Machine Learning Engineer Nanodegree

Capstone Proposal

Lilit Sargsyan
March 30th, 2018

Proposal

Domain Background

Sharing your thoughts online about things you care can be difficult. The threat of abuse and harassment online results on many people stop to express themselves and give up seeking different opinions. According to Pew Research Center[3], 27% of American internet users chose not to post something online after seeing someone being harassed. Toxic language makes it hard to discuss important issues.
The Conversation AI[1] team, a research initiative founded by Jigsaw[2] and Google are working on tools to help improve online conversation. Their research aims to help increase participation, quality and empathy in online discussions. One area of focus is the study of negative online behaviors, like toxic comments (i.e. comments that are rude, disrespectful, or otherwise are likely to make participant to live the conversation).
There is an online competition going on Kaggle[4] to help to solve one of the aspects of this problem.

Problem Statement

The problem that we are going to solve in this project is connected with finding toxic comments in online discussions. One of approaches to solve this problem involves people facilitating discussions, but this is time consuming and requires a large workforce, that's why

[1] https://conversationai.github.io/
[2] https://jigsaw.google.com/
[3] http://www.pewinternet.org/2017/07/11/online-harassment-2017/
[4] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

we need machine learning methods to do it. The model I'm going to build will detect toxic comments and assign to it a type of toxicity like threat, obscenity, insults, and identity-based hate. I'm thinking of adding one more feature for normal comments i.e. when all the other features are zero.

Datasets and Inputs

In this project I'll be using data from Kaggle.com Toxic Comment Classifier competition. It is a public dataset of comments from Wikipedia's talk page edits. Dataset contains large number of Wikipedia comments whit the id which have been labeled by 6 toxicity sub-types (reasons why something might be considered toxic). The labeled annotations are based on asking 5000 crowd-workers to rate Wikipedia comments according to their toxicity (likely to make others leave the conversation).
The types of toxicity are:

- Toxic
- Severe_toxic
- Obscene
- Insult
- Identity_hate

Solution Statement

The solution to this problem is to build a machine learning algorithm that is capable of detecting toxic comments and predicting probability for each of the six possible types of toxicity (toxic, severe_toxic, obscene, threat, insult, identity_hate) for it.

Benchmark Model

So far there are publicly available models served through Perspective API for this problem but they don't show types of toxicity. My algorithm will show probability for each toxicity type besides finding toxic comments.

The leader board on Kaggle shows achieving ROC AUC score 0.97 and higher which will be the aspirational target. Though achieving ROC AUC score higher than 0.96 is more realistic.

Evaluation Metrics

The evaluation metrics for my algorithm is ROC AUC (area under receiver operating characteristic curve). ROC is a set of {tp rate, fp rate} where tp is the true positive rate (positives correctly classified divided by total positives) and fp is true negative rate (negatives correctly classified divided by total negatives). ROC graphs are two-dimensional graph in which tp rate is plotted on the Y axis and fp rate is plotted on the X axis. AUC is the area under ROC curve.

http://people.inf.elte.hu/kiss/13dwhdm/roc.pdf

Project Design

For this project I'll create an algorithm that will take text data i.e. comments as input and return probability for each toxicity level for that text. I'm thinking to use neural networks (convolutional and recurrent layers). For that I'm going to use keras library.
Before starting with the algorithm I'll do data cleaning by getting read of id column and rows that don't contain comment if such exist. After that split data into training and testing sets by using test_train_split from sklearn library using types of toxicity as labels.
Then I'll be using Tokenizer helper function from keras.preprocessing to create an index of the tokenized unique characters (number of unique words will vary 10000-20000). To turn rough text data (comments) into vectors where each entry shows how many time a word accurse in the comment (here the order of words is set and we can chose max number of word to be considered.
After data preprocessing I'll build architecture using keras convolutional and recurrent layers, compile and then train the model. I'll be using different architectures but the layers I'm considering to use are Dense, Embedding Conv1D, MaxPooling1D, GlobalPool1D, LSTM, GRU and Dropout.
Then check how well my algorithm is doing based on ROC AUC score.

Keep tuning parameters need for each step described here until and do additional data preprocessing if needed until I get the best result for the benchmark.

Link to the review