Toxic Comment Classification

Final Report

COMPSCI 273P Machine Learning and Neural Networks
Cutuli, Andre, cutulia, 803641441
Trinh, Linh, thitt2, 49172220
Van Riper, Brenda, bvanripe, 96124814
March 24, 2023

1 Introduction

Discussing things you care about can be difficult. The threat of abuse and harassment online means that many people stop expressing themselves and give up on seeking different opinions. Platforms struggle to effectively facilitate conversations, leading many communities to limit or completely shut down user comments.

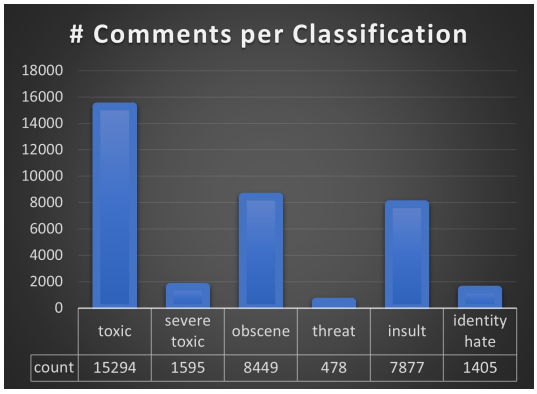
Our area of focus is the study of negative online behaviors, like toxic comments (i.e. comments that are rude, disrespectful or otherwise likely to make someone leave a discussion). We will build a model that's capable of detecting different types of of toxicity like threats, obscenity, insults, and identity-based hate. From a Kaggle competition we will be using a dataset of comments from Wikipedia's talk page for edits. The aim is to make online comments more discussion based, productive, and respectful. Our classification of comments are labelled as one or more of these six categories: toxic, severe, obscene, threat, insult, and identity hate. We have a total of 160,000 training data points and 153,000 in our test data.

We approached this problem knowing cleaning the data would be a huge part of this. After the cleaning we chose to do a Long Short-Term Memory model because it is especially useful in capturing the context and relationship between words in a sentence, or comment in our case. This is particularly important for understanding the meaning of the text and detecting nuances in the language that could indicate toxicity.

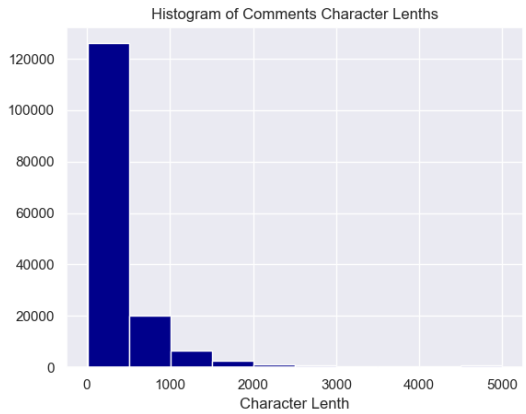
2 Data Exploration

The training data is a collection of 160,000 Wikipedia comments ranging from not toxic at all, to identity hate. 90% of the comments are completely non toxic and the other 10% is classified into one or more of these six categories - toxic, severe, obscene, threat, insult, and identity hate. There are no null values in the train or test cvs files so we do not need to purge any data. In Figure 1a, you can see the distribution of the classifications on the training data, it can be classified as one or more of the classes.

The comments range from one character to up to 5,000 characters with the mean at around 500. The histogram of the character counts, as seen in Figure 1a, for both the training and test data look almost exactly the same. There was a limit set on comment character count to 5,000 for those who had to classify the training set in the competition file.



(a) Bar graph containing the distribution of comment classification among the training dataset.



(b) Histogram of distribution of character count in training data.

Figure 1: Data exploration graphs

3 Data Preprocessing and Feature Design

3.1 Data Preprocessing

For the Toxic Comment dataset we had to do a large amount of data cleaning. Our data preprocessing is how we narrow down the options for feature selection as it determines what words are worthwhile or which can be ignored. Initially we removed punctuation, and used substitutions for contractions, abbreviations, and changing all of our data from uppercase to lowercase. Next, we did text normalization; tokenization to get an array of unique words to removed stop words (i.e. ”and”, ”in”, ”the”,”is”, ”be”, etc.), then snowball stemming. Snowball stemming is a series of transformations that shorten a word of its suffixes and prefixes to get the root word. An example our data cleaning is -

- **Original Comment:** ”omg!! you are an admin and you protect the people who vandalise the ””Bulgars”” article. these attackers are pan-slavic ultra-nationalists and they dont want to see the word beginning with ””Turk””. you help them remove reliable sources. Old Bulgars are changing into slavs by your hand... great!”
- **Comment After Cleaning:** ”oh god admin protect people vandal bulgar article attack panslav ultranation not word begin turk help remove rely source old bulgar change slav hand great”
- **Comment After Text Normalization:** [bulgar, oh, god, admin, protect, peopl, vandalis, bulgar, articl, attack, panslav, ultranationalist, want, see, word, begin, turk, help, remov, reliabl, sourc, old, bulgar, chang, slav, hand, great]

While cleaning the data we noticed comments have random numbers at the end of the message (i.e. .89, .205, .38, .27) and during our cleaning process it merges with the word next to it. This is a potential concern because this may cause unwanted ”spelling errors” that may not get removed during the data cleaning.

3.2 Feature Design

In toxic comment classification, feature design involves selecting and engineering features that can help a machine learning model accurately identify toxic comments.

- **Bag of Words:** This approach involves creating a list of all the words in the training dataset and then counting how many times each word appears in a comment. This creates a vector of word counts for each comment, which can be used as a feature for classification.

comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate	text_clean	tokenized	stopwords_removed	snowball_stemmer	lancaster_stemmer
Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0	explanation\nwhy the edits made under my usern...	[explanation, why, the, edits, made, under, my...	[explanation, edits, made, username, hardcore,...	[explan, edit, made, usernam, hardcor, metalli...	[expl, edit, mad, usernam, hardc, metallic, fa...
D'aww! He matches this background colour I'm s...	0	0	0	0	0	0	daww he matches this background colour i am se...	[daww, he, matches, this, background, colour, ...	[daww, matches, background, colour, seemingly,...	[daww, match, background, colour, seem, stuck,...	[daww, match, background, colo, seem, stuck, t...
Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0	hey man i am really not trying to edit war it ...	[hey, man, i, am, really, not, trying, to, edi...	[hey, man, really, trying, edit, war, guy, con...	[hey, man, realli, tri, edit, war, guy, consta...	[hey, man, real, try, edit, war, guy, const, r...
"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0	\nmore\ni cannot make any real suggestions on ...	[more, i, can, not, make, any, real, suggestio...	[make, real, suggestions, improvement, wondere...	[make, real, suggest, improv, wonder, section,...	[mak, real, suggest, improv, wond, sect, stat,...
You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0	you sir are my hero any chance you remember wh...	[you, sir, are, my, hero, any, chance, you, re...	[sir, hero, chance, remember, page]	[sir, hero, chanc, rememb, page]	[sir, hero, chant, rememb, pag]

Figure 2: The table shows some of our cleaned comments after tokenization, stopped words being removed, and snowball stemmer being applied.

For example, consider if we applied bag of words to comment we had cleaned and normalized earlier:

- **Comment:** "[bulgar, oh, god, admin, protect, peopl, vandalis, bulgar, articl, attack, panslav, ultranationalist, want, see, word, begin, turk, help, remov, reliabl, sourc, old, bulgar, chang, slav, hand, great]"
- **Vector of Word Frequencies:** [1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1]

4 Model Exploration

4.1 Long Short-Term Memory (LSTM)

Long Short-Term Memory is a type of recurrent neural net (RNN) that does particularly well at analyzing text data; it was our first choice of implementation because of its powerful pattern recognition. Using LSTM in our model gives it the power to remember or forget certain words in the comments, letting it decide what parts of information are important enough to keep. Understanding and detecting long term patterns is great for natural language processing in the type of classification we are doing with toxic comments. LSTM also has the advantage that it can process inputs of any comment length and adapt to each individual comment, and it is scalable. or LSTM we used a validation split of 20% of the training data.

In the training set, the model achieved a loss of 0.0363 and an accuracy of 0.9733. Our model was able to predict the correct label for 97.33% of the comments in the training set, and the average loss per comment was 0.0363. In the validation set, the model achieved a loss of 0.0527 and an accuracy of 0.9931. This means that our model was able to predict the correct label for 99.31% of the comments in the validation set, and the average loss per comment was 0.0527.

	id	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	00001cee341fdb12	0.023601	0.000065	0.004357	0.000208	0.005602	0.001977
1	0000247867823ef7	0.001035	0.000001	0.000298	0.000009	0.000242	0.000024
2	00013b17ad220c46	0.004661	0.000031	0.001897	0.000138	0.001767	0.000308
3	00017563c3f7919a	0.027194	0.000014	0.003585	0.000051	0.004405	0.000218
4	00017695ad8997eb	0.800999	0.000745	0.083566	0.000864	0.307945	0.005127

Figure 3: Output table for test data predictions

4.2 Logistic Regression (LR)

Logistic Regression is used for binary classification problems by fitting a logistic function to our data that maps the features to a probability. While training the model will find the best set of weights that minimize the error between predicted and the actual categories by using gradient descent. Once it is trained it calculates the probabilities of a comment being toxic using the logistic function, then applies a threshold value to determine final classification.

The logistic regression model using grid search and cross-validation achieved an average training score of 0.99 and average validation score of 0.97 , with an additional ROC-AUC score of 0.966 indicating a good model performance and has a high ability to distinguish correctly between classes. A higher training score might indicate slightly over-fitting but the difference is relative small, which suggest that over-fitting is not served.

	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	1	1	1	1	1	1
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0

Figure 4: Output table for test data predictions

5 Adaptation to Under- and Over-fitting

- **Early Stopping -** Applied Early Stopping as a safety measure to prevent overfitting due to being unsure of how many epochs this model needs. Early Stopping will occur if the model goes beyond 1 or 2 epoches with no improvements. The model is set to run 8 epochs; however, it stops on epoch 3 or 4.
- **Dropout -** We utilized dropout in the model to randomly ignore 10% of the neurons in the model during training as to reduce reduce interdependent learning among units.
- Keeping the model simple enough with 5 layers (including input and output) to achieve high performance while keeping the model complexity low enough to not risk potential over-fitting.
- The limiting the maximum amount of features to 15,000 for the model while performing extensive data cleaning and reprocessing to reduce the word library down to the most significant words that the model would benefit from.

6 Statement of Collaboration

As a group we discussed how we would approach this problem and what was the best technique to clean the data in order to make it workable to implement models. We all worked together on the data cleaning and then split off - Andre pioneered the LSTM model while Linh engineered our Logistic Regression model. Brenda assisted where needed with the models, did the data exploration, and spearheaded the report document and graphics.