Multi-Label Online Toxic Comment Classification

Ziyan Lai[1]
zl1532@nyu.edu

Weiyao Xie[1]
wx455@nyu.edu

[1]Tandon School of Engineering
New York University
https://github.com/aojiu/toxic_comment_classification

Abstract

With the increasing freedom of online speeches and the enriched online platforms, we found that abusive languages are becoming more and more common or even severe among the online environment. The source of these abusive languages can be individuals that intend to humiliate others, while it can also be commercial or political purposes to depreciate their counter-parties (Erjavec and Kovačič, 2013). Since some of the comments are evolving into newer forms of expressions, previous methods like auto-detecting and blackmailing may not work well any more to address this problem. In this paper, we propose to add extra neural network layers on top of BERT (Devlin et al., 2018), a state-of-the-art model, to classify different types of abusive languages and to create a better online environment.

1 Introduction

Transformers models like BERT and biLSTM models like ELMO have been shown to be empirically robust and powerful when dealing with corpus acceptability, question answering, and text labeling. Based on BERT's characteristic of truly bidirectional and its masked language modeling mechanism, comparing to ELMO (Ethayarajh, 2019) which essentially uses a concatenation of left-to-right LSTM and a right-to-left LSTM, BERT could be used to generate more accurate word representations. Therefore, we chose BERT as a reliable foundation for our task, and hope it could also provide us efficient parameters for our further experiments by adding more layers on top of BERT's fine-tuned layer. There are existing different approaches to fine-tune BERT to fit various target tasks. For example, it is shown to be efficient to experiment with different learning rates to determine which is the best for decreasing testing errors for specific tasks . In addition, researches proved that it is beneficial to fine-tune BERT on a larger sentiment domain, rather than on a partial domain, based on the finding that fine-tuning on a larger domain can lead to the benefit of generally better model performance and less training time on the downstream tasks (Sun et al., 2019).

In this paper, we used the Toxic Comment Classification dataset from Kaggle to deal with this problem of multi-class abusiveness classification. The dataset was collected from Wikipedia comments that are labeled as human raters for toxic behavior. We first fine-tuned the massive dataset with BERT, using the entire 159,571 training examples, where each example includes a comment text and its corresponding labels, and tested it with 31,990 validation examples to generate a score as our baseline. The original dataset has six labels. We screened the six original labels into three, which are "Insult", "Threat", and "Toxic". Then we added one addition label, "IsAbuse", which is a binary indicator that tells whether the comment is abusive or not. We first fine-tuned only on the label "IsAbuse" with BERT, and extracted the last hidden layer as trained word embeddings for our further experiments. Different approaches have been made before, like adding CNN or LSTM layers (Zhang et al., 2019). In our research, we used a two-layer neural network as our classifier. We also experimented with the impact of using our fine-tuned word embeddings that is train on the binary indicator "IsAbuse" versus the bert-base-uncased pretrained embeddings.

2 Related Work

Our task is to classify whether a comment is online harassment or not using a highly imbalanced multi-label dataset from Kaggle. We will briefly introduce some of the studies that we find helpful

during our research.

2.1 Identify Types of Abuse

To better address online abusive comments, we need to first define what should be treated as abusive languages. Current techniques have primarily focused on overt or obvious hate speech, but have not focused enough on other myriad forms of toxic languages. Apart from insults and hate speech, which are the two categories that current approaches are narrowly focused on, there exits other common types of abusive languages, like threats microaggression (Jurgens et al., 2019). To address this problem, we selected the dataset from Kaggle. The original dataset has 6 labels: Toxic, Severe Toxic, Obscene, Threat, Insult, and Identity Hate. These labels has covered a comprehensive spectrum of abusiveness. However, since our research's main focus was on labeling the most conspicuous and malicious comments that needed to be handled most urgently, we filtered the six labels into three, and they are Insult, Toxic, and Threat. In specific, we merged the original "toxic" and "severe toxic" into one label, "Toxic", and omitted labels "Obscene" and "Identity Hate". We hope these three picked labels can accurately address the most abusive comments, while not losing generality from the previous complete spectrum of labels. We also expect to reduce our training time with this smaller number of labels.

2.2 Two-Stage Classifier

In order to better investigate whether certain comments may hurt a specific group of people and how large the damage comments could cause, a two-stage classifier is used to put harmful comments further into three categories(cite sharifirad). The study aims to bring harassment towards females to people's attention, and want to use divided categories to better help social media to lower the damage of toxic comments on the community, while lowering the possibility of restricting people's freedom to post their thoughts. After collecting data, URL, emoji and other noise sources are removed from texts. Then pretrained embeddings like Word2vec, FastText, etc are used to convert texts into tokens. Recurrent Nueral Network(Sherstinsky, 2018) and Concolutional Nueral Network are applied respectively to obtain the classification results measured using accuracy. The combination of FastText and CNN has achieved 93% accuracy in the experiment.

Another research on Tweet comments has used a two-stage classifier (Jha and Mamidi, 2017), where they organized and trained first on binary labels, for example, benevolent and non-benevolent, and then further trained the non-benevolent examples based on 3 other classifiers. We find this method to be relatively efficient in improving the performance of the model. Therefore, we decide to follow this method of two-stage training process. In specific, we added a binary label "IsAbuse" to our dataset as a general indicator of abusive language and trained on it as our first-stage classifier. We expect this step could help to improve our model's performance, since the result after this stage should have a general judgement or preference on whether each individual comment is abusive or not, and will therefore help to better classify the specific categories of abusiveness in the next stage.

2.3 BERT + CNN Layers

For this specific task of toxic classification, a study by Roman Orac that uses word embeddings from bert-base-uncased and feeds into CNN layers has reached AUC scores of 95.4% for "Toxic", 95.9% for "Insult", and 94.6% for "Threat"[1]. It is worth to mention that this model does not incorporate any fine-tuned word embeddings, but directly uses bert-base-uncased pretrained word embeddings as input of the CNN layers. The AUC scores are significant enough to say the combination of BERT and CNN layers is doing a decent job. However, We intend to experiment with several simpler layers to see if it can have an competitive result comparing to the more intricate CNN layers.

3 Data Preprocessing

The original dataset has 6 labels. While reducing and filtering it to 3, which are "Toxic", "Insult", and "Threat", we added a binary label "IsAbuse". This label will be marked as "1" if any of the other three label is "1", which means the specific comment shows positive to some degree of abusiveness. With the intention of making a two-stage classifier, we first fine-tuned on this binary label with BERT, and extracted the last hidden layer as word embeddings for the next stage training. In

[1] https://towardsdatascience.com/identifying-hate-speech-with-bert-and-cnn-b7aa2cddd60d

the second stage, we trained on each of the three label separately to further determine which category of abusive the comment belongs to.

Imbalanced dataset has been a very common problem in machine learning and deep learning. When dealing with a dataset with highly imbalanced multilabel classification, it is important to have a good strategy to balance the data. While using AUC and ROC score as a metric in a task, an effective way to increase the performance is to oversample the minority that can almost eliminate the imbalance (Buda et al., 2017). In order to minimize the detrimental effects of imbalanced dataset, we randomly oversample the minority class for "IsAbuse". The ratio of 0:1 for the label is 143720:15851. Therefore, for our training set, the probability to select a "1" is approximately 10 times higher than selecting a "0" to the data loader. By doing so, we can roughly achieve 1:1 ratio of the labels. For other labels, since they are not mutually exclusive, we do not apply this method on them.

The classifier used to classify toxic comments can be very sensitive to typos and spelling mistakes. The experiment done on Google's toxic comments detector has shown that by making slight changes on the texts, like changing the spelling of certain words. The toxicity score will change dramatically (Hosseini et al., 2017). Therefore, we cleaned our data using regular expressions to correct some frequent typos and spelling mistakes.

4 Model

We followed a two-stage classification for our model. We first fine-tuned BERT Base using over 15,000 text comments only on label "IsAbuse". After this step, we took the last hidden state and use it to transform comments into word embeddings. Then we feed these embeddings to a 2-layer neural network. The neural network is consisted of two linear layers and a ReLU activation function between them. A 0.5 dropout rate is used to prevent overfitting. We trained each of the four labels separately with this classifier. The model outputs one probability. In the evaluation part, we calculated the Area Under the Curve so that we can measure how well the model can predict and distinguish both positive class and negative classes.

For the error function, we use Binary Cross Entropy loss because when dealing with multi-label text classification, BCEloss tends to outperform many other loss functions, for example cross entropy loss(Liu et al., 2017). Before applying BCEloss, we need to convert the output of the linear layer to the range from 0-1 using sigmoid function:

Binary loss punishes the model when a True label is predicted to have low possibility and a False label is predicted to have a high possibility. For example, for comment i, the label is True. we just need to look at the first part of the loss, becasue the second part is 0. If the model predicts a possibility close to 0. Then log(sigmoid) will be close to negative infinity which causes a huge loss. If the possibility is close to 1, log(1) is close to 0 which makes the loss close to 0. Same logic for the other cases.

5 Experiments

In this section, we present our model's results and the comparison between different configurations.

5.1 BERT$_{BASE}$

BERT$_{BASE}$ is defined as the pretrained "bert-base-uncased" configuration. We first used BERT$_{BASE}$ with two linear layers on top of it as our model. Without fine-tuning BERT, the AUC score output by our model is higher than both works we mention in section 2.2 and 2.3, except for the score of "Threat". The results are shown in Table 1. With the use of two linear layers, our model can avoid the potential problem of overfitting, which might exists in CNN. However, BERT$_{BASE}$ performs poorly on "Threat" label, which is possibly caused by the extremely imbalanced data. The dataset only has around 400 positive cases for "Threat".

5.2 Fine-Tuning BERT

We keep the same neural network structure as our previous model, and we fine-tuned BERT using "IsAbuse" label. The AUC score generally experiences a slight improvement comparing to the results in 5.1. There is a significant increase for "Threat" label. The AUC increased by more than 20%, which is the highest among all the labels after using fine-tuned BERT. We think the increased AUC score can be attributed to the understanding of context of the comment: when

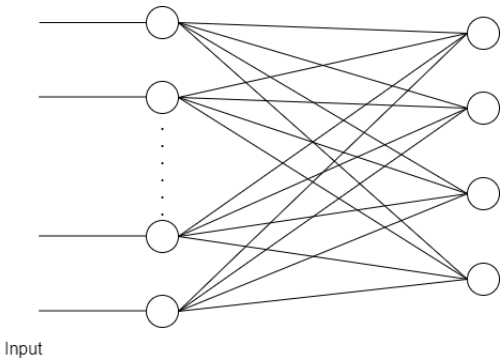

Figure 1: The Structure of the model we use. BERT followed by two linear layers with ReLu activation function in between.

Model	Threat	Insult	Toxic	IsAbuse
BERT+CNN	0.9467	0.9599	0.9606	—-
BERT$_{BASE}$+NN	0.7515	0.9475	**0.9694**	0.9457
BERT$_{Tuned}$+NN	**0.9909**	**0.9766**	0.9678	**0.9695**

Table 1: Test set results. Our implementations of the neural network remains the same. The only difference is whether or not the BERT model is fine tuned. As we can see, the combination of neural network can achieve close AUC score with CNN. However, fine tuning BERT has the best results among these three models.

Model	Threat	Insult	Toxic	IsAbuse
BERT+Uncleaned	0.9909	**0.9766**	0.9678	**0.9695**
BERT+Cleaned	**0.9940**	0.9676	**0.9707**	0.9694

Table 2: The Cleaned and Uncleaned data is fed into the same model respectively. Their results are displayed.

the model can better understand the comment, it can predict positive case better.

6 Ablation Study

The classifier used to classify toxic comments can be very sensitive to typos and spelling mistakes. The experiment on Google's toxic comments detector has shown that by making slight changes on the texts, for example, changing the spelling of certain words, the toxicity score can change dramatically (Hosseini et al., 2017). Therefore, we cleaned our data using regular expressions to correct some frequent typos, to convert some full-width digits to normal digits, to remove urls and images, and to remove special characters. The code we used to handle these noises comes from (zake7749, 2018). The cleaned data is only used for fine-tuned BERT, and we compared the results with those produced by uncleaned data. The comparison is shown in Table 2. From the results we can see that the cleaned data cannot guarantee an improvement on AUC score. While there are small boosts on "Threat" and "Toxic", there are also slight decreases on "Insult" and "IsAbuse". We speculate the reason is that by removing some special symbols or images, we are changing the context of the comments, which results in a slight decrease in the performance. Our cleaning process does not include any spelling check, so we are not helping the model to understand the context correctly. The noise we removed can be trivial to BERT.

7 Conclusion

Compared with previous work using CNN on top of BERT, we found that using neural networks with less parameters can achieve similar or slightly better results. Fine-tuned BERT using only the conclusive "IsAbuse" label can increase the AUC score of the model and outperform the original model, especially for the extremely unbalanced label. However, cleaned data does not seem effective in improving the existing model. Our further study direction can be correcting the spelling of words and grammar to see if it can further improve our model's performance.

8 Collaboration Statement

Ziyan Lai - Data Preprocessing: Extracted informative labels, cleansed dataset with usable data; Fine-tuning with BERT: Modified transformer

source code for our research purposes, fine-tuned and created first-stage classifier for the model.

Weiyao Xie - Dataloader: Handled imbalanced dataset with equal appearance ratio; Model: Implemented BertClassifier class with two linear layers on top; Implemented Dataloader, train and evaluate functions; Investigated loss function and metrics.

References

Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. 2017. A systematic study of the class imbalance problem in convolutional neural networks https://doi.org/10.1016/j.neunet.2018.07.011.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Karmen Erjavec and M.P. Kovačič. 2013. Abuse of online participatory journalism in slovenia: Offensive comments under news items. *Medijska Istrazivanja* 19:55–73.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings.

Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving google's perspective api built for detecting toxic comments.

Akshita Jha and Radhika Mamidi. 2017. When does a compliment become sexist? analysis and classification of ambivalent sexism using twitter data. In *Proceedings of the Second Workshop on NLP and Computational Social Science*. Association for Computational Linguistics, Vancouver, Canada, pages 7–16. https://doi.org/10.18653/v1/W17-2902.

David Jurgens, Libby Hemphill, and Eshwar Chandrasekharan. 2019. A just and comprehensive strategy for using NLP to address online abuse. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, pages 3658–3666. https://doi.org/10.18653/v1/P19-1357.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. pages 115–124.

Alex Sherstinsky. 2018. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *CoRR* abs/1808.03314. http://arxiv.org/abs/1808.03314.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification?

zake7749. 2018. Deep toxic.

C. Zhang, M. Cai, and X. Zhao. 2019. Research on case preprocessing based on bert -cnn-lstm model. In *2019 20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*. pages 253–258.