A Neuro-NLP Induced Deep Learning Model Developed Towards Comment Based Toxicity Prediction

Kulaye Shreyal Ashok
Data Analyst,
Cognitive Business Intelligence
Tata Consultancy Services
Mumbai, India
shreyalkulaye17@gmail.com

Kulaye Aishwarya Ashok
Dept. of Systems
Mumbai Educational Trust
Mumbai, India
aishkulaye@gmail.com

Shaikh Mohammad Bilal Naseem
Dept. of IT
S K Somaiya College
Somaiya Vidyavihar University
Mumbai, India
mohammadbilal@somaiya.edu

2022 5th International Conference on Advances in Science and Technology (ICAST) | 978-1-6654-9263-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICAST55766.2022.10039597

***Abstract*—The comments sections of online forums and social media platforms have become the new playing field for cyber harassment. Correspondingly, various organizations and companies have decided to abolish toxic and nasty comments altogether to avoid this kind of issue. To protect authorized and genuine users from being exposed to comments which contain offensive language on online mediums or social media platforms, organizations have started flagging such comments and they are blocking those users who are using unpleasant forms of language. Most of the organizations use computerized algorithms for instinctive discovery of comment toxicity using machine learning and artificial intelligence based systems. In the present research study, we have tried to build multi headed comment toxicity detection models. We have built three toxicity detection models using deep learning techniques and compared the accuracy and results. We have also developed a menu driven interface which will help to link machine learning models which is uncomplicated for non programmers and this connection of model to interface will be convenient for making interactive programming interfaces with great accuracy and operationality.**

Keywords— toxic comment classification, deep learning, sequential, nlp, lstm, cnn, tensorflow, keras, machine learning

I. INTRODUCTION

The Internet has created some great opportunities for making connections. This connectedness has saved our lots of time, entertained us and embraced our lives in innumerable ways. Online platforms that aggregate user content are the foundation of knowledge sharing on the Internet. But the catch is that not all users on the Internet are intended to participate and support nicely, and some see it as an avenue to release their rampage, humility, and preconceptions. However, there is a dark side to the internet, which appears to bring out the defeat in people. The worst of the internet comes with offering free commenting and it is getting toxic each and every year [2]. People often comment on posts, articles and it seems that, no matter what the point of view is, someone will find a way to connect it to a personal attack, politics or a conspiracy theory and they will have no problem posting the comment. Firstly, it is important to understand what toxicity is. Toxicity has a negative effect. Toxicity is an ill mannered, insolent, unpleasant or obstructive statement that is plausible to make authors or users quit a discussion. Toxicity can be polymorphic. It may have many forms like harassment, insult, racism, hate speech, profane language, terrorism and frauds [10]. Users deserve a safe and civil environment to exchange ideas, emotions, feelings and workflow without the fear of toxicity taking over. Positive feedback and Quality conversations are essential for supportability because they lead the way to productive discourse that keeps users engaged and spending more time on online platforms. Social media platforms can create their own rules for their community and use trending technologies to remove toxicity and create quality and healthy conversations. Many organizations have already started an initiative which chooses to eliminate toxic and unpleasant comments altogether to avoid this kind of problem. To protect authorized and genuine users from being exposed to comments which contain offensive language on online forums, social media platforms, companies and organizations have started flagging such comments and they are blocking those users who are using unpleasant forms of language [3]. Several techniques are developed to protect internet users from becoming sufferers of online harassment and cyberbullying.

In the present research work we have tried to explain how we have built multi headed toxic comment detection models with the trending techniques. We have used techniques from neural networks and deep learning. We have built three multi headed models which includes Sequential model, Long Short Term Memory (LSTM) [19] model and Long Short Term Memory (LSTM) using Convolutional neural networks (CNN). We have tried to incorporate the usage of new techniques, libraries and frameworks which will make it interesting and serve higher potential. Idea behind using multi headed technique is that it will try to suggest a toxicity of comment with its higher accuracy. Also by getting the result of various algorithms we can learn in depth working of algorithms with various cases. We have also built a user interface for the convenience of lay public who can directly use it without having a programming background. Such a detection system can have a very high potential for companies, organizations, social media platforms and online forums in the present and the future.

II. PROBLEM STATEMENT

A. *Statement of a problem*

To analyze the dataset and train it using various algorithms and then build a machine learning model that's appropriate for detecting various kinds of comment toxicity and developing a user interface for the same.

B. *Existing systems and Literature review*

Spread of toxic content from online platforms has attracted many researchers into techniques of computerized recognition in recent years. However, two most important constraints still exist. First limitation is the absence of aid for

multiple labeled toxic comment recognition and second is incomprehension of the influence of the instabalized and noisy data values on such jobs. Since there is a high requisite to resolve the real world problem of toxic comment detection, it has become a diligent quest area with many recently proposed methods and studies [3] [4]. Firstly startups and social media managers started doing it manually but it becomes difficult for the platforms who have a huge number of users. So, there was a need for an automated comment classifier technique which will complete the detection task accurately and it will result in saving so much time and effort. An enabling a computerized automated model for this process is to distinguish toxic comments into different types of labels based on severity. We can call this job 'toxic texts identification', or 'toxic comment classification', and it mainly includes an immense set of latest studies on detecting nasty comments and resentment [3], vulgar and brutish language [5], and cyberbullying [6].

Many researchers and data scientists come up with their detection models. However, while their approaches and models resolve some of the problems and cases others still remain unresolved and the correct path for further study is required. This literature work has been compassed using the systematic literature review (SLR) manner which is explained in detail in the research paper of Kitchenham, Barbara & Charters, Stuart [1]. There exists public models for comment toxicity detection but they don't serve higher accuracy and precision. Many of the models are available on Kaggle too. These existing models are built using Ai and ML algorithms such as logistic regression [15], Naive Bayes [16] [17], Decision Tree [29], Support Vector Machine (SVM) [18], Random Forest, Know Nearest Neighbors (KNN), etc. Some of them have also used deep learning models which includes Natural Language Processing (NLP) [9], BERT embedding, Convolutional Neural Network (CNN), Long Short Term Memory (LSTM) [20] etc. Some of the research papers and studies based on toxic comment detection models are described below.

TABLE I. LITERATURE REVIEW

Sr. No.	Paper Title	Author	Year	Advantages	Limitations
1.	"Toxic Comment Detection in Online Discussions" [21]	Julian Risch, Ralf Krestel	2020	Given an idea how to build and implement classes based on toxicity detection models [21].	Strategies are mentioned, classes of toxicity are well defined but lacking in giving practical implementation steps.
2.	"Towards Non-Toxic Landscapes: Automatic Toxic Comment Detection Using DNN" [22]	D'Sa, Ashwin Illina, I. Fohr, Dominique	2020	Proper explanation and in depth working of BERT and DNN algorithms are provided [22].	Implementation methodology is not explained.
3	"Toxic Comment Classification" [23]	Ravi, Pallam Batta, Hari Yaseen, Greeshma	2019	They have used Receiver Operating Characteristic (ROC) along with Area under the curve (AUC) graphs as a test metric and explained the results of them [23].	Results with different algorithms are shown with ROC but somewhere it is lacking in giving practical implementation steps.
4	"Challenges for Toxic Comment Classification: An In-Depth Error Analysis" [24]	Betty van Aken, Julian Risch, Ralf Krestel, Alexander Loser	2018	Literature review is performed and covers most of the techniques which are used for toxicity detection models [24].	In depth Literature review of papers along with implementation techniques could be more useful.
5	"Machine learning methods for toxic comment classification: a systematic review" [25]	Androcec, Darko.	2020	Systematic review is performed which contains a list of papers text classification models [25].	It would have been great if Techniques used for implementation are mentioned.

III. PROPOSED MODEL

Steps for building ML model for toxicity Prediction:

Step 1. Data Gathering

Gathering or selection of the existing dataset is the initial move apropos to the actual forming of any ML model. This first step literally describes how good the ML model will be. The volume of the dataset plays a decisive part in model building. The better and more data we get, the better the model's accuracy will be. The dataset for building the toxic comment classification model was acquired from the Kaggle competition site and it included the training set as well as the test set [27]. The Conversation AI team [26], Google and Jigsaw innovative research team [27] are working on the tasks to assist refined online discussions. Main area of pivot is the research based on the field of toxic comments. So far, many researchers and scientists have designed various open source systems and methods which are provided by the Perspective API [28], counting comment toxicity. The training dataset is of size 159572 and the testing dataset is of 153165. The good part of this dataset is it is a huge realtime wikipedia dataset. It also has labels which specify types of nasty comments which includes Toxic, Severe toxic, Insult, Obscene, Threat and Identity hate. The existing models still make errors and have less accuracy and functionality.

Step 2. Data Analysis

This step involves description of data and visualization of data. For this step we have carried out Exploratory Data analysis using various data visualization techniques. We have summarized the correlation so that we get an idea how data fields are related to each other.

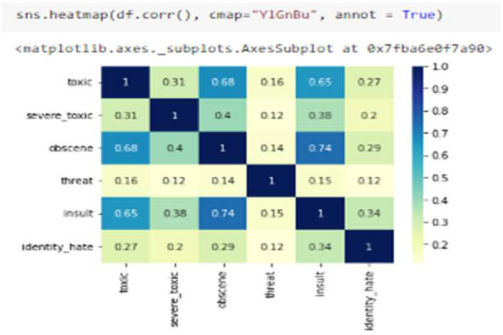

Fig. 1. Heatmap of correlation

We have plotted the heatmap in a diagrammatic way to understand the correlations in a better way (Fig 1). We have also plotted Venn diagrams for performing exploratory data analysis on labels on comment dataset. It includes a combination of each and every label present in the data values. The outcome of the venn diagram is as follows.

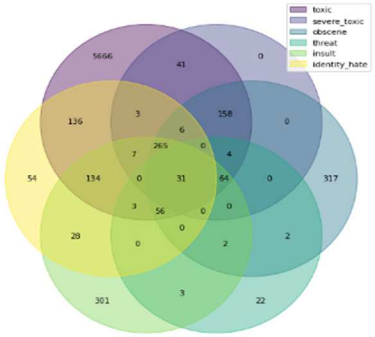

Fig. 2. Venn diagrams of dataset

Step 3. Data Pre-processing

Pre-processing and transformation of data is a key step in model building [30]. The identification of nasty comments is a kind of natural language processing and classification case. It aims to detect the given note into categories of instances (classes, labels, or types) [11] [9]. We have followed the preprocessing workflow as shown below.

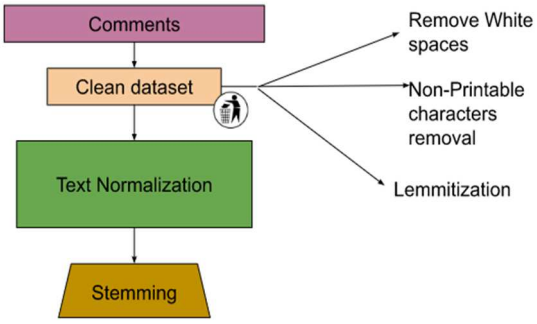

Fig. 3. Data Preprocessing methodology

Pre-processing step reduces the data complexity which can be done using stopword removal and stemming in text based data. Feature extraction is a method used to choose instances with a multi-scaled angle for model utilization. After this ML algorithm is enforced on the training dataset using the features which are extracted from it. Then those extracted characteristics are used to supervise a model to differentiate between homogeneous, hidden examples illustrated in the dataset. In our model we have used maximum features of '300000' for the vocab and the sequence length is '2000'. Majority of the existing systems aim on only one type of data which includes hate speech classification [8] [9]. Kumar et al. [12] organized the first study for examining classes of toxic comments and abusive contents including aggression, and cyber bullying. Distinct kinds of toxic comments can differ in the tasks structuring which usually shares homogeneous features. Schmidt and Wiegand [13] have included some of the very commonly used characteristics in toxicity classification. The most frequently used attributes are letters, words, capitalisation, content length, character n-grams, and in the background of online forums and social media sites, hashtags and uniform resource locator (URL). Word and character generalizations include features such as clustered words [14], lemmatization and word embeddings [15].

Step 4. Model Building and training

For model building we have chosen some deep learning techniques. Deep Learning in Neural network models works best with text data and applications of RNN like NLP will be very useful. The first model which we built was the sequential model developed using tensorflow. A tensorflow Sequential model works best for a plain stack of layers where each data input layer has exactly one input tensor and one output tensor. But the problem with the sequential model in our case is it has given good precision results around 83% but the accuracy parameter decreases because of sequence list. The model was giving good results because of higher precision but the overall model must score good accuracy.

```
print(f'Precision: {pre.result().numpy()}, Recall:{re.result().numpy()}, Accuracy:{acc.result().numpy()}')

Precision: 0.8302785754203796, Recall:0.6663724780082703, Accuracy:0.4413239657878876
```

Fig. 4. Sequential model score

Because of the higher precision and lower accuracy problem of sequential models, we thought to build one more model which will result in higher accuracy. We have again chosen a deep-learning model and trained the data using the training dataset and the validation dataset. Since we are working on a Natural Language Processing based multi headed model, it is the best case to choose Long Short Term Memory model (LSTM) as compared to the rest of the deep learning models [20]. LSTM models work very similarly to RNN models. The LSTM model comes with one major difference that hidden layers of the modeling layer updates are restored by memory cells. This makes LSTM the best model at research findings and exposing a wide range of dependencies in text based data which is imperative for sentence structures.

For building the LSTM model, we have imported the 'Talos' library. Talos library is basically a deep learning which radically changes the ordinary Keras, TensorFlow by fully regenerating the hyperparameter tuning and evaluation of the model. We have used the Scan function to perform GridSearchCV and found the best parameters that would give us the highest accuracy. This training took a lot of time. It literally took six to seven hours to find out the best parameters. But after completion of the scan model has chosen the best finely tuned parameters for building the model. The model scored the accuracy around 94% and after choosing the best finely tuned parameters it secured accuracy of around 99% over the span of two epochs. The model scored the accuracy around 94% and after choosing the best finely tuned parameters it secured accuracy of around 99% over the span of two epochs.

```
model_info_1=model_1.fit(x_train,y_train, epochs=2, batch_size=32,  validation_data=(x_val, y_val))

Epoch 1/2
3990/3990 [==============================] - 498s 124ms/step - loss: 0.0970 - accuracy: 0.9363 - val_loss: 0.0697 - val_accuracy: 0.9947
Epoch 2/2
3990/3990 [==============================] - 510s 128ms/step - loss: 0.0712 - accuracy: 0.9940 - val_loss: 0.0634 - val_accuracy: 0.9947
```

Fig. 5. LSTM Model Accuracy

We achieved the best results with the LSTM model, but we still tried to implement a Convolutional neural network in the LSTM model for experimental purposes. We have a Convolution 1 dimensional layer with the features of embedding such as Maximum Pooling, Denser Network, Batch normalization and dropout. LSTM scored the accuracy of 93.10% which was slightly lesser than the normal LSTM model. The CNN model works best in image based pattern recognition. In text based classifications NLP based LSTM models work the best.

Let us compare the graphs of the LSTM and LSTM CNN model and understand why LSTM is performing better than LSTM CNN in our case. We have visualized the accuracy and loss values of LSTM and LSTM-CNN models during the entire training process. The graphs give us a fair idea about the quality and accuracy of our deep-learning models, and whether models have been appropriately trained or. Trends in the accuracy and the loss values during every epoch can be seen in the images below.

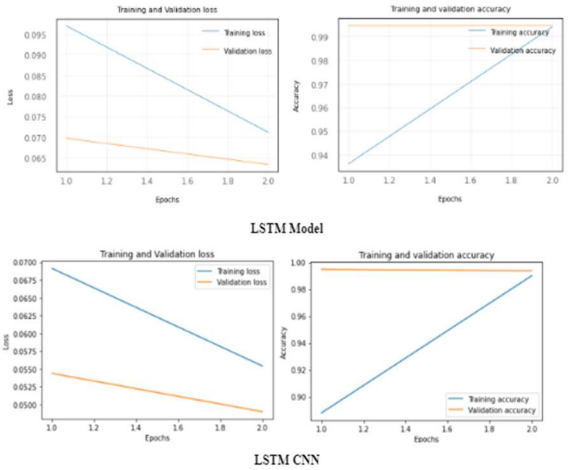

Fig. 6. Loss and Accuracy of LSTM vs LSTM CNN

Step 5. Testing

We will now see the prediction results given by the LSTM model. The model was successfully developed and it is giving very accurate results. The model does not provide the comment toxicity results as yes or no whereas it gives multi labeled output which can be very useful for companies and organizations to understand user behavior.

```
toxicity_level('go jump off a bridge jerk')

Toxicity levels for 'go jump off a bridge jerk':
Toxic:         47%
Severe Toxic:  1%
Obscene:       15%
Threat:        2%
Insult:        23%
Identity Hate: 7%
```

```
toxicity_level('Not bad')

Toxicity levels for 'Not bad':
Toxic:         7%
Severe Toxic:  0%
Obscene:       2%
Threat:        0%
Insult:        2%
Identity Hate: 0%
```

Fig. 7. Testing results

Step 6. Deployment

The model is deployed to the user interface using the gradio library. Gradio is the python library which provides the fastest way to deploy your machine learning model with a friendly web interface. This gradio based User interface is easy to develop and can be deployed on the web browsers in a fraction of seconds. It is highly responsive. We can also deploy this for permanent purposes using python flask and AWS. Screenshot of the user interface is provided in the figure below.

Fig. 8. Testing results

IV. SURVEY ANALYSIS

We have carried out a survey study as a part of this research work. The purpose of this survey is to analyze the need for an automated toxic comment detection system. We have asked the survey questions based on Negative Online Comments and Cyberbullying. The form was submitted to the users of online forums and social media platforms. We have collected the response from 241 individuals and the age was in the range of 15 to 65. The synopsis of some of the responses is shown in charts below.

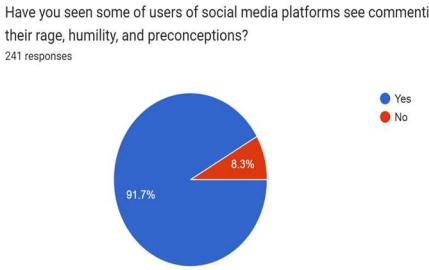

Fig. 9. Analysis 2

91.7% of the audience agreed that some users of social media platforms see commenting as an avenue to release their rage, humility, and preconceptions.

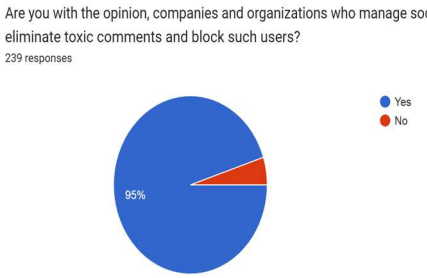

Fig. 10. Analysis 6

Fig.10 shows that 95% of the audience are of the opinion that companies and organizations who manage social media platforms should eliminate toxic comments and block such users.

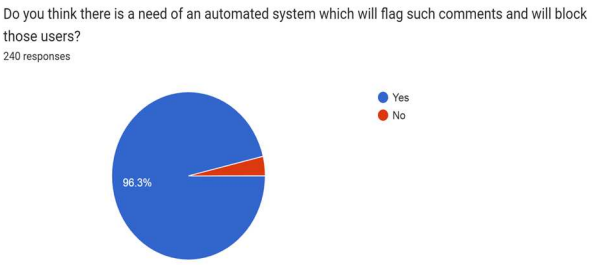

Fig. 11. Analysis 7

As shown in Fig. 11, Most of the audience i.e. 96.3% of the audience agreed that there is a need for an automated system which will flag such comments and will block those users.

As per all the responses recorded from the survey, the report concludes the need and positive response for an automated system which will flag toxic comments and will block those users.

V. CONCLUSION

The ultimate aim of this research work is to build the multi headed Comment Toxicity model. We have successfully built three models which are Sequential Tensorflow. LSTM and LSTM CNN. We can claim from the results observed after the successful completion of the model building that the LSTM Model performs better than the hybrid LSTM-CNN Model. The LSTM CNN model loses marginally to the traditional LSTM DL model. The higher accuracy of the LSTM shows that it is the right choice for the model building of Toxic Comment Classification. By the reference of our research study anyone can successfully construct a deep learning model and can build a menu driven simple interface for the lay public. This system has very high potential for companies, organizations, social media platforms and online forums in the present and the future.

ACKNOWLEDGMENT

It is a matter of great glory to work on this research study on “Multi-headed Comment Toxicity Model developed using Deep Learning Techniques". I would like to express my deep and sincere gratitude to my professor Dr. Shaikh Mohammad Bilal Naseem who introduced me to the world of Machine Learning and Data Science. They are the one from whom I have learnt how programming concepts are derived from real life and how Machine Learning can be used to solve real world problems. Lastly, I would like to thank my co-author, friends and relatives who actively motivated me and supported me to carry out activities related to research work.

REFERENCES

[1] Kitchenham, Barbara & Charters, Stuart, “Guidelines for performing Systematic Literature Reviews in Software Engineering”, ResearchGate (2007).

[2] Waseem Z., Davidson T., Warmsley D., Weber I., Understanding Abuse: A Typology of Abusive Language Detection Subtasks. In 1st Workshop on Abusive Language Online. Vancouver (2017).

[3] Gambäck, B., Sikdar, U., Using convolutional neural networks to classify hate-speech. In Proceedings of the First Workshop on Abusive Language Online, pages 85–90. Association for Computational Linguistics (2017).

[4] Chu T., Jue K., Wang M., Comment Abuse Classification with Deep Learning. Stanford University (2017),

[5] Nobata C., Tetreault J., Thomas A., Mehdad Y., Chang Y.: Abusive language detection in online user content. In Proceedings of the 25th International Conference on World Wide Web, pages 145–153 (2016).

[6] Dadvar M., Trieschnigg D., Ordelman R., de Jong F., Improving cyberbullying detection with user context, In Proceedings of the 35th European Conference on Advances in Information Retrieval, ECIR 13, pages 693–696, Berlin, Heidelberg, Springer-Verlag (2013).

[7] Badjatiya P., Gupta S., Gupta M., Varma V.: Deep learning for hate speech detection in tweets. In Proceedings of the 26th International Conference on World Wide Web Companion, pages 759–760 (2017).

[8] Burnap P., Williams M., Us and them: Identifying cyber hate on twitter across multiple protected characteristics, EPJ Data Science, 5(11):1–15 (2016).

[9] Kwok I., Wang Y., Locate the hate: Detecting tweets against blacks. In Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, pages 1621–1622. AAAI Press (2013).

[10] Park J., Fung P., One-step and two-step classification for abusive language detection on twitter. In ALW1: 1st Workshop on Abusive Language Online, Vancouver, Canada, Association for Computational Linguistics (2017).

[11] Schmidt A., Wiegand M., A survey on hate speech detection using natural language processing. In International Workshop on Natural Language Processing for Social Media, pages 1–10. Association for Computational Linguistics (2017).

[12] Kumar R., Ojha A., Malmasi S., Zampieri M., Benchmarking Aggression Identification in Social Media, First Workshop on Trolling, Aggression and Cyberbullying (2018).

[13] Schmidt A., Wiegand M., A survey on hate speech detection using natural language processing. In International Workshop on Natural Language Processing for Social Media, pages 1–10. Association for Computational Linguistics (2017).

[14] Warner W., Hirschberg J., Detecting hate speech on the world wide web. In Proceedings of the Second Workshop on Language in Social Media, LSM '12, pages 19–26. Association for Computational Linguistics (2012).

[15] Djuric N., Zhou J., Morris M., Grbovic M., Radosavljevic V., Bhamidipati N., Hate speech detection with comment embeddings. In Proceedings of the 24th International Conference on the World Wide Web, pages 29– 30. ACM (2015).

[16] Kwok I., Wang Y.: Locate the hate: Detecting tweets against blacks. In Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, pages 1621–1622. AAAI Press (2013)

[17] Chen Y., Zhou Y., Zhu S., Xu H., Detecting offensive language in social media to protect adolescent online safety. In Proceedings of the 2012 ASE/IEEE International Conference on Social Computing and 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust, pages 71–80, Washington, DC, USA, IEEE Computer Society (2012).

[18] Xiang G., Fan B., Wang, L., Hong J., Rose C., Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In Conference on Information and Knowledge Management, pages 1980–1984. ACM, (2012).

[19] Badjatiya P., Gupta S., Gupta M., Varma V., Deep learning for hate speech detection in tweets. In Proceedings of the 26th International Conference on World Wide Web Companion, pages 759–760 (2017).

[20] Zhang Z., Robinson D., Tepper J., Detecting hate speech on twitter using a convolution-gru based deep neural network. In Proceedings of the 15th Extended Semantic Web Conference, ESWC18, pages 745–760. Springer (2018).

[21] Risch Julian & Krestel Ralf, Toxic Comment Detection in Online Discussions. 10.1007/978-981-15-1216-2_4, ResearchGate (2020).

[22] D'Sa Ashwin & Illina I. & Fohr Dominique, Towards Non-Toxic Landscapes: Automatic Toxic Comment Detection Using DNN, Researchgate (2020).

[23] Ravi Pallam & Batta Hari & Yaseen Greeshma, Toxic Comment Classification. International Journal of Trend in Scientific Research and Development. Volume-3. 24-27. 10.31142/ijtsrd23464. (2019).

[24] Betty Van Aken, Julian Risch, Ralf Krestel and Alexander Loser¨, Challenges for Toxic Comment Classification: An In-Depth Error Analysis. In Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), pages 33–42, Brussels, Belgium. Association for Computational Linguistics. https://aclanthology.org/W18-5105. 2018.

[25] Androcec Darko, Machine learning methods for toxic comment classification: a systematic review, Acta Universitatis Sapientiae, Informatica. 12. 205-216. 10.2478/ausi-2020-0012. (2020).

[26] Conversational AI, Github IO: https://conversationai.github.io/

[27] Dataset: Toxic Comment Classification Challenge | Kaggle. (n.d.). Jigsaw Kaggle. https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge.

[28] Perspective API. (n.d.). Perspective API. https://perspectiveapi.com/

[29] Harsh H. Patel, Purvi Prajapati, "Study and Analysis of Decision Tree Based Classification Algorithms", International Journal of Computer Sciences and Engineering(ICSE), Researchgate, Vol.-6, Issue-10, October 2018.

[30] "Why Data Preparation is So Important in Machine Learning", Machine Learning Mastery, Jason Brownlee, June 2020 https://machinelearningmastery.com/data-preparation-is-important/