the linear layer and the tanh activation function for classification purposes. To obtain richer semantic information features of the model and improve the performance of the model, we first processed the output of the last three layers of XLM-RoBERTa through DPCNN, and then concatenate it with the original output of XLM-RoBERTa (Pooler output) to get a new and more effective feature vector, and then input this feature vector into the classifier for classification. As shown in Figure 1.

4 Experiment and results

4.1 Experiment setting

In this experiment, the pre-training model I used was XLM-RoBERTa-base. After adding the DPCNN module, we began to set the experimental parameters. We set the learning rate as 2e-5, the maximum sequence length is 256, and the gradient steps are set to 4. The batch size is set to 32, as shown in table 2. In the training process, we used five-fold stratified cross-validation to make the proportion of data of each category in each subsample the same as that in the original data and finally obtained the optimal result through the voting (Onan et al., 2016) system, as shown in Figure 2.

4.2 Results

After the evaluation by the organizer, we obtained the weighted average F1-score in the three languages, as shown in table 3. Our team's F1-score is 0.69, ranked 6th place for the Kannada language. For the Malayalam language, our team's F1-score

	Kannada		Malayalam		Tamil	
Label	Train	Validation	Train	Validation	Train	Validation
Not-offensive	3544	426	14153	1779	25405	3193
Not-in-indented-language	1522	191	6205	163	1454	172
offensive-targeted-individual	487	66	239	24	2343	307
offensive-targeted-group	329	45	140	13	2557	295
offensive-untargeted	212	33	191	20	2906	356
offensive-targeted-other	123	16	0	0	454	65

Table 1: Train and Validation datasets description.

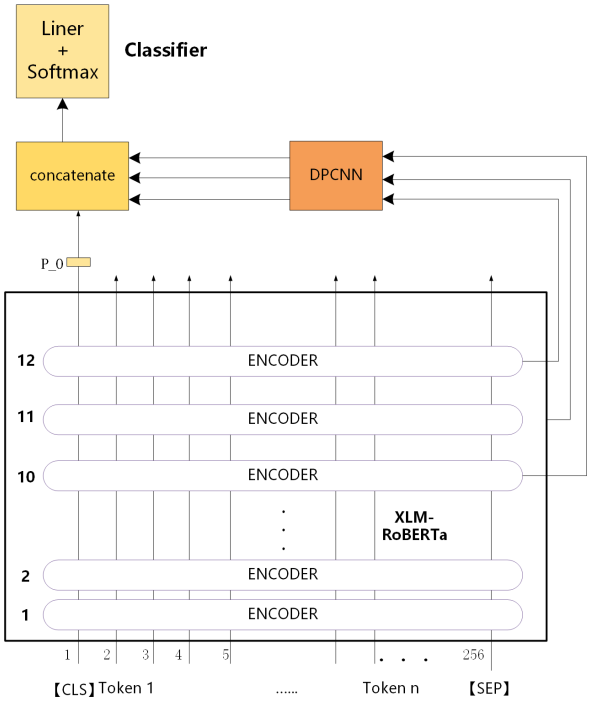

Figure 1: Schematic overview of the architecture of our model

maximum sequence length	learning rate
256	2e-5
gradient steps	**batch size**
4	32

Table 2: Details of the parameters

	Kan	Mal	Tam
Best F1-score	0.75	0.97	0.78
Our Precision	0.65	0.91	0.75
Our Recall	0.74	0.94	0.77
Our F1-score	**0.69**	**0.92**	**0.76**
Rank	6	6	3

Table 3: the results of our methods.

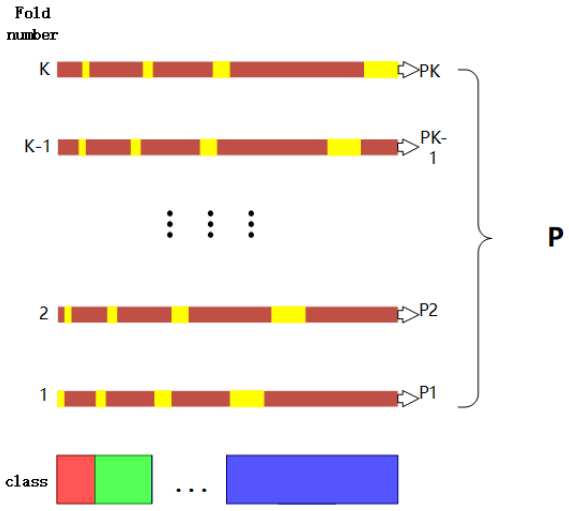

Figure 2: Voting system

is 0.92 ranked 6th place, and for the Tamil language, our team's F1-score is 0.76 ranked 3rd place.

5 Conclusion

In this paper, we describe our system in the task of offensive language identification for Tamil, Malayalam, and Kannada language. In this model, the XLM-RoBERTa pre-training model is used to extract semantic information features of the text, and DPCNN is used to further process the output features. At the same time, the hierarchical cross-validation method is used to improve the training effect. The final results show that our model achieves satisfactory performance. In future work, we will try to adjust the structure of the new model, so as to improve its effect more significantly.

References

Mohamed Abdellatif and Ahmed Elgammal. 2020. Offensive language detection in arabic using ulmfit. In

Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, pages 82–85.

Bharathi Raja Chakravarthi. 2020. HopeEDI: A multilingual hope speech detection dataset for equality, diversity, and inclusion. In *Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media*, pages 41–53, Barcelona, Spain (Online). Association for Computational Linguistics.

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.

Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. 2012. Detecting offensive language in social media to protect adolescent online safety. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pages 71–80. IEEE.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11.

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.

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

Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 562–570.

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.

Aytuğ Onan, Serdar Korukoğlu, and Hasan Bulut. 2016. A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. *Expert Systems with Applications*, 62:1–16.

Anil Ozdemir and Reyyan Yeniterzi. 2020. Su-nlp at semeval-2020 task 12: Offensive language identification in turkish tweets. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2171–2176.

Zeses Pitenis, Marcos Zampieri, and Tharindu Ranasinghe. 2020. Offensive language identification in greek. *arXiv preprint arXiv:2003.07459*.

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

Peter Prettenhofer and Benno Stein. 2010. Cross-language text classification using structural correspondence learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 1118–1127.

Tharindu Ranasinghe and Marcos Zampieri. 2020. Multilingual offensive language identification with cross-lingual embeddings. *arXiv preprint arXiv:2010.05324*.

Amir H Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. 2010. Offensive language detection using multi-level classification. In *Canadian Conference on Artificial Intelligence*, pages 16–27. Springer.

Shengli Sun, Qingfeng Sun, Kevin Zhou, and Tengchao Lv. 2019. Hierarchical attention prototypical networks for few-shot text classification. In *Proceedings of the 2019 Conference on Empirical Methods*

in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 476–485.

Shardul Suryawanshi and Bharathi Raja Chakravarthi. 2021. Findings of the shared task on Troll Meme Classification in Tamil. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

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.