Project#4: Toxic Spans detection

Shivam Kumar [1], **Jaydeep Goyal**[2], **Prajwal H G**[3]
[1]170668, [2]170324, [3]170481
[1]EE, [2]EE, [3]CSE
{krshivam, jdgoyal}@iitk.ac.in, prajwal@cse.iitk.ac.in

Abstract

We have built a model capable of detecting toxic spans within the text . Our model is useful for moderating and censoring online, Twitter arguments. We are able to highlight toxic spans that can assist human moderators who often need a full report to answer the question "why the statement is toxic?". We have applied and tested various techniques ranging from lexicon searching methods by using Kaggle data set of bad words to transformer-based and NBOW-2 models on the data set provided by CodaLab. We have observed that BERT based transformer model has achieved the highest accuracy till now. In the future, we aim to ensemble these models together and research further to achieve better performance.

1 Introduction

Online arguments on various platforms like Twitter, Facebook are useful and can provide healthy discussions and insights. But with the increasing use of abusive and toxic words, this motive is hampered. We intend to provide a model that is able to identify the **toxic spans** in a statement which can be used to censor and monitor these conversations. Our model identifies the spans which make the statement toxic.

Previously a lot of work has been done to classify a statement as toxic and non-toxic , but less work has been done on predicting the spans which make the respective statement toxic. Our model is inspired by SemEval'19 task 6(Zampieri et al., 2019) and NER tagging approaches(Ranasinghe et al., 2020). We propose three approaches to detect the toxic spans in the sentence.The first approach employs some transformer based models(Devlin et al., 2018) which tokenize each word into multiple tokens and each token is labelled as toxic and not toxic, based on these labels the whole word is predicted as toxic or not toxic and thus the toxic span is reported. The second approach uses a simple Lexicon searching model that searches for every word in a list of abusive words and reports them in case of a match. Our third approach predicts a toxic score for each word using the Neural Bag-of-Words model and thus by selecting the appropriate threshold the toxic span is reported.Our forth approach uses classic Bi-Lstm many-many approach.

2 Problem Definition

Our model will extract a list of toxic spans, or an empty list, per text. A toxic span(by ipavlopoulos, 2021) is defined as a sequence of words that attribute to the text's toxicity. For example in the text: *"This is a stupid example, so thank you for nothing a!@#!@."* It comprises two toxic spans, *"stupid"* and "a!@#!@", which have character offsets from 10 to 15 (starting the counting from 0) and from 51 to 56 respectively. So, our model predicts the following list for this text: [10,11,12,13,14,15,51,52,53,54,55,56].

3 Related Work

We have not found any exactly same work but some references inspire us to propose some approaches in the direction aligned to solve the problem statement like a neural model "Pathologies of Neural Models Make Interpretations Difficult"(Feng et al., 2018), which iteratively removes the unimportant words from the text and the remaining words are considered important. Unimportant words and removed after a single forward-backward pass using the following method. For each word in the sentence ,the dot product of its word embedding and gradient of the output with respect to continuous vector that represents the word embedding is calculated and with this input gradient the unimportant words are removed.

Other works like "Learning Word Importance with the Neural Bag-of-Words Model" (Sheikh et al., 2016)where NBOW-2 which proposed, a feed forward network with BOW input, unlike NBOW model which learns word vectors specialised for the classification task, NBOW-2 improves on this model and gives information about which word is important for the given task.
There is a lot of work done on text classification (offensive/not-offensive) like in the SemEval'19 task-6 (Zampieri et al., 2019) the task-1 was offensive language identification and some of the proposed approaches, and most of the best performing approaches used transformer model such as BERT with models such as LSTM, for example in the NULI (Liu et al., 2019a) paper, they used transfer learning and BERT with LSTM to get the best result for task-1.They processed data set according to language behaviors on social media and made a bidirectional encoder representation.Also NER-tagging inspired our Bi-LSTM model some solid work is done in this field as in (Chiu and Nichols, 2016) and some articles as in (Nair, 2016).

4 Corpus/Data Description

In our models we have used three different types of data sets. The Lexicon searching model uses the kaggle data set of bad bad words which contains the list of toxic/ abusive words.,NBOW uses data set from (Zampieri et al., 2019) and the BERT model utilizes the training data from the CodaLab which contains the labelled toxic spans in 7939 sentences. The corpus statistics for the same are as below:

Statistics	Words	Characters
Total	258388	1624086
Mean	35.95	204.57
Minimum	1	4
Maximum	192	1000

Some key features about the data set are:

- 485 sentences report empty spans.
- 7.5% words are reported as toxic.
- 40.1% words in the data are stop words.

5 Proposed Approach

We have majorly tried 4 approaches one is a transformer model using BERT,other is lexicon model ,another a neural bag of words approach and a bi-Lstm Model.Their specification and description is as follows:

5.1 BERT

We used a transformer with a linear layer at top of the hidden states output namely Bert For Token Classification from pretrained models of Hugging face(Devlin et al., 2018). We inspired this model from basic NER tagging tasks to classify a word based on Named and Entity recognition.

Experimental Setup
Training BERT model
BERT tokens tokenize words from their vocabulary using word piece tokenization into one or more number of words. So, in cases of sub-words, it was important for them to retain their toxicity.
We first split the input text based on the white spaces, and then create a label set of the splitted words, consisting of $1'$s and $0'$s, on whether they are toxic or not. Then each split word is tokenized individually and each subtokenized word is mapped to original indices of input sentence. Also, each element of the input label list is extended the same number of times as the number of sub words into which word was tokenized. The tokenized words, ids and attention mask is fed into BERT model.

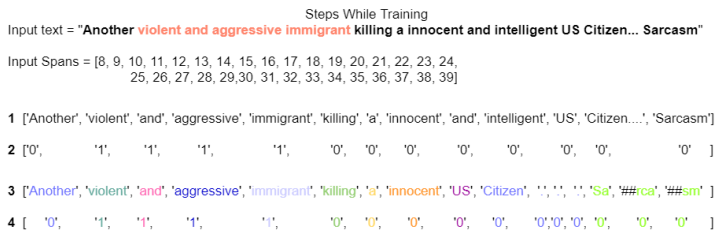

Figure 1: Input while training

Predictions
After the model is fine tuned and the classifier is trained, while predicting the output we again split the input and then tokenize each words and along with create a list containing each token's indices in initial input text.
The prediction is obtained for all the tokens by feeding the sentence into and then if any token is predicted as toxic, then it's original parent word from which the given token was obtained, is marked as toxic. The span is then derived for the the toxic predicted word.

5.2 LEXICON

In the lexicon searching model we have used a list of bad words which contains 1616 common abusive words and used this list to classify every word in our input sentences to toxic or not toxic. The words

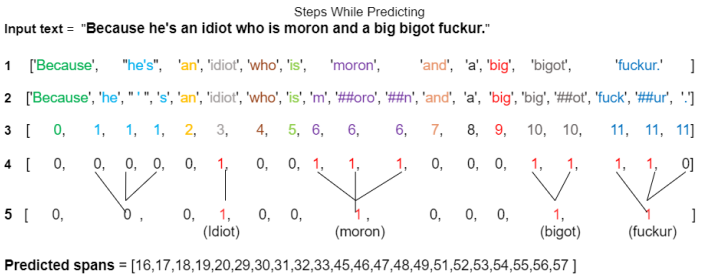

Figure 2: Steps of feeding the input & predicting spans

classified as toxic are then reported by calculating their indices and identifying the relevant spans.

5.3 NBOW-2

This approach is based on the paper **Learning Word Importance with the Neural Bag-of-Words Model**(Sheikh et al., 2016).
A new version of classic NBOW model, the NBOW-2 is used to identify the span of toxic words. This uses a fully connected feed forward network wit BOW(Bag of words) input.To learn the importance of words(word importance weights) in the context of the task we use weighted sum composition of the text X as follows:

$$z = \frac{1}{|X|} \sum_{w \in X} \alpha_w v_w \tag{1}$$

α_w is the scalar word importance weights for each word $w \in X$. Learning task specific word vectors ensure the words which drive the classification are given more importance.α_w are obtained by vector named a as follows:

$$\alpha_w = f(v_w.a) \tag{2}$$

f scales importance in range [1,0] ,soft-max activation function is used for f.
The vector a is randomly initialised and learned along with word vector and other parameters.
The model is trained used stochastic gradient descent with forward pass and will use equation 1 and 2 with the output estimates of soft-max to predict the word weight-age.

Pipeline

Pre-processed the training data & split it into train and test. Collected data for non-toxic data set and toxic and positive sentences(toxic) are labeled 1 and negative(non-toxic)data is labeled 0. Output is in the format $word\_number$ and this is processed and with a cutoff of 0.6 is processed and labeled as 1 or 0 denoting toxic or non toxic.

5.4 Bi-LSTM

We use keras Birectional Lstm , with mean squared loss and relu activation along with merge-mode of concatenation. Keras many-many LSTM prediction inspired this model with BILSTM NER Tagging.

Training

First we mapped word to number and tagged every word corresponding to 1 if its offensive 0 otherwise.Then we used the length of bi-lstm input to be 60 as it was the median+standard deviation for sentences smaller we used padding in both input and output of training data.Then using the proper format we trained the model in the following way, first embedding using all the words in the dictionary , after embeding each token is converted into a vector of 64 dimensions,then a BiLSTM layer that takes a recurrent layer(Lstm) as an argument then the outputs are concatenated this makes the output 128,Then an LSTM layer followed by a time distributed layer with dense as every output will affect the next.
This model is used by varying the parameters,also 'relu' activation was used. We trained for 5 epochs and got 0.4029 f1 score on validation data.

5.5 spaCy NER

We trained our model on another baseline provided by the organizers by adding a new entity type TOXIC on the spaCy's existing pretrained NER model.

6 Experiments and Results

We experimented with four variants of transformers namely BERT large cased, BERT base cased, RoBERTa large(Liu et al., 2019b) & XLNet large cased (Yang et al., 2020) and found out that the best performing model is BERT large cased which has been trained for batch size = 32, token length = 100 and 4 epochs.
We also tried the Lexicon searching model with the Twitter data set.
We experimented in NBOW-2 model with varying data sets of previous editions of OffensEval and various cutoffs and we observed that by using the positive data set of SemEval'19s task and negative data set of SemEval'21s task with the cut off of 0.6 showed the best performance.

The results of all these experiments are tabulated in table 2.

Models	F1-score
spaCy's NER tagging	**0.611**
BERT(Large-Cased)	**0.564**
BERT(Small-Cased)	0.553
RoBERTa(Large)	0.518
XLNet(Large-Cased)	0.551
NBOW-2	0.108
Bi-Lstm	0.403
Lexicon	0.19

Evaluation

We use F1 score to evaluate the output of our model,We compute the F1 score of a system A_i with respect to the ground truth G as follows.

Our model A_i returns a set $S^t_{A_i}$ of characters offset,for parts of the post found to be toxic and G^t be the character offsets of the ground truth annotations of t.

$$F^t_i(A_i, G) = \frac{2.P^t(A_i, G).R^t(A_i, G)}{P^t(A_i, G) + R^t(A_i, G)} \quad (3)$$

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

$$R^t(A_i, G) = \frac{|S^t_{A_i} \bigcap S^t_G|}{S^t_G} \quad (5)$$

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 data-set T to obtain a single score for system A_i.

7 Error Analysis

The Lexicon model performed better than the baseline model provided by the organizers and can be further used as classical approach to find the toxic words which matches to the dictionary. But, the Kaggle data set (Words, 2017) used, contained limited and few bad words, due to which many words being toxic were unable to be mapped to the dictionary. So, better dictionary may provide us with comparatively higher accuracy. Also, it failed to mark words with different spelling or special characters as toxic.

For our transformer model, the results were quite impressive and many surrounding words which when isolated are not toxic but are being correctly classified as toxic in a sentence according to its context. Also, it has been successful to an extent to predict words with spelling mistakes and special characters. BERT large cased performed best among all the transformers. There were many instances when when the predictions were very wrong, such as predicting extra words as toxic and failing to give accurate predictions in case of lengthy sentences. The reason for this is the huge variance of the data set, which contained texts of too small lengths to very large lengths. The Bert model is unable to feed all the tokens. But one possible approach to incorporate this, is to split paragraphs into multiple rows of individual sentences along with the character offset by the same amount counting the spans. The NBOW-2 model does not give a reasonable toxic score to the nouns due to their vicinity in the toxic words in the given data set.

8 Individual Contribution

Name	Contribution
Shivam Kumar	Implemented BERT, RoBERTa and XLNet models. Implemented code for mapping of tokens with original words and mapped the labels with the tokens while training. Currently doing the CRF head on top of BERT model, and trained our model on new baseline provided by organizers
Jaydeep Goyal	Studied previous literature, Implemented Lexicon searching model, Inspected data sets and corpus statistics, Completed preprocessing steps and built auxiliary functions like evaluate F1 score.
Prajwal H G	Implemented NBOW-2 model form (Sheikh et al., 2016), implementing a new word removal model from (Feng et al., 2018), implemented Bi-Lstm model.

9 Future Work

We have completed the baseline models now we intend to ensemble these models together. We will also look for the possibility of improving accuracy through some further pre-processing steps. We faced some troubles due to the increased length of the tokens in the BERT model hence we will be

looking at these concerns in the future.

Name	Future work
Shivam Kumar	Incorporate pre-processing steps in BERT, Reduce token length and experiment with head layer like attach CRF layer, Apply some NER techniques
Jaydeep Goyal	Explore models like RoBERTa and LSTM, Finalize pre-processing steps and ensemble models together
Prajwal H G	Improve Bi-Lstm and use CRF with it,Improving Bert model by using Some classic models on top of Bert

10 Conclusion

We tried four different models as baseline and found that the transformer model has the highest accuracy (F1 Score). Moving forward we will improve this model and try to solve some of the problems we are facing in the data such as, the length of some of the paragraphs are long and can't be directly used in the BERT and pre-processing the data properly is a necessity.
We have built Lexicon searching and NBOW-2 models, however they do not give us good results and are not capable of performing independently but they capture useful insights and we hope that in future ensemble multiple models together and get better results. Also we have Used B-Lstm models which did good for a classical model without using a trasformer we intend to use and improve this model.

11 Presentation Feedback

During the presentation, we were asked about the difference in the SemEval'20 task 6 and this problem statement. We have solved the problem of identifying the toxic spans within a statement whereas SemEval'20 task 6's problem was to classify the whole statement into toxic and not toxic.
One related question to the problem statement was whether we are reporting the toxic spans at word level or character level which was clarified by us that, while we are returning the whole span of toxic words the format of reporting is at character level. Thus we are reporting the indexes of the characters which are toxic.
Some questions were asked about our model, one of them was how we are assigning weights in our NBOW-2 model and do we need any external corpus like sentiwordnet for toxic words. This (word-weight) is found as described in the model α_w using a vector *a* which is learned over the epochs.

$$\alpha_w = f(v_w.a) \tag{6}$$

Where f uses soft-max activation and $v_w.a$ is dot product between *a* vector and word vector v_w. During prediction the model uses stochastic gradient descent along with the learned *a* vector to predict the word weights.
Few questions were asked about the BERT model which are as follows:

- Are we considering neighbouring words and assigning toxicity to them as well?
  We explained that although we are not specifically using the neighbouring words condition, however it has been observed from our results that the model is predicting neighbour words also as toxic but it is not advisable to account the neighbouring words as in our examples our accuracy is reducing due to this extra prediction. Like in the example - "Burn in hell where you belong, you filthy lying treasonous piece of shit. And take every one of your GD enabling pigs with you." Only "filthy" and "pigs" must be labelled as toxic but our model is also labelling "lying" and "treasonous" as toxic as well which is hampering the accuracy.
- Do our model assigns the same toxic score to the same words irrespective of whether they are used in different sentences/context or not?
  Our model does not gives a score to different words rather it identifies a word that is being labelled as toxic under what context and then

while predicting for the testing input it checks if the word is displaying the toxic content and if yes it labels it toxic.

- How are we combining the labels of different tokens of a word?
  We have not shared the exact details of the same in the presentation but we mentioned that if all the tokens are non toxic then the word is not toxic, similarly if all the tokens are toxic then the resulting word is also labelled as toxic. However when there are some toxic labels and some non toxic labels we intend to try some experiments like predicting the label of the word as toxic or some function of the labels of its constituent tokens etc.
- How are we dealing with lengthy paragraphs?
  We have mentioned this in the future work and we hope to split the paragraphs into constituent statements and then feed these sentences into the BERT model.
- How are we dealing with the noisy sentences, are we using any other data sets?
  We currently are not relying on some other data sets but we hope that our model is robust enough to deal with noisy sentences.

Some questions were related to our future work. One question was asked about how we are planning to pre-process the sentences containing special characters. We proposed two solutions for this task. First check the distance of these words containing special characters to the distance of the bad words and if this distance is less than some threshold then we can classify this word as toxic. Secondly we will use the word embedding of Twitter data and compare the words containing special characters with these word embedding and if they are mapped to toxic words, the word is pre-processed to this toxic word.

We were asked about how we will ensemble our multiple models together. As we have just completed our baseline models we are looking for the approaches to ensemble multiple models together but we are currently not aware of how we will tackle this problem.

We thank Sagnik for suggesting us about the Negation scope and NQ detection problem which closely relates to our problem statement and we hope to explore this technique in upcoming future.

References

Jason P. C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns.

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.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.

Organized by ipavlopoulos. 2021. Semeval 2021 task 5: Toxic spans detection. Ongoing sem eval task, https://competitions.codalab.org/competitions/25623.

Ping Liu, Wen Li, and Liang Zou. 2019a. 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, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach.

Snehal Nair. 2016. Named-entity recognition (ner) using keras bidirectional lstm. Without removing stop words Bi-Lstm, https://towardsdatascience.com/named-entity-recognition-ner-using-keras-bidirecti

Tharindu Ranasinghe, Alistair Plum, Constantin Orasan, and Ruslan Mitkov. 2020. Rgcl at semeval-2020 task 6: Neural approaches to definition extraction.

Imran Sheikh, Irina Illina, Dominique Fohr, and Georges Linarès. 2016. Learning word importance with the neural bag-of-words model. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 222–229, Berlin, Germany. Association for Computational Linguistics.

Bad Bad Words. 2017. Bad words. Data retrieved from Kaggle, https://www.kaggle.com/nicapotato/bad-bad-words.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

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 (Of-

fensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.