583 Final Project Report

Yu Hong, Chen Liao

December 2019

1 Summary

We joined an inactive Kaggle competition with late submission about detect the toxic comments among lots of regular comments. The final model we chose is LSTM, which implement pretrained Glove and Gensim dictionaries to build embedding matrix. We preprocessed the data on Google Colab with 12G GPU and 4G memories and ran the model on Kaggle kernel with 5G memories and 13G GPU. The performance is evaluated by Jigsaw biased AUC. Due to late submission, we don't have a public leaderboard score ; In the private leaderboard, our score is 0.92550, considered the scores , we ranked 1953 out of 2596 teams.

2 Problem Description

Problem: the problem is to build a regression model to detect the toxic comments. This is a binary regression problem. The competition is at https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/data.

Data: the data are comments with different labels created by some experts. The target label is the degree of the toxicity, and there are some sub-labels described the type of the toxic comment. The number of training sample size is 1804874 the range of the toxic score is [0,1]. The training data is imbalanced, where 95% of the samples are not toxic. The result of the prediction will be evaluated by Jigsaw Biased Score on Kaggle, so we will also report this score.

Challenge: the data is imbalanced, there are some meaningless columns in the data, most of the dirty words in the comments were masked by *asterisk or have other undetectable symbols like emoji, to get a better result we need to remove them from the data and restore the changed word.

About the mismatch of proposal and this report: Initially we chose the competition Understanding-Cloud-from Satellite Images (https://www.kaggle.com/c/understanding-cloud-organization). However we found that Google Colab is unable to handle a dataset over 10G, so we switch to Identify the Toxic Comments (https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification)

About the small epoch and the Metric report: Due to the constraint of notebook commitment, we are not able to train the model with enough epoch.

Due to the dataset is highly biased, we will not report accuracy, instead we will report the Jigsaw bias score, which is the evaluation metric of Kaggle.

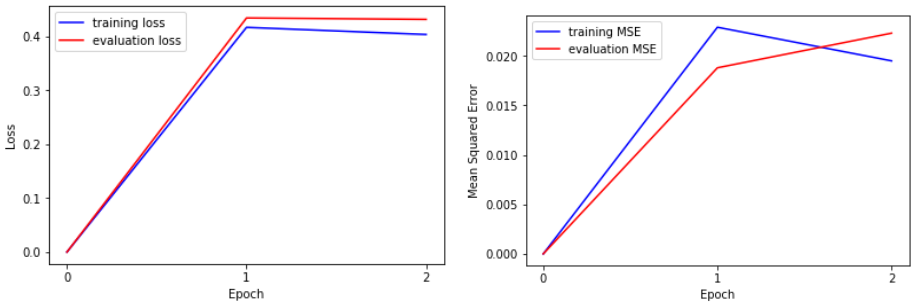
3 Solution

Model: the model we finally chose is LSTM model, and the embedding dictionaries we chose are Genism-300d and Glove-300d, these two dictionaries can convert a word into a 300-dimension vector. More advanced description about these two dictionaries can be found at: https://en.wikipedia.org/wiki/Gensim and https://nlp.stanford.edu/projects/glove/

Implementation: We implemented a bi-directional LSTM with dropout using Keras with Tensorflow backend. Our code is available at https://github.com/MadderPete/handmade_algorithm/blob/master/Toxic%20Comment%20LSTM.html and https://github.com/MichelleCLiao/CS583Fall/blob/master/cnn%26resnet.html. Due to the competition submission constraint (kernel less than 9hr for GPU or 20 hr for CPU), we did the preprocessing in Google Colab (GPU runtime), then trained the model in Kaggle kernel (GPU). We preprocessed the data on Google Colab with 12G GPU and 4G memories and ran the model on Kaggle kernel with 5G memories and 13G GPU. Then It takes 2 hours to train the model. Baseline models were run locally on MI laptop with Intel i7 CPU, 16G Memory, NVIDIA GeForce GTX 1060(laptop-version) 4G GPUs, and Macbook Air with Intel Core i5 CPU, 8G Memory.

Setting: the loss function is binary cross-entropy, The optimizer is Adam. The evaluation metrics is mean squared error.

Model performance: We tuned the model and ran it with 2 full epochs on 1.8 million samples. We randomly chose 20% of the training set as evaluation data and the rest of 80% as training data. Figure 1 plots the convergence curve of the 80% training data and 20% evaluation data.



4 Compared Methods

CNN. We used Convolutional Neural Network (Keras). Using CNN we expected the model can extract some key words that makes a comment toxic, such as some

dirty words or some racism words. The Jigsaw biased score is 0.755.

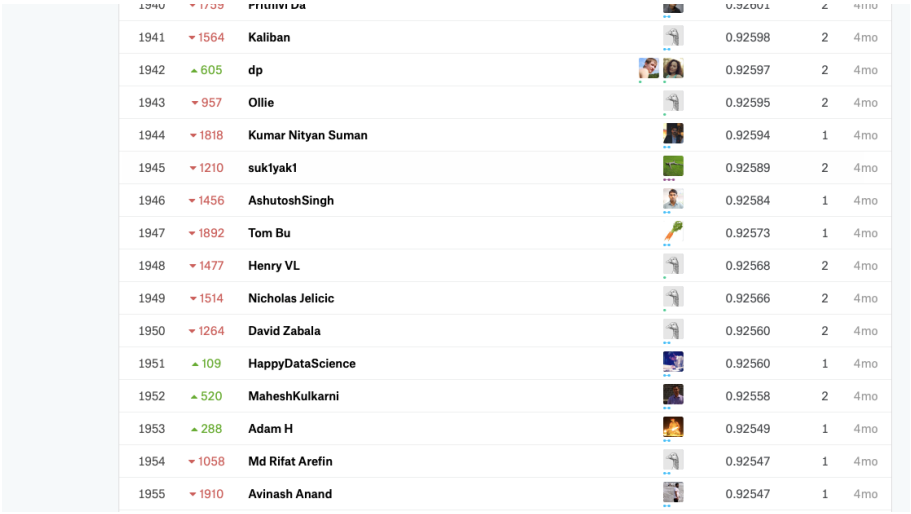
ResNet. We used the structure of ResNet to train the model. This structure is to enhance the power of key word extraction of shallow CNN. The Jigsaw biased score is 0.71.

SVM. We used the Support Vector Machine provided by sklearn. The Jigsaw bias score is 0.501.

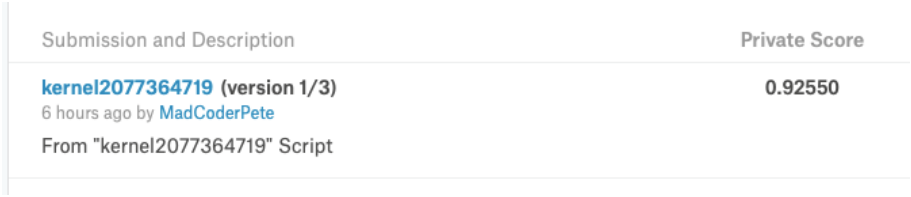
GBR. We used the Gradient Boosting Regressor provided by sklearn. The Jigsaw bias score is 0.51.

5 Outcome

We anticipated an inactive competition. Our public score is, with a rank of XX, and our private score is 0.9255, with a rank of 1953 out of 2596 teams.



1941	▼1564	Kaliban	0.92598	2	4mo
1942	▲605	dp	0.92597	2	4mo
1943	▼957	Ollie	0.92595	2	4mo
1944	▼1818	Kumar Nityan Suman	0.92594	1	4mo
1945	▼1210	suk1yak1	0.92589	2	4mo
1946	▼1456	AshutoshSingh	0.92584	1	4mo
1947	▼1892	Tom Bu	0.92573	1	4mo
1948	▼1477	Henry VL	0.92568	2	4mo
1949	▼1514	Nicholas Jelicic	0.92566	2	4mo
1950	▼1264	David Zabala	0.92560	2	4mo
1951	▲109	HappyDataScience	0.92560	1	4mo
1952	▲520	MaheshKulkarni	0.92558	2	4mo
1953	▲288	Adam H	0.92549	1	4mo
1954	▼1058	Md Rifat Arefin	0.92547	1	4mo
1955	▼1910	Avinash Anand	0.92547	1	4mo



Submission and Description	Private Score
kernel2077364719 (version 1/3) 6 hours ago by MadCoderPete From "kernel2077364719" Script	0.92550