CS230

Unbiased Toxic Comments Prediction using Adversial Bert (Natural Language Processing)

Raghura Kowdeed*
Department of Computer Science
Stanford University
rkowdeed@stanford.edu

1 Introduction

Machine learning touches everyone's life and has huge influence on many decision makings such as which news article to read , which product to buy etc . Given its impact on society , it is very import to understand its pitfalls and address them in a rigorous way . Given that ML models are trained on huge data that is available online with millions of trainable parameters, these models tend to be sensitive to data distribution . The goal of this project is to build a Deep learning Model to classify toxicity of a given sentence. The input to the model is word embedded representation of a sentence and the output would be classifying text as toxic or not . Paper [1] provides details on how this classification training task suffers from biases by associating identity features to toxicity irrespective of content of the text. This failure is caused by the unbalanced train data and model trying to improve the train accuracy ignoring false positive bias. Aim of this work is to improve the accuracy of model and remove unintended biases from it.
To start with , I use a model based on Distil Bert [2] architecture to train on this data and do error analysis to gain insights on model behaviour. this base model is further improved by adding adversarial layer which helped in de-biasing the model.
the code for this project is available here `https://github.com/Raghuramkowdeed/cs230`

2 Related work

Fairness in AI, specially in natural language processing is a topic of interest given its widespread usage across internet . Paper [3] points out that NLP models picks up biases from word embedding and proposes a method to de-bias the word embedding. Paper [1] discusses this problem in detail proposing new metrics to quantify model fairness and attributes model bias to skew in data and proposed data augmentation as a solution. [4] and [5] proposed adversarial learning based architecture [6] to constrain model to be fair.

3 Dataset and Features

The Data and Problem statement are taken from the following kaggle competition.
`https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/overview`.
This data set contains 1700k train samples, 170k test samples . each entry contains comment text and

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

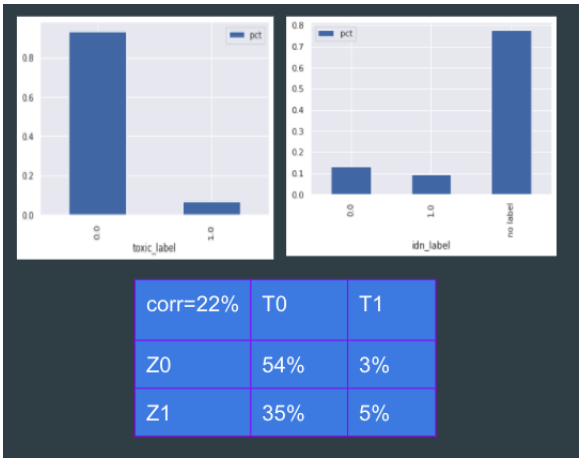


Figure 1: data stats

toxicity label associated with it . It also contains several subclass toxicities ,identity features (men, women etc)
Here is an example from data set :

- Text : haha you guys are a bunch of losers.
- Target (toxicity) : 0.89
- Insult (meta attribute) : 0.88
- Female (identity) : 0.0
- Few more attributes

Below are useful statistics on the data

- dataset is highly unbalanced with 90 percent of negative examples
- only 25 percent of data contains information on identity features . identity features can be used to de-bias the model .
- there is significant positive correlation (22 percent) between identity feature and target, hinting at potential issue with data distribution.

4 Data Sampling

Given the unbalanced nature of dataset , i use the following the data sampling technique to train model on each batch.

- pre-process comment-text by removing special characters, tokenizing etc . use 0.5 as threshold to assign 0, 1 label.
- assign max identity score across all subgroups as sentence identity score.
- split train data into two , one containing positive samples, other containing negative samples. note that negative dataset is ten times larger than positive dataset.
- create a batch by taking half of the samples from positive dataset sequentially and rest from sampling randomly from negative dataset.
- this approach ensures that each batch has balanced labels .
- by end of the epoch, model is trained on all positive samples and fraction of negative samples,
- since samples are randomly chosen from negative dataset, model is trained on different negative samples in every epoch.
- within both positive and negative samples, data is resampled again to get equal number of samples with low and high identity scores.

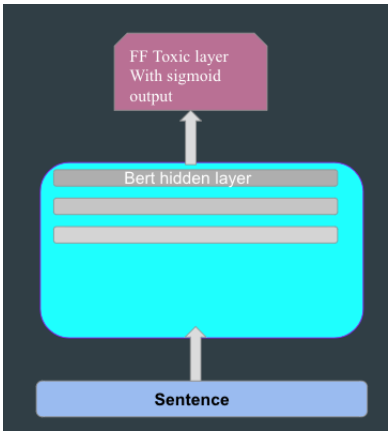


Figure 2: base model

5 Methods

5.1 Bert + FF layer

The model (fig 2)uses pre-trained Distil-Bert [7, 2] (trained on glue task `https://gluebenchmark.com/`) as the back bone to classify the sequence. (from library`https://github.com/huggingface/transformers`)

- given a sentence, tokenizer splits into words and assign ids to it .these word embeddings are fed into series of self attention layers .
- take the word embedding of last 4 hidden layers and concatenate them .
- apply feed forward network with sigmoid activation to output the probability of toxicity for each word.
- assign max score of the words as sentence toxic score. max pooling makes sense for this task as any word with high toxicity makes sentence toxic.
- train the model by minimizing mse loss .

list of important hyper parameters

- learning rate
- dropout prob
- number of hidden outputs from distil-bert to pass into logistic layer = 4
- pooling of word embedding into sentence embedding
- dimension of feed forward network .
- choice of loss function (MSE vs BCE)

after training models with different set of hyper parameters , the following set of hyper parameters were chosen 3.
$MSElossfunction$,$dropout = 0.25$, $learning_rate = 3e-06$, $epochs = 20$. $FFlayers = 1$

5.2 Adversarial Bert + FF layer

adversarial bert (fig 4) architecture same as previous model plus identity prediction layer. This model minimizes both toxic label loss and identity label loss. The idea here is to train the hidden layers of the Bert without loading on to identity features .The shared Bert layers are updated to minimize the loss associated with toxic labels while maximizing the identity labels loss .

Loss func	pooling	AUC
MSE	First token	0.94
MSE	Max	0.965
BCE	First token	0.935
BCE	**Max**	**0.96**

Figure 3: hyper parameters metrics

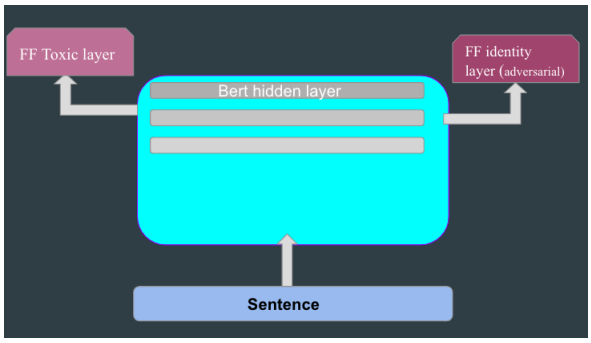
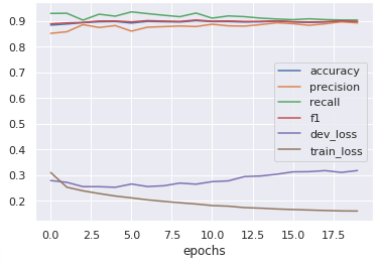
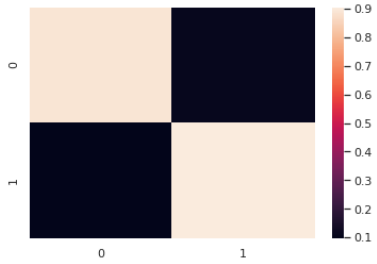


Figure 4: adv model

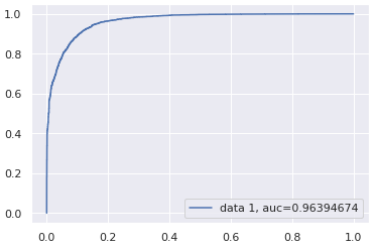


[loss]

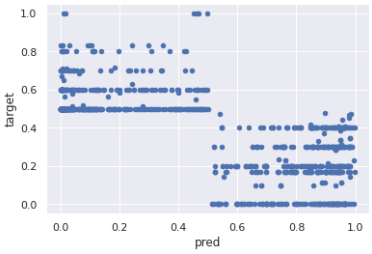


[conf mat]

Figure 5: Model Results



[auc]



[target vs pred]

Figure 6: Model Results

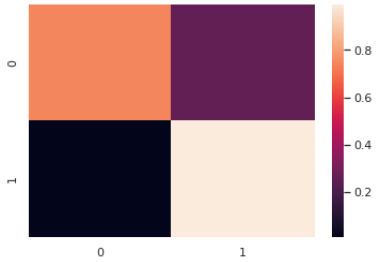


Figure 7: conf mat with high idn

Metric	Base Bert	Adv Bert
AUC	0.96	0.96
Kaggle Score	0.929	0.93
False Neg	0.11	0.16
False Pos	0.10	0.07
False Neg with iidn_Score=1	0.01	0.2
False Pos with iidn_Score=1	0.26	0.12

Figure 8: metrics

6 Result

Fig 5 shows base model train loss , dev loss and other metrics plots. this plot ensures model convergence with no over fitting . below are base model's metrics .

- accuracy 0.89
- precision 0.89
- recall 0.9
- f1 score 0.89
- auc 0.96
- kaggle score 0.92

Since this project is based on Kaggle competition, model is evaluated in kaggle leader board. this submission got a score of 0.92 while the highest score in leader board is 0.94. Both false positive , false negatives are around 10 percent. figure 6 is the scatter plot between target and pred values for the mis-classified labels. most of these false positives are due to model bias towards identity . for example, model pred score for below sentence is 0.9 due to presence of religious word even though sentence is not labelled as toxic.
'so punishing the baby by killing it in the womb is a christian stance.'
Indeed model has high false positive rate (25%) (refer fig 7) within samples with high identity score (> 0.5).
Adversarial model performance is similar to Base model on the dev set fig 8 .But the performance varies when dev data conditioned on high identity score (fig 8, 7, 6). Base model has high false positive rate on samples with high identity score,where as adversarial model has low FPR on this subset of data but has high high FNR . This behaviour excepted as adversarial model is tuned to make FPR lower . since these two models complement each other , i tried ensemble of two models with equal weight. The ensemble model outperformed both models with kaggle score of 0.934.

7 Conclusion and Future Work

After train the bert model for sentence classification , it was indeed conformed that model shows undesired behaviour on the subset of data. Adding the adversarial layer prevents the model from becoming unfair at the expense of high FNR. As part of future work, It would be interesting to see

- more hyper parameter tuning to see if model performance improves
- explore other state of the art NLP models such as GPT-2 to train on this task
- fix the data skew issue using data augmentation and richer data sources
- exploring novel loss functions, metrics that account for bias issues.
- build a systematic framework to identity and address data distribution issues coming up with techniques that make deep learning models more robust to data distribution issues .

References

[1] Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Measuring and mitigating unintended bias in text classification. 2019.

[2] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

[3] Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings, 2016.

[4] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning, 2018.

[5] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H. Chi. Data decisions and theoretical implications when adversarially learning fair representations, 2017.

[6] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

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