Multi-Label Text Classification With Imbalanced Dataset

Ziyan Lai[1]
zl1532@nyu.edu

Weiyao Xie[1]
wx455@nyu.edu

[1]New York University
https://github.com/aojiu/toxic_comment_classification.git

Abstract

With the increasing freedom of online speeches and the enriched online platforms, we found that online abusive speeches 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 become a commercial or political purpose to depreciate their counter-parties (Erjavec and Kovačič, 2013). Since some of the comments are evolving into newer forms of expressions, some of the previous methods like auto-detecting and blackmailing may not work well to address this problem. In this paper, we proposed 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 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, it could generate more accurate word representations comparing to ELMO (Ethayarajh, 2019), which essentially uses a concatenation of left-to-right LSTM and a right-to-left LSTM. Therefore, we chose BERT to set a reliable baseline for our task, and it can also provide us efficient parameters for our further experiments to add layers on top of BERT's fine-tuned layer. There are existing approaches of ways to fine-tune BERT to fit 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 a partial domain, given that it turns out to perform better, and requires 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 the problem of multi-class sentiment classification. The dataset was collected from Wikipedia comments that are labeled as human raters for toxic behavior. We fine-tuned 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. Because our main goal was to label the abusive or malicious comments, we filtered them into three labels and added one binary indicator label. To further improve the performance, we construct additional neural network layers on top of BERT. Different approaches have been made before, like adding CNN or LSTM layers (Zhang et al., 2019). However, we intend to experiment with several simpler layers to see if it has an ideal improvement to the result. We extracted the last layer of hidden-state as an input and fed it into a non-linear layer and a linear neural network to experiment with further improvements. Then we compared the F1 scores between the plain BERT fine-tuned model and our experiment on two additional layers, and observed some improvements.

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

For our research to better address online abusive problem, we need to first define what should be counted 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 that has 6 labels, including toxic, severe toxic, obscene, threat, insult, and identity hate to cover a more comprehensive spectrum of abusiveness. With the intention of speeding up the training process without losing generality, we then merged "toxic" and "severe toxic" into one label, and omitted labels "obscene" and "identity hate".

We also added a binary label "IsAbuse". Previous researches on Tweet comments have used a two-stage classifier, 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 (Jha and Mamidi, 2017). In our case, we want to speed up the computation process by making a one stage classifier that can both determine whether a comment is abuse and returns the category of abusing it belongs to. Therefore, we added the label "IsAbuse" to the original dataset as a general indicator of abusive language, which will be marked as 1 if any of the other 3 labels is 1. This binary indicator was also expected to improve the performance of our model, since it expands the correct territory of the prediction. After identified our target abusive language types, we refined the domain into four labels, and they are: "toxic", "threat", "insult", and "IsAbuse".

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(Sharifirad, 2019). 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(cite RNN) and Concolutional Nueral Network(cite CNN) are applied respectively to obtain the classification results measured using accuracy. The combination of FastText and CNN has achieved 93% accuracy in the experiment.

3 Data Preprocessing

The dataset we use has 6 labels, while reducing it to 3, we added a binary class "IsAbuse". Instead of making a two-stage classifier, we want our model to function as a one stage classifier that can both determine whether a comment is abuse and returns the category of abusing it 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 the categories, 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.

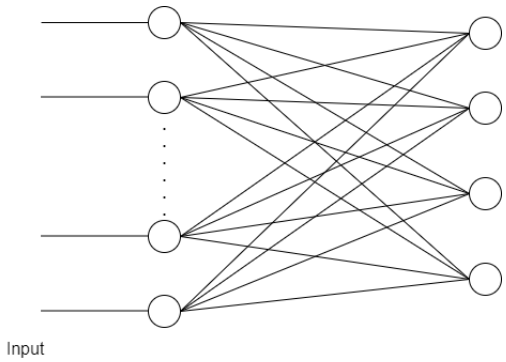

Figure 1: Ground truth tree for a premise sentence in SNLI's training dataset.

4 Model

We have two steps in our model. We first fine-tune BERT Base using 15000 text comments with 4 labels. After fine-tuning BERT, we take the final hidden state of comments and input the embedding to a 2-layer neural network. The model outputs 4 classes, each represents the probability of a class. We use a threshold to determine the final output should be 0 or 1. The threshold is fine-tuned using Grid Search.

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:

$$\sigma(x) = \frac{1}{1 + exp(-x)}$$

Then we compute the loss:

$$l = \frac{1}{n}\sum_{i=1}^{n}[y_i \cdot log\, x_i + (1 - y_i) \cdot log(1 - x_i)]$$

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.

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.

Sima Sharifirad. 2019. Nlp and machine learning techniques to detect online harassment on social networking platforms .

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

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.