YNU_qyc at MeOffendEs@IberLEF 2021:The XLM-RoBERTa and LSTM for Identifying Offensive Tweets

Yuanchi Qu[0000−0002−0971−1795], Yanhua Yang[0000−0003−1508−4318], and Gang Wang[0000−0003−3721−0952]

School of Information Science and Engineering
Yunnan University, Yunnan, P.R. China
qychi@foxmail.com

Abstract. Our team (qu) participate in the classification tasks of MeOffendEs@IberLEF 2021. The task aims to distinguish offensive Spanish in online communication. We only participate in subtask 3 which aims to identify the offensive tweets (non-contextual Mexican Spanish). To solve the problem, we mainly put forward the model (XLM-R combined with LSTM). In our model, the F1 score is 0.6676 and the precision is 0.7433.

Keywords: offensive · XLM-RoBERTa · LSTM · K-folding ensemble.

1 Introduction

With the popularization of the Internet, online communication becomes a important part of life. While social networking brings great convenience to human beings, it also has huge risks and hidden dangers [1]. One risk arises from online comments (offensive), which may hurt netizens, or even cause long-term harm to the victims, leading them to depression and suicide [2]. Therefore, offensive comments can be detected and analyzed which have a positive effect on every Internet user. Many social media and technology companies are studying the identification of offensive speech in the network, and the most important thing is how to improve the effectiveness of identification [3].

The main task of MeOffendEs@IberLEF 2021 [4] is to analyze and detect offensive Spanish on social networks, hoping to find a better solution to identify offensive Spanish and its categories, and explore the impact of metadata on the identification task [5]. The mission is divided into four subtasks and two different Spanish languages (universal Spanish and Mexican Spanish). The goal of subtask 1 is to classify generic Spanish comments (non-contextual) into multiple categories (OFP: Offensive, target is a person. OFG: Offensive, target is a group of people or collective. NOM: Non-Offensive, but with inadequate language. NO:

IberLEF 2021, September 2021, Málaga, Spain.

non-offensive). The goal of subtask 2 is also to do multiple classifications of comments, but the difference is that subtask 2 provide metadata. The goal of subtask 3 is to classify Mexican Spanish tweets (non-context) into offensive and non-offensive tweets. Subtask 4 is also a dichotomy for Mexican Spanish tweets, but the difference between subtask 4 and subtask 3 is that subtask 4 will provide metadata. Metadata associated with tweets include date, retweet count, bear count, reply status, quote status and so on [6].

In this competition, we only take part in subtask 3 (Non-contextual binary classification for Mexican Spanish). For subtask 3, we mainly use the XLM-R model combined with LSTM (Long-Short Term Memory). The XLM-R model is a combination of the XLM model and RoBERTa model. Relatively speaking, the new model increases the number of model's languages and expands the number of train set, which greatly improve the performance of the downstream tasks. At the same time, the LSTM network [7] can further improve the model accuracy. Finally, we use the k-fold ensemble method to get higher scores.

The structure of the rest in this paper is as follows: The second part is the related research work on the recognition and classification of offensive language. The third part is the relevant description of the data set and the construction of our model. The fourth part is the summary and analysis of the experimental results.

2 Related Work

The identification and classification of offensive language in social media have always existed. As early as 2009, Yin [8] identified and classified the harassment behaviors that appeared in Web2.0. The author's idea was to extract the features of N-Grams, TF-IDF, semantic and other features from the text at first, and classified the text by SVM (Support Vector Machine). This method achieved excellent results at that time. For now, it has to be admitted that traditional machine learning methods still play a key role in the recognition and classification of offensive language. However, the traditional machine learning method also has a major drawback, which relies excessively on features and model parameters selection. With the development of Deep Learning, more and more neural network models make outstanding achievements in the classification of offensive language. At first, after Kim [9] applied CNN (Convolutional Neural Networks) to text classification, many text recognition models related to CNN are produced in the NLP (Natural Language Processing

dict.cnki.net). In the field of text detection and recognition, we have to talk about the LSTM model. RNN (Recurrent Neural Network) [10] plays an important role in the field of NLP, but it has the shortcoming of long-term dependence, and LSTM is designed for RNN's shortcomings [11]. Chakrabarty's, Zhang's and Srivastava's experiments prove the significant contribution of the LSTM model in text classification [12].

In 2018, Google introduced a new NLP model: the BERT model [13]. The BERT model is excellent in many fields of NLP. Subsequently, the BERT model

is also used in the recognition and classification of offensive language in social networking. A large number of facts prove that the effect of BERT model is indeed better than the general model. But in the cross-lingual realm, BERT still has many disadvantages.

Although the BERT model can be trained in multiple languages, it also has this irreparable disadvantage: different languages cannot learn from each other and communicate with each other. The XLM offers its solution for this shortcoming. By training different kinds of languages under the same model, the model can absorb the information from different languages. [14].

RoBERTa's improvement over traditional BERT is mainly reflected in the following three aspects: (1) BERT uses static masking. RoBERTa adopts the dynamic masking. (2) RoBERTa cancels the NSP task relative to BERT. (3) Compared with BERT, RoBERTa increase the batch size [15].

3 Methodology and Data

3.1 Data description

In subtask 3, we use the data set provided by MeOffendEs@IberLEF 2021 that is collected from Mexican-Spanish data on Twitter, and the data are manually tagged (offensive and non-offensive). There are training data (5060 pieces), validation data (76 pieces) and test data (2183 pieces). The ratio of offensive data items and non-offensive data items is about 5:2, and the data distribution is uneven.

3.2 XLM-R and LSTM Model

Although Spanish is the fourth most spoken language in the world, there are relatively few resources about it. So we need to use cross-lingual model for this task. As shown in Figure1 below, the XLM-R model+LSTM model is adopted in this paper. The experimental steps of this model are as follows:

(1) Encode.We add [CLS] and [SEP] for subsequent classification tasks and separation between sentences. (Language embeddings+Position embeddings+Token embeddings)

(2) After the text data is encoded and input to the 12 hidden layers and the hidden state sequence output by the last layer is obtained (last hidden state).

(3) Input the hidden state sequence to LSTM and GRU and we can get the output (H_N, H_AVERAGE, H_MAX).

(4) We input P_O (Pooler output), H_n, H_average, and H_max into the classifier. The H_n is the second output returned by GRU: the hidden state of the last time step. H_AVERAGE is the average-pooling of GRU output. H_MAX is the max-pooling of GRU output.

At the same time, we also use the comparison model, only using the XLM-R model, and not combined with LSTM. The experimental steps of this model are as follows: Firstly, we get pooler output (P_O) from input data through hidden

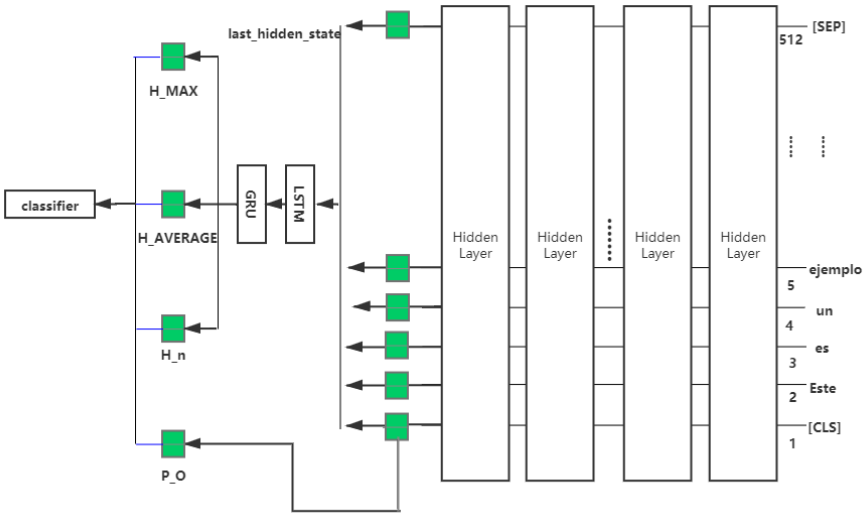

Fig. 1. The model: XLM-R combine with LSTM

layer calculation. P_O is a two-dimensional vector. Secondly, we transform P_O into a three-dimensional vector and input it into Bi-GRU to get the result of a three-dimensional vector. Finally, we transform the result of three-dimensional vector into two-dimensional vector and input it into the classifier and we can get the labels of tweets. The results of the two comparative experiments (the parameters of the two models are consistent) in the development set are shown in the following table 1. As shown in the following table, the model combined with LSTM has more excellent performance than the other.

Table 1. The evaluation results of the two models

Model	K-folding Ensemble	Precision	F1 score
XLM-R	No	0.6142	0.6023
	Yes	0.6823	0.6536
XLM-R + LSTM	No	0.7058	0.6632
	Yes	0.7536	0.7256

3.3 K-folding Ensemble

In this article, the training data set only have 5,060 pieces. And the amount of data is relatively small, so we use the K-folding ensemble method to improve the

performance of the model. The idea of the K-folding ensemble method comes from k-folding cross-validation. We divide the data set into K in total, use K-1 for training, and use the remaining one as a validation set. Repeat K times in total to ensure that each piece of data can become a validation set. Finally, we will accumulate these K results to find the average value as the final result. The K-folding ensemble method is shown in Figure 2.

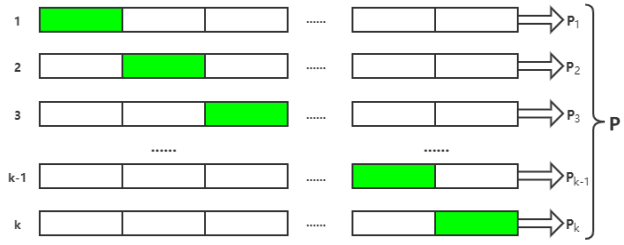

Fig. 2. The k-fold ensemble method

The purpose of using the K-fold ensemble method is to extract different features of all data as much as possible in the case of existing training data, which can effectively improve the generalization ability of the model. We use validation set to test the effect of k-fold ensemble method. In the Table 1, compared with the method without k-fold ensemble method, the test result in F1 score increased by 0.0624.

4 Experiment and Results

4.1 Data preprocessing

We observe the data set and find that the data downloaded from Twitter are not processed and looked very messy. In order to enable the model to obtain data features and get a good training model, we further process the data. The specific process is as follows: (1) All user names are replaced by "#username". Under normal circumstances, we usually think that usernames do not contain emotions, so to prevent the influence of usernames on model training, we replace all usernames. (2) All abbreviations are expanded. (3) There are emojis in unprocessed sentences. Computers usually cannot recognize these emojis. The emojis are replaced with the corresponding words by emotion lexicons [16] to express the corresponding emotion. (4) Change all uppercase to lowercase, which can unify the standard and facilitate model training. (5) Remove stop words and repeated words.

4.2 Experiment setting

The platform used in this experiment: Intel CORE i7 CPU (16G), hard disk 1T, NVIDIA RTX 3080Ti. The operating system is Windows 10. The editor is Pychrom2020. The framework of Deep Learning is PyTorch. The setting of hyper-parameters has a great influence on the performance of the model. The following Table 2 shows the settings of the experimental hyper-parameters:

Table 2. The value of hyper-parameters.

Parameter	Value
max_seq_length	80
learning_rate	2e-5
batch_size	64
dropout	0.4
epoch	10
Gradient_accumulation_steps	4

4.3 Result

The evaluation criteria of subtask 3 are precision, recall ,and F1 score. In this mission, the partial results and the baseline shown in Table 3 below.

Table 3. The Evaluation results of subtask 3 and the baseline.

Team	Precision	Recall	F1 score
vic_gomez	0.76	0.653295	0.702619
qu	0.743333	0.605978	0.667665
Baseline	0.719298	0.41	0.522292

In the Table 3, the results of team vic_gomez, team qu and the baseline of task 3 are included. The team vic_gomez win the first place in the subtask 3. And our team-name is qu. In the competition subtask 3, the ideal results were not achieved. We think there may be two main reasons for the poor results: (1) The data set is unbalanced. In the training data set, there are only 5060 items, and the ratio of offensive data to non-offensive data is 5:2. This model has poor adaptability to unbalanced training data. Because the linear classifier used in this model is biased to most classes, it causes the deviation of the model. (2) The parameters' design of the model are unreasonable. The results of the model in the train set are excellent, but the results in the test set are not the best. So the model may cause over-fit problem.

5 Conclusion

In this paper, we mainly propose a model (XLM-R and LSTM) for offensive detection and classification in Mexican Spanish (non-contextual). And the k-fold ensemble method is adopted to improve the generalization ability of the model. We think that there is still room for improvement in model performance. In the next research, we will further improve the optimization model (solve the problems caused by data imbalance by over-sampling and under-sampling), and improve the efficiency of recognition and classification.

References

1. Whittaker, E., Kowalski, R.M.: Cyberbullying via social media. Journal of school violence **14**(1), 11–29 (2015)
2. Pasi, G., Piwowarski, B., Azzopardi, L., Hanbury, A.: [lecture notes in computer science] advances in information retrieval volume 10772 —— deep learning for detecting cyberbullying across multiple social media platforms **10.1007/978-3-319-76941-7**(Chapter 11), 141–153 (2018)
3. 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)
4. 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)
5. 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)
6. Mehdad, Y., Tetreault, J.: Do characters abuse more than words? In: Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue. pp. 299–303 (2016)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
8. Yin, D., Xue, Z., Hong, L., Davison, B.D., Kontostathis, A., Edwards, L.: Detection of harassment on web 2.0. Proceedings of the Content Analysis in the WEB **2**, 1–7 (2009)
9. Kim, Y.: Convolutionalneuralnetworksforsentence classification
10. Gambäck, B., Sikdar, U.K.: Using convolutional neural networks to classify hate-speech. In: Proceedings of the first workshop on abusive language online. pp. 85–90 (2017)
11. Zhang, Y., Xu, B., Zhao, T.: Cn-hit-mi. t at semeval-2019 task 6: Offensive language identification based on bilstm with double attention. In: Proceedings of the 13th International Workshop on Semantic Evaluation. pp. 564–570 (2019)
12. Srivastava, S., Khurana, P.: Detecting aggression and toxicity using a multi dimension capsule network. In: Proceedings of the Third Workshop on Abusive Language Online. pp. 157–162 (2019)

13. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
14. Lample, G., Conneau, A.: Cross-lingual language model pretraining. arXiv preprint arXiv:1901.07291 (2019)
15. Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R.: Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). arXiv preprint arXiv:1903.08983 (2019)
16. Majumder, P., Patel, D., Modha, S., Mandl, T.: Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In: Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation (2019)