Project13: Toxic Spans Detection

Archit Bansal[1], **Abhay**[2], **Samyak Jain**[3]
[1]180134, [2]180014, [3]180661
[1]EE, [2]CSE, [3]CSE
{architb, kabhay, samyak}

Abstract

In this work, we present the approaches we have tried so far and our findings for the SemEval 2021 Task 5 - Toxic Spans Detection. The main aim of the task is to detect the spans that make a text toxic, on the dataset annotated from the toxic posts of Civil Comment dataset. We have discussed three different approaches to solve the problem and have analysed their results for the given task. Among the presented methods, the best model achieved an F1-score of 0.5 on the dev set. Furthermore, we discuss our future plans to build a more robust model and achieve higher a score.

1 Introduction

The internet user base has grown to over 4 billion and this boom has brought with it the issue of ensuring a safe environment for communication on the internet. The sheer amount of data being generated nowadays means that manual content moderation is not possible and therefore the focus has shifted to tackling the issue using machine learning methods.

Various toxicity detection datasets (Wulczyn et al., 2017; Borkan et al., 2019) and models (Pavlopoulos et al., 2017; Liu et al., 2019; Seganti et al., 2019) have been successfully developed over the years to tackle the issue of moderation. However, most of these research works focus on identifying whole comments or documents as either toxic or not, and in a semi-automated setting where the human moderators might have to deal with lengthy comments, a model generated unexplained toxicity score can be frustrating. In order to tackle this issue, the SemEval 2021 Task 5 : Toxic Spans Detection introduces the task of accurately identifying toxic spans in a post which can give the human moderators a lot more insight about what actually contributes to the toxicity of the text.

To address this task, we will be presenting three different approaches which include model agnostic methods for explaining text classifier results (Ribeiro et al., 2016; Li et al., 2017), word level Long Short Term Memory(LSTM) models trained on Glove embeddings and fine-tuned Bidirectional Encoder Representation from Transformer (BERT) (Devlin et al., 2019). The fine tuned BERT model currently performs the best on the validation data.

The rest of this paper is arranged as follows, section 2 formally introduces the problem statement, section 3 lists previous work that has been done in related to this task, section 4 contains a description of the dataset, section 5 introduces our proposed approaches in further details followed by our results and error analysis in sections 6 and 7 respectively. We conclude with the individual contributions, our future plan and a conclusion in sections 8, 9 and 10 respectively.

2 Problem Definition

The shared task under SemEval 2021 Task 5 : Toxic Spans Detection is to extract a list of toxic spans for each toxic text in the dataset. Here, by toxic spans the organizers are referring to a sequence of words that contribute to the text's toxicity. The systems are expected to return a list of the character offsets of each character in the detected spans (following 0 indexing) for each text. Therefore, the problem is clearly a span detection task which can also be approached as a sequence labelling task.

The metric used for evaluating the systems is the character level F1 score averaged over each post.

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

$$P^t(A_i, G) = \frac{|S^t_{A_i} S^t_G|}{S^t_{A_i}} \; R^t(A_i, G) = \frac{|S^t_{A_i} S^t_G|}{S^t_G}$$

Here, for the F1 score F_1^t of text t, $S^t_{A_i}$ refers to the set of character offsets returned by system A_i and G^t refers to the ground truth character offsets.

3 Related Work

As our task involves detection of toxic spans in a text, we present the related work in two parts: (i) Offensive Language detection and (ii) Span Detection. Due to the abundance of literature in both these areas, we focus our attention to those studies which we deem as pertinent to the current work.

Offensive Language Detection: Different abusive and offense language identification problems have been explored in the literature ranging from aggression to cyber bullying, hate speech, toxic comments, and offensive language. (Davidson et al., 2017) reported hate speech detection results using word n-grams and sentiment lexicon. Recent contributions of offensive language detection comes from the SemEval-2019 Task 6 OffensEval. The task was based on the OLID dataset ((Zampieri et al., 2019a)) and featured three sub tasks - (i) Identification (ii) Categorization, and (iii) Target identification. (Zampieri et al., 2019b) concluded that most of top performing teams either used BERT (Liu et al., 2019) or an ensemble model to achieved SOTA results for the corresponding subtask.

Span detection Span detection/identification tasks form a substantial part of applied NLP. It includes numerous tasks like named entity recognition NER (Nadeau and Sekine, 2007), chunking (Sang and Buchholz, 2000), keyphrase detection(Augenstein et al., 2017) , or quotation detection(Pareti, 2016). An abundance of model architectures have been implemented for these tasks including range of models like token classification models, probabilistic models, conditional random fields, recurrent neural networks and transfer learning techniques. (Papay et al., 2020) showed that presence of BERT component in the model is highest positive predictor for most of the span identification tasks since it is robust and largely independent of span or boundary distinctiveness effects.

4 Corpus/Data Description

For this dataset, the organizers have randomly exrtacted 10K comments from a pool of crowd-rated toxic comments derived originally from the Civil Comments Dataset. Each comment was then annotated by three crowd raters who extracted toxic spans from the text and the character offsets of toxic spans extracted by the majority were retained. The inter-annotator agreement was also calculated for a trial run of 35 posts using five crowd-raters. A Cohen's kappa score of 0.61 was achieved which indicates moderate consistency in the annotations.

The training data provided by the organizers consisted of a total of 7939 texts which we have further split into training, validation and test sets for evaluation purposes using a 80:10:10 split. After analysing the training split, we found that each example on an average consists of about 37 words and 206 characters. The longest post in the dataset contains 200 words in comparison to shortest single word example.

5 Proposed Approach

5.1 Preprocessing

Tokenization We first used a word level tokenizer to tokenize our data as we required a method to map the final token level output of our models to their character offsets while still maintaining the original form of the words for performing sub-word tokenization later on. We used the TreebankWord Tokenizer from NLTK library which is a rule based tokenizer that uses regex to tokenize text and also returns the offset span for each token.

Data Cleaning After tokenizing the text, we proceeded to perform some data cleaning operations on these tokens such as removing emojis (as they are not treated as toxic spans in any of the data samples), expanding contractions, lowercasing data and dealing with punctuation.

5.2 Methodology

Representation Erasure The first method that we tried out was using a model (Li et al., 2017) that can explain the output of a neural network at the word level. We decided to take the Bi-LSTM model as the base model which we trained upon a custom made subset of data from the Civil Comments Dataset for the task of classifying each text as toxic/non-toxic. A tokenized text sequence was fed to an embedding layer which used pre-trained Glove embeddings, the sequence was then passed through a Bi-LSTM layer which mapped each text into a feature vector which was fed to a fully connected layer with sigmoid activation for binary classification. We then used this model to compute the importance of each word as the relative change of the log-likelihood of the correct label for a text when a particular word is erased.

$$I(d) = \frac{S(e,c) - S(e,c,-d)}{S(e,c)}$$

Here $I(d)$ is the importance score of word d, $S(e,c)$

is the log-likelihood of text e for class c and $S(e, c, -d)$ if the log-likelihood for text e for class c with word d erased. The words which had an importance score above a set threshold were included in the toxic span for that text.

Bi-LSTM Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) is a powerful extension of recurrent neural networks. Our second approach was based on using a Bi-LSTM model for sequence labelling task, i.e. we classified each token in the text as toxic/non-toxic by using a linear head on top of the Bi-LSTM layer. Each text was first tokenized and fed to an embedding layer which uses pretrained Glove embeddings. These sequences were then passed to a Bi-LSTM layer which returned a hidden state vector for each token in the sequence, which were then passed through a fully connected layer with sigmoid activation function to perform binary classification on each token. The training was done using the Adam optimizer which was able to converge the Binary Cross Entropy loss after 10 epochs.

BERT Bidirectional Encoder Representation from Transformer (BERT) (Devlin et al., 2019) is a pretrained multihead transformer that has been able to achieve SOTA performance for many NLP tasks. As BERT has been trained on a large corpus of data, transfer learning using BERT leads to excellent results in spite of our small dataset. We use the BERT model for token classification task by employing a linear layer on top of the BERT model which takes the hidden layer output of each token and performs binary classification. As we are using a pretrained model, its vocabulary is already defined and hence as a first step we tokenize our preprocessed texts using the BertTokenizer before passing on the sequences to the model. The model was finetuned using the AdamW optimizer which uses weight decay as regularization, along with a linear scheduler to reduce the learning rate throughout the epochs. These optimizations helped the model converge in the first 2 epochs before overfitting.

6 Experiments and Results

Although we are expected to evaluate our model on the basis of average f1 score calculated from the character offsets of toxic spans, here we evaluate our model by calculating f1 score for predicted token labels. Improving the accuracy for token wise labels would essentially mean that we are predicting correct tokens that are toxic, which is further equivalent to detecting corresponding toxic spans. For our first method, we had to create a custom dataset of toxic/non-toxic texts for sequence level classification. We extracted around 100K data samples for each class from the Civil Comments Dataset from which our competition dataset has been extracted to ensure consistency in distribution. We then trained the Bi-LSTM models on this dataset for varying hyperparameter settings. The final results were computed on the validation split of our competition dataset and are listed in Table 1.

Pretrained Embedding	Bi-LSTM layers	Dropout Rate	Val F1 Score
glove	1	0.5	0.0915
glove	2	0.5	**0.0934**
glove*	2	0.5	0.0921
gtwitter	1	0.7	0.0926
gtwitter**	1	0.7	0.0918
gtwitter*	1	0.7	0.0919

Table 1: Here glove and gtwitter stands for glove.6B.300D and glove.twitter.27B.200D resp.
* - Trained on lowercase dataset
** - UNK token not initialized to zero

Pretrained Embedding	Bi-LSTM Layers	Hidden Dimension	Val F1 Score
NA	2	64	0.25
glove	2	64	0.32
gtwitter[1]	3	64	0.38
gtwitter	2	64	**0.413**
gtwitter	2	128	0.40

Table 2: **BiLSTM**: glove and gtwitter stands for glove.6B.300D and glove.twitter.27B.200D resp.

Next, we experimented on the token wise Bi-LSTM classifier with various configurations. We use Bi-LSTM with different word embeddings and variations in hyperparameters like no. of Bi-LSTM layers and hidden dimensions. The results on validation split are summarised in Table 2.

For the BERT token classification method, we finetuned three different pretrained BERT models (3). The BERT-Base-Uncased and BERT-Large-Uncased models were trained with the originally preprocessed dataset. For the BERT-Base-Cased model, while preprocessing the data we excluded the lowercasing and pucntuation removal steps.

Pretrained Model	Val F1 Score	Val Acc
BERT-Base-Uncased	0.492	92.34
BERT-Large-Uncased	0.488	**92.47**
BERT-Base-Cased	**0.501**	92.2

Table 3: BERT Finetune Results

7 Error Analysis

The results we have obtained till now have brought to light some problems that we will have to resolve. First of all, the data annotations are not uniform in what they mark as toxic spans. We have observed complete sentences being marked as toxic just because of the presence of a few toxic words in them. These irregularities in the annotations make it difficult for the model to generalize on the data.

Analyzing the performance of our models, two things became clear. First of all our token classification data is skewed and therefore we see that in spite of achieving a very high accuracy for the task, the F1 score is only around 0.5 for our top performing model as we see in Table 3. Another issue is that as we can see from the confusion matrices(Fig 1,2), the model puts more emphasis on correctly classifying the non-toxic tokens. We are planning to use class weights in the loss function to tackle this issue.

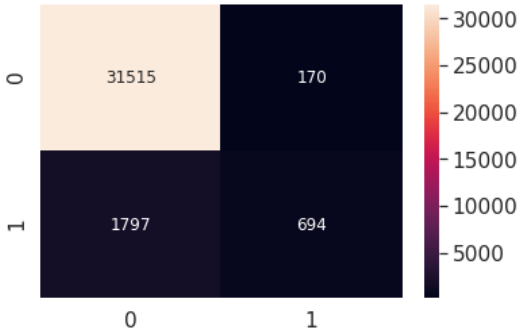

Figure 1: Confusion matrix for best BiLSTM for token wise classification model on validation set

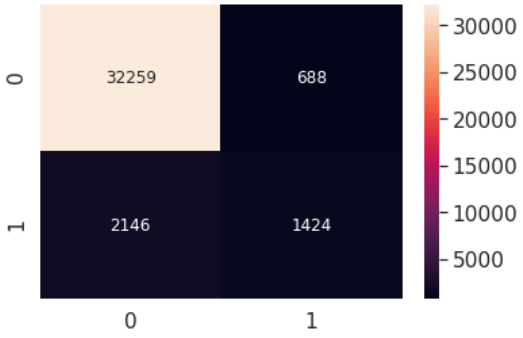

Figure 2: Confusion matrix for BERT-Base-Cased model on validation set

8 Individual Contribution

The contributions of each team member are summarised in Table 4.

Name	Contributions
Abhay	BiLSTM, Literature Review,Project Paper
Archit Bansal	Representation Erasure, BERT Literature Review, Project Paper
Samyak Jain	Dataset cleaning and Preprocessing Literature Review

Table 4: Memberwise Contributions

9 Future Work

To further improve upon the results we have achieved till now, we plan to try out a few different strategies before the final presentation. We are working on improving upon the baseline BERT results by using BPE tokenizer and trying out adding BiLSTM and CRF heads over it (Archit). We also expect to improve upon the results we got from BERT by using newer transformer models like RoBerta/XLNet (Abhay). We plan to complete both of these tasks before the end of the month. After that, we are looking to try out the idea of few shot learning since our dataset is small and is not consistent in itself. Apart from that, we are also looking at incorporating adversarial training as the data is pretty noisy and considering that the test dataset would be from the same distribution, training on adversarial examples would likely improve the robustness of our model. We plan to collectively complete these two tasks by 20 Nov. and then finish of the project by writing a paper in the last two weeks of the semester leading up to the final presentation in December.

10 Conclusion

The task of detecting toxic spans in text is a novel one and there is no doubt about how important successfully completing this task can turn out to be for online content moderation. The data gathered from online platforms tends to be noisy and corrupted as we can see from our dataset and therefore we require clever data preprocessing techniques to make our models perform better. Not only that, annotating spans in textual data is time consuming and therefore it is important to be able to extract as much as possible from the given data, and hence transfer learning approaches are important to achieve better performance. In light of these issues, other techniques like few-shot learning and adversarial loss training are also worth exploring.

11 Presentation Feedback

We were asked the following questions during our presentation.

1. The first question was a query about the evaluation metric we are using to report the current results on BERT model. As discussed in section 6, we are currently reporting our results from token level predictions.

2. The other question was about how are we keeping a check on over fitting. We are currently using model checkpoint method to check over fitting of our models. One of the student suggested an easy alternative to use early stopping technique.

3. We were asked another question regarding the benefit of using a Bi-LSTM head over the BERT model. The suitable explanation to this is that LSTMs can help in improving our results on longer spans and those with distinct boundaries since the gates inside LSTM could prevent gradient vanishing problem, to memorise the long time dependency.

4. We were also asked about why we chose to go with a model agnostic approach like Representation Erasure instead of a white box approach. We are currently using a black box approach as it was a baseline recommended by the competition organizers. We will consider looking into white box approaches for neural network understanding as they might provide better results.

References

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. *arXiv preprint arXiv:1704.02853*.

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.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *arXiv preprint arXiv:1703.04009*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2017. Understanding neural networks through representation erasure.

Ping Liu, Wen Li, and Liang Zou. 2019. Nuli at semeval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Sean Papay, Roman Klinger, and Sebastian Padó. 2020. Dissecting span identification tasks with performance prediction. *arXiv preprint arXiv:2010.02587*.

Silvia Pareti. 2016. Parc 3.0: A corpus of attribution relations. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3914–3920.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1125–1135, Copenhagen, Denmark. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144.

Erik F Sang and Sabine Buchholz. 2000. Introduction to the conll-2000 shared task: Chunking. *arXiv preprint cs/0009008*.

Alessandro Seganti, Helena Sobol, Iryna Orlova, Hannam Kim, Jakub Staniszewski, Tymoteusz Krumholc, and Krystian Koziel. 2019. NLPR@SRPOL at SemEval-2019 task 6 and task 5: Linguistically enhanced deep learning offensive sentence classifier. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 712–721, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the type and target of offensive posts in social media. *arXiv preprint arXiv:1902.09666*.

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