hub at SemEval-2021 Task 5: Toxic Span Detection Based on Word-Level Classification

Bo Huang, Yang Bai, Xiaobing Zhou*
School of Information Science and Engineering
Yunnan University, Yunnan, P.R. China
⋆Corresponding author:zhouxb@ynu.edu.com

Abstract

This article introduces the system description of the hub team, which explains the related work and experimental results of our team's participation in SemEval 2021 Task 5: Toxic Spans Detection. The data for this shared task comes from some posts on the Internet. The task goal is to identify the toxic content contained in these text data. We need to find the span of the toxic text in the text data as accurately as possible. In the same post, the toxic text may be one paragraph or multiple paragraphs. Our team uses a classification scheme based on word-level to accomplish this task. The system we used to submit the results is ALBERT+BILSTM+CRF. The result evaluation index of the task submission is the F1 score, and the final score of the prediction result of the test set submitted by our team is 0.6640226029.

1 Introduction and Background

From the popularization of the Internet to the first year of the mobile Internet in 2011, the number of online social media and social media users has continued to grow. In the context of the ever-expanding user base, coupled with the free and interactive features of online social media communication. Therefore, online social media has exposed many issues worthy of our attention, such as the lack of communication standards and the out-of-control of information dissemination, which makes the dissemination of online social media prone to various negative functions (Baccarella et al., 2018).

The task of toxic span detection is to detect the span of text with toxic information in the text (Pavlopoulos et al., 2021). The goal of the task is to predict the beginning and ending character positions in the text as accurately as possible. Reviewing the content in online media can effectively avoid the spread of a series of negative information such as cyber violence, cyberbullying, and false news. Audits are essential to promote healthy online discussions. However, the content and number of posts in social media are too large, and the manual review method obviously cannot achieve a good effect. Therefore, in combination with the development of modern technology, achieving a semi-automatic audit is the best solution.

2 Related Work

There are many different kinds of methods for identifying negative information in social media, but usually, these methods mainly focus on supervised learning. Simple SurfaceFeatures similar to the bag of words model can provide very clear and easy-to-understand information in text processing tasks. The general approach of this method is to merge multiple larger n-grams into a feature set (Nobata et al., 2016; Djuric et al., 2015). Use the artificial neural network method to train the word embeddings in the corpus. The purpose is to use the distance between vectors to indicate the semantic similarity of different words. Djuric et al. proposed a method of directly using embedding to represent the entire text and showed us the effect of this method (Sun et al., 2019).

Negative information in the text can be detected by the above methods. But more information needs to be obtained according to the context. The pretrained language model based on the Transformer architecture has great advantages both at the word level and in context information (Wang et al., 2019). Therefore, in this task, we try to combine the pretrained language model to complete the detection of toxic content.

3 Data and Methods

In this section, we introduce the data provided by the task organizer team to the participating teams, as well as the models and methods we use.

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 904–908
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

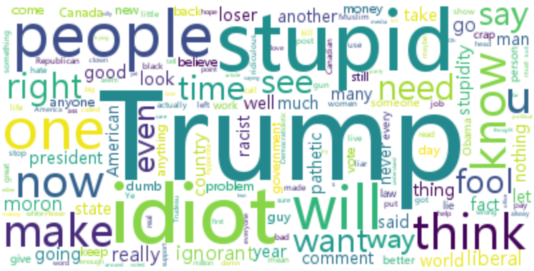

Figure 1: A word cloud diagram of the training set text data provided by the task organizer team. The result shown in the figure is the data after removing the stop words.

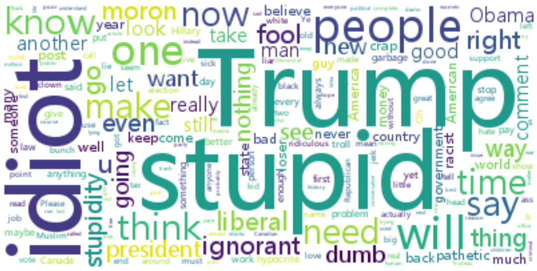

Figure 2: A word cloud diagram of the test set text data provided by the task organizer team. The result shown in the figure is the data after removing the stop words.

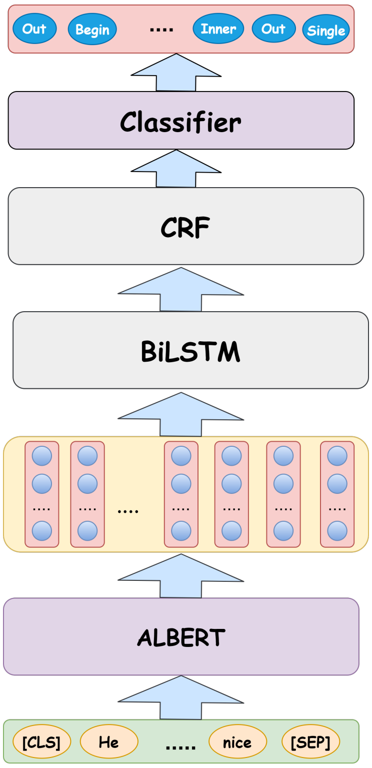

Figure 3: The model structure and data flow we used in the task.

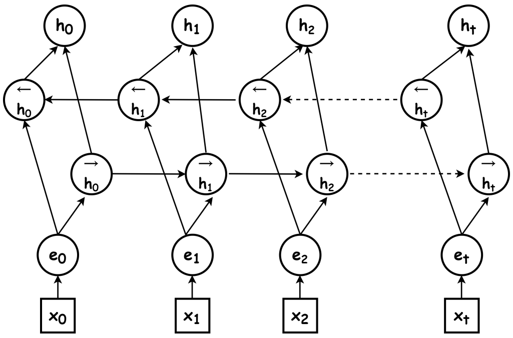

Figure 4: The BiLSTM structure and data flow

3.1 Data Description

The task organizer team provides each team with training data sets and test data sets related to the "Toxic Spans Detection" task. The training data set consists of two parts, one is the text data of the post, and the other is the index position of the span of Toxic Spans. Each post corresponds to an index span data. There are some posts in the training set that do not contain toxic content, and there are one or more pieces of toxic content in the remaining posts. Also, in the index range of these toxic content, it may be a phrase, a sentence, or a word. The length of the post is not the same. Compared with the training data set, the test set only contains the text data of the posts. We need to use our method to predict the index span of the toxic content of posts in the test set. Table 1 shows the sample data of the data we used in the task.

There are 7939 and 2000 pieces of data in the training set and test set, respectively. We visualize the text data in the training set and the text data in the test set using word cloud graphs. The word cloud image clearly shows us the characteristics of word frequency distribution in the text data set. Regardless of the text data in the training set data or the text data in the test set data, some insulting vocabulary, as well as some neutral vocabulary and human names (Trump) appear most. Those sentences with insulting words are usually detected as text with toxic spans. Some short sentences composed of words with neutral meanings and other phrases may also be recognized as text with toxic spans. The reason is because these sentences combined with some background information will convey unfriendly information. Figure 1 and Figure 2 show the word frequency information in the training set and the word frequency information in the test set.

Data category	Text	Spans
train	““““who do you think should do the ***killing***?”””	[32, 33, 34, 35, 36, 37, 38]
train	“CROOKED Trump = GUILTY as hell. ***pathetic***”	[32, 33, 34, 35, 36, 37, 38, 39]
train	He is a scary maniac with a psychopath attitude.	[]
test	This ***idiot*** has no clue.	[5, 6, 7, 8, 9]

Table 1: Part of the data samples in the training set and test set provided by the task organizer team.

3.2 Methods

In our system, we use pre-processed data as the input to ALBERT. The architectures of ALBERT base and BERT base are both composed of 12-layer Transformer(Devlin et al., 2018; Lan et al., 2019). Compared with the BERT model, the result of the original embedding parameter P is the product of the vocabulary size V and the hidden layer size H. ALBERT factorizes the Embedding matrix by using a lower-dimensional embedding space of size E and then project it to the hidden space.

$$V * H = P \qquad \rightarrow \qquad V * E + E * H = P \quad (1)$$

Different from H=E in BERT, when $H \gg E$, the number of parameters of ALBERT has a significant reduction. Another big difference from BERT is that ALBERT's default decision is to share all parameters across layers (Lan et al., 2019). Based on these improvements, the training effect of ALBERT is better than that of BERT. In terms of memory usage, the ALBERT pre-training model is also smaller than the BERT pre-training model.

Based on the structural characteristics of the RNN artificial neural network, the RNN network has great advantages in processing text data (Zaremba et al., 2014). But in the actual training process, a simple RNN network is difficult to converge. Because the loss value of the RNN network is continuously accumulated as the text sequence increases. Compared with the RNN network, LSTM artificial neural network has great advantages in model convergence and processing long text (Gers et al., 1999; Olah, 2015). LSTM is mainly composed of two key points, one is the cell state, the other is the gate unit. The information learned by the LSTM unit will be directly stored in the cell state, and we can input and update the value in the cell state. The gating unit plays a role in controlling how to update the value in the cell state. These gating units are composed of forget control gate, input control gate, and output control gate. The key to the composition of the gating unit is the sigmoid function.

In our system, first, we use the preprocessed data as the input of ALBERT. Then, use the output of ALBERT as the input of BiLSTM. Next, the output of the BiLSTM model is used as the input of CRF. Finally, a classifier is used to classify the output results of CRF. The classifier needs to classify each word in the text into one of four different categories. These four categories are the beginning of the text span, the inside of the text span, the outside of the text span, and the toxic span formed by a single word. The architecture of BiLSTM, our model architecture and data flow can be seen in Figure 3 and Figure 4.

4 Experiment and Results

In this section, we will introduce the data preprocessing methods and experimental settings we used in the task and the final results.

4.1 Data Preprocessing

Because our model and method are to classify content at the word level. So we preprocessed the text data provided by the task organizer team. Preprocessing mainly involves dividing all words in each post into one of four categories (Begin, Out, Inner, Single). These four categories represent our specific description in Section 3.2, paragraph 4. Then split the processed training set into a new training set and a validation set. The split rule is to randomly extract part from the training set as the validation set, and the ratio of the training set to the validation set is 8: 2.

4.2 Experiment setting

We use preprocessed data as input to the model. During the training process, we adjust the parameters of the model according to the results of the model on the validation set. The learning rates used by the ALBERT-base, BiLSTM, CRF and classifier modules in the model are not the same. The learning rate used by ALBERT-base and BiL-

STM is 3e-5, and the learning rate used by CRF and classifiers is 1e-4. The maximum length of sentences input in the model is fixed at 120 words. The choice of this length comes from the length of the text in the data and the memory size of the GPU. Sentences that do not reach 120 words in length will be supplemented with zeros. Sentences longer than 120 words will be deleted. The epoch and batch during training are 10 and 32, respectively. The optimizer used in our experiment is Radam (Liu et al., 2019).

4.3 Results evaluation method

The evaluation index announced by the task organizer team is the F1 score. Let system A_i return a set $S^t_{A_i}$ of character offsets, for parts of the post found to be toxic. Let G be the character offsets of the ground truth annotations. We compute the F1 score of system A_i with respect to the ground truth *G* for post *t* as follows, where $| \cdot |$ denotes set cardinality.

$$F_1^t(A_i, G) = \frac{2 \cdot P^t(A_i, G) \cdot R^t(A_i, G)}{P^t(A_i, G) + R^t(A_i, G)} \tag{2}$$

$$P^t(A_i, G) = \frac{|S^t_{A_i} \bigcap S^t_G|}{|S^t_{A_t}|} \tag{3}$$

$$R^t(A_i, G) = \frac{|S^t_{A_i} \bigcap S^t_G|}{|S^t_G|} \tag{4}$$

If S^t_G is empty for some post *t* (no gold spans are given for *t*), we set $F1^t(A_i$, *G*) = 1 if $S^t_{A_i}$ is also empty, and $F1^t(A_i$, *G*) = 0 otherwise. We finally average $F1^t(A_i$, *G*) over all the posts *t* of an evaluation dataset *T* to obtain a single score for system A_i (Da San Martino et al., 2019).

4.4 Results

In the final results list announced by the task organizer team, a total of 91 team results are presented in the list. Our team's F1 result score was 0.6640226029, ranking 37th. Table 2 presents the result score of our system on the validation set and the result score on the test set. Compared with the results of the top 3 teams in the ranking, our result is 0.0442802224 different from the optimal result. Table 3 shows the scores of the top three teams and our team on the test set.

Data	**F1 score**
Validation set	0.6821240031
Test set	0.6640226029

Table 2: The scores obtained by our system on the validation set and test set. The validation set comes from 20% of the training set provided by the task organizer team.

team	**F1 score**	**Rank**
HITSZ-HLT	0.7083028253	1
S-NLP	0.7077035474	2
hitmi&t	0.6984762534	3
hub(our method)	0.6640226029	37

Table 3: In the result list released by the task organizer team, the top 3 submitted test set prediction results scores and our submitted test set prediction results scores. A total of 91 participating teams submitted the prediction results of the test set.

5 Conclusion

This paper presents the system description submitted by our team to SemEval 2021 Task 5: Toxic Spans Detection. Our goal is to use our system to detect the span of toxic content as accurately as possible. We use a classification scheme based on word-level to complete the task. The system combines the pre-training language model (ALBERT) and BiLSTM+CRF commonly used in NLP tasks. The results we submitted proved the feasibility of our system, but compared with the optimal results, our method still has room for improvement. In future work, we will try to improve our methods to achieve better results.

References

Christian V Baccarella, Timm F Wagner, Jan H Kietzmann, and Ian P McCarthy. 2018. Social media? it's serious! understanding the dark side of social media. *European Management Journal*, 36(4):431–438.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeno, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. 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 5640–5650.

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*.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web*, pages 29–30.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 1999. Learning to forget: Continual prediction with lstm.

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 preprint arXiv:1909.11942*.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2019. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153.

Christopher Olah. 2015. Understanding lstm networks.

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*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Chenguang Wang, Mu Li, and Alexander J Smola. 2019. Language models with transformers. *arXiv preprint arXiv:1904.09408*.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.