Toxic Comment Classification

Usama Zafar
dept. Computer Science and Technology
Tsinghua University
Beijing, China
firstauthor@i1.org

Abdalwhb Abdalwhab 2019380044
dept. Computer Science and Technology
Tsinghua University
Beijing, China
Abdalwhab.Bakheet@gmail.com

Abstract

The ABSTRACT is to be in fully-justified italicized text, at the top of the left-hand column, below the author and affiliation information. Use the word "Abstract" as the title, in 12-point Times, boldface type, centered relative to the column, initially capitalized. The abstract is to be in 10-point, single-spaced type. Leave two blank lines after the Abstract, then begin the main text. Look at previous CVPR abstracts to get a feel for style and length.

1. Introduction

This project aims at tackling the Jigsaw Multilingual Toxic Comment Classification [1]. Where a toxic comment is defined as anything rude, disrespectful or otherwise likely to make someone leave a discussion. If these toxic contributions can be identified, then we can plan and execute polices like removing them or notifying their corresponding users. Eventually, this will help us taking a step towards having a safer, more collaborative internet.

2. Related Work

3. Problem Definition

4. Proposed Methods

Figure 1 shows our BERT-based[3] baseline model, we started by a specialized pre-trained multilingual model called m-BERT. We used the pretrained weights to initialize both the tokenizer and the model.

For the pre-processing, we pad all sentences to the same length and add a [CLS] token at the beginning (a special token used as a placeholder to get a vector embedding representing the whole sentence). Then attention mask for each sentence is generated to clarify which tokens represents real words and which are just padded junk. Then every sentence is passed through a 12 transformers layers each has a size of 768, and with 12 attention heads. Each layer (including the last layer) produces a vector embedding for each word, and another vector embedding representing the whole sentence and pass it to the next layer. In this implementations, we neglect all the words embedding and only use the 768 dimensional vector representing the sentence and use it for classification. We started with simple 2 Fully Connected layers (FCs) for classification. First one, use ReLU activation and the other uses Sigmoid function.

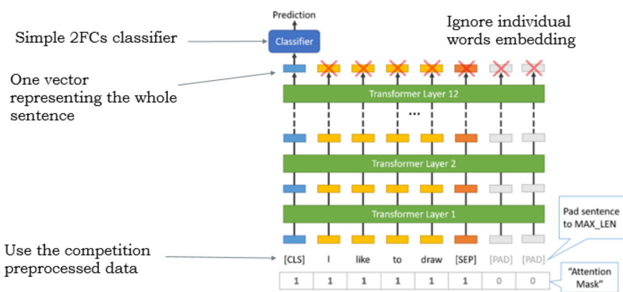

Figure 1. BERT-based Baseline Model.

Although BERT is the best possible model known to us, the performance can still be improved by data pre-processing. So we decided to split the work along two branches, exploring better models, and improving pre-processing of available data while using BERT or BERT like architecture. We tried bunch of other models such as:

- Bidirectional LSTMs with pretrained glove model for embedding
- Using the BERT model itself we also tried bunch of other things such as:
- Trying a more compact version of BERT (6 layers instead of 12)
- Use different pretrained BERT models
- Preprocess the data ourselves instead of using the competition pre-processed data

- Ensemble multiple predictions

Furthermore, to improve the obtained performance in the competition, we apply simple ensemble technique by taking the weighted average of our top models predictions. Figure 2 introduce the diagram for the ensemble technique.

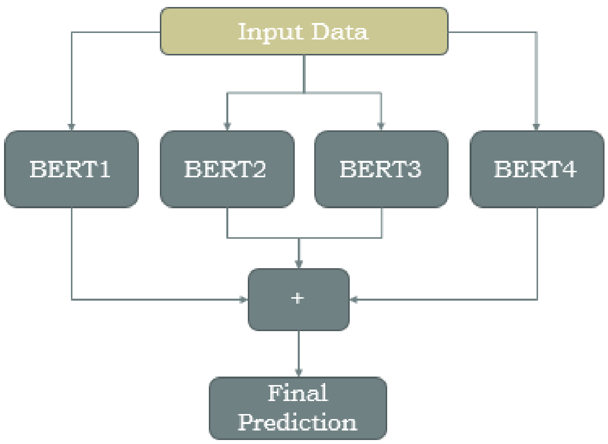

Figure 2. Ensemple couple of models.

5. Experiments

5.1. Dataset

We based our work on the competition dataset. The primary data for the competition consists of comments classified as toxic or non-toxic (0 and 1). The dataset is divided into train, validation and test splits. The train set's comments are entirely in english and come either from Civil Comments or Wikipedia talk page edits. The validation and test data's comments are composed of multiple non-English languages[1]. They also, include the data from their other competition "Jigsaw Unintended Bias in Toxicity Classification"[2] as an auxiliary dataset, but the dataset has 6 classifications for comments labels instead of just two.

The organizers of the competition provided the dataset in two format; raw text files, and pre-processed data ready for use by BERT models, but they used a small maximum sequence length of only 128 tokens.

5.2. Evaluation

To evaluate a model, depending on the experiment, we may train it on the primary training dataset, auxiliary dataset and validation dataset, but we never let the model train on the test dataset. In fact, the labels for test dataset of the competition are not provided, so the only way to get the score of your model is by submitting the predictions to the competition website. So, that is what we did.

Submissions to the competition are evaluated on area under the receiver operating characteristic (ROC) curve (AUC) between the predicted probability and the observed target.

5.3. Exp.1: Baseline m-BERT

For the first experiment, we tried to train our baseline model (the m-BERT-based model) for 5 epochs on the main training dataset and evaluate it on the text dataset.

After this baseline experiment, we decided to evaluate a more compact version of BERT called DistilBERT with special pretrained weights on multilingual data called distilbert-base-multilingual-cased. This model has only 6 hidden layers instead o the 12 layers by m-BERT.

We only used one FC layer with one neuron and sigmoid activation function to do the classification. With the memory limitations on Kaggle environment we adopted this smaller model to be able to add on top of it our own ideas.

Then we investigated improving this model in two directions, one that utilizes the auxiliary dataset for training with early stopping, while the second one only uses the primary dataset.

5.4. Exp.2.A Utilizing The Auxiliary Dataset

For this experiment, we tried to utilize the auxiliary dataset by converting its labels to be in the range 0,1 by applying a threshold of 0.5 (the recommended threshold from [2] for mapping their labels to toxic and non-toxic). Then we merged this data with the primarily train dataset process it by ourselves with a maximum sequence length of 100 tokens.

We then trained the model for 10 epochs on the combined training data, and 20 epochs on the validation data while early stopping utilized to stop training whenever the performance on the validation data is not improved after one epoch.

5.5. Exp.2.B: No Auxiliary Dataset

Without utilizing the auxiliary dataset, with maximum sequence length of 192. Training on the primary dataset for 3 epochs and on the validation dataset for another 3 epochs.

5.6. Exp.3.A: Improved Classifier

Based on Exp.2.A in this experiment, we increased the maximum sequence length to 120 token (which is still less than the maximum sequence length used in Ep.2.B because processing of the auxiliary dataset requires a lot of memory), and improved the classifier by adding two layers before the final classification layer. The first one, is FC layer with 256 neuron, and ReLU activation function. While the second is a dropout layer with dropout probability of 0.2.

5.7. Exp.4.A: Train on Primary and Half The Auxiliary Datasets Separately

In this experiment we tried to utilize the available memory more efficiently by explicitly releasing any unneeded object. This allowed us to process longer sequences without truncating it. In fact, we changed the maximum sequence length to 250.

We started, by training on the primary training data for 5 epochs, then process around half of the auxiliary data and train on it for 4 epochs. Lastly, the model is trained on the validation data for 3 epochs before it is used for evaluation.

5.8. Exp.5.A: Train on Primary and Auxiliary Datasets Separately

We increased the maximum sequence length to 250 and same setup as Exp.4.A, but after training on half of the auxiliary dataset, we process and train on the other half before training on validation dataset.

5.9. Exp.6.A: Ensemble Multiple Predictions

In this experiment, we take the average of predictions from our top performing models that were trained on both primary and secondary datasets.

5.10. Other Failed Experiments

We tried to implement a Bidirectional Long-Short-Term Memory (BiLSTM) based model with pretrained glove model for embedding, but the experiment failed couple of times. Some of the failures were due to the law speed internet connection and the fact that we have to use VPN to be able to connect to Kaggle website and run our experiments. In addition, the time limit for the project did not allow us to investigate in all the directions we want, so we focused on BERT-based models.

Furthermore, we tried to use the Off-shelf BertForSequenceClassification to tackle the classification problem but failed for the same above reasons.

6. Results

A brief summary of our submissions is presented in Table1. As expected we managed to improve the performance as we improve the model and pre-processing. Except for Exp.5.A where training on the whole auxiliary dataset did not improve the performance, which we think is because the model starts to overfit on it, while it is too different from the training dataset.

Implementation	ROC Score
Exp.1: Baseline m-BERT	0.8236
Exp.2.A: Utilize the auxiliary dataset	0.8613
Exp.2.B: No auxiliary dataset	0.8653
Exp.3.A: Improved classifier	0.8782
Exp.4.A: Primary/$\frac{1}{2}$auxiliary separately	0.8835
Exp.5.A: Primary/auxiliary separately	0.8668
Exp.6.A: Ensemble multiple predictions	0.8851
Pre-Trained Transformer [HuggingFace] v2	0.8782
Pre-Trained Transformer [HuggingFace] v3	0.8835
Ensemble [m-BERT + Pre-trained] v1	0.9269
Ensemble [m-BERT + Pre-trained] v2	0.9315
Ensemble [m-BERT + Pre-trained] v3	0.9318

Table 1. Brief summary of our submissions

7. Conclusion

References

[1] Jigsaw multilingual toxic comment classification. https://www.kaggle.com/c/jigsaw-multilingual-toxic-comment-classification. Accessed: 2020-06-13.

[2] Jigsaw unintended bias in toxicity classification. https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/data. Accessed: 2020-06-13.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.