YNU-HPCC at SemEval-2021 Task 5: Using a Transformer-based Model with Auxiliary Information for Toxic Span Detection

Ruijun Chen, Jin Wang and Xuejie Zhang
School of Information Science and Engineering
Yunnan University
Kunming, China
Contact: chenrj@mail.ynu.edu.cn, {wangjin, xjzhang}@ynu.edu.cn

Abstract

Toxic span detection requires the detection of spans that make a text toxic instead of simply classifying the text. In this paper, a transformer-based model with auxiliary information is proposed for SemEval-2021 Task 5. The proposed model was implemented based on the BERT-CRF architecture. It consists of three parts: a transformer-based model that can obtain the token representation, an auxiliary information module that combines features from different layers, and an output layer used for the classification. Various BERT-based models, such as BERT, ALBERT, RoBERTa, and XLNET, were used to learn contextual representations. The predictions of these models were assembled to improve the sequence labeling tasks by using a voting strategy. Experimental results showed that the introduced auxiliary information can improve the performance of toxic spans detection. The proposed model ranked 5th of 91 in the competition. The code of this study is available at https://github.com/Chenrj233/semeval2021_task5

1 Introduction

Existing toxicity detection datasets and models classify the entire comment or document and do not identify the range that makes the text toxic. A system that accurately locates the toxicity range in the text is crucial in achieving semi-automatic review. As a complete submission for the shared task, systems are required to extract a list of toxic spans or an empty list per text. We define a sequence of words that attribute to the text's toxicity as the toxic span. Table 1 shows two toxic spans, "stupid" and "a!@#!@," which have character offsets from 10 to 15 (counting starts from 0) and from 51 to 56, respectively. Systems are then expected to return the offset list for this text.

Text
This is a stupid example, so thank you for nothing a!@#!@.
Offset List
[10,11,12,13,14,15,51,52,53,54,55,56]

Table 1: Example of toxic spans detection shared task.

The main purpose of this task is to identify the toxic spans in a given text; this can be transformed into a sequence labeling task in natural language processing. Unlike normal sequence labeling tasks, this task is more challenging because the toxic spans in the text may involve a word, phrase, or even a sentence. Traditional methods used to address the problem of sequence labeling include conditional random fields (CRF) (Lafferty et al., 1999), combined models of both long-short-term memory and CRF (LSTM-CRF) (Gupta et al., 2019), and bidirectional encoder representation from transformers (BERT) (Devlin et al., 2019).

In this study, we use BERT, ALBERT (Lan et al., 2019), RoBERTa (Liu et al., 2019), and XLNET (Yang et al., 2019) to solve this problem. Compared with the conventional model, our model adds auxiliary information to improve the performance in this task. After a simple analysis of the text data, it can be found that not all the words in the toxic span have a toxic meaning, and some toxic meanings occur in a specific context or semantic conditions. Therefore, if the tokens can be classified with the auxiliary information such as sentence representation, the performance of the model will improve. The results of the experiment prove that some of the proposed methods are effective. By using ensemble learning, we merge the results of the BERT, ALBERT, RoBERTa, and XLNET models into the final prediction, obtaining the 5th rank out of 91 and a F_1 score of 0.696.

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 841–845
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

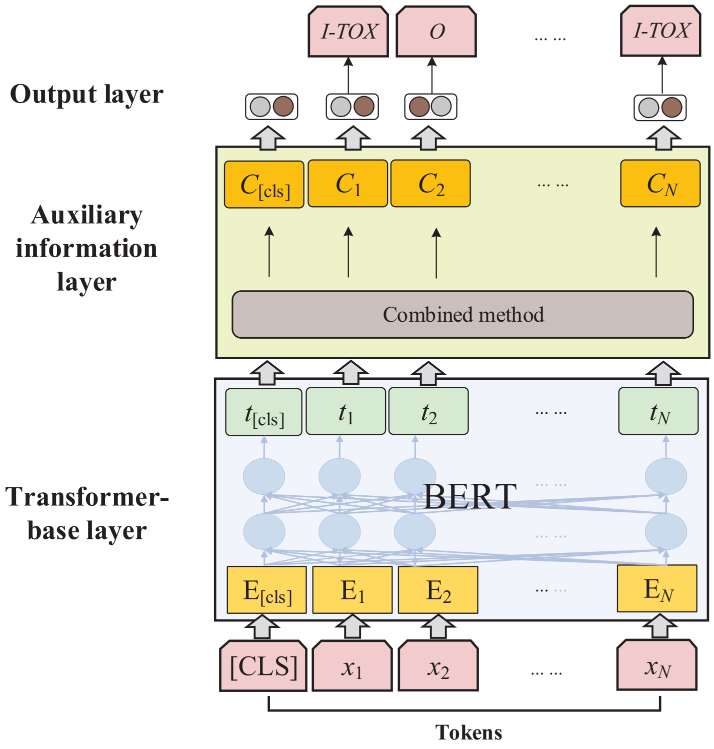


Figure 1: Overall architecture of the proposed transformer-based model with auxiliary information.

The remainder of this paper is organized as follows. Section 2 describes the specific structure of the adopted model. The experimental results are summarized in Section 3. Finally, Section 4 presents the conclusions of the study.

2 Transformer-based Model with Auxiliary Information

Figure 1 shows the architecture of the proposed model, which consists of three layers: a transformer-based layer, an auxiliary information layer, and an output layer. The transformer-based layer can be BERT, ALBERT, RoBERTa, XLNET, or any other transformer-based model. In the auxiliary information layer, several approaches are applied to combine token representation. The combined token representations are used in the output layer to output the label of each token.

2.1 Transformer-based Layer

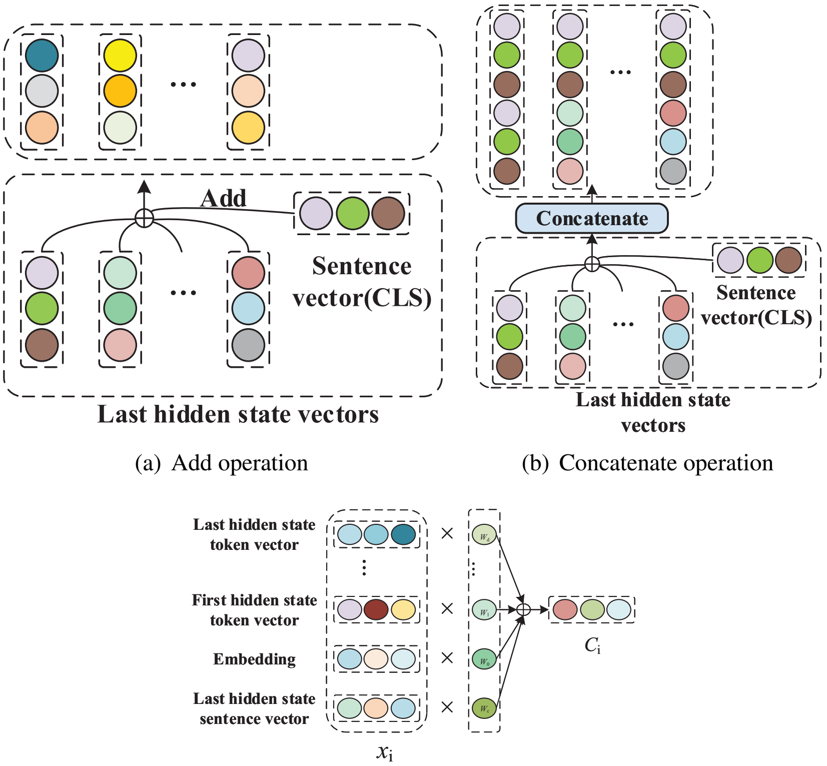
The transformer-based layer is the first part of the model. The purpose of this layer is to obtain the representation of tokens and the entire text. For illustration, we can use the BERT-large (Devlin et al., 2019) model to produce token representations from each layer. With BERT-large, 25 layers of token representation vectors can be obtained: one embedding representation and twenty-four hidden states. Unlike previous methods, 25 layers of token representation vectors are combined by using several methods in the next layer. The representations produced by the transformer-based layer are then fed into the next layer.

2.2 Auxiliary Information Layer

The traditional method directly passes the token representation vectors to the classification layer. To improve the performance of the model, we attempt to combine token representation vectors and the sentence representation vector in different ways. Figure 2 depicts the attempted methods, which are described as follows:

- **Method 1.** Token vector of the last layer and the sentence vector.
- **Method 2.** Token vector of the last layer concatenated with the sentence vector.
- **Method 3.** Linear combination of the token vector of each layer.
- **Method 4.** Linear combination of the token vector of each layer and the sentence vector.

The combined representation of tokens passes on to the next layer.



(a) Add operation

(b) Concatenate operation

(c) Linear combination if $W_C = 0$; linear combination plus sentence vector if $W_C = 1$

Figure 2: Different types of representations in auxiliary information layer.

2.3 Output Layer

The output layer is a fully connected dense layer with softmax activation. It aims to classify whether a token belongs to the toxic span in a text. The combined representation of each token passed by the auxiliary information layer is the input of this layer, and the output layer predicts the labels for the candidate tokens. The loss function of the proposed model is the categorical cross-entropy.

3 Experimental Results

In this section, we present the comparative results of the proposed model.

3.1 Dataset

During the competition, we used only the data (Pavlopoulos et al., 2021) provided by the task organizer for the experiments. This task involves trial data (which include 689 posts and spans), training data (which include 7939 posts and spans), and test data (which include 2000 posts). We used the training data as the training set and trial data as the validation set. We needed to find the subscript offset set of the toxic spans of each post in the test data.

As this is a sequence labeling task, a common data preprocessing method is to use the BIO tagging format. We observed better performance when the IO tagging format was adopted during the actual training process. Therefore, our output layer was a two-classification layer that outputs the probability of a token belonging to a toxic span.

3.2 Evaluation Metrics

For this task, we employed the F_1-score metric (da San Martino et al., 2020) to evaluate the responses of a system participating in the challenge.

For each post, t_i, the predicted span was a set, S_i, of character offsets and G_i was the character offset of the groundtruth annotations of t_i. The F_1 score of ti was calculated as follows:

$$F_1(S_i, G_i) = \frac{2 * P(S_i, G_i) * R(S_i, G_i)}{P(S_i, G_i) + R(S_i, G_i)} \qquad (1)$$

where $P(S_i, G_i)$ and $R(S_i, G_i)$ are respectively precision and recall scores defined as follows:

$$P(S_i, G_i) = \frac{|S_i \cap G_i|}{|S_i|} \qquad (2)$$

Transformer model	Auxiliary information	Validation set	Test set
BERT-large	None	0.649	0.679
	Add sentence vector	0.670	0.679
	Concatenate sentence vector	0.671	0.671
	Linear combination	0.661	0.683
	Linear combination plus sentence vector	0.672	0.679
ALBERT-xlarge	None	0.659	0.675
	Add sentence vector	0.665	0.665
	Concatenate sentence vector	0.656	0.668
	Linear combination	0.648	0.667
	Linear combination plus sentence vector	0.657	0.670
RoBERTa-large	None	0.656	0.676
	Add sentence vector	0.620	0.662
	Concatenate sentence vector	0.610	0.673
	Linear combination	0.663	0.667
	Linear combination plus sentence vector	0.667	0.667
XLNET-large	None	0.659	0.679
	Add sentence vector	0.674	0.674
	Concatenate sentence vector	0.669	0.669
	Linear combination	0.674	0.675
	Linear combination plus sentence vector	0.678	0.681

Table 2: F_1-score of different models on validation set and test set.

$$R(S_i, G_i) = \frac{|S_i \cap G_i|}{|G_i|} \qquad (3)$$

If G_i is empty for some post t_i, we set $F_1(S_i, G_i) = 1$ if S_i is also empty and $F_1(S_i, G_i) = 0$ otherwise. Finally, we averaged $F_1(S_i, G_i)$ over all posts t_i.

3.3 Implementation Details

Each model was fine-tuned for eight epochs. We used the Adam (Kingma and Ba, 2015), AdamW (Loshchilov and Hutter, 2017), and Stochastic Gradient Descent (SGD) algorithm for optimization. The final one used was AdamW with a learning rate of $5e-6$.

In the training process, we attempted to use the cross-entropy loss, focal loss (Lin et al., 2020), and Dice loss (Li et al., 2020). The results on the validation set showed that the focal loss and Dice loss are better than the cross-entropy loss. This may be due to an imbalance between the toxic and nontoxic categories in the text. In order to compare with the baseline model, we finally used the cross-entropy loss function to train all models.

3.4 Comparative Results

We used BERT, ALBERT, RoBERTa, and XLNET as the transformer-based layers. The model exhibiting the best performance on the validation set in the eight epochs was used to predict the spans on the test set in the competition. The results on the test set are presented in Table 2. The model that performed the best on the test set over the eight epochs is also presented in Table 2.

In terms of the performance on the validation set, the BERT and XLNET models with the auxiliary information layer are better than those without. Method 4, mentioned earlier, achieves the highest F_1 score. In case of ALBERT, only method 1 improves the performance. Methods 3 and 4 can improve the performance of RoBERTa.

Regardless of the performance on the validation set, the F_1 score increases by 0.004 when using method 3 in the BERT model and increases by 0.002 when using method 4 in XLNET. The auxiliary information layer does not improve the performance of ALBERT and RoBERTa.

The results show that the performance of the best-performing model on the validation set is significantly different from that of the best-performing model on the test set. The reason for this difference may be the inconsistent data distribution of the validation and test sets.

However, the results indicate that when the validation set is not appropriate, the auxiliary information layer can effectively improve the performance of the baseline model on the validation set. The BERT and XLNET models are the most suitable for the auxiliary information layer.

4 Conclusion

In this paper, we introduce the method we used in SemEval-2021 Task 5. We improved the performance of the basic model by reducing the number of categories for each token, selecting the appropriate loss function, adding some additional information to the representation vector of the tokens during classification, and finally obtaining a model that can detect the toxicity in a text. Our experimental results showed that adding auxiliary information to the original token representation vector is helpful in sequence labeling tasks.

In addition, we found that the model has some limitations. After analyzing the prediction results, we observed that although the model can learn the representation of each token well, token classification errors can occur when some tokens are toxic without the entire text being toxic. One possible solution for this is to add a text classification task to train the model.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants Nos. 61702443, 61966038 and 61762091.

References

Giovanni da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2020. Fine-grained analysis of propaganda in news articles. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, (January):5636–5646.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1(Mlm):4171–4186.

Pankaj Gupta, Khushbu Saxena, Usama Yaseen, Thomas Runkler, and Hinrich Schütze. 2019. Neural architectures for fine-grained propaganda detection in news. *arXiv*.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15.

John Lafferty, Andrew Mccallum, and Fernando Pereira. 1999. Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data Abstract. 2001(June):282–289.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv*, pages 1–17.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020. Dice Loss for Data-imbalanced NLP Tasks. pages 465–476.

Tsung Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. 2020. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv*, (1).

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv*.

John Pavlopoulos, Léo Laugier, Jeffrey Sorensen, and Ion Androutsopoulos. 2021. Semeval-2021 task 5: Toxic spans detection (to appear). In *Proceedings of the 15th International Workshop on Semantic Evaluation*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. *arXiv*, (NeurIPS):1–18.