Non-contextual Binary Classification for Mexican Spanish with XLM and CNN

Suidong Qu[0000−0002−7274−5891], Qinyu Que[0000−0001−6688−7896], and Shuangjun Jia[0000−0001−8315−5662]

Yunnan University,Yunnan,P.R.China
icat@mail.ynu.edu.cn

Abstract. This article first introduces the development of deep learning in natural language processing in recent years. Then the related work is briefly explained. We participat in the classification tasks of MeOffendEs@IberLEF 2021 Subtask 3: Non-contextual binary classification of Mexican Spanish. In this task, our work involves categorizing tweets as offensive or non-offensive tweets in the OffendMEX corpus. The method in this article first uses XLM to obtain semantic feature information, and then extracts the features again through convolutional neural networks. Focal Loss is used in the model to improve the classification effect. In this task, our team name is Dong. The precision of our model is 0.6050, the recall is 0.5361, and the F1 score is 0.5685.

Keywords: Binary Classification · Offensive Language Detection · XLM.

1 Introduction

In recent years, with the rapid development of Internet-based media, a large number of websites with social services as the main content have emerged, such as Facebook, Twitter and Weibo. Their birth makes information dissemination no longer centered on media, but centered on users. The cost for people to share and obtain information has become very low, but at the same time offensive language is also flooding people's vision. Offensive language is text content that can irritate individuals or groups, including hate language, personal attacks, harassment, ridicule, etc. In recent years, research related to speech abuse has received more and more attention. Preventing the abuse of offensive language is of great significance to maintaining social harmony. The basis for effectively avoiding speech abuse is to quickly, automatically and accurately identify offensive language. In this binary classification task of MeOffendEs@IberLEF 2021 [1, 2]: Subtask 3 , we need to classify the data set of short texts based on Twitter as offensive or non-offensive tweets in the OffendMEX corpus.

Section 2 of this article briefly introduces the progress and current situation of work related to natural language processing. Section 3 describes the data

IberLEF 2021, September 2021, Málaga, Spain.

set used in this mission. Section 4 illustrates our method, which describes the preprocessing steps and main structure of the model. Section 5 describes the adjusted hyper-parameters applicable to this model and our experimental results. We summarized our work in Section 6.

2 Related Work

With the application and development of deep learning in many industrial and commercial fields and the rapid improvement of computer computing performance, deep learning models are becoming more and more easily applicable to various scenarios, including the task of detecting offensive short texts. Deep neural networks have a more complex network structure than traditional machine learning methods, and can dig out high-dimensional feature information from samples by self-learning the characteristics of samples. In 2017, Thomas Davidson et al. [3] conducted a research on automatic hate speech recognition based on Twitter text. Experimental results showed that the effects of logistic regression and linear SVM (Support Vector Machines) are significantly better than classifiers based on naive Bayes and decision trees. In 2017, Shervin Malmasi et al. [4] used linear SVM as a classifier to detect hate speech in Twitter text. They used character-based and vocabulary-based N-Grams features with different N values. In 2016, Zhang et al. [5] used a two-way gated neural network to extract grammatical and semantic information in Twitter text, and used a pooling operation to extract contextual features in historical data. In 2017, Pinkesh Badjatiya et al. [6] used CNN (Convolutional Neural Network) and LSTM (Long Short-Term Memory) methods to detect hate speech in Twitter text. In 2017, Ji Ho Park et al. [7] used a hybrid model combining character CNN and word CNN for the particularity of tweets. In 2019, Shiwei Zhang et al. [8] used a two-way long and short-term memory network combined with an attention mechanism to detect mocking comments in tweets. In 2019, Pushkar Mishra et al. [9] used CNN to detect the abuse of speech in combination with user personal information and user social information. The model can capture the characteristics of individual language behavior and group structure. In 2020, Baruah and Das et al. [10] used pre-training BERT [11] for feature extraction of context and target text. We participate in this task for Mexican Spanish and propose a method that includes XLM [12] and CNN models. It can integrate the advantages of two models to enhance the effectiveness of the binary classification.

3 Data and Resources

This task is Subtask 3 in MeOffendEs@IberLEF 2021. Our task is to classify tweets as offensive or non-offensive tweets in the OffendMEX corpus. The training data used in our task is provided to each participant by the task organizer. These data are collected from Mexican Spanish on Twitter and have been tagged.

4 System Description

4.1 Data Preprocessing

In order to make the model better understand the semantic relationship of sentences during the training phase, we remove punctuation marks, numbers, and emoticons during preprocessing. This approach does not affect the effect of the model in this classification task, and can reduce the difficulty of training the model. BERT inserts an identifier [CLS] and a separator [SEP] into each sentence sample during training. Among them, [CLS] is added at the beginning of the sentence, and [SEP] is added at the end of the sentence. [SEP] separates two sentences, so that the sample forms a structure of [CLS] + sentence + [SEP].

4.2 Model Description

Our model includes XLM and CNN models. Since BERT was proposed, it has proved that the pre-training model is very effective in processing natural language understanding tasks, and it has achieved good results in a variety of natural language processing tasks. We consider that BERT mainly focuses on the natural language processing tasks of a single language, so in this task we used Hugging Face's implementation of XLM model. XLM is a Cross-Lingual version developed by Facebook on the basis of BERT. XLM has the same structure as BERT, so it is also an encoder composed of multiple Encoder structures in Transformer [13].

When training XLM, we use XLM to encode any sentences into a shared embedding space. The input sequence will be masked. The process is to select 15% of the input sequence for random masking, 80% of which are replaced by [Mask], 10% are replaced by random words, and the remaining 10% are still correct words. The Fig.1 below shows the basic structure of the model.

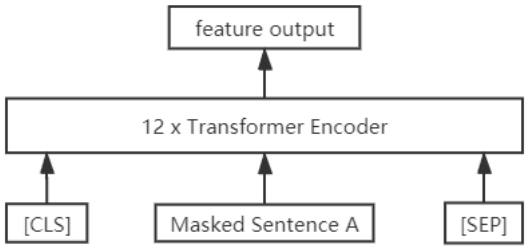

Fig. 1. The XLM model captures the feature information of the sentence.

The focus of this classification task is to extract the core semantic information contained in the sentence. We input the feature information extracted by XLM

into CNN. We initialize it before entering CNN. The CNN layer can effectively reduce the number of parameters, thereby reducing the difficulty of training, and at the same time extract low-level local features into higher-level features. In the convolutional neural network we designed, the Batch size is set to 8, the size of the convolution kernel is 2, 3, 4, and 5, and the dropout is set to 0.2. Dropout is added to reduce the possibility of overfitting. In order to accelerate convergence and alleviate the problem of gradient disappearance, we use the Batch Normalization layer to normalize each batch of data. We use the ReLU activation function inside the CNN, and use the sigmoid activation function for the final output.

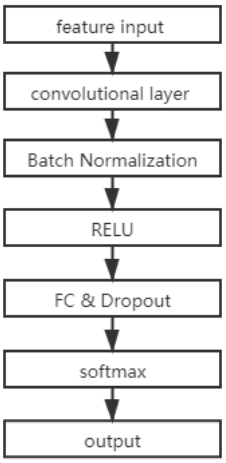

Fig. 2. The convolutional neural network

When some previous models used the traditional cross-entropy loss function, those models ignored the difference in the degree of optimization of the model between positive and negative samples. In this task, the proportion of positive samples and negative samples in the training data set is unbalanced. Therefore, this model adoptes the Focal Loss function[14] based on the cross-entropy loss function in fine-tuning. This method can reduce the weight occupied by a large number of simple negative samples in the process of training and optimization, and put the model's focus on sparse difficult samples. This method is added to the classification model of this article, and and are set to 2 and 0.75 respectively, which can improve the classification effect.

5 Hyper-parameters and Results

In this classification task, our model is based on PyTorch. After many experiments, the training parameter values are shown in Table 1. For this task, our model is composed of XLM and CNN, and the Adam [15] optimizer is used. In the internal and output of CNN, we use ReLU activation function and sigmoid activation function respectively.

Table 1. Hyper-parameters

hyper-parameters	Values
learning rate	1e-5
per gpu train batch size	64
Gradient accumulation steps	6
num train epochs	8
max seq length	100
dropout	0.2

In this task, the scores obtained by our model are shown in Table 2. The organizer's evaluation indicators for this task are precision, recall and F1 score. The precision of our model is 0.6050 (ranked 10th, 0.31 lower than the highest score). The recall is 0.5361 (ranked 9th, 0.16 lower than the highest score). The F1 score is 0.5685 (ranked 9th, 0.14 lower than the highest score).

Table 2. Evaluation results

	Our model	Model 1	Model 2	Model 3
Macro precision	0.6050(10)	0.7600(4)	0.6733(7)	0.9183(1)
Macro recall	0.5361(9)	0.6533(5)	0.6966(1)	0.3141(12)
Macro F1 score	0.5685(9)	0.7026(1)	0.6847(3)	0.4683(12)

From Table 2 we can see that although our macro precision is not high compared to model 3, our model has better stability. Comparing madol 1 and model 2, our macro recall and F1 score are relatively low. We plan to use Focal Loss to improve macro recall, but the effect is limited. From the perspective of the data set, we find that the amount of offensive and non-offensive text is very different. In addition, the semantics of some texts are ambiguous, which can cause interference. This situation makes our model more biased towards the side with more data when fine-tuning, and the model may have over-fitting during training. These situations ultimately lead to the model's poor performance in the test set.

6 Conclusion

This article mainly describes the overall idea and optimization scheme of non-contextual offensive detection and classification for Mexican Spanish. In the model implementation, we use the XLM pre-training model as a word vector model, and construct a convolutional neural network in downstream tasks to extract deep semantic information. The Focal Loss function is introduced into the model to improve the classification performance. Among these evaluation indicators, this method is not optimal compared with other methods, indicating that this model has certain disadvantages. Although the downstream task captures some local information through CNN, its extraction ability is limited. In the next stage of research, we can add sentence-level attention information to improve the detection results. In addition, we consider adjusting the depth of the CNN in the model and the number of epochs to improve the model's characterization ability.

References

1. Montes, M., Rosso, P., Gonzalo, J., Aragón, E., Agerri, R., Álvarez-Carmona, M.Á., Álvarez Mellado, E., Carrillo-de Albornoz, J., Chiruzzo, L., Freitas, L., Gómez Adorno, H., Gutiérrez, Y., Jiménez-Zafra, S.M., Lima, S., Plaza-de Arco, F.M., Taulé, M. (eds.): Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2021) (2021)
2. Plaza-del-Arco, F.M., Casavantes, M., Jair Escalante, H., Martin-Valdivia, M.T., Montejo-Ráez, A., Montes-y-Gómez, M., Jarquín-Vásquez, H., Villaseñor-Pineda, L.: Overview of the MeOffendEs task on offensive text detection at IberLEF 2021. Procesamiento del Lenguaje Natural **67**(0) (2021)
3. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated hate speech detection and the problem of offensive language (2017)
4. Malmasi, S., Zampieri, M.: Detecting hate speech in social media (2017)
5. Zhang, M., Yue, Z., Fu, G.: Tweet sarcasm detection using deep neural network
6. Badjatiya, P., Gupta, S., Gupta, M., Varma, V.: Deep learning for hate speech detection in tweets. In: Proceedings of the 26th International Conference on World Wide Web Companion. pp. 759–760 (2017)
7. Ji, H.P., Fung, P.: One-step and two-step classification for abusive language detection on twitter. In: Proceedings of the First Workshop on Abusive Language Online (2017)
8. Zhang, S., Zhang, X., Chan, J., Rosso, P.: Irony detection via sentiment-based transfer learning. Information Processing Management **56**(5), 1633–1644 (2019)
9. Mishra, P., Tredici, M., Yannakoudakis, H., Shutova, E.: Abusive language detection with graph convolutional networks (2019)
10. Baruah, A., Das, K., Barbhuiya, F., Dey, K.: Context-aware sarcasm detection using bert. In: Proceedings of the Second Workshop on Figurative Language Processing (2020)
11. Bert: Pre-training of deep bidirectional transformers for language understanding (2018)
12. Conneau, A., Lample, G.: Cross-lingual language model pretraining. In: Advances in Neural Information Processing Systems. vol. 32, pp. 7057–7067 (2019)

13. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. vol. 30, pp. 5998–6008 (2017)
14. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(2), 318–327 (2020)
15. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. Computer Science (2014)
16. Miao, Y., Ji, Y., Peng, E.: Application of cnn-bigru model in chinese short text sentiment analysis. In: Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence (2019)