CS 4501: NLP Project Proposal

Andrew Lin
University of Virginia
Charlottesville, VA
acl2mf@virginia.edu

Kaleb Getachew
University of Virginia
Charlottesville, VA
kg9rv@virginia.edu

Abstract

Within these past couple of years, children and many adolescents have been spending large amounts of their time on the internet, switching between their cell phones and computers. It is no wonder that children learn a lot of their profanity and toxic comments from the internet, this can spoil their young minds and skew their sense of morality. By diminishing the impact of the internet's toxicity, the internet will be slightly more inclusive and less toxic for everyone.

1 Introduction

The motivation behind this project is as the world becomes more digital, and people are using technology at a younger age, it is important to stay safe on the internet. People on social media may find messages in their DMs to be very hurtful to read or see a post that may cause them discomfort. In some serious cases, this bullying may cause someone to commit serious self-harm or cause a mental disorder. This project aims to combat cyberbullying, so that people on social media can have a more enjoyable time.

2 Electronically-available resources

Kaggle provies a brief description accompanied by the files at https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data. The resource above will be the focal point of the entire project regarding toxicity analysis.

3 Data

The dataset we plan to use is a toxic comments dataset. These comments come from Kaggle, but are originally from Wikipedia's talk page edits. This topic was brought to light by the Conversation AI team, which is a research initiative founded by Jigsaw and Google to build tools that help improve online conversation.

These comments have then been labeled by human raters for toxic behavior. Within our dataset, there are a total of 8 columns; 6 of which are binary columns of the toxicity levels: Comment-text (string): Comment from Wikipedia Id (string): Unique identifier Binary Toxicity Columns (int): Toxic Severe-toxic Obscene Threat Insult Identity-hate

The existing dataset contains 158,802 unique Wikipedia comments. The dataset is tentatively to be split into training, testing and validation sets using a 70-15-15 split. This should yield an ideal split for the purpose of training and testing the model.

Toxic Class	Count
Identity Hate	1,405
Insult	7,877
Obscene	8,449
Severe Toxic	1,595
Threat	266
Toxic	15,294
Total	34,886

4 Proposed Methods

The proposed solution to this problem is logistic regression. This is because sequence prediction or text generation is not the focus for this project. A good predictor to predict if a comment is toxic and in what way(s) is what is needed.

A logistic regression is a method where the output is a probability of the specific class. The benefits of logistic regression are the interpretability, is nonparametric, and is not as prone to overfitting data.

The evaluation criteria we will evaluate for this logistic regression can be recall, precision, accuracy, and the ROC curve. Recall will tell us the

fraction of true positives the model got correct and precision will tell us out of the predicted positives, how many were true positives. Accuracy can give us a general idea of how well our model is getting things right and the ROC curve can tell us how the model can perform under different thresholds. Also, we can look at the AUC and see how well the model is doing.

5 Implementation Details

The data itself was pulled from the Kaggle competition. However, the data we used was cleaned by another Kaggle user who posted the cleaned comments dataset. That data cleaning was mainly fixing misspellings, getting rid of punctuation, and spelling out abbreviations. Some of the preprocessing performed involved some minor adjustments to better process the data. Contractions were replaced with both components of the word, all components of any abbreviated obscenities were fully written out, symbols were removed, comments were all placed in lowercase, as well as general grammar and spelling was corrected. In addition, we removed certain columns from our dataset like the 'id' and 'set' columns which represented a unique identifier for each comment and classified the record as 'train' or 'test', respectively. Finally, we used the base nltk English stop words package to remove all of the common words and phrases from our comments.

We got features from the TfidfVectorizer() from sklearn and sentiment, subjectivity, length of the cleaned comment, and flags for profanity and "you". To avoid RAM maxing out, we sampled 100,000 rows from the dataset and used that as our main dataset. Before feeding the data into the models, we made a numerical pipeline that standardized the values of all columns. The vocabulary size for the data we used was 127,464.

The main package that was used to perform machine learning was sklearn. In addition, to sklearn, however, we utilized the nltk, pandas, and numpy packages for various operations. Although we did not use the actual code, the dataset that we received was previously cleaned by Zafar (Kaggle), which aided us in the cleaning process.

6 Experimental Results

Base Logistic Regression Models

	Class	AUC	Accuracy	Recall	Precision	F1 Score
0	Identity Hate	0.677998	0.991000	0.359259	0.500000	0.418103
1	Insult	0.579192	0.991567	0.159259	0.623188	0.253687
2	Obscene	0.809678	0.975000	0.624533	0.872174	0.727866
3	Severe Toxic	0.611566	0.989900	0.225589	0.478571	0.306636
4	Threat	0.634597	0.997367	0.269663	0.631579	0.377953
5	Toxic	0.796037	0.951867	0.603497	0.847741	0.705065

Weighted Logistic Regression Models

	Class	AUC	Accuracy	Recall	Precision	F1 Score
0	Identity Hate	0.838822	0.964233	0.711111	0.161752	0.263555
1	Insult	0.922269	0.940500	0.903704	0.121817	0.214694
2	Obscene	0.936197	0.958733	0.910959	0.571931	0.702690
3	Severe Toxic	0.947544	0.965433	0.929293	0.213622	0.347388
4	Threat	0.917492	0.947167	0.887640	0.047763	0.090648
5	Toxic	0.853714	0.899167	0.797552	0.482547	0.601292

The models we tested were a base logistic regression model and a weighted logistic regression model. Initially, we expected the Random Forest models to perform better with our data due to the size and high-dimensionality of our data, however, we did not have enough RAM on Google Colab to fully implement the model.

The base logistic model had near 100% accuracy, however the AUC and recall scores for most of the classes were sub-par. This model would be useless to deploy to combat toxic comments since the recall on the model for the classes range from 16% to 62%. The reason for the poor performance was that our data-set was heavily imbalanced. Since our data contained disproportional amounts of toxic to neutral comments, we had to use other methods to remedy this. To combat the imbalanced dataset, we used a weighted logistic regression model that helped penalize incorrect predictions of the minority class, in this case if a comment was one of the toxic classes, more harshly. The results of this model were very good. The AUC and Recall scores drastically improved, but our precision score suffered. The goal of the model is to stop toxic comments from getting posted, so we believe that precision is not as much of a concern as recall.

The weighted logistic regression model performed the best in classifying severe toxic comments and performed and the worst for classifying identity hate.

Overall, the results of the weighted logistic re-

gression exceeded our expectations. We did not have high hopes of getting such a high AUC and recall score for some of the classes, especially since some of the labels were less than 1% of the dataset, but the weighted logistic regression was able to account for the discrepancy.

Here is the balance of our labels for the dataset we used:

```
Name: sentiment, dtype: int64
0.0    90459
1.0     9541
Name: toxic, dtype: int64
0.0    99114
1.0      886
Name: identity_hate, dtype: int64
0.0    95067
1.0     4933
Name: insult, dtype: int64
0.0    94753
1.0     5247
Name: obscene, dtype: int64
0.0    99023
1.0      977
Name: severe_toxic, dtype: int64
0.0    99705
1.0      295
Name: threat, dtype: int64
```

7 Our Code

Here is the link for the code we made for this project: Our Code

References

Jigsaw/Conversation AI. 2017. Toxic comment classification challenge.

UNICEF. 2021. Cyberbullying: What is it and how to stop it. *UNICEF*.

Zafar. 2017. Cleaned toxic comments.