Lone Pine at SemEval-2021 Task 5: Fine-Grained Detection of Hate Speech Using BERToxic

Yakoob Khan, Weicheng Ma, Soroush Vosoughi
Department of Computer Science, Dartmouth College
{yakoob.khan.21,weicheng.ma.gr,soroush.vosoughi}@dartmouth.edu

Abstract

This paper describes our approach to the *Toxic Spans Detection* problem (SemEval-2021 Task 5). We propose **BERToxic**, a system that fine-tunes a pre-trained BERT model to locate toxic text spans in a given text and utilizes additional post-processing steps to refine the boundaries. The post-processing steps involve (1) labeling character offsets between consecutive toxic tokens as toxic and (2) assigning a toxic label to words that have at least one token labeled as toxic. Through experiments, we show that these two post-processing steps improve the performance of our model by 4.16% on the test set. We also studied the effects of data augmentation and ensemble modeling strategies on our system. Our system significantly outperformed the provided baseline and achieved an F1-score of 0.683, placing Lone Pine in the 17th place out of 91 teams in the competition. Our code is made available at https://github.com/Yakoob-Khan/Toxic-Spans-Detection

1 Introduction

The promotion of respectful discourse has always been a core tenet of civilized societies. The Cambridge dictionary defines hate speech as "public speech that expresses hate or encourages violence towards a person or group based on something such as race, religion, sex, or sexual orientation." Online platforms enable malicious actors to hide behind a cloak of anonymity and surreptitiously post toxic comments that are a "menace to democratic values, social stability and peace" (United Nations). To combat this problem, such platforms often employ human moderators to address offensive content that goes against community standards. However, moderators are unable to manually keep pace with the large volume of user-generated content today. This motivates the development of natural language processing systems to automatically detect hate speech and ensure that online platforms remain healthy and inclusive for all.

There has been extensive research on hate speech detection, with the creation of large datasets (Wulczyn et al., 2017) and the use of pre-trained text representations (Devlin et al., 2019) for varied modeling approaches. Competitions on offensive language identification (Zampieri et al., 2020) have further attracted attention to this topic. Prior work has hitherto focused on classification at the document-level based on various taxonomies, such as whether a given text contains offensive language or if it is targeted towards an individual or group. This line of inquiry does not identify the toxic spans that ascribe a text as hate speech. Doing so will assist human moderators to efficiently locate offensive content in long posts and elucidate further insight into hate speech explainability.

This motivated the *Toxic Spans Detection* task (Pavlopoulos et al., 2021) where systems are asked to extract the list of toxic spans that attribute to a text's toxicity. Consider the following example[1]:

Text	Because he's a **moron** and **bigot**. It's not any more complicated than that.
Span	[15, 16, 17, 18, 19, 27, 28, 29, 30, 31]

Table 1: A sample example from the task dataset.

As there are two toxic spans in the above text, systems are tasked to extract the character offsets (zero-indexed) corresponding to the sequence of toxic words. This is a challenging task as classification at the word-level is inherently more difficult than at the document-level. The intentional obfuscation of toxic words, use of sarcasm and the

[1]We caution readers that the examples included in this work contain explicit language to illustrate the severity and challenges of hate speech detection.

subjective nature of hate speech further adds complexity to the problem.

Our contributions to the task is threefold:

1. We propose **BERToxic**, a system that fine-tunes a pre-trained BERT model with additional post-processing steps to achieve an F1-score of 0.683, placing Lone Pine in the 17th place out of 91 teams in the competition.

2. We study the effects of simple data augmentation strategies on our system and find that they yield no improvement in classification performance.

3. We examine late fusion and multi-task learning neural architectures and conclude that they under-perform compared to the standalone BERT model for this task.

2 Our Approach

2.1 Baselines

To have a better sense of our final system's performance, we initially examined two baseline models. First, we created a trivial model that randomly predicts each character offset of a text as toxic if its $\rho > 0.5$, drawn from a continuous uniform probability distribution.

To have a stronger baseline model, we fine-tuned the off-the-shelf spaCy NER model provided by the task organizers. This model consists of a multi-hash embedding layer (feed-forward sub-network) that uses sub-word features and an encoding layer consisting of a CNN and a layer-normalized maxout activation function. The model uses a transition-based algorithm that assumes that the "most decisive information" regarding the entities "will be close to their initial tokens", with a loss function that optimizes for whole-entity accuracy.

2.2 BERToxic

We framed the toxic spans detection task as a sequence labeling problem and leverage the **B**idirectional **E**ncoder **R**epresentation from **T**ransformers (Devlin et al., 2019) model to extract rich feature representations from the input texts.

The first step in the BERToxic system pipeline (Figure 1) was to tokenize the text inputs and generate the word embeddings using BERT's WordPiece tokenizer. This sub-word tokenization algorithm (Schuster and Nakajima, 2012) tokenizes a word like `"moron"` into `["mo","##ron"]` and we

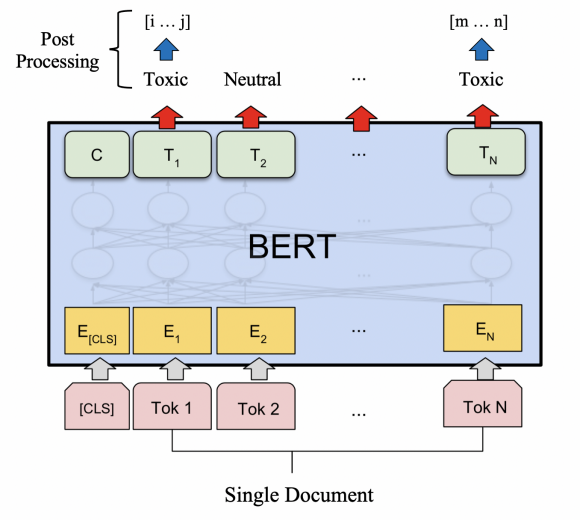

Figure 1: The BERToxic model architecture. Image modified from (Devlin et al., 2019).

ensured that the ground truth labels were preserved across all tokens of a word. As BERT uses absolute position embeddings, we padded shorter sequences with `[PAD]` tokens on the right side such that all tensor inputs are set to equal the maximum sequence length observed for batched parallelized training. Long sequences were truncated to 512 tokens, the maximum sequence length allowed by BERT. As the data was obtained from online comments that are generally shorter in nature, the truncation procedure was not needed in this task but nevertheless served to handle long sequences if present. We also stored the mapping

$$\mathcal{M} : t_i \mapsto (start_i, end_i)$$

of each token to its relative character offsets in the original string, used for outputting the toxic span predictions at the post-processing stage.

We performed all of our experiments using the $\text{BERT}_{\text{BASE}}$ model architecture that consisted of 12 layers, 768 hidden size, 12 self-attention heads and $109M$ parameters. The $\text{BERT}_{\text{LARGE}}$ model was not explored in this work due to its compute-intensive nature. Our intuition suggested that letter casing could be helpful for this task as proper nouns (e.g *Muslim*) can be used offensively, so we selected the cased model for our experiments. A token classification head containing a linear layer was applied on top of the final hidden-states output, with a label prediction of 1 denoting a toxic token, 0 otherwise. For each token t_i labeled as toxic, we utilized $\mathcal{M}$ to output all character indices in the range of $(start_i, end_i)$ inclusive as the toxic span

of this token.

Additionally, our system performed two post-processing steps to refine the boundary predictions. Consider the following tokenized sequence:

$$t_1, \cdots, t_i, t_{i+1}, t_{i+2}, \cdots, t_n$$

First, for any two consecutive tokens t_i and t_{i+1} whose prediction labels are toxic, we output the character indices in the range of $(end_i + 1, start_{i+1} - 1)$ inclusive as toxic as well. This had the effect of including the delimiter characters between consecutive toxic words, thereby detecting toxic phrases. Second, recall that BERT's WordPiece tokenizer could split a word into multiple tokens, say t_i, t_{i+1} and t_{i+2}. If at least one token was predicted toxic by the model, our system assigned a toxic label to all constituent tokens of this word. This achieved coherence in the prediction of toxic words and phrases, thus avoiding incomplete word piece issues.

We also attempted to vary the thresholds of the confidence scores before SoftMax for toxic token predictions but observed no improvement in performance.

2.3 Data Augmentation

Data augmentation is widely used to improve the generalization of models by acting as a regularizer to reduce overfitting. While various sophisticated techniques exist to artificially enhance the size and quality of the training set without collecting additional manually labeled examples, we chose to apply the set of **E**asy **D**ata **A**ugmentation (EDA) techniques (Wei and Zou, 2019) to generate synthetic training data for this task.

The four operations in EDA are Synonym Replacement (SR) using WordNet (Miller, 1995), Random Insertion (RI), Random Swap (RS) and Random Deletion (RD) of words in a document. Shorter documents are disproportionately more affected by these operations if a fixed number of words are modified per document. To ensure that all documents experienced the augmentation strength proportionately, the number of words n modified was varied based on the document length l using the formula $n = \alpha \cdot l$, where α is a hyperparameter that indicates the percentage of words changed per document. Each operation was applied once per document and care was taken to ensure that the ground truth labels were preserved.

Our experiments revealed that the recommended value of $\alpha = 0.1$ was too low for this task and we observed small but consistent improvements as α increases. Furthermore, we noted that the SR technique alone leads to better performance than using all four operations to create the augmented training set for this task.

We also attempted data augmentation using an external dataset, HateXplain (Mathew et al., 2020), that contains $20,148$ documents with word-level annotations that we processed to conform to this task's data format. Each document consisted of 2 - 3 annotations and we used their intersection to maximize the inter-annotator agreement in constructing the ground truth labels. HateXplain's annotation strategy appeared to be different and included labeling pronouns, conjunctions and stop words as toxic when located between offensive words. We removed such toxic labels so that the external dataset annotation was more similar to this task. When our task dataset was augmented with the full external dataset, the model experienced underfitting, while removing all the non-toxic labeled documents from the external dataset alleviated the issue to some extent.

2.4 Ensemble Modeling

Ensemble modeling is an approach where multiple different models are trained and their predictions are aggregated. By adding bias to counter the variance of a single model, this line of work has been shown to improve the predictive performance of a system (Liu et al., 2019). While numerous ensemble modeling techniques like boosting, bagging, etc. exist, we investigated two techniques of interest: late fusion and multi-task learning.

We reframed the problem as a binary classification task and trained a sequence classifier to predict whether a given sentence is toxic. In the late fusion approach, we utilized NLTK's tokenizer to split each document into sentences. If a sentence contained a ground truth toxic span, we assigned the toxic class label 1, 0 otherwise. In this way, a binary classification dataset was created to separately fine-tune a pre-trained BERT sequence classifier. We hypothesized that token labels should be predicted toxic only if the corresponding sentence was classified as toxic as well. Late fusion was performed at the prediction phase, where both the sequence and token classifiers voted in the predictions by having the former model filter toxic sentences on which the latter model made final toxic span predictions.

Rather than fine-tuning the two models separately, we also investigated if multi-task learning (MTL) improved the predictive performance of the ensemble model. We hypothesized that a training regime where the two classifiers were learned jointly could be useful as the knowledge gained in learning one task could benefit the other. To perform MTL, we fine-tuned an MT-DNN (Liu et al., 2019) model where the text encoding lower BERT layers are shared across the two tasks while the top layers are task-specific.

3 Experiments

The following sections describe the experimental set-up of our work.

3.1 Dataset

The task data was sourced from the Civil Comments dataset (Borkan et al., 2019), which contains public comments made between 2015 - 2017 that appeared on approximately 50 English-language news sites across the world. As the original dataset contained only document-level class labels, the task organizers selected a subset of the data for crowd-sourced toxic spans annotation. For the data split, we chose to fine-tune our models using the entire provided training dataset ($N = 7939$) to maximize performance, validate using the trial dataset ($N = 690$), and submit our predictions using the test data ($N = 2000$). The test labels were withheld during the evaluation phase of the competition and were only released afterward.

3.2 Evaluation Metric

To evaluate the performance of the models, the task organizers employed a variant of the F1-score (Da San Martino et al., 2019). For a document d, define S_d as the set of toxic character offsets predicted by a system and G_d as the set of ground truth annotations. Then the F1-score of the system with respect to ground truth G for d is defined as

$$ {F_1}^d(G) = \frac{2 \cdot P^d(G) \cdot R^d(G)}{P^d(G) + R^d(G)} $$

where

$$ P^d(G) = \frac{|S_d \cap G_d|}{|S_d|} $$

$$ R^d(G) = \frac{|S_d \cap G_d|}{|G_d|} $$

If a document has no ground truth annotation ($G_d = \emptyset$), or the system outputs no character offset prediction ($S_d = \emptyset$), we set

$$ {F_1}^d(G) = \begin{cases} 1 & G_d = S_d = \emptyset \\ 0 & otherwise \end{cases} $$

We finally take the arithmetic mean of ${F_1}^d(G)$ over all the documents of an evaluation dataset to obtain a single F1-score for the system.

3.3 Implementation Details

We utilized the PyTorch framework for the development of our system, HuggingFace's transformers library (Wolf et al., 2020) for the BERT-based models and Microsoft's implementation of the MT-DNN model. All models were trained on Google Colab Pro's High-RAM environment using a single NVIDIA P100 GPU. The training policy used the following hyper-parameters: batch size of 16, sequence length of 512, weight decay of 0.01. For optimization, we used Adam with a learning rate of 5e-5 and a linear warm-up schedule over 500 steps. Except for MT-DNN[2], all our models were fine-tuned for approximately 2 epochs and we practiced early stopping by monitoring the dev F1-score to reduce overfitting. The EDA experiment was performed with $\alpha = 0.8$ using only the SR technique. All other hyper-parameters were set to their default values according to HuggingFace's implementation. We set a random seed for all our experiments and open-sourced the code for reproducibility.

4 Results

On the following page, Figure 2 visualizes the precision-recall curves[3] of all the models and Table 2 summarizes their performance metrics. Figure 3 shows the confusion matrix of our best performing BERToxic model and Table 3 highlights selected predictions that it made.

An interesting observation we noted from Table 2 was that the F1 scores for the test set were higher than the dev set for many of the models. We hypothesize that this is because the models have an inductive bias to predict shorter toxic spans, evidenced by the average ground truth span length of 7.2 in the test set and 14.7 in the dev set.

[2]MT-DNN was fined-tuned for 3 epochs with a batch size of 8.

[3]The curves for the spaCy and BERT multi-task model are less detailed due to the ambiguity in obtaining the probability scores from their respective implementations, necessitating the use of their predicted labels instead.

Model	Dev			Test		
	Precision	Recall	F1	Precision	Recall	F1
Random	0.143	0.463	0.175	0.089	0.413	0.122
SpaCy	0.692	0.588	0.595	0.664	0.686	0.656
BERToxic	0.781	0.678	0.681	0.683	0.732	**0.683**
+ EDA	0.787	0.683	0.684	0.681	0.725	0.678
+ HateXplain	0.792	0.674	0.681	0.683	0.721	0.678
BERT late fusion	0.733	0.636	0.639	0.675	0.709	0.669
BERT multi-task	0.744	0.629	0.634	0.665	0.694	0.656

Table 2: A summary of the performance of all our models, reporting the precision and recall scores along with the F1 evaluation metric used for the competition. The BERToxic model outperformed the strong spaCy baseline by 4.16% on the test set, placing Lone Pine in the 17th place out of 91 teams. In comparison, the top-ranked submission achieved an F1-score of 0.708. The experiments revealed that our data augmentation and ensemble modeling strategies did not outperform the standalone BERT model.

Our proposed system performed well at the toxic spans detection task, showing strength in identifying profanity and common toxic words like "idiot" and "stupid". The model identified the obfuscation of offensive words and successfully detected hate speech from such adversarial cases (Example 1).

1.	Kill this *F'n W*ore* on site.
2.	.. how I am an ignorant fool ..
3.	Nazi boneheads deserve being punched.
4.	@ remoore Shut up, *racist*.
5.	*Cruz is a piece of garbage a globalist fraud*

Table 3: Selected examples obtained from the test set. BERToxic's predictions are shown in red while ground truth annotations are *italicized*.

The error analysis revealed that the system lacked nuance as it would sometimes classify toxic words used in neutral contexts (Example 2). It is also worth mentioning that there was considerable noise in the ground truth annotations. Our manual inspections concurred with the model's predictions that some words and phrases were used in offensive contexts but the annotators thought they were neutral (Example 3 and 4). Furthermore, we observed some inconsistencies in the labeling scheme as some annotations spanned entire sentences (Example 5) while others only highlighted a few words in the sentence. These issues point to the subjective nature of hate speech and the challenges involved in its fine-grained classification.

We found through our ablation studies of data augmentation that generating synthetic data using

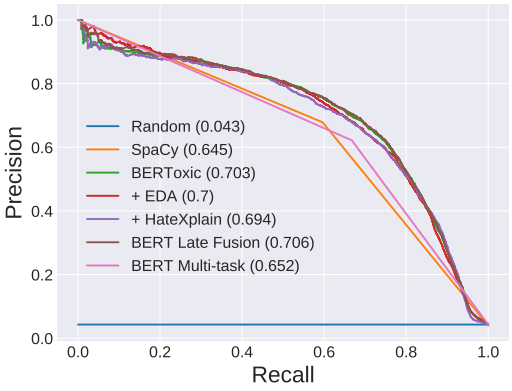

Figure 2: Comparison of the precision-recall curves of all the models at the token level on the test set. The area under the curve is enclosed within parentheses.

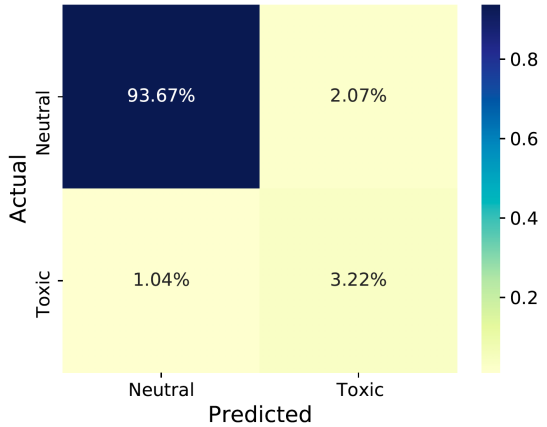

Figure 3: A confusion matrix of the BERToxic system at the token level, revealing insights about the classification performance in each category and highlighting the imbalance of the class labels in the test set.

the EDA techniques did not improve the performance of the system. This suggested that the dataset size does not appear to be the limiting factor affecting the performance of BERT in this task. Using HateXplain's external dataset, we learned that different data sources and annotation guidelines can introduce noise that hurts the performance of models.

Finally, the ensemble modeling strategies we explored did not outperform the standalone BERT model. The late fusion technique performed slightly better than the spaCy baseline, but it seemed that the sequence classifier made errors on similar parts of the input space as the token classifier. The multi-task learning approach underperformed compared to late fusion, suggesting that the sequence labeling and classification tasks are not closely related enough to benefit their joint training.

5 Conclusion and Future Work

In this work, we have proposed BERToxic, an empirically powerful system that performed fine-grained detection of hate speech. We found that our exploration of data augmentation and ensemble modeling strategies did not outperform the standalone model. The error analysis revealed that BERT lacked nuance in understanding the use of offensive words in neutral contexts and encountered boundary detection issues when faced with noisy ground truth annotations.

Future avenues of work could address these limitations and explore other transformer-based models to develop more robust hate speech detectors. We hope that our findings inspire more creative approaches towards fine-grained detection of hate speech so that online discourse can remain healthy and inclusive for all.

Acknowledgments

Yakoob Khan is thankful to be supported by the Stamps Scholarship funded by Dartmouth College and the Strive Foundation.

References

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.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, 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 5636–5646, Hong Kong, China. Association for Computational Linguistics.

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.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

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*.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

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*.

M. Schuster and K. Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6383–6389, Hong Kong, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1391–1399, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

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 the Fourteenth Workshop on Semantic Evaluation*, pages 1425–1447, Barcelona (online). International Committee for Computational Linguistics.