Toxic Comment Identifier

Introduction to Machine Learning (CptS 437), Spring 2021
Professor: Dr. Diane Cook
Hien Duong | Eric Furukawa | Keuntae “Kurtis” Kim

Introduction

- The internet today is easily accessible and widely available to everyone around the world. With so many people accessing the internet and leaving their comments, moderating every single individual’s comment can be a very time consuming and demanding task. Thus, the main goal and motivation of this project is to answer the question; How can we use machine learning to automate this process?
- There are two main goals for our project. The first goal is to be able to classify any comment as being either toxic or not toxic. Once this goal has been accomplished we will incrementally increase categories that will represent the level of toxicity of a comment.
- Our group hypothesizes that the Naive Bayes Classifier will have high accuracy in classifying these comments based on word counts. NBC is known to perform well on such text recognition problems and the concept of toxic language can likely be traced to specific words. However there may be issues in certain edge case comments such as innuendos, chess related comments (being falsely classified as racial hate speech), and comments in different languages or even text formating.

Dataset

- We will be using an internet comment dataset from Kaggle. This dataset already has the training and the testing data split.
- The relevant features are the user ID, comment text, and 5 different categories that classify the comment’s level of toxicity. Each category has a binary value, where 0 denotes false and 1 denotes true.
- The dataset is already cleanly formatted. Our task will be to clean, remove and preprocess any irrelevant features correctly from the csv(comma-separated values) files.
- https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data

Methods

- The first method will be how we accomplish our first goal.
- We are going to first transform the comments into a bag of words. We are then going to transform the bag of words into a tf-idf matrix.
- We will be using the Multinomial Naive Bayes Classifier to fit the training data.
- Once we have our classifier, we will predict the labels for our testing data. Once we have the predicted labels, we will compare them with ground truth and obtain our accuracy.
- The second method will be how we accomplish our second goal.
- The second method is similar to the first method. However, the difference now is that we are going to classify the comments for 5 different levels of toxicity. These levels of toxicity are toxic, severe toxic, threat, obscene, insult and/or identity hate.
- Each one of these categories will have a binary value associated where 0 denotes false and 1 denotes true. If a comment has 0 for all of these categories, it is considered not toxic, which we labeled as clean.
- We will have a classifier for each of these level of categories.
- We will use the same testing method mentioned in the first method to determine the accuracy of each classifier.

Results

- The results are currently for the first goal only
- The classifiers are classifying comments as either toxic or not toxic reporting the accuracy classifiers.
- SKLearn Multinomial Naive Bayes had an accuracy of about 0.91 on the training set and 0.95 on the testing set, when no weights are involved. This shows that there was not overfit and our classifier was able to accurately predict whether a comment is toxic or not.
- It is notable that the testing accuracy was higher than the training accuracy as this is rarely the case. This indicates a great model.
- When weights were involved, SKLearn Multinomial Naive Bayes had an accuracy of about 0.96 on the training set and 0.76 on the testing set. This is a sign of overfit and the results were worse than we expected.

Future Works

- Currently we only have our first goal met.
- In the future we will implement our second goal of classifying comments based on 5 different categories (toxic, severe_toxic, obscene, threat, insult, and identity_hate).
- We will also implement our own Multinomial Naive Bayes Classifier to compare with the SKLearn Multinomial Naive Bayes Classifier.
- Lastly, there may be room to experiment with further data cleanup/modification to improve accuracy, like expanding the stop words list or finding and weighting words commonly creating false positives.

```
Unweighted
------- Training dataset -------
Accuracy:          0.9094565302551514

f-macro:           0.5748256942664751
--------------------------------

------- Testing dataset --------
Accuracy:          0.9507521349664412

f-macro:           0.5468925373435732
--------------------------------


Unweighted finished, now doing Weighted.


Weighted
------- Training dataset -------
Accuracy:          0.955707607531796

f-macro:           0.8974025751476375
--------------------------------

------- Testing dataset --------
Accuracy:          0.7616411167114988

f-macro:           0.5418974842071688
--------------------------------
```