HITMI&T at SemEval-2021 Task 5: Integrating Transformer and CRF for Toxic Spans Detection

Chenyi Wang[*], **Tianshu Liu**[*], **Tiejun Zhao**[†]
Machine Intelligence and Translation Laboratory,
Harbin Institute of Technology, Harbin, China
wcy708708@126.com, liutianshu99@163.com, tjzhao@hit.edu.cn

Abstract

This paper introduces our system at SemEval-2021 Task 5: Toxic Spans Detection. The task aims to accurately locate toxic spans within a text. Using BIO tagging scheme, we model the task as a token-level sequence labeling task. Our system uses a single model built on the model of multi-layer bidirectional transformer encoder. And we introduce conditional random field (CRF) to make the model learn the constraints between tags. We use ERNIE as pre-trained model, which is more suitable for the task accroding to our experiments. In addition, we use adversarial training with the fast gradient method (FGM) to improve the robustness of the system. Our system obtains 69.85% F1 score, ranking 3rd for the official evaluation.

1 Introduction

With the prosperity of the Internet, it is easier and easier for people to get information and publish their opinions online. However, sometimes users' opinions can be offensive to others. Because toxic posts will have a negative impact on the network environment, and manual identification is time-consuming and expensive, automatic detection of these behaviors has attracted researchers' attention.

After adapting the hate-speech problem to the problem of word sense disambiguation, an approach to detect hate speech in online text is presented (Warner and Hirschberg, 2012), which uses template-based strategy to generate features and an SVM classifier to identify whether the text is toxic or not. In SemEval-2020 Task 12 (Zampieri et al., 2020) and SemEval-2019 Task 6 (Zampieri et al., 2019), which also related to offensive statements, transformer-based methods were the most popular approaches for their great advantages in learning word representations in context.

In Semeval-2021 task 5: Toxic Spans Detection (Pavlopoulos et al., 2021), the organizers use posts from the publicly available Civil Comments dataset (Borkan et al., 2019), which already comprises post-level toxicity annotations. After manual annotation, character-level annotation results are obtained, which are the toxic spans we need to locate. The task extends the prior work by identifying spans that make a text toxic, which can better explain why posts are offensive rather than just giving a system-generated unexplained toxicity score.

We model the task as a sequence labeling task because toxic spans are contextually influenced. Our model is in Transformer-CRF architecture, and we try different pre-trained models as the transformer's initialization to fine-tune model suitable for toxic spans detection. The Conditional Random Fields (CRF) (Lafferty et al., 2001) allows the model to learn the constraints between tags. We also use the Fast Gradient Method (FGM) (Miyato et al., 2016) as adversarial training strategy, which applies perturbation to word embedding to enhance the robustness of the model.

The paper is organized as follows: Section 2 briefly introduces the Toxic Spans Detection shared task. Section 3 talks about our system, including pre-processing and post-processing. Section 4 shows our experiment results. Finally, the conclusion and future work are drawn in Section 5.

2 Toxic Spans Detection

The research of automatic offensive language detection has gained attention in the past decade. Instead of just classifying the whole comments or documents, the Toxic Spans Detection task requires the system to detect the spans that make a text toxic.

∗ Equal contribution.
† Corresponding author.

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 870–874
Bangkok, Thailand (online), August 5–6, 2021.

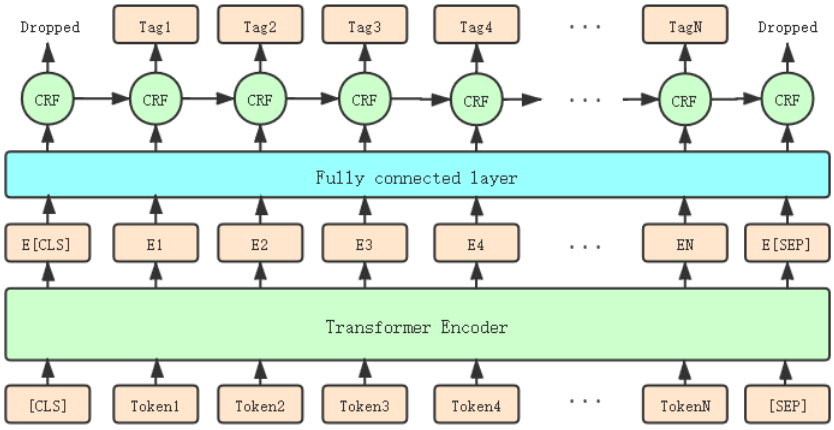

Figure 1: Model structure of Transformer-CRF. In our system, the transformer encoder is ERNIE. "[CLS]" and "[SEP]" are the special input tokens, "TokenN" means the Nth token in tokenized text. "E[CLS]", "E[SEP]" and "EN" means the output of "[CLS]", "[SEP]" and the Nth token's embedding after Transformer Encoder. The output of Transformer-CRF is "TagN", which is "B-toxic", "I-toxic" or "O".

People often judge offensive sentences in terms of words, therefore the toxic spans in this task are always associated with words. In this task, the input sentence may contain no toxic span, which means it is not offensive. On the other hand, there may be more than one word that shows the author's malice in the sentence. Considering toxic spans are contextually influenced, we model the task as a token-level sequence labeling task and use BIO tagging scheme.

3 System description

Our system uses a single model to get the result, which is in Transformer-CRF architecture. In our system, we utilize a pre-trained contextualized language model, namely ERNIE 2.0 (Sun et al., 2019), to identify the toxic tokens. And we introduce CRF to learn the constraints between tags. In addition, we use adversarial training with the FGM to improve our performance.

3.1 Data pre-processing

Toxic spans detection is a character-level task. Considering transformer architecture requires token-level inputs, we choose token-level sequence labeling instead of character-level. When converting character-level sequence labels to token-level, the tokenizer we use is the one used in pre-training. And we lowercase the text before tokenized.

After tokenizing the text, we tag the tokens. If one of the token's spans is tagged as toxic in the original dataset, considering the tag of the previous token, the token will be tagged as "B-toxic" or "I-toxic" in pre-processed data. If the token's spans are not tagged as toxic in the original dataset, the token will be tagged as "O" in pre-processed data. Some typical examples are shown in Table 1.

3.2 Transformer-CRF architecture

Our system uses a single model to get token-level predictions. The model is in Transformer-CRF architecture. As shown in Figure 1, it consists of three components: transformer encoder, fully connected layer, and a CRF layer.

First, the transformer encoder is used to extract representations of the input tokens. Based on multi-layer bidirectional transformer encoder, we use ERNIE as pre-trained model to encode. During pre-training, transformer-based language models always use inputs with special tokens (such as [CLS] or [SEP]). Therefore, after tokenizing text into tokens, we insert "[CLS]" at the beginning of the token list and "[SEP]" at the end to make our input closer to what it would be when ERNIE was pre-trained. When we train the model, the tag of "[CLS]" and "[SEP]" is "O", and we drop the tags of them during predicting.

After we get the embeddings of tokens, the representations are fed into a fully connected layer to

Origin tags	Origin data	Tokens and tags
8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, ...	Another violent and aggressive immigrant killing a innocent and intelligent US Citizen....	another O violent B-toxic and I-toxic aggressive I-toxic ... ...
7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17	What a knucklehead. How can anyone not know this would be offensive??	what O a O kn B-toxic ##uck I-toxic ##le I-toxic ##head I-toxic ... ...

Table 1: Typical Examples, where "Origin tags" means the toxic spans given in the dataset, "Origin data" means the comment given in the dataset, "Tokens and tags" means the tokens and tags after tokenized and tagged.

reduce dimension. Finally, the CRF layer decodes the reduced representations into the most probable tag sequence using the Viterbi algorithm. Using only the fully connected layer may output illegal tags (e.g. "... O I-toxic O ..."), while introducing the CRF layer allows the model to learn the constraints between tags. When fine-tuning the pre-trained model, the layer closer to the input is more likely to learn more simple features. We want to modify the deeper weights more to adapt to the target task. Therefore, we use different learning rates for different parts of the network. The transformer encoder has a lower learning rate, while the fully connected layer and the CRF layer have a higher learning rate.

We train the model maximizing the log-likelihood of the given sequence of tags, as compared to the gold training labels. Except that, we use the FGM as an adversarial training strategy to improve the performance of our system. FGM applies perturbation to word embedding to enhance the robustness of the model.

Given a token sequence $S = t_1, t_2, \ldots t_N$ as input where N denotes the sequence length. The process for the model to obtain token-level results is as follows:

$$e_i = transformer\_encoder(t_i) \quad (1)$$

$$out_i = W_o e_i + b_o \quad (2)$$

$$tag_i = crf(out_i) \quad (3)$$

Where t_i is the current token, and e_i denotes the output of transformer encoder. Then e_i is fed into fully connected layer to reshape into out_i. After that, the CRF layer use out_i to get token-level result tag_i.

3.3 Post-Processing

Our model produces token-level predictions, but detecting toxic spans within the text is a character-level task. To get the results, we use the opposite way of pre-processing to get spans from model's predictions. Particularly, we assume the blank between toxic tokens should be tagged as toxic, too. This is observed from the spans of the dataset.

4 Experiment

4.1 Dataset

We trained our models by SemEval-2021 Task 5 training data which consists of 7,939 samples with a total of 139,115 toxic spans. We convert the span from character-level to token-level, as described in section 3.1. Then we get a total of 31,114 toxic tokens, with an average of 3.92 per sample. And each sample contains an average of 48.50 tokens.

4.2 Metric

Let system A return a set S_A^t of character offsets, for the post t that found to be toxic. Let S_G^t be the set of character offsets of the ground truth annotations of t. We compute the F1 score of system A with respect to the ground truth G for post t as follows, where $|\cdot|$ denotes set cardinality.

$$P^t = \frac{|S_A^t \cap S_G^t|}{|S_A^t|} \quad (4)$$

Method	F1
BERT-CRF	0.6933
ELECTRA-CRF	0.6944
ERNIE-CRF	**0.6985**
ERNIE-CRF w/o-adv	0.6964
ERNIE-CRF nltk-preprocessing	0.6557

Table 2: Results on the official evaluation testing data. "w/o-adv" means no adversarial training. "nltk-preprocessing" means using NLTK for pre-processing.

$$R^t = \frac{|S_A^t \cap S_G^t|}{|S_G^t|} \tag{5}$$

$$F_1^t = \frac{2 \cdot P^t \cdot R^t}{P^t + R^t} \tag{6}$$

If S_G^t is empty for some post t (no gold spans are given for t), we set $F_1^t = 1$ if S_A^t is also empty, and $F_1^t = 0$ otherwise. We finally average F_1^t over all the posts t of an evaluation dataset T to obtain a single F_1 score for system A.

$$F_1 = \frac{\sum_{t \in T} F_1^t}{|T|} \tag{7}$$

4.3 Experiment Settings

We try different pre-trained model as the transformer's initialization such as BERT (Devlin et al., 2018), ELECTRA discriminator (Clark et al., 2020) and ERNIE 2.0 (Sun et al., 2019). We find that the models initialized with ERNIE 2.0 always achieve better performance. So we select ERNIE 2.0 as the transformer's initialization, which has 768 hidden units, 12 heads, 12 hidden layers.

For other parameters, we use streams of 256 tokens, a mini-batch of size 32, transformer's learning rate of 3e-5, CRF's learning rate of 1e-3, the epoch of 3, and the random seed of 42.

4.4 Testing Results

As shown in Table 2, we build five systems including: (1) **BERT-CRF** means model using BERT (Devlin et al., 2018) as the transformer's initialization; (2) **ELECTRA-CRF** means model using ELECTRA discriminator (Clark et al., 2020) as the transformer's initialization; (3) **ERNIE-CRF** means model using ERNIE 2.0 (Sun et al., 2019) as the transformer's initialization; (4) **ERNIE-CRF w/o-adv**; (5) **ERNIE-CRF nltk-preprocessing**. All models are Transformer-CRF architecture. All models use adversarial training with the FGM except **ERNIE-CRF w/o-adv**. All models convert character-level sequence labels to token-level with tokenizers used in pretraining, except for **ERNIE-CRF nltk-preprocessing**, which uses NLTK (Bird et al., 2009) for pre-processing.

Table 2 shows the overall performances of our models on the official evaluation testing data. The ERNIE based model achieves better performance than both the BERT based model and the ELECTRA based model. We conjecture that ERNIE 2.0 is pre-trained through multi-task learning, which allows it to capture lexical, syntactic, and semantic information, such as named entities and discourse relations. And this makes it more suitable for Toxic Spans Detection task.

Adversarial training proved to be effective. Although only a small improvement has been achieved, experiments show that FGM can always achieve a stable improvement. Adversarial training adds some perturbation to the input, which makes the model more robust and has a better performance on the unknown test set.

ERNIE-CRF achieves more than 4 points improvements over **ERNIE-CRF nltk-preprocessing**, which proves the importance of choosing the correct method to convert character-level sequence labels to token-level. We conjecture that there is a mismatch between the word segmentation results of NLTK and the input required by the Transformer model.

4.5 Attempts with no obvious improvement

It is worth mentioning that the best performance of our system is based on a single model. We tried model ensemble to improve the performance of the single model but failed. Due to the limitation of time and submission, we did not find an ensemble method with obvious improvement.

The BiLSTM network has a strong ability to capture long-term dependencies of the input sequence for sequence labeling task (Huang et al., 2015). We tried to add BiLSTM layer after transformer. But this resulted in overfitting, and the training speed

was greatly reduced.

We also tried to combine the information from word embedding with the information from the deep layer. It is proved that, in the machine translation task, the low layers of the network are more focused on lexical information, while deeper layers pay more attention to word meaning (Belinkov et al., 2017). In toxic spans detection task, we consider the information from the lower layers to be important. So we concatenate the word embedding layer with the output of ERNIE, but this didn't work.

5 Conclusion and Future Work

The paper describes our system at SemEval-2021 Task 5, which integrating Transformer and CRF for Toxic Spans Detection task. It is shown that, in this task, using ERNIE as pre-trained model achieves better performance in Transformer-CRF architecture. We convert the character-level sequence labeling task into token-level, and prove the importance of preprocessing method. We also use adversarial training with the FGM to improve the robustness of the system. Our system achieves the third F1 score in official evaluation.

Since the best performance of our system is based on a single model, we are planning to find an effective ensemble method to improve the performance of the single model in the future.

References

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? *arXiv preprint arXiv:1704.03471*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced metrics for measuring unintended bias with real data for text classification. *CoRR*, abs/1903.04561.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2016. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*.

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*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2019. Ernie 2.0: A continual pre-training framework for language understanding. *arXiv preprint arXiv:1907.12412*.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the second workshop on language in social media*, pages 19–26.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

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). *arXiv preprint arXiv:2006.07235*.