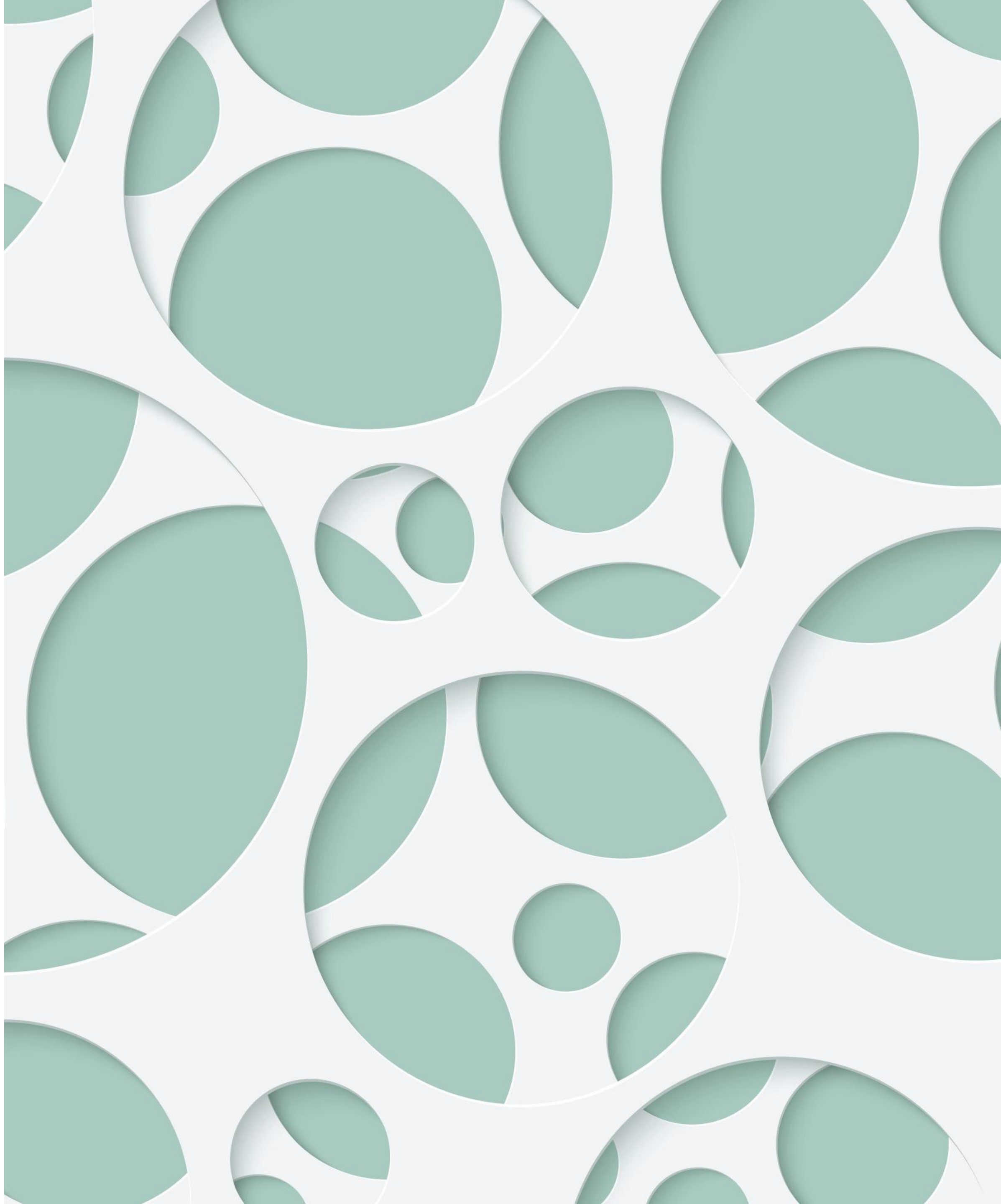
WIKIPEDIA TOXIC COMMENT CLASSIFICATION

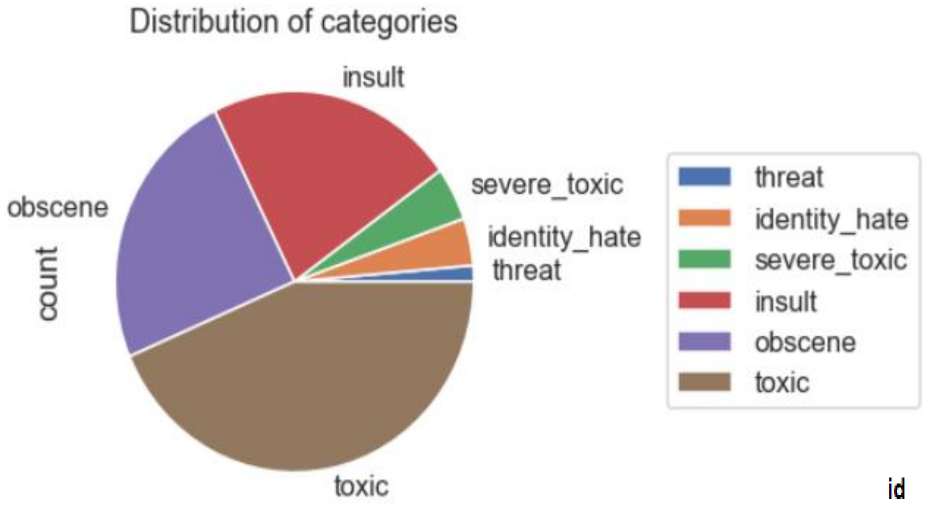
Karthik Enjeti

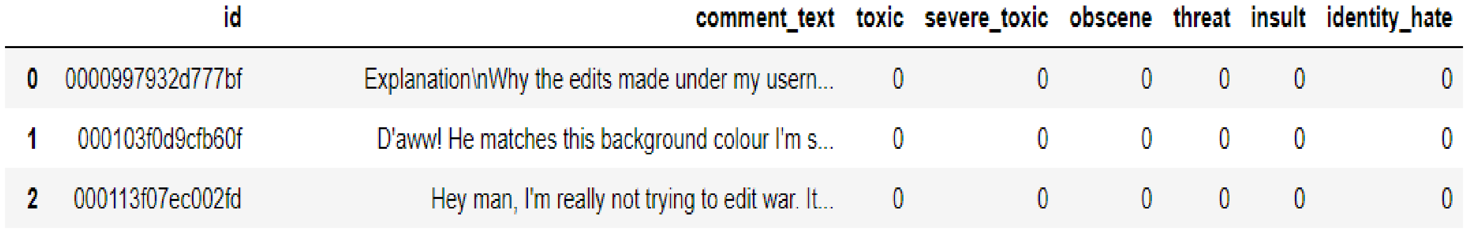
enjeti.k@northeastern.edu



DATASET

- The dataset is obtained from Kaggle (https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview) and has been put together by Jigsaw and Google.
- It consists of comments from Wikipedia talk pages which are classified into 6 categories of toxicity namely toxic, severe-toxic, obscene, threat, insult and identity-based hate.
- The dataset consists of 312,735 comments which have been collected between 2011 and 2015 and have been tagged by groups of people.

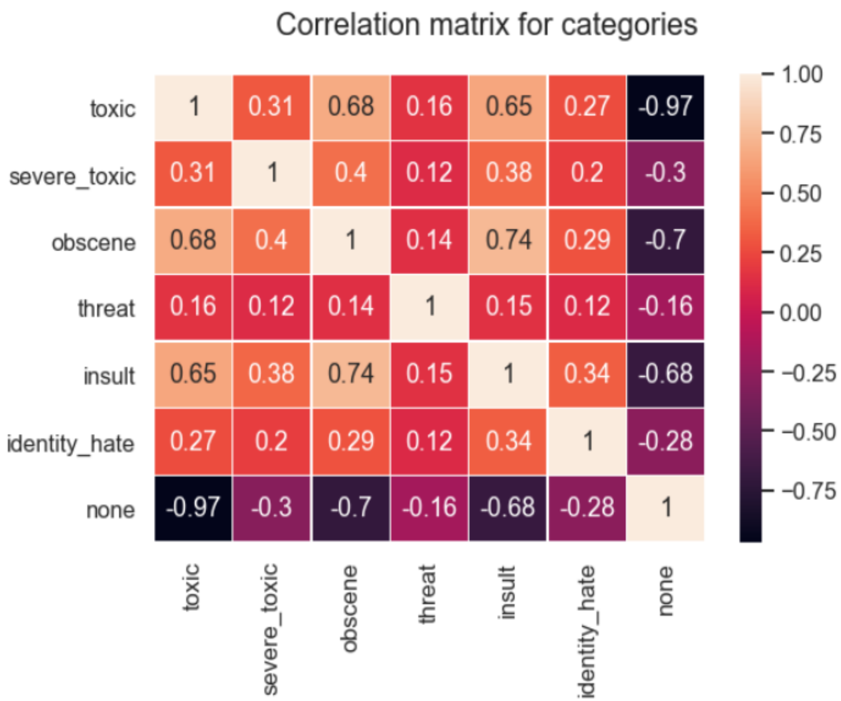




	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0

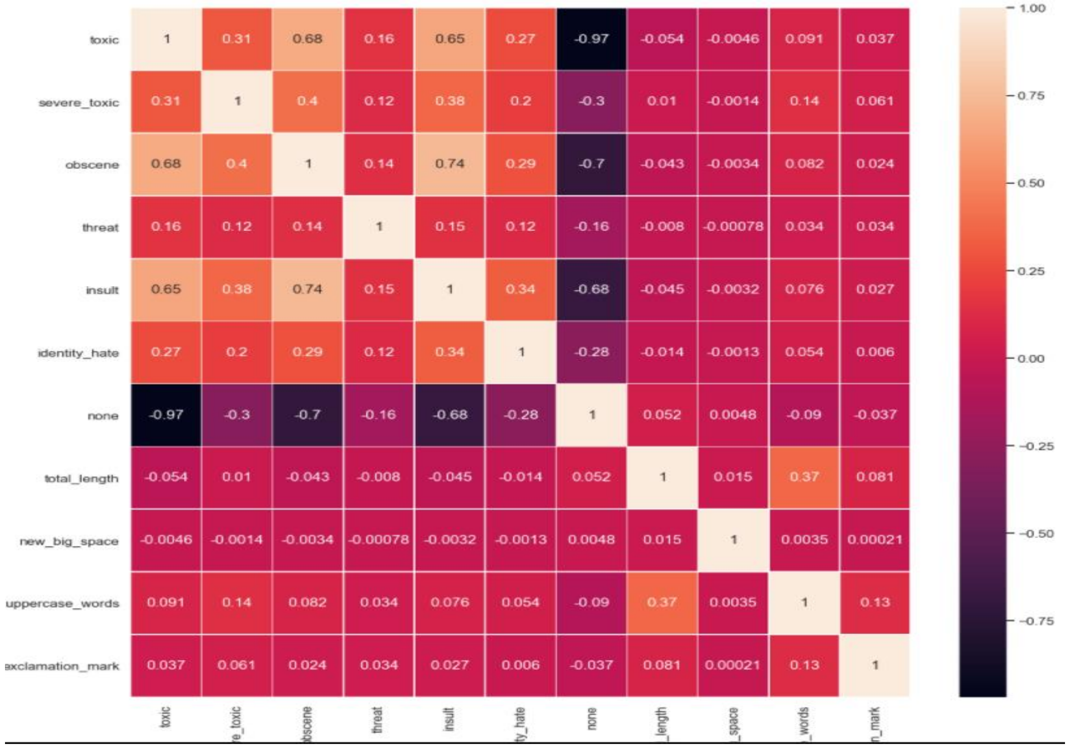
EDA

- Correlation between toxic categories
- Deriving features such as length of comments, exclamation mark, uppercase words and large spaces in text and observing if any relation to toxic categories



Correlation matrix for categories

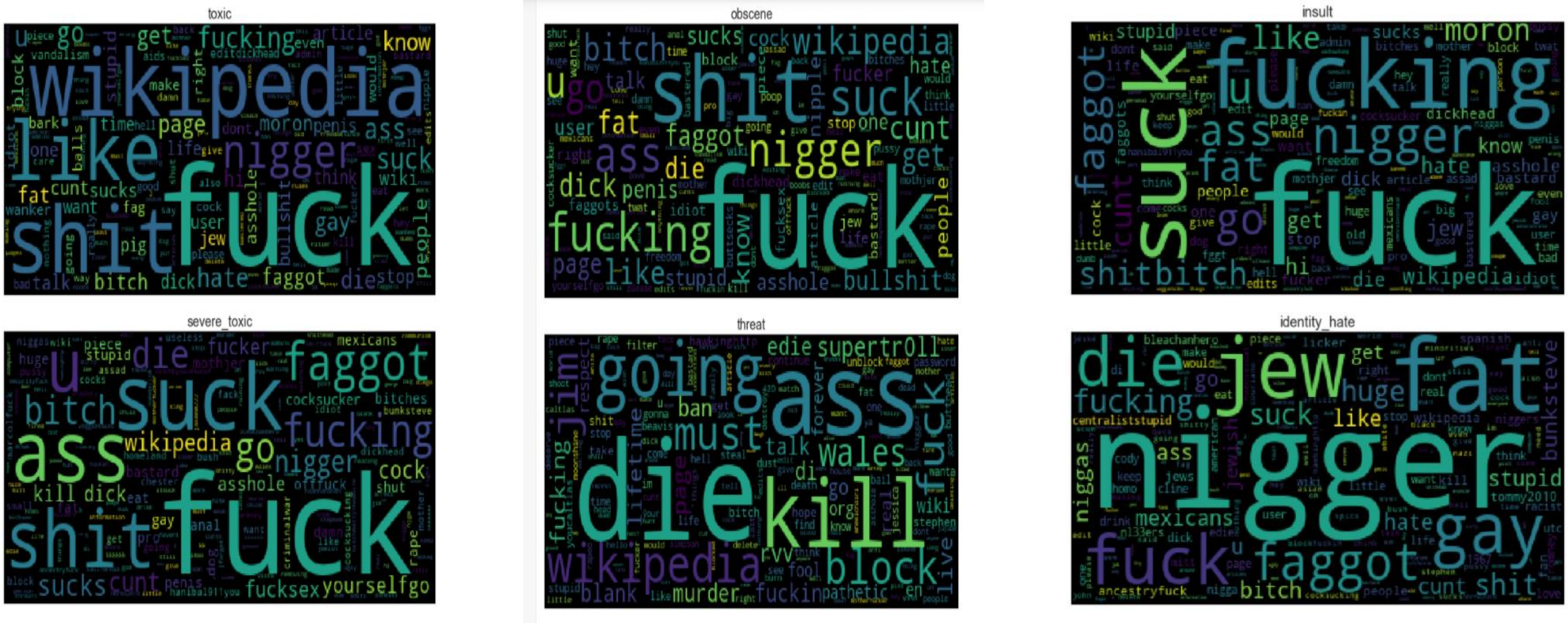
	toxic	severe_toxic	obscene	threat	insult	identity_hate	none
toxic	1	0.31	0.68	0.16	0.65	0.27	-0.97
severe_toxic	0.31	1	0.4	0.12	0.38	0.2	-0.3
obscene	0.68	0.4	1	0.14	0.74	0.29	-0.7
threat	0.16	0.12	0.14	1	0.15	0.12	-0.16
insult	0.65	0.38	0.74	0.15	1	0.34	-0.68
identity_hate	0.27	0.2	0.29	0.12	0.34	1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



	toxic	re_toxic	bscene	threat	insult	ty_hate	none	_length	_space	_words	n_mark
toxic	1	0.31	0.68	0.16	0.65	0.27	-0.97	-0.054	-0.0046	0.091	0.037
severe_toxic	0.31	1	0.4	0.12	0.38	0.2	-0.3	0.01	-0.0014	0.14	0.061
obscene	0.68	0.4	1	0.14	0.74	0.29	-0.7	-0.043	-0.0034	0.082	0.024
threat	0.16	0.12	0.14	1	0.15	0.12	-0.16	-0.008	-0.00078	0.034	0.034
insult	0.65	0.38	0.74	0.15	1	0.34	-0.68	-0.045	-0.0032	0.076	0.027
identity_hate	0.27	0.2	0.29	0.12	0.34	1	-0.28	-0.014	-0.0013	0.054	0.006
none	-0.97	-0.3	-0.7	-0.16	-0.68	-0.28	1	0.052	0.0048	-0.09	-0.037
total_length	-0.054	0.01	-0.043	-0.008	-0.045	-0.014	0.052	1	0.015	0.37	0.081
new_big_space	-0.0046	-0.0014	-0.0034	-0.00078	-0.0032	-0.0013	0.0048	0.015	1	0.0035	0.00021
uppercase_words	0.091	0.14	0.082	0.034	0.076	0.054	-0.09	0.37	0.0035	1	0.13
exclamation_mark	0.037	0.061	0.024	0.034	0.027	0.006	-0.037	0.081	0.00021	0.13	1

Word Analysis

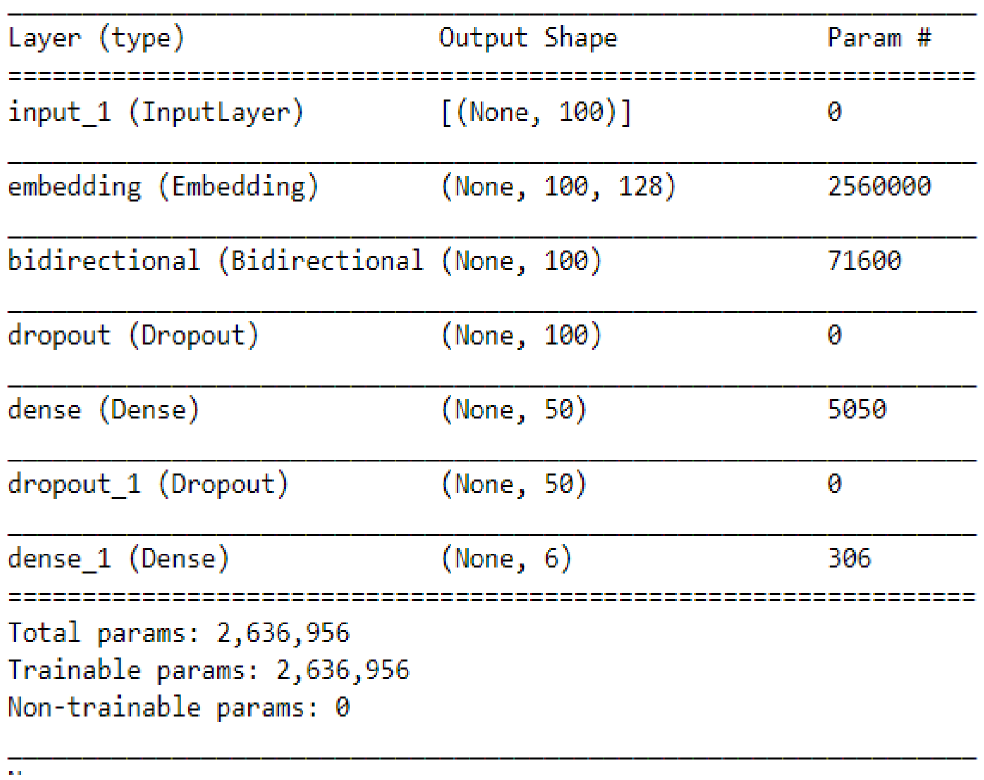
Relevant words which most frequently appear based on the toxic category, which yielded the below results



Bidirectional-LSTM

Data Pre-processing and Model structure

As part of pre-processing data, we tokenize the text and then performing padding as comments are of different lengths and to avoid complications as we progress. We have set max length of vector as 100.



```
_________________________________________________________________
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

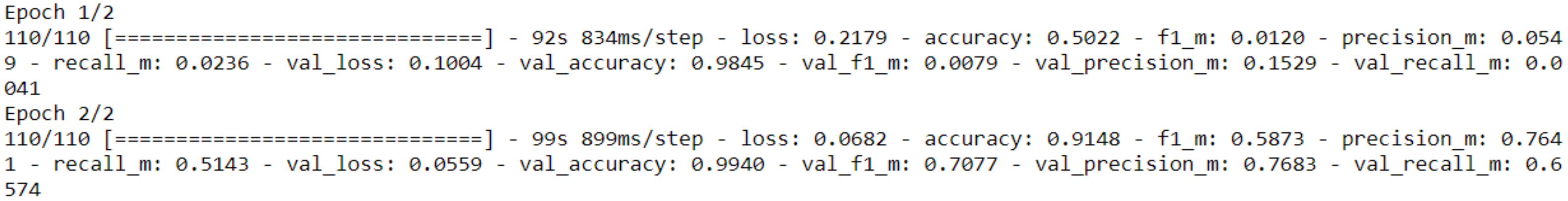
Optimizer: Adam

Model Tuning and Result

Tuning

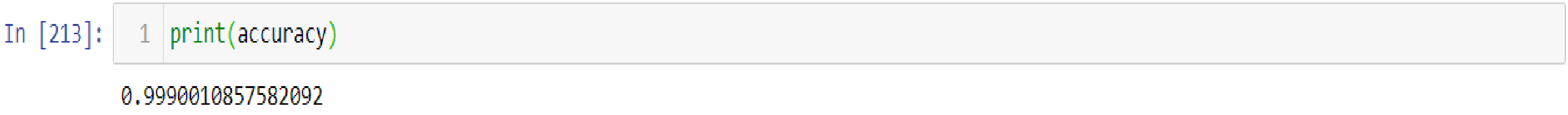
- Batch Size: 24,32,100,1024
- Epochs: Number of iterations (Early Stopping)
- Activation Function: Relu, tanh, sigmoid

Training Results



```
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
```

Testing Results



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

0.9990010857582092
```

Conclusion

- A deep neural network is far more reliable for diverse cases than a simplistic model based of word frequencies.
- Of the available neural network models, BERT, LSTM’s and SVM’s perform the best for text processing tasks such as multi-class text classification.

Scope and Improvements

- We have taken only English comments into consideration, but online content consists of different languages. Often a toxic comment can consist of cusswords from multiple languages.
- A more reliable method is needed to identify spelling correction for cusswords in comments
- Offensive content is not always in text but also in the form of audio, gifs, videos and hyperlinks. Although we know of multiple manners in tackling these individually, we must see how to integrate these together.

References

- 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/