Toxic Comment Classification Challenge

CONTENTS

1) Problem Statement:

The Conversation AI team, a research initiative founded by Jigsaw and Google (both a part of Alphabet) are working on tools to help improve online conversation. One area of focus is the study of negative online behaviors, like toxic comments (i.e. comments that are rude, disrespectful or otherwise likely to make someone leave a discussion). So far they've built a range of publicly available models served through the Perspective API, including toxicity. But the current models still make errors, and they don't allow users to select which types of toxicity they're interested in finding (e.g. some platforms may be fine with profanity, but not with other types of toxic content).

In this competition, you're challenged to build a multi-headed model that's capable of detecting different types of of toxicity like threats, obscenity, insults, and identity-based hate better than Perspective's current models. You'll be using a dataset of comments from Wikipedia's talk page edits. Improvements to the current model will hopefully help online discussion become more productive and respectful.

2. Data used:

The train dataset has 95,851 rows and 8 columns. The test dataset has 226,998 rows and 2 columns.

Variable used

1. Train column names: id, comment_text, toxic, severe_toxic, obscene, threat, insult, identity_hate
2. Test column names: id, comment_text

3. Exploration of Variable:

The predictor is a single variable in the form of a free text comment.The entire corpus of comments has 6.47 million words.6 variables/columns are used for toxicity classification of the comment.

Summary of TOXIC

0 144277

1 15294

Summary of - SEVERE_TOXIC

0 157976

1 1595

Summary of - OBSCENE

0 151122

1 8449

Summary of - THREAT

0 159093

1 478

Summary of - INSULT

0 151694

1 7877

Summary of - IDENTITY_HATE

0 158166

1 1405

6 variables/columns are used for toxicity classification of the comment.Percentage of each classification are:

'toxic 9.64%, severe_toxic 1.01%, obscene 5.33%, threat 0.32%, insult 4.97%, identity_hate 0.85%'

Class "threat" is the most rare

'Comments with more than one class selected: 5957'

For the 9,237 toxic comments, these are the percentages from other classes that overlap with toxic:

'toxic 100%, severe_toxic 100%, obscene 94%, threat 95%, insult 93%, identity_hate 92%'

We learn that every severe_toxic comment is also toxic, also all other classes are for the most part subsets of the toxic class.

For the 965 severe_toxic comments, thsese are the percentages from other classes that overlap with severe_toxic:

'toxic 10%, severe_toxic 100%, obscene 18%, threat 25%, insult 17%, identity_hate 22%'

For the 5,109 obscene comments, these ar the percentages from other classes that overlap with obscene:

'toxic 52%, severe_toxic 95%, obscene 100%, threat 65%, insult 78%, identity_hate 75%'

For the 305 threat comments, these are the percentages from other classes that overlap with threat:

'toxic 3%, severe_toxic 8%, obscene 4%, threat 100%, insult 4%, identity_hate 8%'

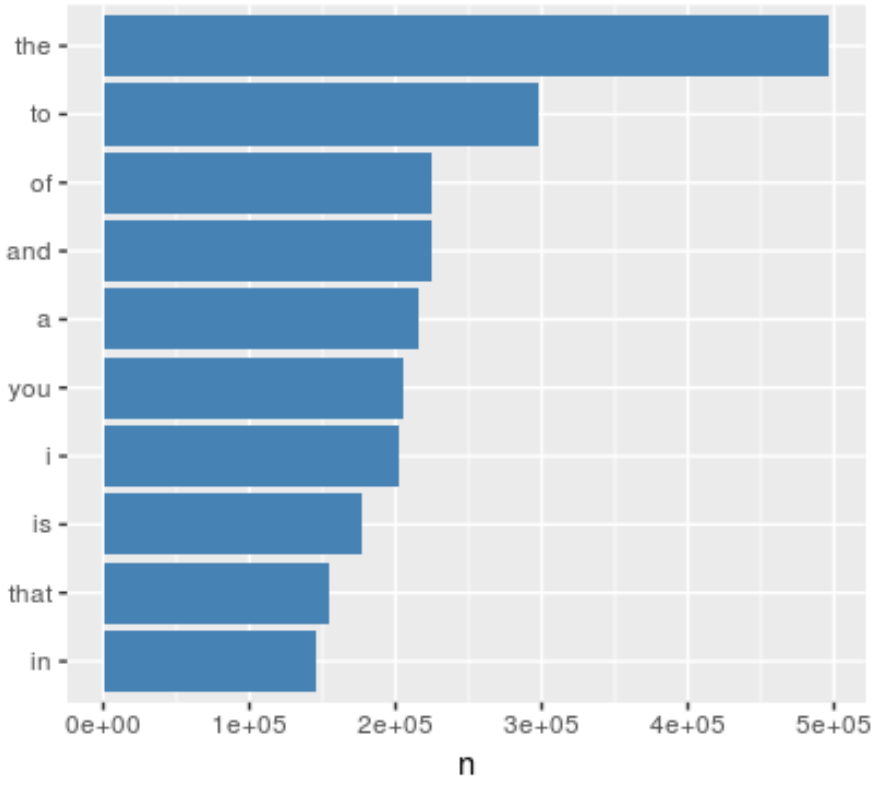
For the 4,765 insult comments, these are the percentages from other classes that overlap with insult:

'toxic 48%, severe_toxic 86%, obscene 73%, threat 66%, insult 100%, identity_hate 83%'

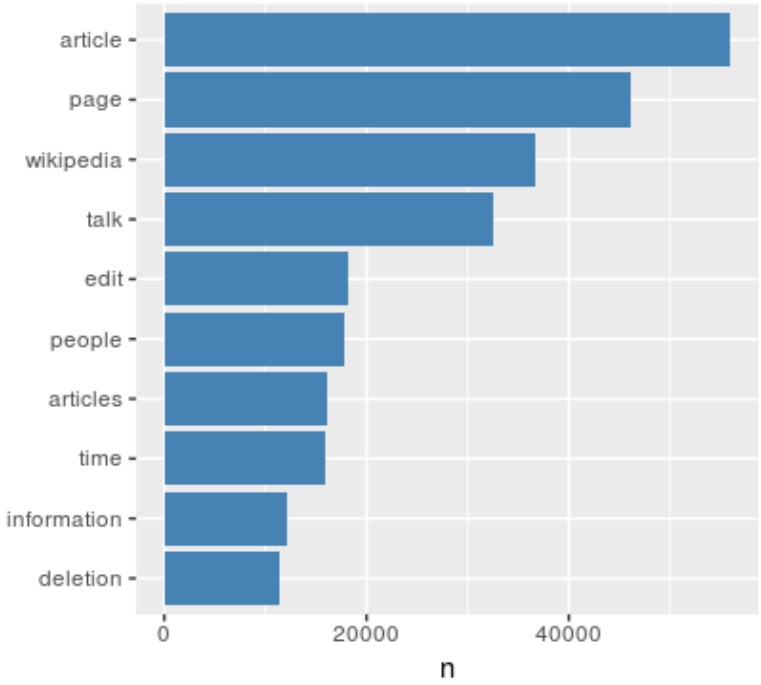
For the 814 identity_hate comments, these are the percentages from other classes that overlap with identity_hate:

'toxic 8%, severe_toxic 18%, obscene 12%, threat 21%, insult 14%, identity_hate 100%

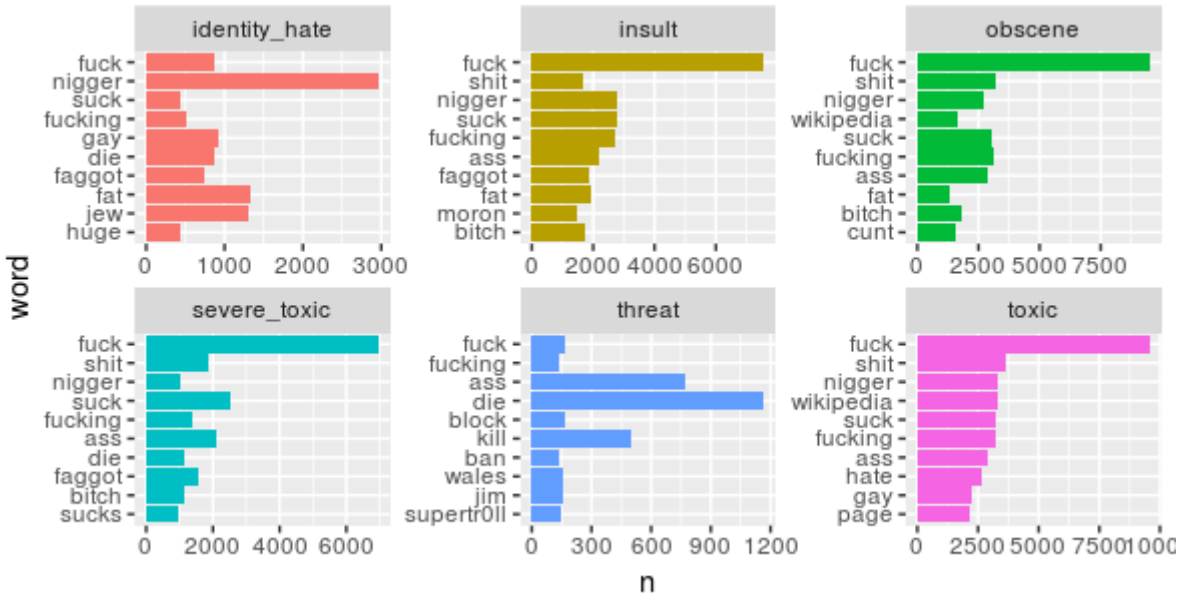
The entire corpus of comments has 6.47 million words. Let's take a look at the top 10 words, before doing any data cleaning.



The top 10 words in the corpus after removing stop words.



Top words for each class



Wordcloud for toxic

Wordcloud for severe_toxic

Wordcloud for obscene

Wordcloud for threat

Wordcloud for insult

Wordcloud for identity_hate

4. Preprocessing of the Variable:

Character variable are preprocessed before modelling.Various library are used for the preprocessing – tidyverse,text2vec,tokenizers etc.Stops words,punctuations,smileys etc were all removed. Unique terms and corresponding statistics were collected using create_vocabulory function. Tf-idf was calculated.Sparse matrix was created on which data was modelled.

5. Building Predictive Models

The evaluation metric for this prediction are evaluated on the mean column-wise ROC AUC. In other words, the score is the average of the individual AUCs of each predicted column.

XGBoost: XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient**,** flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way

Glmnet:XGBoost is an R package which provides Lasso and elastic-net regularized generalized linear models. It features extremely efficient procedures for fitting the entire Lasso or elastic-net regularization path for linear regression, logistic and multinomial regression models, Poisson regression and the Cox model.

Both model combined were used to predict the toxicity of different classes.

The score of mean column-wise ROC AUC - 0.9785

THANK YOU