Lee at SemEval-2020 Task 12: A BERT model based on the maximum self-ensemble strategy for identifying offensive language

Junyi Li, Xiaobing Zhou*, Zichen Zhang
School of Information Science and Engineering
Yunnan University, Kunming 65091, P.R. China
*Corresponding author, zhouxb@ynu.edu.cn

Abstract

This article describes the system submitted to SemEval 2020 Task 12: OffensEval 2020. This task aims to identify and classify offensive languages in different languages on social media. We only participate in the English part of subtask A, which aims to identify offensive languages in English. To solve this task, we propose a BERT model system based on the transform mechanism, and use the maximum self-ensemble to improve model performance. Our model achieved a macro F1 score of 0.913(ranked 13/82) in subtask A.

1 Introduction

In recent years, due to the rapid development of mobile internet and social media platforms, people have begun to use a variety of social media to share their lives, such as Facebook and Twitter. People share their something like events on social media, and this behavior can receive good or bad comments. Some bad reviews slowly evolve into offensive language. A large amount of offensive language is flooded in social media, and filtering out offensive language has become an important thing. Because manual screening is very time consuming and can cause symptoms such as post-traumatic stress disorder, many studies have focused on automating this process.

Task 12 of Semeval 2020 (Zampieri et al., 2020) is the second edition of task 6 of Semeval 2019 (Zampieri et al., 2019b). The purpose of this task is to identify online offensive language . Its goal is to use computational methods to identify offensive and hate speech in user-generated content on online social media platforms. We can use this method to prevent the abuse of offensive words in social media. This task gives us some social media platforms and categorizes the content

In this task, we only participate in subtask A (Rosenthal et al., 2020): Recognizing offensive language. For this task, we use deep learning methods. We mainly use BERT model based on Transformer mechanism and model ensemble. After data processing, we input the generated word vector into the pre-trained model. In addition, we also adopted the method of model ensemble to optimize the prediction probability of the results in order to get better results. Facts have proved that this approach is an effective method.

The rest of our paper is structured as follows. Section 2 describes data preparation. Models are described in Section 3. Experiments and evaluation are described in Section 4. The conclusions are drawn in Section 5.

2 Data Preparation

The organizers provided training and test sets (Mubarak et al., 2020), containing 9073396 and 3887 sentences respectively. We note that the data set format of task 12 is different from that of the previous year (Zampieri et al., 2019a). Therefore, before we perform data preprocessing, we need to process the data format. The data format of the obtained task 12 is shown in figure 1.

We did not get a clear label data. We get the data about the label including AVG and STD. Where AVG is the average confidence of a particular instance predicted by several supervised models. STD is

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 2067–2072
Barcelona, Spain (Online), December 12, 2020.

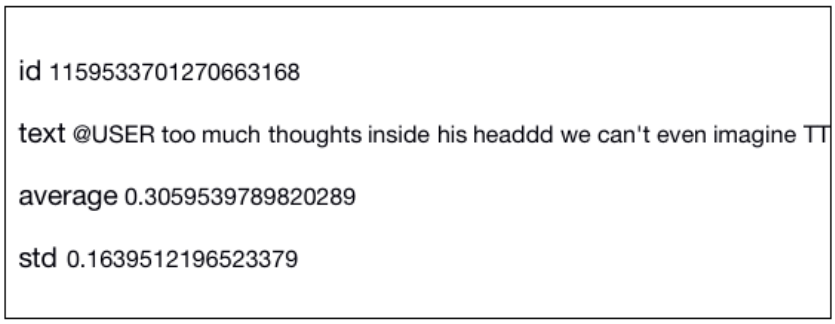

Figure 1: An example from the SemEval 2020 Task 12 dataset

the standard deviation of the confidence of a particular instance from AVG. In order to get the label, we need to calculate the confidence of the models to get the value of the label. For the characteristics of the data format, we have two calculation methods to calculate the confidence of the data. The calculation formula is shown below.

$$S_1 = AVG + STD \tag{1}$$

$$S_2 = AVG \tag{2}$$

where S_1 and S_2 represent the calculated confidence. (In the remaining of this article, we use S_1 and S_2 to denote these two methods)

Since we only participated in subtask A, and this is a binary task. Therefore, we define the sentence with the calculated confidence greater than 0.50 as the label 1. Where the label 1 represents offensive language, on the other hand, the label 0 represents non-aggressive language. The processed data format is shown in Figure 2. After modifying the data format, we can start data preprocessing.

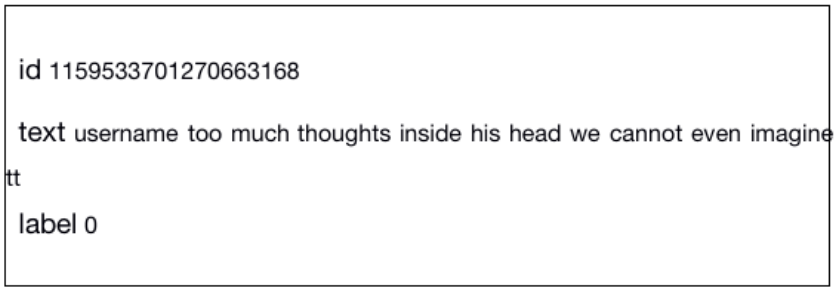

Figure 2: An example from the modified data set in SemEval 2020 Task 12

The text from the tweets is inherently heavily useless. Tweets are processed using the tweettokenize tool (Vydiswaran et al., 2019). Cleaning the text before further processing helps to generate better functionality and semantics. We perform the following preprocessing steps.

- All of @user is replaced with username.Username mentions, i.e. words starting with @, generally provide no information in terms of sentiment.Hence such terms are removed completely from the tweet.
- We know that some repeated symbols have no meaning. As a result, repeated periods, question marks and exclamation marks are replaced with a single instance with the special mark ”repeat” added.
- All contractions were changed to complete parts. This helps the machine understand the meaning of words (for example:”there’re” changed to ”there” and ”are”.

- Twitter data contains a lot of emojis. Emojis can cause the number of unknown words to rise, which can lead to poor pre-training effects. Emoticons (for example, ":(", ":" ",": P "and emoticons, etc.) are replaced by emotional words with their own meaning. This will improve the pre-training effect

- Generally, words have different forms according to the change of context. However, different forms of words will cause ambiguity in pre-training and affect the effect of pre-training. Lexicalization, through WordNetLemmatizer (Nielsen, 2011) to restore language vocabulary to the general form (can express complete semantics).

- Tokens are converted to lower case.

- In Twitter, there are always different themes. Each topic has its own topic tag. Usually, these hashtags are indicated by the symbol "#". However, we know that the symbol "#" has no meaning. Therefore, the "#" symbol is removed and the word itself is reserved for hashtags.

3 Models

After data preprocessing, we can start training the model. In this task, the model that we use is the BERT model based on the transform mechanism. In order to improve the effect of the model, we also adopted the model ensemble method to improve the model effect. Facts have proved that this has indeed been improved. The model is shown in Figure 2. In the following, we will introduce the details of the model we use.

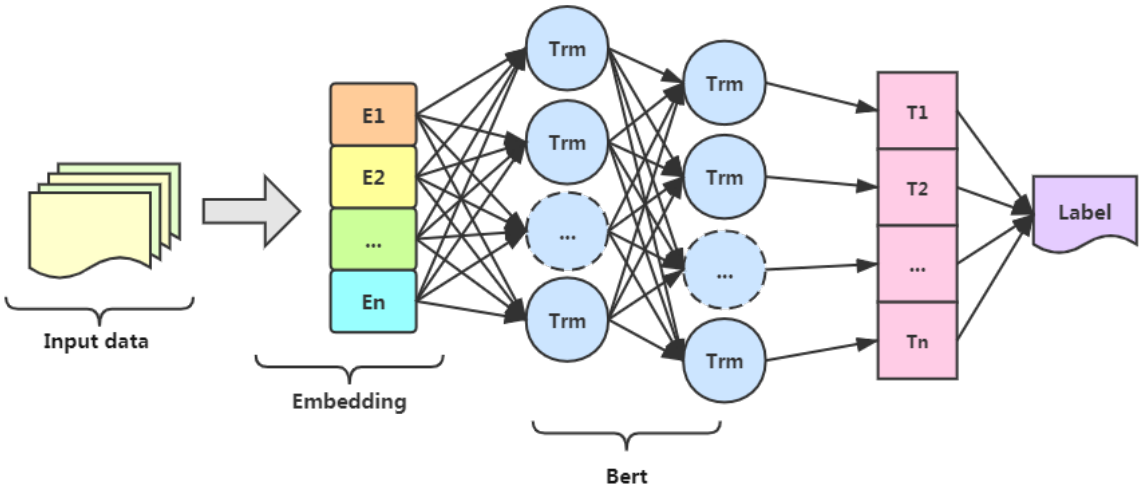

Figure 3: The architecture of the bert model in our task

3.1 Embeddings

The model performs 3 Embeddings: the word embedding (Shuang et al., 2020); the position embedding (You et al., 2019) and segment embedding. The word embedding encodes the information of each word, and the position embedding is similar to the word, mapping a position into a low-dimensional dense vector. Segment embedding indicates whether the currently encoded words belong to the same sentence. Therefore, it all corresponds to an embedding vector. Segment embedding of the same sentence is shared so that it can learn information belonging to different segments.

3.2 BERT

Google company introduces a new language representation model called bert, which stands for Bidirectional Encoder Representations from Transformers. Unlike other language representation models, BERT is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT (Devlin et al., 2018) model can be fine tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task specific architecture modifications.

In this article, we use the BERT model for training. We can get a better result. However, we found that the single-use BERT model has a limited improvement effect. Therefore, we need to find ways to improve the model to get better results

3.3 Max Self-ensemble

Usually, the method of using bert pre-trained model in the task is to fine-tune the model. This method can significantly improve the performance of the model. Therefore, looking for a good fine-tuning strategy is worthy of attention.

In this article, we hope to make the best use of the BERT model through better fine-tuning strategies without using external data or knowledge to achieve the best task performance. BERT models are usually fine-tuned using stochastic gradient descent (SGD) methods. In fact, fine-tuning the performance of bert is usually sensitive to different random seeds and orders of the training data, especially if the last training sample is noise. In order to alleviate this situation, the integration method is widely used to combine multiple fine-tuning models because it can reduce overfitting and improve model generalization. The ensemble (Wang et al., 2018) BERT model usually has higher performance than the single BERT model.

We know that common ensemble methods are based on voting. In this article, first, we fine-tune multiple BERT models with different random seeds. For each input, we will output the best prediction and probability made by the fine-tuned BERT, and summarize the prediction probability of each model. The output of the integrated model is the prediction with the highest probability. This ensemble method is called max-voting ensemble (Xu et al., 2020). The formula for the ensemble BERT model we used is shown below

$$MEB = Bert_{vote}(x; s) = Max(\sum_{n=1}^{s} Bert(x_s)) \tag{3}$$

where MEB is the ensemble BERT model and $Bert(x_s)$ represents a fine-tuning of the BERT model.

The BERT model after voting ensemble has improved significantly in our task. Our max-voting ensemble is shown in Figure 4.

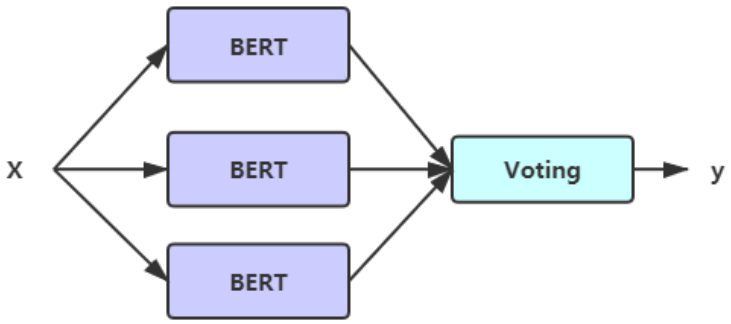

Figure 4: The architecture of the ensemble bert model in our task

where x represents our input and y represents the model ensemble output.

4 Experiments and evaluation

In this task, we use the BERT model to train the task. For the BERT model, the main parameters we focused on are the training step size, batch size, epoch, and learning rate. After learning the parameter adjustment for similar tasks, we fine-tune the model parameters. Our parameter fine-tuning is shown in Table 1.

N	train step	learning rate	batch size	epoch
1	10000	3e-5	16	20
2	20000	4e-5	32	20
3	23800	5e-6	64	30

Table 1: Details of the parameters.

In this task, we use macro F1 to evaluate the quality of the classification system. Our data set uses two methods, S_1 and S_2 , for preprocessing. After fine-tuning the model parameters, we select the data with high score parameters for the maximum ensemble voting fine-tuning in each data set processing method. The results of the model are shown in Table 2.

Status	System	dataset method	MarcoF1
submitted	Majority Baseline	/	0.4193
	BERT	s_1	0.881
	BERT	s_2	0.892
	BERT	s_1+s_2	**0.908**
unsubmitted	MEB	s_1	0.905
	MEB	s_2	0.907
	MEB	s_1+s_2	**0.913**

Table 2: Results with different methods.

where Majority Baseline is the result evaluated by the task organizer.

Due to time reasons, some of our results were not submitted to the system. From the table, it can be noted that the MEB method can significantly improve the effect of the model. This is a useful method for our task.

5 Conclusion

In this task, we mainly use BERT model based on transformer mechanism and model ensemble. After testing, we found that the performance of the single model is slightly worse than the ensemble model. And there are some differences between the results of different parameters of the same model. Our results are still not as satisfactory as the top teams on the leaderboard.

In the future, we will continue to adjust the model, improve the hardware configuration of the computer, collect more external data, and conduct more experiments to obtain better results. Meanwhile, we will try to solve the problem of data imbalance. We will discuss the impact of data preprocessing on model performance. We will continue to carry out model optimization and try more ensemble model methods.

Acknowledgements

This work was supported by the Natural Science Foundations of China under Grant 61463050, the Science Foundation of Yunnan Education Department under Grant 2020Y0011.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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*.

Finn Årup Nielsen. 2011. A new anew: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A Large-Scale Weakly Supervised Dataset for Offensive Language Identification. In *arxiv*.

Kai Shuang, Zhixuan Zhang, Jonathan Loo, and Sen Su. 2020. Convolution–deconvolution word embedding: An end-to-end multi-prototype fusion embedding method for natural language processing. *Information Fusion*, 53:112–122.

VG Vinod Vydiswaran, Grace Ganzel, Bryan Romas, Deahan Yu, Amy Austin, Neha Bhomia, Socheatha Chan, Stephanie Hall, Van Le, Aaron Miller, et al. 2019. Towards text processing pipelines to identify adverse drug events-related tweets: University of michigan@ smm4h 2019 task 1. In *Proceedings of the Fourth Social Media Mining for Health Applications (# SMM4H) Workshop & Shared Task*, pages 107–109.

Jin Wang, Bo Peng, and Xuejie Zhang. 2018. Using a stacked residual lstm model for sentiment intensity prediction. *Neurocomputing*, 322:93–101.

Yige Xu, Xipeng Qiu, Ligao Zhou, and Xuanjing Huang. 2020. Improving bert fine-tuning via self-ensemble and self-distillation. *arXiv preprint arXiv:2002.10345*.

Jiaxuan You, Rex Ying, and Jure Leskovec. 2019. Position-aware graph neural networks. *arXiv preprint arXiv:1906.04817*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1415–1420.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval)*.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.