Toxic Comment Classification Challenge

Wenyi Jones
Department of Computer Science
Stanford University
wenyi503@stanford.edu
https://github.com/pennydew/ToxicFinal

December 15, 2018

Abstract

In this toxic comment classification project, we aim to identify vulgar or toxic language in online comments. The data are of Wiki Page's online comments from Kaggle.com. This is a binary classification of six possible labels - toxic, severe toxic, obscene, threat, insult, and identity hate. After testing, we found that LSTM RNN produced the best performance with 96% accuracy across all six categories.

1 Introduction

There is a great deal of obscene language on the internet nowadays, often because cyber identities cannot be tied easily to real-world identities. This behavior could lead to serious consequences and cause harm to individuals, especially those of the younger and more impressionable generation. Being able to flag and review toxic comments provides a more civil community for people to share opinions and collaborate. We trained a deep neural network (DNN) and recurrent neural network (RNN) using different techniques to classify toxic comments. We investigated and compared both models.

2 Related work

With deep learning taking off in recent years, social media have been investigating ways to use AI algorithms to flag and remove offensive language. Most of previous work apply text classification. Specifically focusing on hateful and/or antagonistic content, Greevy and Smeaton (2004) classified racist content in Web pages using a supervised machine learning approach with a bag-of-words (BoW) as features. A BoW approach uses words within a corpus as predictive features and ignores word sequence as well as any syntactic or semantic content.[1] One interesting approach consists on using style transfer techniques to translate offensive sentences into non-offensive ones[2].

3 Dataset and Features

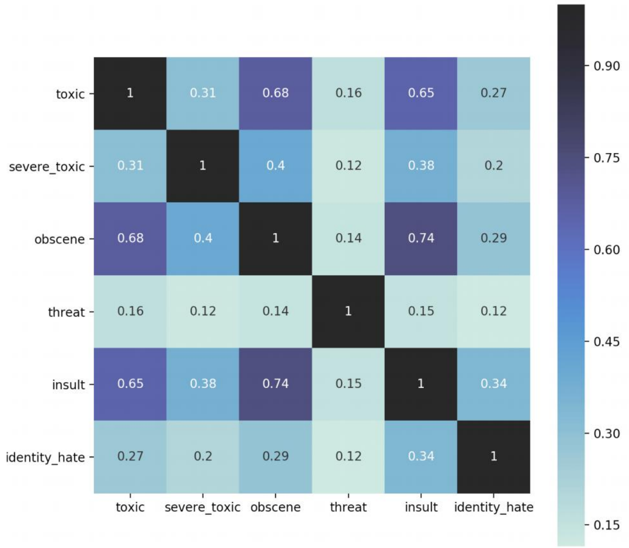


Figure 1: correlation heatmap

The data sets are from a Kaggle competition "Toxic Comment Classification Challenge: Identify and classify toxic online comment[3]." There are about 160,000 examples for both training and testing. Each column is a binary classification of the following six categories: toxic, severe toxic, obscene, threat, insult and identity hate. A comment can be flagged for multiple categories and it is always labelled as "toxic" if any of the other five categories are labelled 1. For example, a label of (1 0 0 0 1 0) indicates this comment was toxic and insulting. An exploration of data shows about 10% of training examples were labelled as toxic or worse.

4 Methods

For all models employed, we first converted words to integers and built a dictionary. We then converted comments to integers and constrained the maximum sentence length to 435. As learned in class, we knew that RNN was commonly used for language predictions, therefore we started with DNN to establish a baseline. We ran the DNN model with different parameters and chose three of the most representative to discuss. For these three DNN models, they all have three fully connected layers and use softmax as activation for the last layer. The RNN has one layer of LSTM followed by 1 layer of fully connected network.

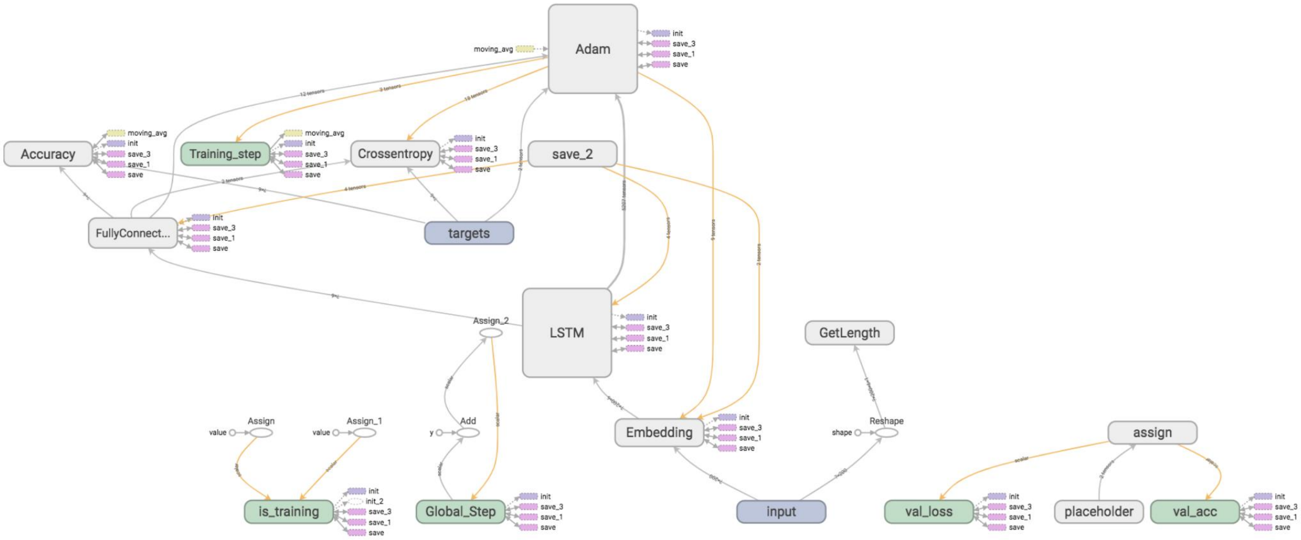
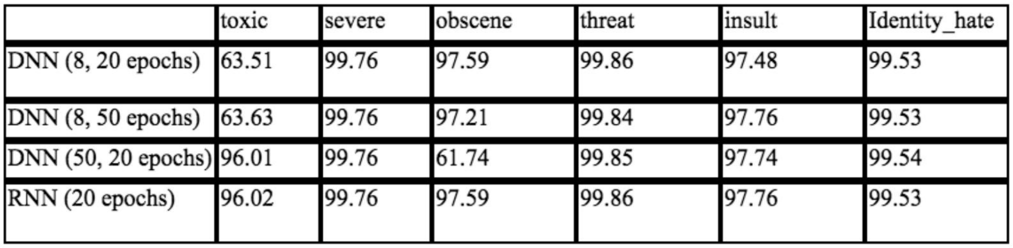


Figure 2: from TensorBoard

5 Results



	toxic	severe	obscene	threat	insult	Identity_hate
DNN (8, 20 epochs)	63.51	99.76	97.59	99.86	97.48	99.53
DNN (8, 50 epochs)	63.63	99.76	97.21	99.84	97.76	99.53
DNN (50, 20 epochs)	96.01	99.76	61.74	99.85	97.74	99.54
RNN (20 epochs)	96.02	99.76	97.59	99.86	97.76	99.53

Figure 3: accuracy results

For the two DNN models with the same size of hidden layers and different epochs, there is virtually no difference in terms of accuracy results. For the third DNN model with 50 fully connected nodes, toxic's accuracy improved dramatically just as much as obscene's accuracy dropped. The LSTM RNN model has the best performance with great accuracy across all six classifications.

6 Conclusion/Future Work

It comes as no surprise that RNN LSTM has the best performance with great accuracy across all six category classifications. LSTM allows us to compute the hidden state and thus great for predicting sequential words. There are many future steps to be taken to improve training and increase accuracy. First of all, better considerations can be taken with data pre-processing. We constrained the maximum sentence length to 435 as 75% of the data have length of less than 435 words. An intuition behind this decision was that toxic words were likely to first appear early rather than late in a sentence. But given more resources, we could remove this constraint. In addition, there were non-English words and characters that could be removed or translated before being passed in for training. We could also apply glove dict words split (e.g SSSfuck $->$ fuck, F...U.c.k $->$ fuck). Secondly, as for models, we expected RNN to perform well and could employ other LSTM RNN models such as pooled RNN and ensemble to achieve diversity.

7 Contributions

I worked on this project by myself and received guidance from class TA Ahmadreza Momeni.

8 References

1 Burnap, Pete & Williams, Matthew (2015) Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & Internet 7(2):223-242*

2 Nogueira dos Santos, Cicero & Melnyk, Igor & Padhi, Inkit (2018) Fighting Offensive Language on Social Media with Unsupervised Text Style Transfer *arXiv:1805.07685*

3 `https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge`