

Classifying Toxic Comments with Natural Language Processing

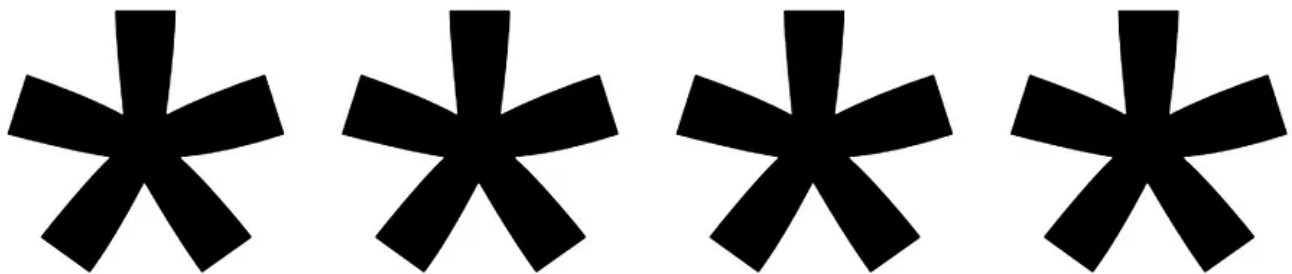


Mars Xiang · [Follow](#)

5 min read · Jul 4, 2020



33



Open in app ↗

[Sign up](#)

[Sign In](#)



Search



Write



Regardless of whether you have a Medium account, Youtube channel, or play League of Legends, you have probably seen toxic comments somewhere on the internet before. Toxic behavior, which includes rude, hateful, and threatening actions, is an issue that stops a productive comment thread, and turns it into a battle.

Needless to say, developing and artificial intelligence to identify and classify toxic comments would greatly help many online groups and communities.

Data

The data for this project can be found on [Kaggle](#). This data set contains hundreds of thousands of comments, each labelled with some of the following traits: toxic, severe toxic, obscene, threat, insult, and identity hate.

The non-toxic comments (with none of labels being true) outnumber the toxic comments approximately 10:1. Here are two examples of a toxic comment, and a non-toxic comment with their labels.

```
["COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK"      1 1 1 0 1 0]
["Your vandalism to the Matt Shirvington article has been reverted.
Please don't do it again, or you will be banned."      0 0 0 0 0 0]
```

Text Processing

Text Processing is transforming and cleaning the text before it is fed into the model for better results. Some common text processing steps include:

- Tokenization — splitting the text into individual words
- Stemming and Lemmatization — converting words into their basic forms
- Stopword Removal — removing common words, such as “to” and “of” that do not contribute to the meaning of the sentence
- Lowercasing — sometimes used for character based models, where tokenization is done on a character level

Since the database was comprised of online comments, many of which would contain misspelled and uncommon words, character level model was chosen. Tokenization, stemming, lemmatiation, and stopword removal were not preformed.

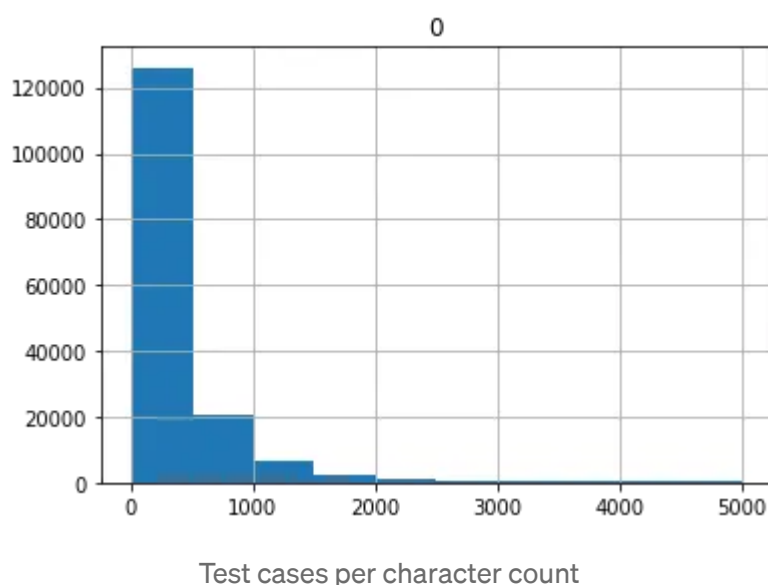
Lowercasing was also not done as consecutive upper case letters are a sign of toxicity.

Data Cleaning

Since the data was unbalanced, oversampling, or adding more examples of the underrepresented class (toxic comments) to the data set was done until the two classes were approximately even.

The data set consisted of a number of unusual characters, such as accented characters, emojis, and Chinese characters. All non-ascii characters were deleted from the examples that contained them. The final character count was 97 unique characters.

Almost all of the test cases were below 500 characters. To convert the data in a form Tensorflow can use, all test cases above 500 characters were deleted. Approximately 20% of the test cases were deleted.



After, all test cases that were below 500 characters were padded with spaces until they were exactly 500 characters long.

Network Architecture

For this project, stacked GRU layers connected to a neural network hidden layer with softmax activation was used to give each test case 6 labels.

- Character Embedding — transforms each character into a meaningful high-dimensional vector. The vector of an uppercase letter would probably be similar to that of a lowercase letter. The vector of a space or punctuation would probably be different from the characters’.
- Stacked GRUs — can work with sequenced data, as the number of layers is equal to the length of the sequence. GRUs can also understand the order of letters. Ultimately, the only output that is passed on to the next layer is the output of the last time step of the last GRU.
- Neural Network — takes the output of the GRU as input and has six sigmoid neurons as output. The six neurons represent the model’s predictions on whether a comment should be labeled with toxic, severe toxic, obscene, threat, insult, and identity hate.

For this project, the embedding layer produced a 128 dimensional vector for each character, and two GRUs with 256 cells each were used.

Training

The model was trained with binary cross-entropy loss and ADAM optimizer, and evaluated with binary accuracy.

The training data set was split into mini-batches of 64 examples. Due to time restrictions, the model was only trained for four epochs in total.

Results

After four epochs of training, the model received a binary accuracy of 98.58% on the test set.

The model seems to use **recognizing swear words and other insults** to decide whether a comment is toxic or not. Whenever “fuck” is a word in the comment, the toxicity and obscene labels jump to around 99%. It makes sense, since all comments with that word are probably labelled as toxic in the data set.

For longer phrases, **capital letters usually have high toxicity** than normal text, as expected, since people use all capital letters to shout.

However, **for shorter phrases, capitalization seems to have a random effect**. The phrase “go aWay” has a toxicity label of 0.059, while “go away” has a label of 0.679, and “GO AWAY” a label 0.312. For shorter phrases, it probably confuses capital letters for many different words, and gets confused.

The model also seems to understand **word phrases** that show toxicity, such as “go away”, “no one likes you”, but is unable to understand how a **complete sentence** is made. “No one likes you so go away” has a toxicity label of 0.769, while “You so go no away one likes” a label of 0.182, and “Likes you no one go away so” a label 0.692.

While the model is not as advanced as humans are, and cannot understand things like sarcasm and implications, toxic comments are generally very obvious to spot. While natural language processing has generally been successful with handling tasks like sentiment analysis, new sub-fields are still far from being solved. Natural language processing is one of the most **exciting and unique** fields in artificial intelligence. I’m excited to see what will come next.

Summary

- Abuse and harassment from toxic comments make it difficult to express your opinions, and to write constructive and productive comments.
- The first step to this project was to perform text processing and data cleaning, which makes the data easier for the model to understand.
- The model consists of a character embedding layer, which creates meaningful vectors from characters, stacked GRUs, since they are naturally effective at processing text, and a softmax layer.

- In the end, the model is able to recognize simple signs of toxicity, and since toxic comments are very straightforward, was able to perform quite well.

Artificial Intelligence

Natural language processing

Machine Learning



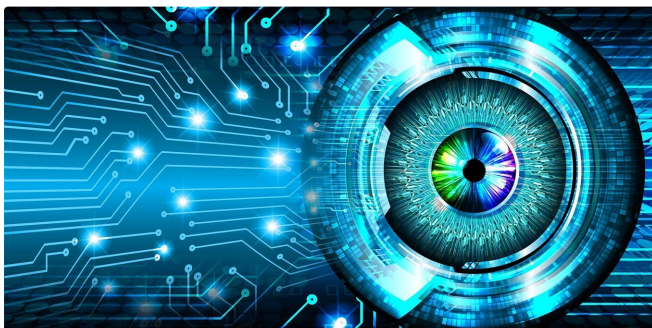
Written by Mars Xiang


Follow

39 Followers

I talk about math and other topics: youtube.com/channel/UCLeCoh8O6YPQ96HP1ttYyeA

More from Mars Xiang



 Mars Xiang in The Startup

Convolutions: Transposed and Deconvolution

Convolutional Neural Networks are commonly used in computer vision problems. But what...



 Mars Xiang in The Startup

Abstractive Text Summarization with NLP

RNNs, LSTMs, and Word Embeddings For Text Summarization