Toxic Comment Identification and Classification using Deep Learning

Submitted to:

Pianalytix

Team Name:

ML Internship Team 7.7

Submitted by:

Raj Maharajwala
Abhishek Salian
Nikita Sonawane
Venkateshwaran Prabakar
Dheeraj kumar k
Mokshit Jain

Abstract

Every day a tremendous amount of data w.r.t to video, audio, images, and text from various social media platforms are generated. Out of which text data is majorly generated in large volumes. This text data contains toxic comments like abuse towards a person community or some threat to some person or a community that can be severe if not identified early and eventually will take an ugly turn. So to address such issues, the severity of toxic statements needs to be identified so that conscious action against such toxic people could be taken. Since the volume of text data is huge we cannot monitor it manually and the ambiguous nature of threat patterns could even be unidentifiable. So in this project we have tried to build a toxic comment identifier to know whether a text or comment is toxic or not, also we are other types of toxicity which will act like a prealarming factor in major social media platforms where the toxic comments can go viral very easily. We have not used any pretrained model. We have trained a simple custom model from scratch. Several experiments were runned and only the best one is mentioned in this brief note. Architecture consists of embedding layers, fully connected layers, BiLSTM cells, dropout and global average pooling. The training accuracy is 97.28 validation accuracy is 99.40 and test accuracy is 90.42 .

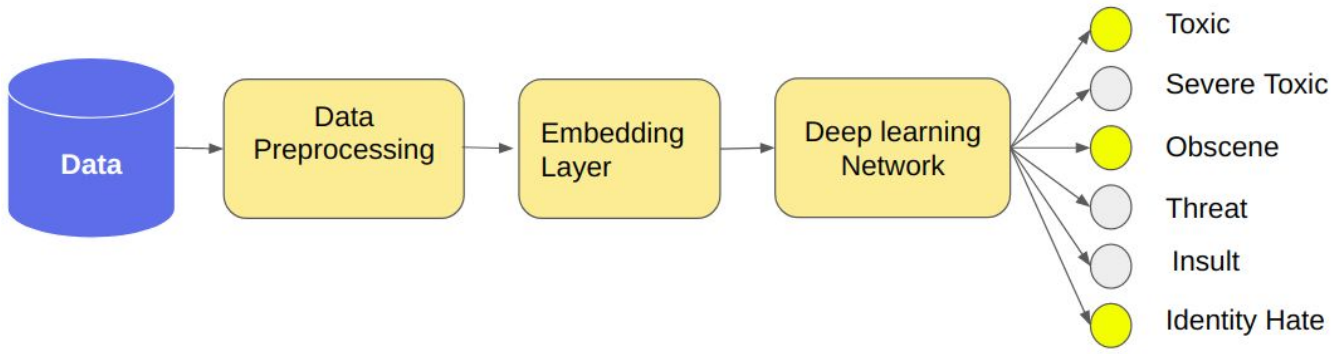
Table of contents

Introduction

Human beings are social animals. In today's world, many of us use social media platforms such as Facebook, Twitter, Youtube, and Instagram to connect with each other. Social media has a direct or indirect effect on our mental health and toxic comments can worsen mental health. Toxic comments can cause personal attacks, online harassment, and cyberbully, serious risk to mental health, and emotional health. Sometimes toxic sentences are used to cause communal disharmony which has worse effects on society. In this era of ever-growing online discussions, learning sources, and social media have increased toxicity in the form of online comments and discussion forums. Toxic comments spread hate and it's ever-growing due to contrasting views of different groups of people covering up under some anonymous username can affect mankind mentally, emotionally, and make them stressed out a lot that they can't even focus in their life. We are developing this project to help maintain social authenticity. We are exploring various neural-based and NLP methods to build an accurate model that can detect diverse types of negative online comments perceived by users or people.

Statement of purpose

The sole purpose of this project is to identify the toxic comments and classify the level or type of toxicity infused in the comments. It is a multi label classification task. Also an extensive study indicating various factors influence toxicity in comments was done.



Note:- Yellow color in diagram is indicate that it is multi label classification

Procedure

3.1 Dataset

We used two dataset one for contraction mapping and another which had toxic and non toxic comments.

3.2 Pre-processing

Text preprocessing is an important step for building machine learning models.

3.2.1. Tokenization

In text preprocessing, the first step is cleaning and preparing text data. Text data must be cleaned before one can use it for modelling. Tokenization is used for cleaning text data. Tokenization is the process of splitting a text object into tokens. The tokens could be words, numbers, symbols, ngrams, characters.

Maximum vocabulary size chosen: 20000

3.2.2. Normalization

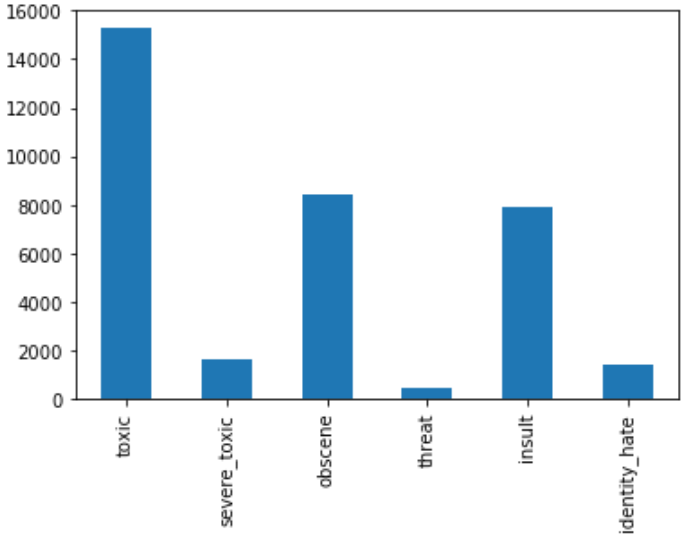
Normalization is the process of converting a token into its base form (morpheme).
It is used to reduce data dimensionality, text cleaning.
There are two types of normalization:
1. Stemming
2.Lemmatization
We have used WordNet Lemmatizer in our project

3.2.3. Contraction Mapping

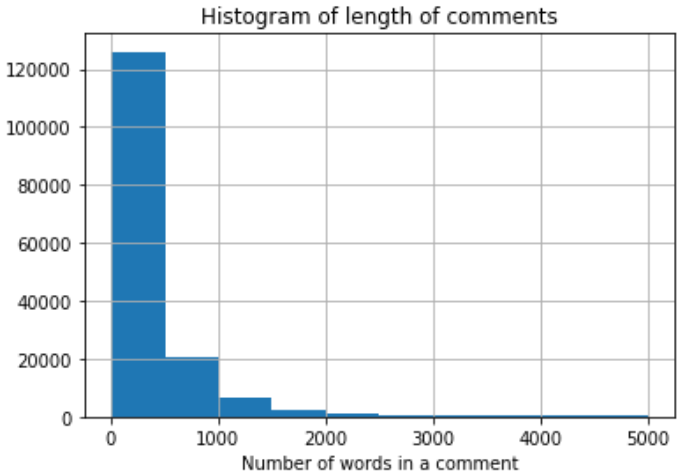
We used contraction mapping to expand the shortened version of words or syllables. The contraction of words is created by removing specific letters and sounds. By doing contraction mapping the words in sentences do not lose any information during the filtering process i.e removing punctuation marks.
For e.g

Contracted Word	After Contraction Mapping
don’t	do not
'aight	alright
you'll	you will

3.3 Data Exploration and visualization



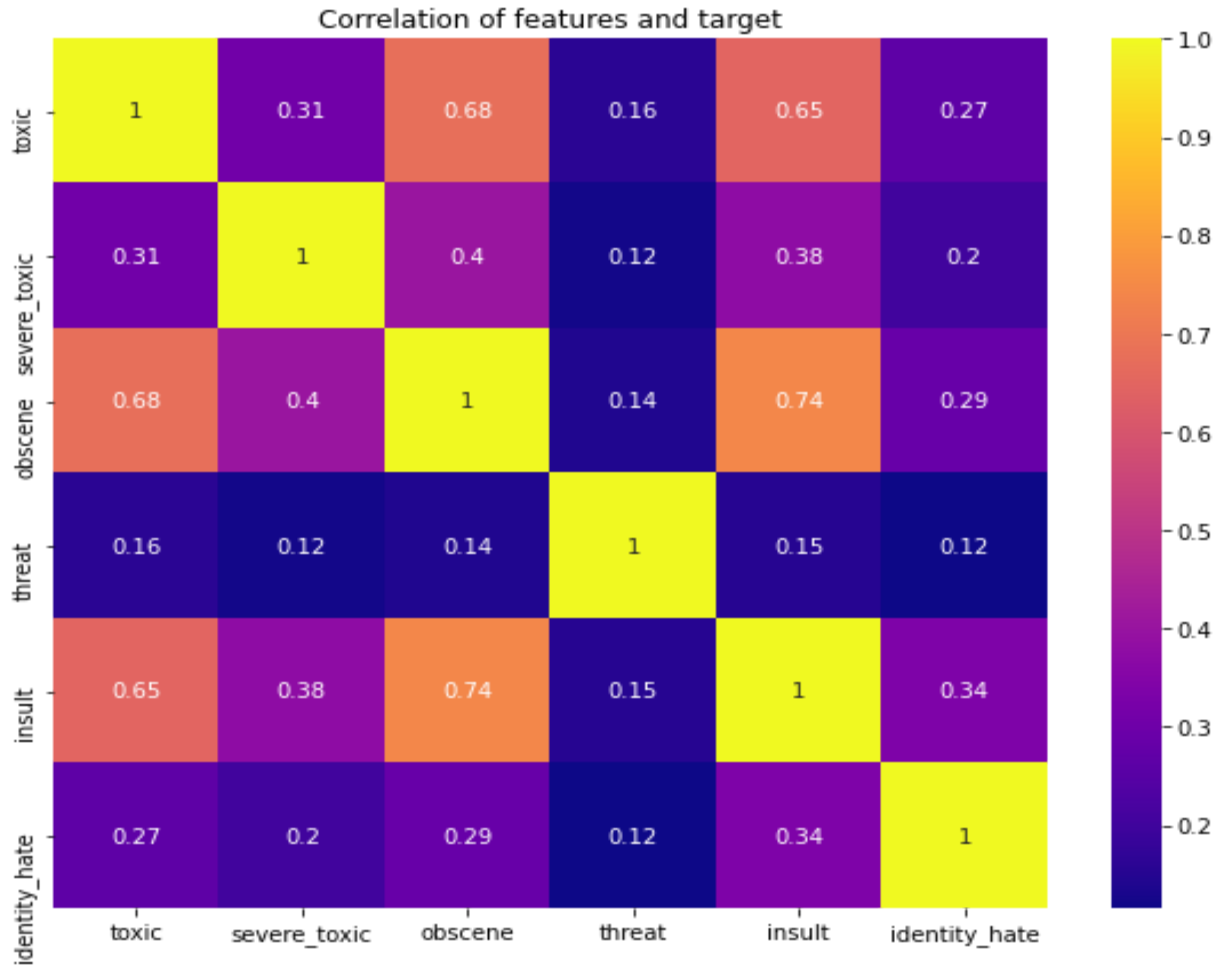
3.3.1 Histogram of length of comment



The histogram shows that most of the comment lengths are within 5000 words.

Statistics of Corpus	**Length**
Average Comment Length	395
Standard Deviation of Comment Length	591
Maximum Length of Comment	5000

3.2.4 Correlation matrix of categories



Correlation matrix is plotted to association association and to quantify strength of different variables in the dataset. By knowing correlation we can select features which are not correlated i.e independent variables. The correlation matrix shows that there is the highest correlation between 'insult' and 'obscene'.

3.2.5 Hyperparameter Selection:

Batch Size Selection: 32 & 64
Optimizer used: ADAM
LOSS Function: Binary Cross Entropy
Thresholding level: Probability of class node >= 0.2 is considered(if p>=0.2 class is mapping to 1 else 0)

Activation function used: ReLU
Last Dense Layer Activation: Sigmoid

Other hyperparameters were kept default.

3.2.6. Model Building

We didn't use any pretrained model and word embeddings. Everything was trained from scratch with a custom build model. Here is the summary of the model that we built.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 400)]	0
embedding (Embedding)	(None, 400, 128)	2560000
lstm_layer (LSTM)	(None, 400, 60)	45360
global_max_pooling1d (Global	(None, 60)	0
dropout (Dropout)	(None, 60)	0
dense (Dense)	(None, 50)	3050
dropout_1 (Dropout)	(None, 50)	0
dense_1 (Dense)	(None, 6)	306

Total params: 2,608,716
Trainable params: 2,608,716
Non-trainable params: 0

3.2.7. Train, Validation, Test Split

The entire corpus of comments was split into 3 sets for train, test, validation. 70% of data was used for training, and 20 % was used for testing, rest 10% was used for validation for building generalized models.

Observations

In this section we share our project result which we have iterated to several experiments only good performing experiment runs will be written in table.

5.1 Results

Model Training	Train(%)	Validation (%)	Test(%)	No. of Epoch
Experiment 1	97.28	99.40	90.42	2
Experiment 2	94.37	98.28	90.34	5

5.2 Simple Flask Framework based Web Application

Conclusion

By building this neural-based system to identify toxic statements, we can track various social media comments which would worsen the communal harmony, a threat to a person, or maybe lead to various social issues. There are several other factors which influence the toxicity, to capture such data would require a lot of social media network analysis data. This can help create healthy communication on various social platforms which has many benefits.

References:

1. Risch J., Krestel R. (2020) Toxic Comment Detection in Online Discussions. In: Agarwal B., Nayak R., Mittal N., Patnaik S. (eds) Deep Learning-Based Approaches for Sentiment Analysis. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-15-1216-2_4
   Link: https://link.springer.com/chapter/10.1007/978-981-15-1216-2_4

2. Challenges for Toxic Comment Classification: An In-Depth Error Analysis

3. Chakrabarty N. (2020) A Machine Learning Approach to Comment Toxicity Classification. In: Das A., Nayak J., Naik B., Pati S., Pelusi D. (eds) Computational Intelligence in Pattern Recognition. Advances in Intelligent Systems and Computing, vol 999. Springer, Singapore. https://doi.org/10.1007/978-981-13-9042-5_16
   Link: https://arxiv.org/pdf/1903.06765

4. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/reports/6837517.pdf

5. [1909.12642] HateMonitors: Language Agnostic Abuse Detection in Social Media

6. Identification and Classification of Toxic Comments on Social Media using Machine Learning Techniques