Project Title: Wikipedia toxic comment classification

Karthik Enjeti

enjeti.k@northeastern.edu

IE 7374

Natural Language Processing

Index

Abstract:

Internet is the new punching bag for some folks where they put out their profanity best and shower all the harsh, offensive and ill-themed words out there for everyone to read.

The inspiration for this is the idea of using machine learning to have better online conversations.

Aim to build a model to detect different types of toxicity like threats, obscenity, insults, and identity-based hate in online comments. To address this, I have chosen a simple method of mapping words corresponding to toxic category and a bi-directional LSTM model which helps classify online content.

By building a reliable and deployable model, we can help moderate the content put out by people on talk pages.

Introduction:

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). This often leads to users refraining from participating in online discussions, over the last decade a lot of improvements have been made in moderating online conversations, but these methods are still prone to error. With advancements in NLP and machine learning techniques, multiple machine learning models and methods have been created to address this issue.

The dataset for this project has been made available by jigsaw and Wikipedia on Kaggle

(https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview).

This dataset consists of Wikipedia talk page comments by users classified into 6 categories of toxicity namely toxic, severe-toxic, obscene, threat, insult and identity-based hate.

The dataset consists of 312,735 comments tagged on the above-mentioned categories.

This data has been collected over the years 2011 through 2015 and has been tagged by humans which has been validated by Jigsaw and Google.

To tackle this issue, I have built to methods:

- A simple interpretable method that tags frequency of words with the toxic category. e.g. “kill” word in a sentence corresponds to “threat”
- A Bi-directional LSTM model that classifies comments into the above mentioned 6 categories.

The model based of simple word frequency tagging is naïve and is not accurate, but it is interpretable to a common man, but the bi-directional LSTM model takes context of word into analysis and is far more reliable but far less interpretable.

Background:

Conversation AI team, a research initiative by Jigsaw and Google through Perspective API have built multiple models to moderate online content. These models are primarily deep neural networks.

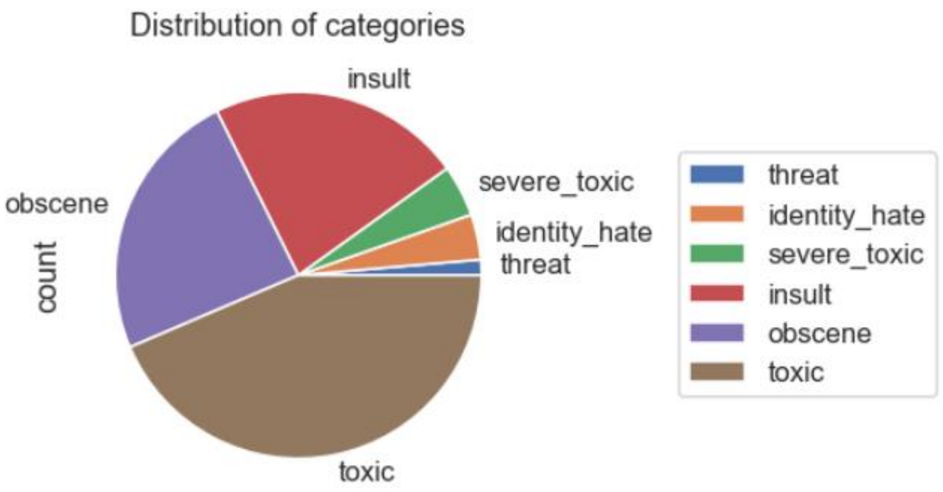
The workflow for these models is partially available on the github page of Perspective API.

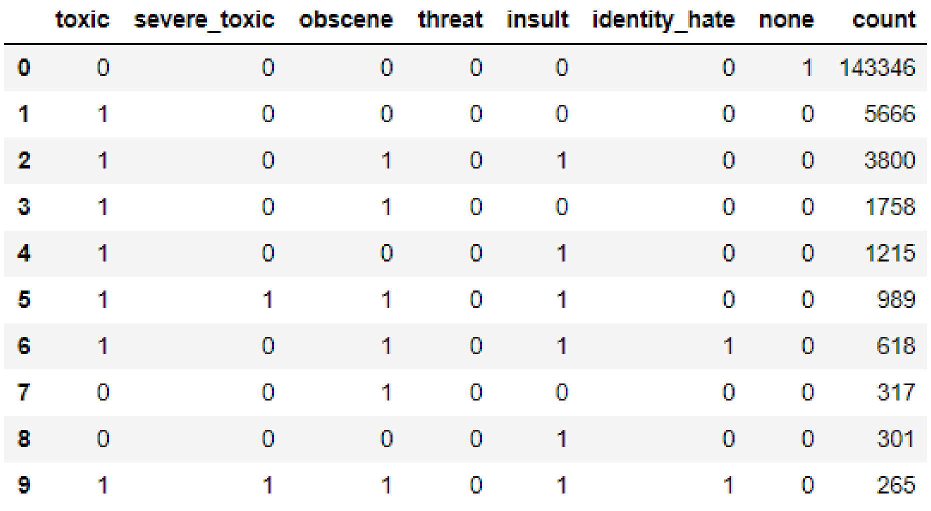
Bi-directional encoder representation of Transformers (Hugging-Face transformers) are current state of art for multi-label classifying using NLP in deep neural networks.

Other popular methods are Bi-directional LSTM, SVM, TF-IDF methods.

Approach:

Initially we will look at the distribution of our data based of the categories we want to classify.



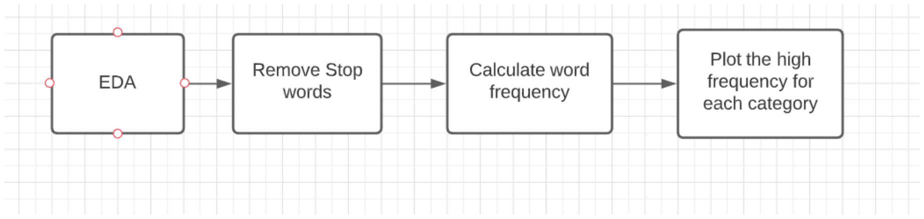


	toxic	severe_toxic	obscene	threat	insult	identity_hate	none	count
0	0	0	0	0	0	0	1	143346
1	1	0	0	0	0	0	0	5666
2	1	0	1	0	1	0	0	3800
3	1	0	1	0	0	0	0	1758
4	1	0	0	0	1	0	0	1215
5	1	1	1	0	1	0	0	989
6	1	0	1	0	1	1	0	618
7	0	0	1	0	0	0	0	317
8	0	0	0	0	1	0	0	301
9	1	1	1	0	1	1	0	265

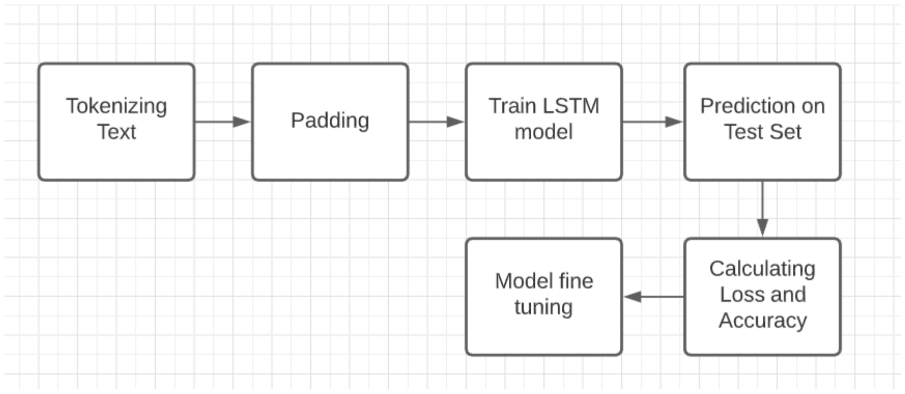
In the second image, “none” represents comments which are not toxic and are acceptable in the online conversations.

As mentioned before we will address this issue through two methods,

- Commonly occuring words based on category of toxicity (Word Analysis)



- Bi-directional LSTM



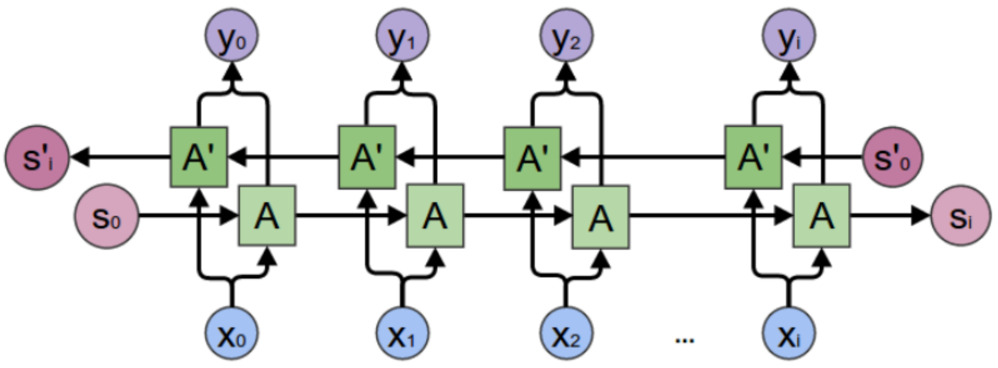
Word Analysis(Method 1)

The aim of this method is to identify most frequent words based off toxic category.

Here we take the raw text and clean it by first converting the text to lower case and then removing words that have low relevance(aka stopwords using NLTK library).

Once we have the cleansed data, we use wordcloud library and calculate frequency of words based on toxicity category and a identify the most frequent words for each category.

Bi-directional LSTM(Method 2)

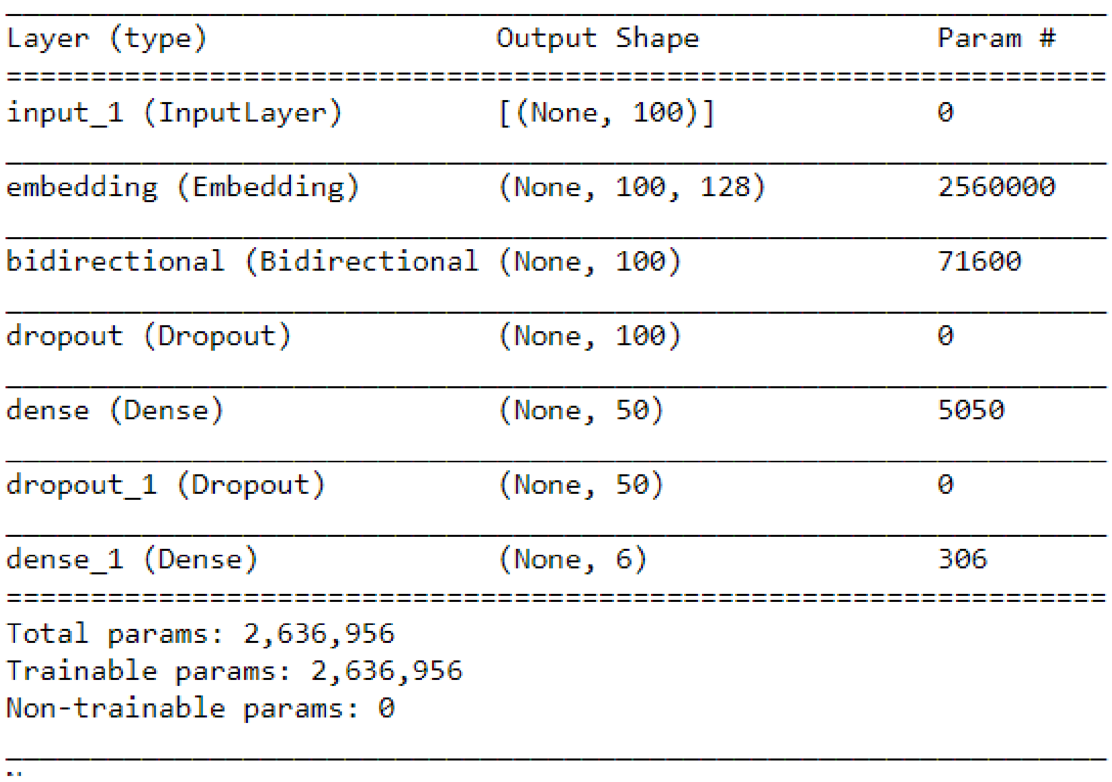


This is a deep neural network method which is far more reliable in classifying text compared to previous method.

As part of pre-processing the data, we tokenize the text and perform padding to the text both training and testing data. The reason we perform padding because the comments are of varying lengths and it is not advisable to have varying word vectors with many null indexes, hence we set max length for vector as 100. This number has been decided based on the average length of comments.

Once these vectors are available, we split them into train and validation datasets and pass them to our model.

The model :



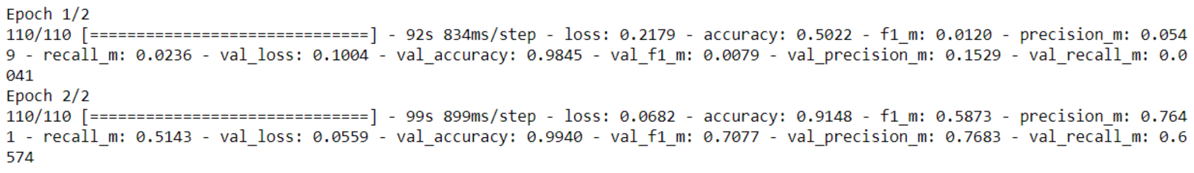
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_1 (InputLayer)         [(None, 100)]             0         
_________________________________________________________________
embedding (Embedding)        (None, 100, 128)          2560000   
_________________________________________________________________
bidirectional (Bidirectional (None, 100)               71600     
_________________________________________________________________
dropout (Dropout)            (None, 100)               0         
_________________________________________________________________
dense (Dense)                (None, 50)                5050      
_________________________________________________________________
dropout_1 (Dropout)          (None, 50)                0         
_________________________________________________________________
dense_1 (Dense)              (None, 6)                 306       
=================================================================
Total params: 2,636,956
Trainable params: 2,636,956
Non-trainable params: 0
_________________________________________________________________
```

Loss-Function: Binary cross entropy

Activation Function: Sigmoid

Optimizer: Adam

We perform early stopping on model to avoid overfitting and get a well-balanced model

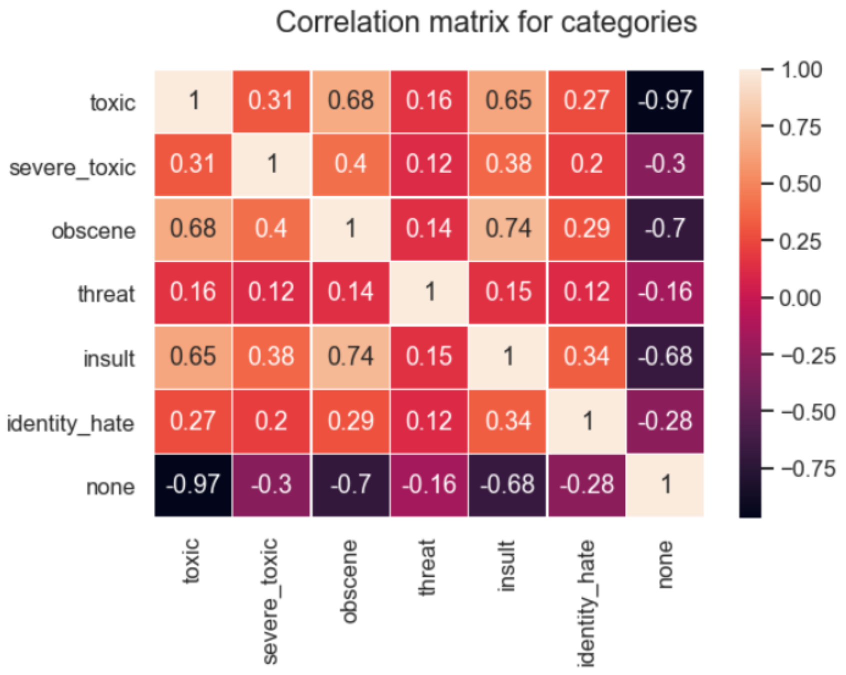


```
Epoch 1/2
110/110 [==============================] - 92s 834ms/step - loss: 0.2179 - accuracy: 0.5022 - f1_m: 0.0120 - precision_m: 0.054
9 - recall_m: 0.0236 - val_loss: 0.1004 - val_accuracy: 0.9845 - val_f1_m: 0.0079 - val_precision_m: 0.1529 - val_recall_m: 0.0
041
Epoch 2/2
110/110 [==============================] - 99s 899ms/step - loss: 0.0682 - accuracy: 0.9148 - f1_m: 0.5873 - precision_m: 0.764
1 - recall_m: 0.5143 - val_loss: 0.0559 - val_accuracy: 0.9940 - val_f1_m: 0.7077 - val_precision_m: 0.7683 - val_recall_m: 0.6
574
```

A major challenge in training neural networks is how long to train them. Too little training will mean that the model will underfit the train and the test sets. Too much training will mean that the model will overfit the training dataset and have poor performance on the test set. We approach this problem by treating the number of training epochs as a hyperparameter and train the model multiple times with different values, then select the number of epochs that result in the best performance on the train or a holdout test dataset. As part of tuning the model we have tweaked the batch sizes, number of epochs and the activation functions and changing the number of layers.

Results:

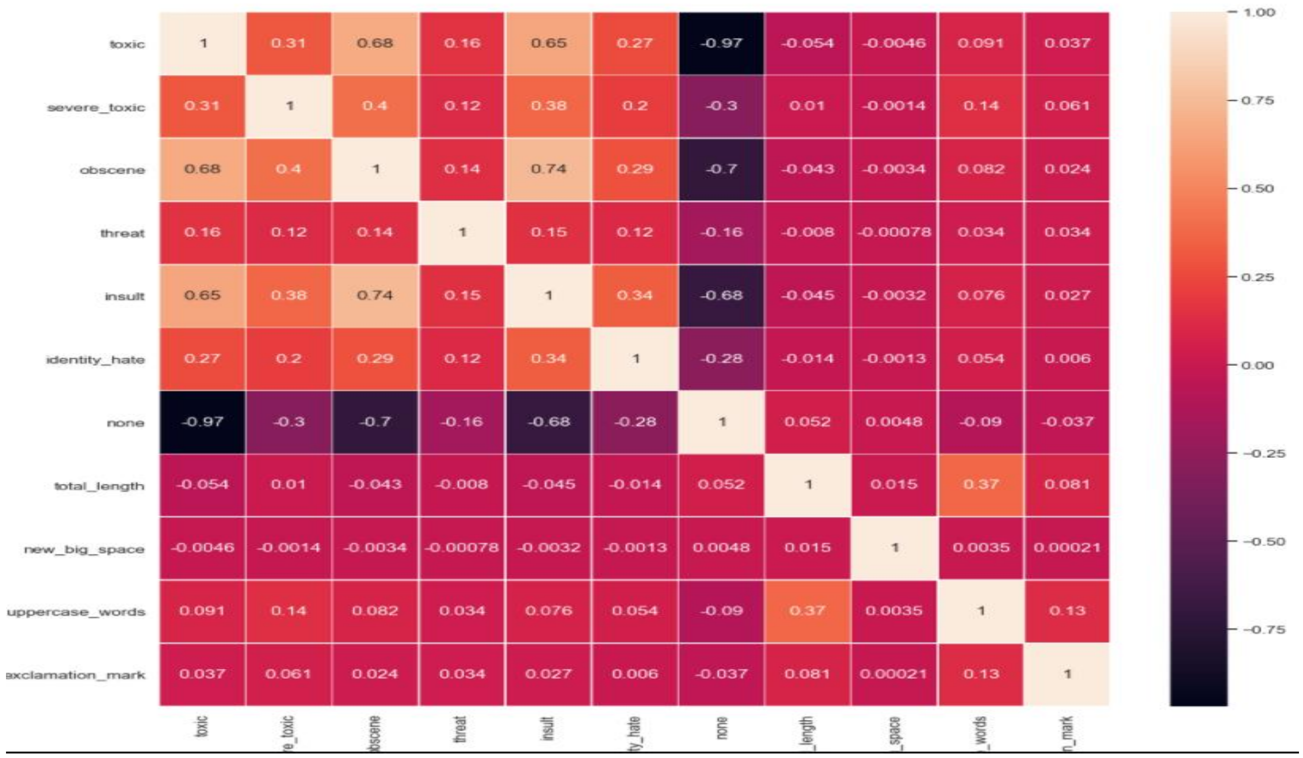
From our initial analysis we notice that these toxic categories are correlated



	toxic	severe_toxic	obscene	threat	insult	identity_hate	none
toxic	1	0.31	0.68	0.16	0.65	0.27	-0.97
severe_toxic	0.31	1	0.4	0.12	0.38	0.2	-0.3
obscene	0.68	0.4	1	0.14	0.74	0.29	-0.7
threat	0.16	0.12	0.14	1	0.15	0.12	-0.16
insult	0.65	0.38	0.74	0.15	1	0.34	-0.68
identity_hate	0.27	0.2	0.29	0.12	0.34	1	-0.28
none	-0.97	-0.3	-0.7	-0.16	-0.68	-0.28	1

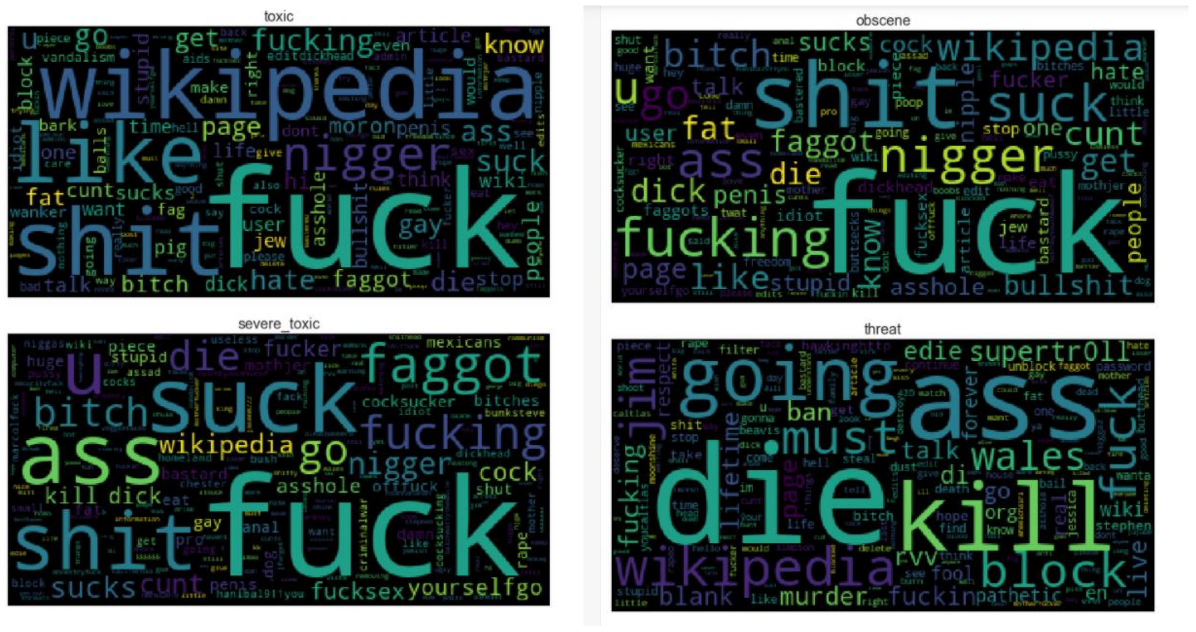
There is a clear distinction between severly toxic and toxic comments in the data which naturally should have been highly correlated. Also we notice that comments that are obscene tend to be toxic too.

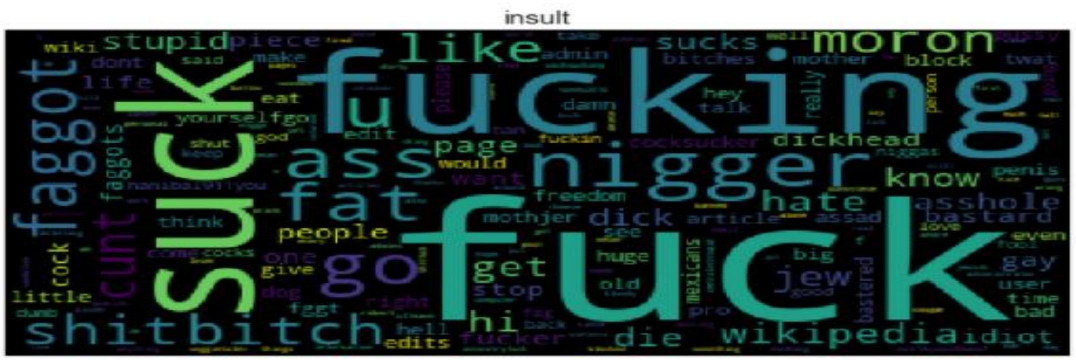
Also comments that are insulting are usually obscene too. With other feature such as length of comment, big spaces in comments, upper case words or exclamation marks, the correlation matrix doesn't yield any connection due to the large corpus of data containing a lot of variation.

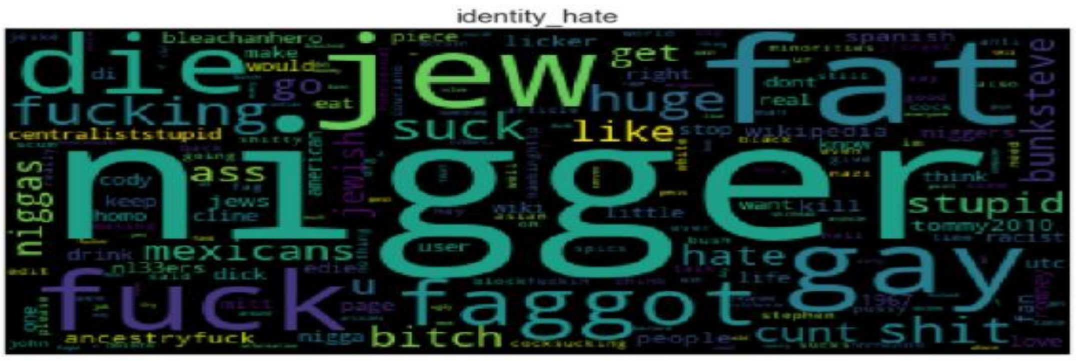


So, we cannot generalize that long comments or comments having a lot of spaces should be moderated.

From our word analysis, the following words occur more frequently for the following categories,







Until here, we have a very naïve way of moderating online comments where we don't take context of comments or sequence of words into analysis.

To provide a more reliable method of classifying comments we have built a bi-directional LSTM.

This model yields the following results on training data

```
Epoch 1/2
110/110 [==============================] - 92s 834ms/step - loss: 0.2179 - accuracy: 0.5022 - f1_m: 0.0120 - precision_m: 0.054
9 - recall_m: 0.0236 - val_loss: 0.1004 - val_accuracy: 0.9845 - val_f1_m: 0.0079 - val_precision_m: 0.1529 - val_recall_m: 0.0
041
Epoch 2/2
110/110 [==============================] - 99s 899ms/step - loss: 0.0682 - accuracy: 0.9148 - f1_m: 0.5873 - precision_m: 0.764
1 - recall_m: 0.5143 - val_loss: 0.0559 - val_accuracy: 0.9940 - val_f1_m: 0.7077 - val_precision_m: 0.7683 - val_recall_m: 0.6
574
```

We achieve a high accuracy of 99.4% on validation data and 99% on test data.

And on the test data

```
In [213]: print(accuracy)

0.9990010857582092
```

This is a highly accurate model that works well for all categories of online comments.

Conclusion:

Classifying comments just based on presence of words is not foolproof and a deep neural network performs reliably. With advancements in deep neural networks , methods such as BERT, LSTM and SVM are considered reliable and yield good results for multi label text classification.

Our dataset comprises of diverse data that has been collected over 4 years and tagged by humans, so there is a general bias in the data and yet the model performs reliably. In online content, toxic comments can be of any form such as images, gifs, videos, audio, text and hyperlinks. Here we - have tackled only text. In some of these comments people have intelligently masked the way they have used cusswords by misspelling them on purpose, to avoid getting caught out. This has been usually handled for common misspellings but not in all cases as there is no norm for the same.

As further scope, we can attempt to find all relevant words with even misspellings and try to moderate audio messages too.

Also, here we have a considered a small subset of online content containing only English words in the comments, but online content can consist of a variety of languages. Building an amalgamation of models that can interact with each other and share information to build a reliable moderation system will be a tough but useful task.

References and Acknowledgement:

https://medium.com/@raghavaggarwal0089/bi-lstm-bc3d68da8bd0

Introduction to Natural Language Processing, Jacob Eisenstein

https://datascience.stackexchange.com/questions/25650/what-is-lstm-bilstm-and-when-to-use-

them
https://machinelearningmastery.com/develop-bidirectional-lstm-sequence-classification-python-keras/

https://machinelearningmastery.com/early-stopping-to-avoid-overtraining-neural-network-models/