Entity at SemEval-2021 Task 5: Weakly Supervised Token Labelling for Toxic Spans Detection

Vaibhav Jain
New Delhi, India
vaibhav29498@gmail.com

Mina Naghshnejad
Oakland, CA, USA
mina.naghshnejad@gmail.com

Abstract

Detection of toxic spans - detecting toxicity of contents in the granularity of tokens - is crucial for effective moderation of online discussions. The baseline approach for this problem using the transformer model is to add a token classification head to the language model and fine-tune the layers with the token labeled dataset. One of the limitations of such a baseline approach is the scarcity of labeled data. To improve the results, We studied leveraging existing public datasets for a related but different task of entire comment/sentence classification. We propose two approaches: the first approach fine-tunes transformer models that are pre-trained on sentence classification samples. In the second approach, we perform weak supervision with soft attention to learn token level labels from sentence labels. Our experiments show improvements in the F1 score over the baseline approach. The implementation has been released publicly.[1]

1 Introduction

The growth of social media platforms has led to an increase in hate speech and abusive language in online communities, primarily due to the anonymity provided on such platforms (Mollas et al., 2020). Since manual moderation is not feasible for the gigantic amount of textual data, automated toxicity detection has received significant attention with numerous datasets being released in recent years (Pavlopoulos et al., 2020). However, most of the existing work on toxicity detection labels the entire comment as toxic or non-toxic and does not provide information about which specific part of the comment is toxic. In practice, human moderators (e.g., news portals moderators) can benefit from information on which character indices of the part of the comment that is toxic instead of just a system-generated unexplained toxicity score per post. Designing models that can accurately locate toxic spans within a text is thus a crucial step towards successful semi-automated moderation. This is challenging because of the scarcity of datasets that are labeled on a segment or token level.

This paper explains our approaches for the SemEval-2021 Task 5 which requires us to identify the character offsets for the toxic spans within a comment (Pavlopoulos et al., 2021). We explore possible techniques for improving the results of the vanilla transformer model by leveraging several available public datasets.

2 Related Work and Background

Sequential adaptation Using pre-trained language models has recently proved to be effective for language understanding tasks (Phang et al., 2018). The supplementary training is particularly beneficial when labeled data is scarce (Phang et al., 2018). A popular transfer learning technique in Natural Language Processing (NLP) has been to pre-train sentence encoder neural networks, such as BERT (Devlin et al., 2019), on unsupervised tasks and then fine-tune the encoders for the target supervised learning task. However, this approach can be found inadequate if the input distribution for the target task is considerably different from that of the corpus used for pre-training. Phang et al. (2019) suggested that training on related data-rich supervised tasks as an intermediate step can help in making the final trained model more robust and effective. This approach is called *Supplementary Training on Intermediate Labeled-data Tasks*. This can also help the model in learning the domain knowledge when in-domain data is available.

Weakly Supervised Learning Zhou (2018) defined *inexact* supervision as a type of weakly super-

[1] https://github.com/vaibhav29498/Toxic-Spans-Detection

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 935–940
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

vised learning in which coarse-grained labels are used to train for a more specific problem. Rei and Søgaard (2018) used this approach for inferring token-level labels by training a sentence classification model. They used a bidirectional LSTM to get a contextual representation $\tilde{h}_i \in \mathbb{R}^m$ for every token w_i, and pass it through a fully-connected layer with hyperbolic tangent activation to obtain $h_i \in \mathbb{R}^n$:

$$h_i = tanh(W_h \tilde{h}_i + b_h) \tag{1}$$

where $W_i \in \mathbb{R}^{n \times m}$ and $b_i \in \mathbb{R}^n$.

They then used a two-layer feed forward neural network followed by a soft-attention layer to get a single-valued attention score a_i for each token:

$$e_i = tanh(W_c h_i + b_c) \tag{2}$$

$$\tilde{e}_i = W_{\tilde{c}} e_i + b_{\tilde{c}} \tag{3}$$

$$\tilde{a}_i = \sigma(\tilde{e}_i) \tag{4}$$

where $W_c \in \mathbb{R}^{p \times n}$, $b_c \in \mathbb{R}^p$, $W_{\tilde{c}} \in \mathbb{R}^{1 \times p}$, $b_{\tilde{c}} \in \mathbb{R}$, and $\tilde{a}_i$ is the normalized attention score. These scores indicate the importance of the tokens towards predicting the sentence class and can be considered as the token-level predictions. Their normalized forms are used for constructing a weighted average sentence representation c, which is used to compute the prediction score y with a value higher than 0.5 indicating a positive class:

$$a_i = \frac{\tilde{a}_i}{\sum_{k=1}^{N} \tilde{a}_k} \tag{5}$$

$$c = \sum_{i=1}^{N} a_i h_i \tag{6}$$

$$d = tanh(W_d c + b_d) \tag{7}$$

$$y = \sigma(W_y d + b_y) \tag{8}$$

where $W_d \in \mathbb{R}^{q \times n}$, $b_c \in \mathbb{R}^q$, $W_y \in \mathbb{R}^{1 \times q}$, and $b_y \in \mathbb{R}$. The authors used a modified loss function to ensure that the model learns high-quality token labels:

$$L = \sum_j [(y^{(j)} - \tilde{y}^{(j)})^2 + \lambda(min_i(\tilde{a}_i)^2 + (max_i(\tilde{a}_i) - \tilde{y}^{(j)})^2)] \tag{9}$$

where $y^{(j)}$ is the predicted score, $\tilde{y}^{(j)}$ is the ground-truth, $min_i(\tilde{a}_i)$ and $max_i(\tilde{a}_i)$ are the lowest and highest attention scores respectively for the j^{th} sentence.

Karamanolakis et al. (2019) used a modified version of this approach to generate segment labels for text review classification problems. The segment embeddings are generated by feeding the word embeddings into a convolutional neural network. The output of the convolution neural network is passed through a single layer with softmax activation to get the segment-level prediction. Additionally, the word embeddings are passed through a bidirectional GRU network with sigmoid attention to find the attention weights. These weights are used to aggregate the segment-level predictions into a single prediction for the entire review.

3 Methodology and Experimental Setup

In this section, we will present our two solutions. Both of our approaches are using BERT pre-trained language model. In the first approach, we first fine-tune the BERT model with additional labeled data explained in Subsection 3.1 and then perform another round of fine-tuning using token classification head with task training data. In the second approach, we apply weak supervision to learn token labels for the additional dataset. We then use the augmented labeled token dataset to fine-tune the token classifier head and use it to predict labels for the task test data set.

3.1 Datasets

As the main data source for training our models, we used SemEval-2021 Task 5 data. Additionally, we used five publicly available English-language datasets for sentence classification to improve our results.

3.1.1 Token-level Labelled Data

We used two token-level labeled datasets, the first being the one provided by the SemEval-2021 Task 5 organizers which are composed of 9,939 English posts along with their toxic spans. The span for a single post is a possible-empty list of character indices that have been marked as toxic by crowd-annotators. The dataset has been divided into training and test sets with 2,000 samples in the latter.

The second dataset used is HateXplain (Mathew et al., 2020) which is composed of 20,148 tweets and Gab posts, each of which is classified as hateful, offensive, or normal, by three annotators. If an annotator masks a post as either hateful or offensive, they are also asked to mark the span (*rationale*) which influenced their decision. We considered a token to be toxic if it was included in at least one annotator's rationale, and excluded all posts with no toxic token, which left us with 11,415 posts.

3.1.2 Sentence-level Labelled Data

We used five other datasets which consisted of only sentence-level labels:

- Jigsaw/Conversation AI toxic comment classification challenge dataset[2] - Composed of Wikipedia's talk page edits, it labels 223,549 posts into zero or more categories of toxicity (toxic, severely toxic, obscene, threat, insult, and identity hate). We bundled each of them into a single category, and 22,468 posts were categorized as toxic.
- Hate speech and offensive language dataset (Davidson et al., 2017) - 24,783 tweets categorized to hate speech, offensive language, or neither. We labeled 20,620 belonging to the former two categories as toxic.
- Online harassment dataset (Golbeck et al., 2017) - Composed of 20,360 tweets out of which 5,285 have been labeled as harassing.
- Impermium dataset for detecting insults in social commentary[3] - 6,594 comments from online forums, out of which 1,742 were identified as insulting to at least one of the participants.
- OffensEval-2020 subtask-A extended test dataset - 5,993 tweets out of which 3,002 have been labeled as offensive.

Merging these five datasets resulted in a collection of 281,279 comments out of which 53,117 were labeled as toxic.

3.2 Approach 1: Sequential fine tuning

In approach 1, we first fine-tune the sentence classification model with additional data. The sentence classification model is built upon the base uncased BERT model, which maps every sentence to a vector of size 768. A two-layer sentence classifier is added to the BERT base model. The first layer of the head is a linear layer with leaky ReLU activation (negative slope of 0.1) that maps it to a vector of size 64. The second layer of the head is a dense layer that maps 64 nodes to a single-valued prediction score with sigmoid activation. The model is shown in Figure 1.

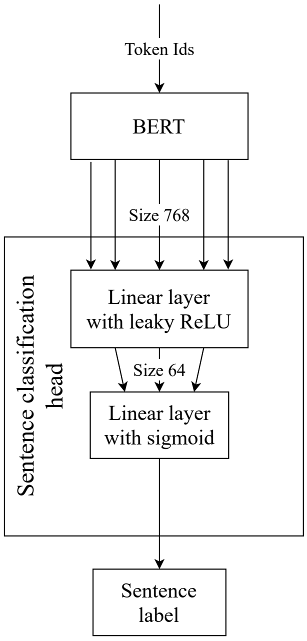
Token Ids
BERT
Size 768
Sentence classification head
Linear layer with leaky ReLU
Size 64
Linear layer with sigmoid
Sentence label

Figure 1: Model for sentence classification

We used the pre-trained BERT language model as the starting point for the BERT model. The layers of sentence classification head were randomly initialized. We applied the WordPiece tokenizer and added the special BERT tokens *[CLS]* and *[SEP]* to each sentence. We removed 5,136 samples as their number of tokens exceeded the base BERT model's maximum limit (512). The model was trained for a single epoch using the AdamW optimizer (Loshchilov and Hutter, 2019) with a batch size of 16.

The fine-tuned model was used to predict token classes. To predict token classes, a token classification was used. The token classification head architecture is similar to the sentence classification head with two linear nodes. The first layer maps 768-dimensional inputs to 64 nodes and applies the ReLu activation function with the negative slope of 0.1. The second layer maps input to a single token prediction score with a sigmoid activation function.

[2] www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data

[3] www.kaggle.com/c/detecting-insults-in-social-commentary/data

3.3 Approach 2 : Augmenting token labels with weak supervision

In our second approach, we first augment our training set with imperfect labels that we learn by weak supervision. We then use the augmented training set to fine-tune the token classifier model and use it to predict the labels of our test data. The model used for augmenting token labels using sentence-level labeled dataset described in Section 3.1.2, is loosely based upon the solution proposed by Karamanolakis et al. (2019). The model is shown in Figure 2. We first obtain the initial token embeddings from a pre-trained BERT model and input it to the token augmentation network shown in Fig 2. The input is connected on one side to token labels as well as to sentence-level predictions through two BiGRU layers. The GRU layers generate an attention weight for every token. These weights are used for finding a weighted average of the token-level predictions, which is used to calculate the prediction for the entire sentence.

Equation 9 is used as the loss function to train the weights of the model. The sentence labels are calculated from token labels following equations 5 to 8. The loss function optimizes the sentence and token labels. first, it makes sure sentence classifications are closest to the sentence labels. Second, it optimizes token labels by considering the minimum values of token labels in each sentence. It makes sure the minimum label of the tokens in a single sentence is zero to make sure all tokens in a sentence do not have a positive sentiment. It makes sure the most toxic token in the sentence has the same label as the label for the sentence. We trained the model for five epochs using the AdamW optimizer with a batch size of 16.

The token labels generated by the model described above are used as additional training data. The BERT base model is initialized with the publicly available pre-trained version. Only those artificially-generated samples in the sentences with the correctly predicted label (prediction score $\geq$ 0.5) are used. We used the prediction scores as the labels for the additional dataset instead of having discrete values of zero and one based on some threshold. We also subtracted the value of 0.1898 from these labels to make their mean equal to that of the labels of the contest dataset.

For generating the character offsets, we considered an entire word as toxic if any single of its subword tokens were identified as toxic.

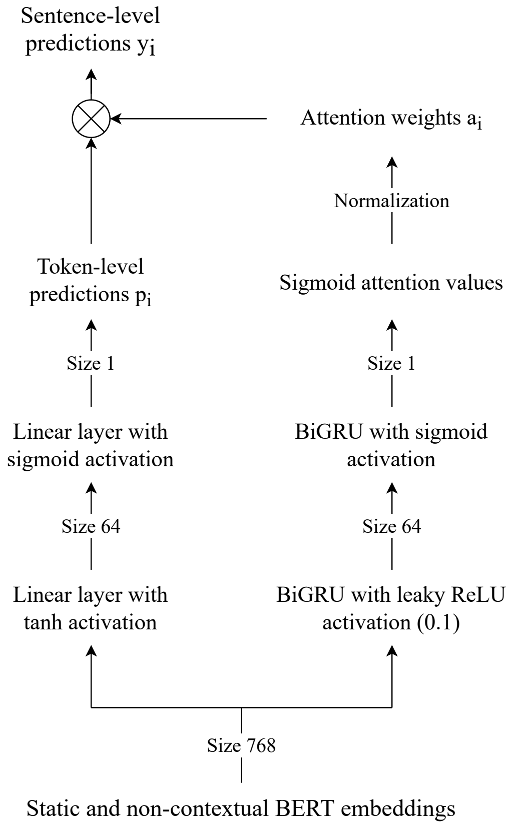
Sentence-level predictions y_i
Attention weights a_i
Normalization
Token-level predictions p_i
Sigmoid attention values
Size 1
Size 1
Linear layer with sigmoid activation
BiGRU with sigmoid activation
Size 64
Size 64
Linear layer with tanh activation
BiGRU with leaky ReLU activation (0.1)
Size 768
Static and non-contextual BERT embeddings

Figure 2: Model for generating token labels

4 Results

Disclaimer: The section contains offensive, obscene, and hateful content; however this is necessary to showcase the results of this work.

The contest organizers used the average F1 score as the performance metric. For a sample S_i, if $\tilde{Y}_i$ and Y_i are the predicted and actual set of character offsets, then the F1 score F_1^i is calculated as

$$F_1^i = \frac{2 \cdot P^i(\tilde{Y}_i, Y_i) \cdot R^i(\tilde{Y}_i, Y_i)}{P^i(\tilde{Y}_i, Y_i) + R^i(\tilde{Y}_i, Y_i)} \quad (10)$$

$$P^i(\tilde{Y}_i, Y_i) = \frac{|\tilde{Y}_i \cap Y_i|}{|\tilde{Y}_i|} \quad (11)$$

$$R^i(\tilde{Y}_i, Y_i) = \frac{|\tilde{Y}_i \cap Y_i|}{|Y_i|} \quad (12)$$

Our best-scoring solution submitted to the contest achieved an F1-Score of 0.6561 and a rank of 49 amongst 91 submissions. After the contest, we excluded the HateXplain dataset from the token-classification training process to make the model more suited for the contest dataset and masked out the padding tokens while calculating $min_i(\tilde{a}_i)^2$ in Equation 9. We also developed a baseline model which does not use any additional datasets by training the pre-trained base BERT model with the token-level labeled dataset described in Section 3.1.1. The training hyper-parameters were the same as our other approaches. The F1 scores for the baseline model and the two approaches discussed in Sections 3.2 and 3.3, namely the *sequential adaptation* and the *weakly supervised learning* approaches have been plotted for various thresholds in Figure 3.

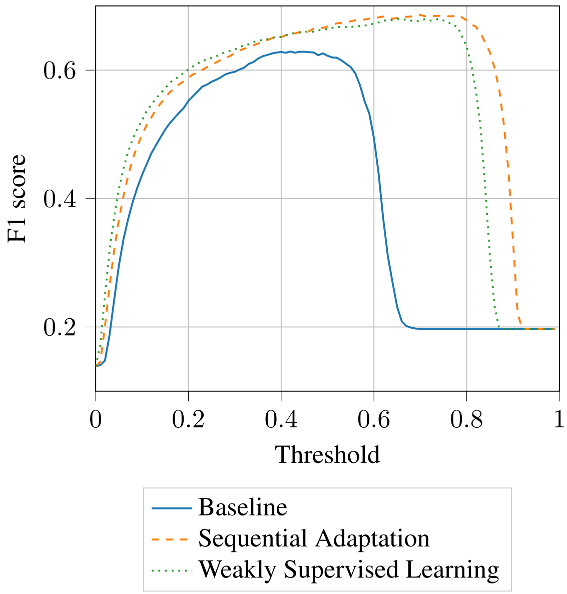

Figure 3: F1 score plotted against threshold

The F1 score at threshold value 0.5 and the maximum F1 score for each approach have been mentioned in Table 1.

Approach	$F1_{0.5}$	$F1_{max}$
Baseline model	0.6224	0.6289
Weakly Supervised	0.6644	0.6799
Sequential adaptation	0.6678	0.6861

Table 1: F1 score metrics for various approaches

Using sentence-level labeled data to incorporate domain-specific knowledge improves the results by a considerable margin. The increase in performance over the baseline model is more profound for the higher threshold value. This can be attributed to the comparatively higher toxicity of the sentence-level labeled datasets when compared with the contest dataset, which makes the models more expert on detecting highly toxic spans. We observe that the sequential adaptation approach outperforms the weakly supervised learning approach. However the latter might be more suitable for large-scale datasets due to less training time: the sentence classification model has to be trained on the entire sentence-labeled dataset in which the majority of the samples are non-toxic, whereas the model for generating token labels is trained only on toxic samples. In our experiments, the former took 4.25 hours for a single epoch of training and the latter took 1.5 hours for five epochs of training. Even though training the token classification model takes more time in the weakly supervised approach due to larger volume of data samples, the training time difference was only 48 minutes. All the models were trained on the Kaggle Notebooks[4] platform using GPUs.

The model for generating token labels (described in Section 3.3) did not perform as we expected. Out of the 52,640 toxic samples, only 7,629 samples were given a toxicity score of more than 0.5, thus greatly reducing the size of the additional dataset for token classification. However it was successful in correctly identifying the toxic spans. For example, the highlighted parts were given a toxicity score of more than 0.9 in the following sample:

> admins **suck!!** the **fucking** admins **suck ass!** i **fucking hate** the people who delete my **fucking shit!!!!!!!!**[5]

5 Conclusion and Future Work

In this paper, we proposed two solutions to improve the baseline transformer model fine-tuning for the span toxicity detection task. In the first approach, we performed sequential fine-tuning with an additional fine-tuning of the sentence classifier with supplemental public data. In the second approach, we augmented the labeled token dataset with weak supervision and then performed the fine-tuning token classification on the augmented dataset. Our experimental results show that the first approach improves the baseline fine-tuning results by a 0.0572 F1 score and the second approach improves the baseline results by a 0.051 F1 score. An interesting future direction is to improve the weak supervision technique - possibly using other objective functions for relating token labels and sentence labels - and multi-task learning.

[4] https://www.kaggle.com/code

[5] This sample is from the Jigsaw toxic comment classification challenge dataset (id `13913f443da71ac6`) and was labelled as *toxic* and *insult*.

Acknowledgements

We would like to thank the task organizers for preparing the dataset and organizing the competition, Jennifer Golbeck and her team at the University of Maryland, College Park for providing access to the online harassment dataset, and the anonymous reviewers for their valuable feedback.

References

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515.

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

Jennifer Golbeck, Zahra Ashktorab, Rashad O. Banjo, Alexandra Berlinger, Siddharth Bhagwan, Cody Buntain, Paul Cheakalos, Alicia A. Geller, Quint Gergory, Rajesh Kumar Gnanasekaran, Raja Rajan Gunasekaran, Kelly M. Hoffman, Jenny Hottle, Vichita Jienjitlert, Shivika Khare, Ryan Lau, Marianna J. Martindale, Shalmali Naik, Heather L. Nixon, Piyush Ramachandran, Kristine M. Rogers, Lisa Rogers, Meghna Sardana Sarin, Gaurav Shahane, Jayanee Thanki, Priyanka Vengataraman, Zijian Wan, and Derek Michael Wu. 2017. A large labeled corpus for online harassment research. In *Proceedings of the 2017 ACM on Web Science Conference*, WebSci '17, page 229–233, New York, NY, USA. Association for Computing Machinery.

Giannis Karamanolakis, Daniel Hsu, and Luis Gravano. 2019. Weakly supervised attention networks for fine-grained opinion mining and public health. In *Proceedings of the 5th Workshop on Noisy User-generated Text, W-NUT@EMNLP 2019, Hong Kong, China, November 4, 2019*, pages 1–10. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2020. Hatexplain: A benchmark dataset for explainable hate speech detection. *arXiv preprint arXiv:2012.10289*.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2020. Ethos: an online hate speech detection dataset. *ArXiv*, abs/2006.08328.

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*.

John Pavlopoulos, Jeffrey Sorensen, Lucas Dixon, Nithum Thain, and Ion Androutsopoulos. 2020. Toxicity detection: Does context really matter? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4296–4305, Online. Association for Computational Linguistics.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

Jason Phang, Thibault Févry, and Samuel R. Bowman. 2019. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks.

Marek Rei and Anders Søgaard. 2018. Zero-shot sequence labeling: Transferring knowledge from sentences to tokens. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 293–302, New Orleans, Louisiana. Association for Computational Linguistics.

Zhi-Hua Zhou. 2018. A brief introduction to weakly supervised learning. *National Science Review*, 5:44–53.