International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT228391

An Intense Study of Machine Learning Research Approach to Identify Toxic Comments

Monika Dandotiya*[1], Dr. Rajni Ranjan Singh Makwana[2], Nidhi Dandotiya[3]
*[1]Department of Computer Science, ITM University, Gwalior, Madhya Pradesh, India
[2]Department of Computer Science, MITS Gwalior, Madhya Pradesh, India
[3]Department of Computer Science, ITM University, Gwalior, Madhya Pradesh, India

Article Info
Publication Issue :
Volume 8, Issue 4
July-August-2022

Page Number : 71-81

Article History
Accepted: 05 July 2022
Published: 14 July 2022

ABSTRACT

A large number of online public domain comments are usually constructive, but a significant proportion is toxic. The comments include several errors that allow the machine-learning algorithm to train the data set by processing dataset with numerous variety of tasks, in the method of conversion of raw comments previously feeding it to Classification models using a ML method. In this study, we have proposed classification of toxic comments using a ML approach on a multilinguistic toxic comment dataset. The logistic regression method is applied to classify processed dataset, which will distinguish toxic comments from non-toxic comments. The multi-headed model comprises toxicity (obscene, insult, severe toxic, threat, & identity-hate) or Nontoxicity Estimation. We have implemented four models (LSTM, GRU RNN, and BiLSTM) and detected the toxic comments. In Python 3, all models have a simple structure that can adapt to the resolution of other tasks. The classification problem resolution findings are presented with the aid of the proposed models. It has been concluded that all models solve the challenge effectively, but the BiLSTM is the most effective to ensure the best practicable accuracy.

Keywords: Machine Learning Toxic Comments, LSTM, GRU, RNN, BiLSTM.

I. INTRODUCTION

a The data in structured and unstructured data are divided into two categories. Structured data are in tabular format & can be read and interpreted simply, while unstructured data are essential for interpretation before they are processed. The text is provided as unstructured data. Text Analytics has numerous applications, such as the differentiation of news articles by the designation of the type of related articles based on post definitions. Text Summarizing is a two- or three-line Summarizing of a broad document, Text Classification involves the classification of a piece of text into one of the types. Language translation means the translation of them into a single language, voice recognition is a comprehension of the human voice and the following orders and several others [1]. The understanding of a

text is part of all analytical texts. In recent years, the industry has focused on natural language processing (NLP). Many methods & techniques to interpret the text are built. The NLTK has been designed to interpret text statistically and to program text data as the NLP Toolkit [2]. Some of the uses of the "Toolkit" for NLP include "tokenization" which means that the expression is broken into sentences; "tokenization" means breaking the sentence into separate terms, and word count defines the presence of the word in text or corpus in particular. The Stop-word is the word in English, it is a preposition with conjunction much of the time. Many applications for text processing exclude stopping words so they do not have any useful information. Text Blob, a regular expression parser, and Beautiful Soup are the other well-known text processing libraries. Text Blob server offers context for the majority of NLP applications along with the NLTK toolkit. In most text applications regular expressions are used, most of the time unwanted text contains unique characteristics and details. Regular techniques are used to delete unwanted unique features found in the data, as well as regular phrases for superior characteristics like dates and times in the text. Spacy [3] has been built into C Python as one of the fastest word processing kits. Spacy helps classify and recognize parts of speech tagging. Spacy as well as the identified objects like name of a user, year, institutional names & money current in a text may be used to identify the speech parts present in the text details with Spacy. Applications for text processing have developed every day and for instance, the entry of text in the search bar as "flights from Mumbai to London". Names like Mumbai and London are recognized by the name of the individual.

Spam filters are also of concern if the incoming email is classified as spam. In the publication sector where much of data is shared in text form, text analytics is most useful. The news articles with a mutual goal may be divided into common groups. Form defining & understanding data in text, sports articles, political articles, economic items & so on can be classified into common groups. Example of an Unsupervised ML (UML) strategy that does not have the target label. The common trends of the data are calculated and a range of identical data points is combined. The latest work in text applications wherever a computer responds to human beings is online catboats [4]. The catboats process and understand the context of the incoming text and respond to the consumer accordingly. In most industries, closed domain catboats succeed in understanding human texts and handling text, and dealing with the situation. Sentiment Analysis (SA)[5]. Is the most important application for companies. Customers share their responses to their product purchase or share their opinions about a specific issue. Several companies like Twitter, Amazon & Facebook want to know the sentiment of the produce or the actual subject. E.g., on Twitter, one can capture the text data using Twitter APIs and then identify the sentiment of the text. Many users who buy products express their thoughts on a product by providing material and experience.

II. The Inception of Text Classification

The text processing, classification & clustering applications are protected by NLP applications. The spam filtering applied in mails is easiest to use text analytics tool. They may classify an incoming email as spam or not depending on content of email. Spam filters are used to categorize email spam by ML or not depend on the text in an email. " Using the TF-IDF to assess Word Relevance in document queries," terms occurrence and made of different frequency methodology are applied to regulate most appropriate words in the corpus for a given query Prepare Your Paper Before Styling [6].

The term frequency is similar to how many periods a word in a article occurs & document frequency shows how often a similar word is used in other forms. measure TF-IDF helps to identify the significance of

words to a given document. High TF-IDF numbers suggest strong relation with a document in which they look and mean that document may be of interest to the user when this term appears in a query. This algorithm aims to improve its relevance to a certain query machine learning algorithm including the NB classifier (NBC), decision trees (DC), the Random Forest (RF), SVM & boosting approach are established to be effective in classification of text complications[7]. To perform the text classification, use the NBC. The classification task is done by the Naive Bayes classification system with the probabilistic distribution. This paper compares their performance empirically to five text firms. Due to its storage components, long short-term memory circuits have become popular for text analytics and can retain sequence data. The research team from Facebook suggested this study called "Bag of Tricks for Efficient Text Classification" [9] to increase the efficiency in representing text data as numerical vectors.

Various architectures of neural n/w are applied in the fields of text processing. The "Very Deep Convolutional Text Classification Networks" study uses a new architecture (VDCN) that is directly used for text processing and uses no convolutions and pooling. Sequentially obtaining a hierarchical image of input through several layers of convolutions and pooling. Unstructured data is text data. Before moving text onto some ML or DL algorithm, data should be converted into a numeric format. Text data is transformed into a vector format in various ways. About approaches use word occurrence and others use word frequency as a means of preserving meaning of a word in a sentence. This initiative utilized three methods of text processing:

- TF-IDF Vectorizer
- Bag of Words
- Word Embedding

A. Bag of Words

A BOW is a method in which a word bag is signified in a given sentence by occurrence or non-appearance of a word. In first instance, this method is a word corpus with all the words applied. The word rate is marked as one, whereas 0 is marked for the absence of a word. This method is also referred to as count victories. The phrase "How you work," for example, is shown in the word bag as shown in table 1: The bag of words is impacted by the disadvantage of meaning loss. The sequence and importance of the document cannot be captured by this technique. This software is used to treat the sentence "Dog bites man" and "Man bites Dog." In addition, the frequency of information repeated most often in the document is not retained.

B. TF-IDF

It, defined as the term occurrence and inverse occurrence of document, is applied to eliminate the downside of the word bag technique. It makes it important that a word has occurred several times in a particular document and that it appears rarely in all such documents. Equation 1 provides the frequency of the term and equation 2 shows the frequency of the inverse document.

$$tf(t,d) = 0.5 + 0.5.\frac{f_{t,d}}{\max\{f_{t',d:t'd}\}} \quad (1)$$

$$idf(t,D) = log\frac{N}{|\{d \in D: t \in d\}|} \quad (2)$$

Whereas $f_{t,d}$ is raw amount., N is total no. of documents = |D| & denominator |{d ∈ D: t ∈ d}|: no. of documents in which word t occurs TF-IDF changes text into vectors TF-IDF works more
Then a bag of words in many cases because it differentiates the Table 1: Bag of Words e.g.

Scores of the TF-IDF model from its counter document. For example, in the article on the sport, words on sport are used and in the article on political news, words on politics such as government, representative, etc. During the fine-tuning of the vectors with the TF-IDF vectorizers, terminology like

a ball, helmet, court, and stadium in the sports article even received a high score in political news articles, with words like government, president, etc. The value of stop words is also reduced by TF-IDF. Stop words are common words in the English language, particularly conjunctions & prepositions, etc.

C. Word Embedding

The scarce representation depended on vector agonizes from the disadvantage of losing the semantic & syntactic relations. Besides that, sparse document or sentence representation leads to errors, when the document set is too large and when processing large volumes of text with sparse representation is computerized very expensively. Every word in a dataset is embedded into feature vectors, this is done by creating an embedding matrix. An embedding matrix is a list of words and their corresponding embeddings. Embeddings usually refer to n-dimensional dense vectors. The embedding matrix is of shape (vocab_size, embed size). Here vocab_size is the number of words in the dictionary that are obtained from the tokenization method and embed size is the number of features into which the words will be embedded. There is a lot of pre-trained word embeddings available with different embedding sizes like the GloVe (Global Vectors for Word Representation), word2vec, Fast text-crawl, etc. This paper uses fast text crawl-300d-2m for the embedding matrix. This embedding matrix is then passed to different algorithms. The challenge for the applications for NLP was to understand the sense in the text also to document the sequence within the sentence. It is one way to preserve semantic and syntactic relationships when a sentence or document is signified in a matrix. The weights are assigned to words surrounding the weight of a word. The algorithms are trained to record syntactic & semantic relationships of words in a sentence, so that term integration can be applied in a wider range of apps, e.g. language translation, voice recognition, sentiment analysis (SA), & many other applications [8]. The relation between the words can be identified using word insertion.

D. BERT

Bidirectional Encoder Representations from Transformers are different from other word embeddings on the basis that, unidirectional word embeddings only lookup words sequentially left-to-right or right-to-left. BERT makes use of the transformer encoder to read a complete series of words at once. Hence it is regarded as bidirectional, or non-directional. This function enables the model to acquire a word context-dependent on the word context.

E. Fast Text

Similarly, as with other word embedding techniques, FastText creates a vector space representing words. Differences are in the word vector representations. Each vector represents the character n-grams of the word. (e.g., “matter” with n = 3, has a FastText representation).

III. Literature Review

Several papers are focusing on a brief analysis of the Aggression by Text is a complex phenomenon, & diverse areas of knowledge try to examine & deal with the described problem. This study focuses on a computer science approach for the detection of aggressiveness, a growing new discipline. Currently, the scientific research of automated recognition of violent writing, utilizing information technology approaches, is expanding. In this research, various linked pieces of genres of literature are employed to convey distinct categories of aggressiveness. Some of those hate cyberbullying, and abusive language toxicity. Image processing (IP) was widely used in medicine.

Deep Learning has gained wide popularity across various fields of interest ranging from a theoretical study in academia to practical implementation in industry. Newly emerging algorithms have extended

applications of Learning Machines to be able to compute vast quantities of data with the aid of hardware implementations and optimized parallel computing techniques available today. With this, Deep Learning (DL) Architecture can be generalized, adapted, and optimized to facilitate effective and precise models. DL is a favoured choice as efficiency improves as the data scales compared to other techniques e.g. regular Machine Learning algorithms. ML approach like SM, RF, etc. When data increases in these algorithms, performance plateaus. For my lung cancer detection research, Deep Learning Neural Networks work well as our dataset is large and can iteratively grow larger as more individuals infer new data into it, resulting in a performance boost.

Rahul et al.(2020) examined the extent of online harassment and categorized the material as correctly as possible into labels to analyze toxicity. Here, they are using six ML algorithms, which they use to resolve the text classification issue and find the best machine-learning algorithm based on their evaluation measurements for the classification of toxic comments. They are aiming at toxicity with extreme accuracy to limit its negative effects, an incentive for companies to take the appropriate steps. Toxic comments on the Web are disrespectful, abusive, or irrational, frequently leading to a discussion among other users. To minimize the disagreement of the people, the risk of online bullying and harassment inhibits freedom of thought. Sites fail to successfully facilitate dialogue and restrict or close user feedback in certain cultures [10].

Mestry et al.(2020) Developed a fast text word embedding technique to use a text-based CNN with word embedding. Rapid Text's findings were more efficient and accurate than that of GLOVE and Word2Vec. Their model seeks to improve the detection of multiple toxicity forms to enhance the experience of social media. Their model classifies these observations into 6 different classes: Toxic, Severe Toxic, Obscene, Threat, Abuses & Hate of Identity. Multi-Label Classification helps us deliver an automatic response to the toxic feedback they face. They have the power to share their opinions & ideas through social networking and online conversations. Manually, it is a very lengthy, exhausting, and unreliable method to define & classify these statements. To address this challenge, they have built a DL algorithm that detects & effectively classifies certain negative content in online platforms for discussion[11].

Shang et al.(2019) Resolved problem is that the videos themselves have no hateful content but unpredictably cause hateful comments from the viewers to identify online despise videos. It is not ideal if despised videos are simply viewed as hateful and removed from sharing platforms. In the future, uploaders of these videos would not be allowed to bring valid & useful videos. Nevertheless, the treatment of these hateful videos would provide hateful users with unwanted opportunities to spread their toxic comments & radical ideologies. With this paper, they developed VulnerCheck, a comprehensive, supervised approach to research structure & semantics of public comment networks On a real-world dataset composed from YouTube, they evaluate VulnerCheck. Outcomes show that their system is efficient & effective in identifying despised videos & beats state-of-the-art expectations significantly. Owing to the increased popularity of online video platforms (Twitter, Vimeo), hateful videos have been a major problem, and lack of rigorous hateful control of the content [12].

Ibrahim et al.(2019) Define a Solution consisting of a set of 3 models: CNN, LSTM, and GRU. The issue is separated into 2 stages: first, they decide whether the input is toxic & then toxicity forms are found in toxicity content. Results of evaluation reveal that the ensemble method proposed has the highest accuracy of all algorithms considered. 0.828 F1 for toxic/non-toxic classification & 0.872 for estimation of toxicity

types. In much of the public online communities, cyberbullying and online harassment has recently been two of the most severe problems. They use data from modifying the Wikipedia discussion page to train multi-label grade students, which detect various forms of toxicity in content generated by users online. To solve data imbalance problems on the Wikipedia data set, they present different strategies for data increase [13].

Chandra et al.(2018) Proposed posts including comments that are racist & malicious be effectively banned & removed to prevent them. This article usages documentation of racist comments as a text to identify racist comments & meaning using an effective ML algorithm. The extent of similarity between a pair of text messages as a source and categorized terms that are either anti-social or discrimination should be identified to detect anti-social content. The method to detect antisocial activity in this article is a methodology focused on the frequency classification of content. Millions of submissions are made daily in form of Facebook tweets, social blogs, and online discussions through various online communities and social media sites [14].

Takeda et al.(2016) suggested a tool for classifying tweets with tree nodes generated by Wikipedia categories in this study. Besides that, through use of a system for tweets related to tourism, they have established a recall system for tourism videos. Many web services like product reviews & Twitter have been sent with brief feedback. Automatic & precise text detection may go to the development of novel web services & systems. Bag-of-word presentation frequency was commonly used as a standard technique for text classification [15].

IV. Proposed Work

This paper aims to detect toxicity (for eg. rudeness, disrespect, or threat) in user comments throughout online communication. Toxic comments are a problem in the current web world. People often indulge in trolling and hatred on social media etc. these days. They propose to identify the toxicity in 6 different languages, and their model uses only very little training data from 3 of these different languages and almost no training data for the other 3 languages. Hence, builds a model to deal with complications such as toxicity, misinformation, and harassment radicalization online continued this project. The focus is on word embedding where word embedding is a depiction of words where words are given a numerical weightage such that similar words from any language will have the same values, to that have similar meanings have a similar representation. Hence, their project can handle 6 different languages easily.

A. Proposed Methodology

We have used three models (LSTM, GRU,RNN and BiLSTM).We used Logistic regression for classification. Using a logistic function to define a binary dependent variable is the simplest form of logistic regression. However, extensions that are more complex exist. To guess the parameters of a logistic regression (or logit regression) model, one uses logistic regression. In this work, we will be using a combination of logistic regression models and four neural network models –LSTM, RNN, GRU, and BiLSTM. Due to this ensemble effect, we expect our system to be robust so that it can identify toxicity in sentences of any given language. For the preprocessing of toxic words, we used the tokenization approach. NLP involves a lot of tokenization. Natural Language is built on tokens. Using a process known as tokenization, text may be broken down into smaller, more manageable chunks. Words, characters, or sub words are all acceptable as tokens. Kaggle multilingual poisonous comments were used to collect the data (jigsaw MLTC).[.we](.we) have worked on the 2nd, 3rd, and 9th datasets.

B. Tokenization Code for Preprocessing

Text is what a token represents. Tokenizer (num words=none) max_len = 1500

token.fit_on_texts (list (xtrain) + list (xvalid))
Trainset = token. texts_to_sequences (xtrain)
Xvalid seq is equal to a token (xvalid)
#zero pad the sequences
Xtrain pad is the sequence. There are a number of functions that may be used to pad sequences (xtrain seq, maxlen=m x len).
Xvalid pad = sequence. Pad sequences
Invalid Sequence, Maximum Length Equals Maximum Length
Token is what word index is.

C. Pre-processing

1) Removal of Punctuation:

All punctuation marks in every comment are indifferent.

2) Lemmatization

The term "lemma" refers to a word's inflected forms, such as distinct verb tenses or singular/plural forms. Inflected variants of the word, such as go and gone, are examples of a lemma. Lemmatization is the process of combining this lemma. As a result, every remark is subjected to lemmatization.

3) Stop words Removal

Stop words are often occurring in everyday words such as posts, prepositions, and so on. As a result, with each comment, stop words are omitted.

4) Training:

Any of the pipelines are free to make predictions on their own. However, in the case of the mark extreme toxic, it is self-evident that unless a statement is found to be toxic, it has no risk of being classified as unadorned toxic. result in all test cases that are not harmful being labelled 0 for the extreme toxic label.

V. RESULTS AND DISCUSSION

In this section, we have compared four approaches and experiments for detecting toxic comments using our proposed methodology.

TABLE 1: THE PERFORMANCE TABLE OF 4 DIVERSE NETWORK

Network type	Layers	Training Loss	Training acc	Val loss	Val acc
LSTM	3	0.1314	0.9514	0.0209	0.96
GRU	3	0.1233	0.9557	0.1123	0.98
RNN	2	0.2209	0.9215	0.3209	0.77
BiLSTM	4	0.1112	0.998	0.1012	0.991

In the above table, the loss in training is the flaw in the data set for training. Validation loss is the error after running the validation collection of data via the trained network. The ratio of the two trains/is valid. Unfortunately, both validation and training errors decline over the period. The lack of validity is the same as the loss of training, but it is not used for updating weights. This is determined in the same manner - by using a loss function to run the network over inputs and compare network outputs with the ground-truth values. The loss of training, the accuracy of training, loss of validity, and validation accuracy are all listed in the above table.

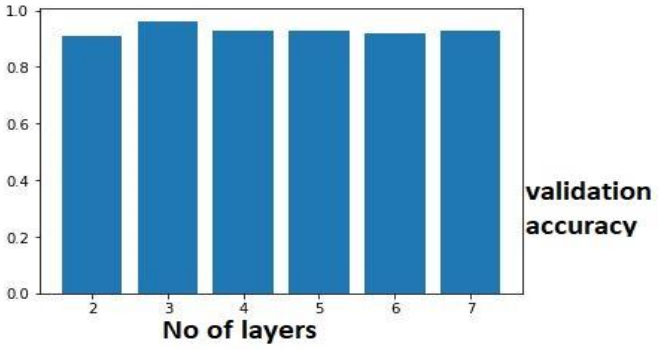

Figure 1. Performance of LSTM with varying number of layers

As can be seen in the graph above, as the number of convolution layers was increased the training accuracy increased but the validation accuracy started reducing. Hence using the “early stopping” method we stopped at the appropriate number of layers where validation accuracy was maximum. As can be seen, that happened in the above figure with layers = three.

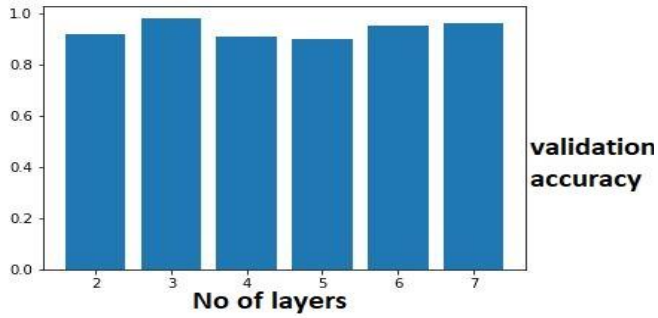

Figure 2: Performance of GRU with varying number of layers

In above graph, Using the "early stopping" method we stopped at layers = 3

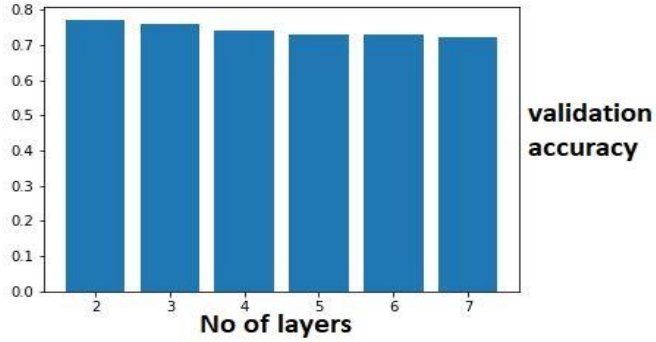

Figure 3. Performance of RNN with varying number of layers

In above graph, using the "early stopping" method we stopped at layers = 2.

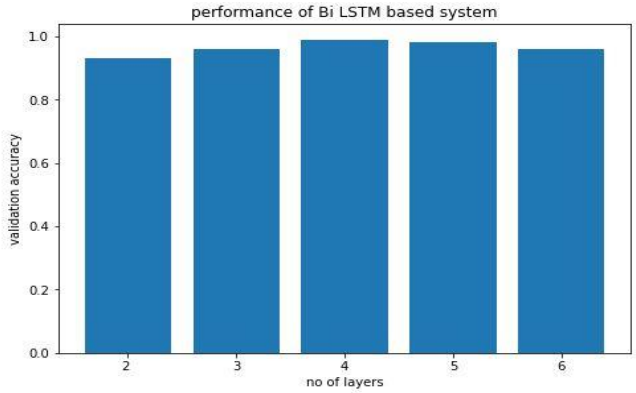

Figure 4. Performance of BiLSTM with varying number of layers

The above figure shows the BiLSTM performance based on a number of layers. By traversing the input data twice (i.e., left-to-right and right-to-left), bidirectional LSTMs (BiLSTMs) allow additional training. The research issue is whether BiLSTM outperforms standard unidirectional LSTM with additional training capabilities.

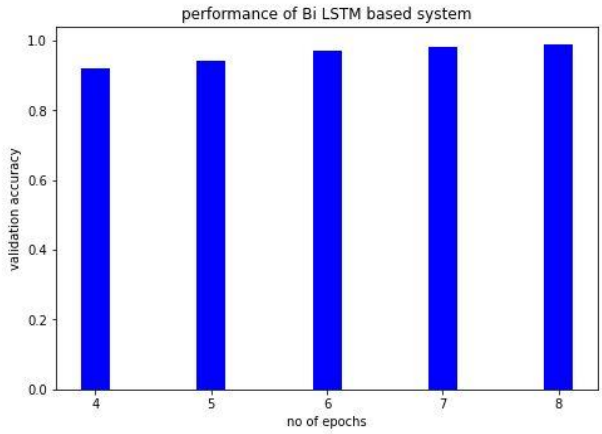

Figure 5. Performance of BiLSTM with varying number of Epochs

The above figure shows the BiLSTM performance based on number of epochs. By traversing the input data twice (i.e., left-to-right and right-to-left) based on epochs, bidirectional LSTMs (BiLSTMs). allow additional training

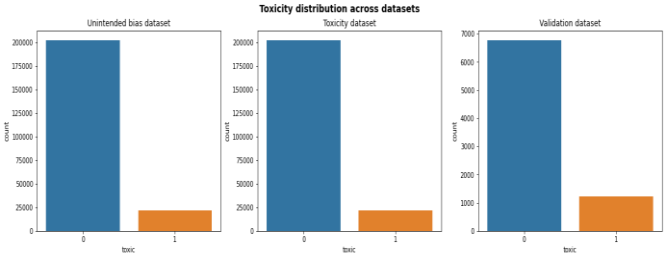

Figure 6. Toxicity distribution across the dataset

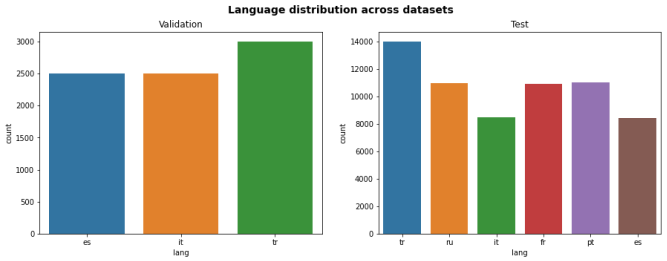

Figure7. Language Distribution across dataset

The above two graph shows the Toxicity distribution across the dataset and Language Distribution across dataset in which, we use four datasets to evaluate the prediction of specific toxicities.

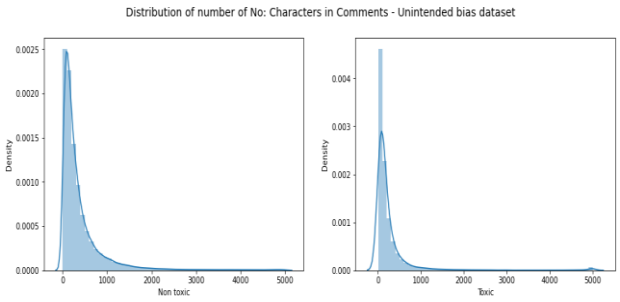

Figure 8. Distribution of Number of No: Character in Comments for unintended bias dataset Comments-Unintended bias Dataset

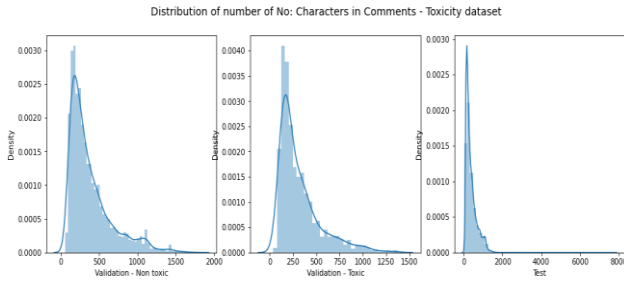

Figure 9. Distribution of Number of No: Character in Comments for Toxicity dataset

The above graph shows the Distribution of Number of No: Characters in Comments for Toxicity dataset. This study offered me some extremely useful insights into the distribution of my data. The next step was to undertake pre-processing of the data. The amount of the data supplied was decent enough for excellent analysis but not simple enough to work with.

```
model.fit(xtrain_pad, ytrain, epochs=5, batch_size=64*strategy.num_replicas_in_sync) #Multiplying by Strategy to run on TPU's

Epoch 1/5
2795/2795 [==============================] - 5626s 2s/step - loss: 0.1804 - accuracy: 0.9383
Epoch 2/5
2795/2795 [==============================] - 5736s 2s/step - loss: 0.1333 - accuracy: 0.9530
Epoch 3/5
 396/2795 [===>..........................] - ETA: 1:22:38 - loss: 0.1165 - accuracy: 0.9560
```

Figure 10. Loss and Accuracy in terms of Epochs Comments

The above graph shows the Loss and Accuracy in terms of Epoch Comments. Loss per epoch was seen when training my neural network using the no or Tensor Flow. Above figure shows the accuracy and loss values in terms of epoch which define that our proposed approach is efficient with existing approach.

Many algorithms have been developed in the past to prevent the problem of SPAM e-mails. These days there is a heavy flow of social communication chats as well as sites. People use forums like stack overflow, where people discuss popular computer science-related technical details. Similarly, people in the corporate world, doctors, and lawyers have got their social network channels over which they arrange technical talks meetings, etc. And in such an environment, identifying toxic comments is easy for human beings but it is difficult for computers. As a consequence, all data must be transformed into this form before it can be processed by the computer and returned to us. As a result, we must first evaluate and vectorise the input data, as well as extract characteristics from the text, into categories.

The dataset has 2 lakh images. We have used the multilingual toxic comments classification approach. In this approach, we have used 6 languages in which we have to find the toxic comments from these 6 languages. For pre-processing we have used tokenization and for classification, the logistic regression approach has been used. We have implemented the model using STM, GRU RNN, and BiLSTM method. The training and testing ratio is 2 lakh:7 Thousand. Accuracy on the training set is referred to as "Training Accuracy" (TA). For example, Validation Accuracy describes the test set accuracy.

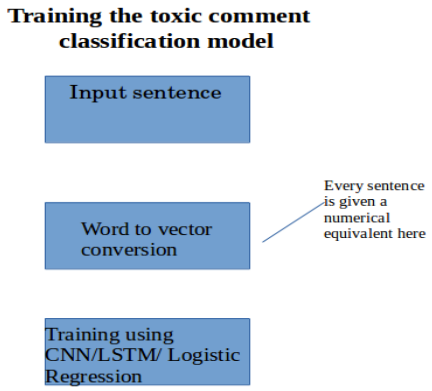

Figure 11. The classification model for training and toxic comments

The above figure shows the classification model for training and toxic comments. These include vulgarity, threats, personal insults, and references to one's identity as a target.

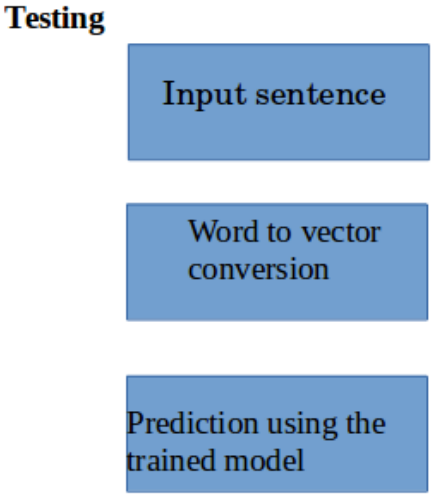

Figure 12. The classification model for Testing and toxic comments

The above figure shows the classification model for testing and toxic comments. It seeks to examine any piece of text and identify various sorts of toxicity.

VI. CONCLUSION

This study used a combination of Machine Learning and Natural Language Processing (NLP) to identify toxicity in user comments. LSTM, GRU RNN, and BiLSTM are the suggested models for online abusive comments categorization. Because of the collected findings, it can be concluded that BiLSTM is the most successful method for both testing and training. The goal of this study is to utilize logistic regression to create a model that can automatically determine if a remark is hazardous or not. As a result, this work attempts to construct a multi-headed model that can identify various forms and levels of toxicity. Toxic-classified comments can be collected and pre-processed for training and testing purposes. The Grid Search Algorithm may be used to construct a more robust model for every pipeline in the future, using the same dataset as the Machine Learning Algorithms to acquire more accurate results and classifications. This means that future results can be improved by, for example, varying experimental parameters like max features and maxlen.

VII. REFERENCES

[1]. Almerekhi, H., Jansen, B. J., Kwak, H., & Salminen, J. (2019). Detecting toxicity triggers in online discussions. HT 2019 - Proceedings of the 30th ACM Conference on Hypertext and Social Media. https://doi.org/10.1145/3342220.3344933

[2]. Berk, E., & Filatova, E. (2019). Incendiary News Detection Enis. Proceedings of the Thirty-Second International Florida Artificial Intelligence Research Society Conference.

[3]. McCallum, A., & Nigam, K. (1998). A Comparison of Event Models for Naive Bayes Text Classification. AAAI/ICML-98 Workshop on Learning for Text Categorization. https://doi.org/10.1.1.46.1529

[4]. Serban, I. V., Sankar, C., Germain, M., Zhang, S., Lin, Z., Subramanian, S., Kim, T., Pieper, M., Chandar, S., Ke, N. R., Rajeshwar, S., de Brebisson, A., Sotelo, J. M. R., Suhubdy, D.,

Michalski, V., Nguyen, A., Pineau, J., & Bengio, Y. (2017). A deep reinforcement learning chatbot. In arXiv.

[5]. Liu, P., Qiu, X., & Xuanjing, H. (2016). Recurrent neural network for text classification with multi-task learning. IJCAI International Joint Conference on Artificial Intelligence.

[6]. Ramos, J. (2003). Using TF-IDF to Determine Word Relevance in Document Queries. Proceedings of the First Instructional Conference on Machine Learning.

[7]. McCallum, A., & Nigam, K. (1998). A Comparison of Event Models for Naive Bayes Text Classification. AAAI/ICML-98 Workshop on Learning for Text Categorization. https://doi.org/10.1.1.46.1529

[8]. The amazing power of word vectors. (2018).

[9]. Rahul, Kajla, H., Hooda, J., & Saini, G. (2020). Classification of Online Toxic Comments Using Machine Learning Algorithms. 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 1119–1123. https://doi.org/10.1109/ICICCS48265.2020.9120939

[10]. Rahul, Kajla, H., Hooda, J., & Saini, G. (2020). Classification of Online Toxic Comments Using Machine Learning Algorithms. 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 1119–1123. https://doi.org/10.1109/ICICCS48265.2020.9120939

[11]. Mestry, S., Singh, H., Chauhan, R., Bisht, V., & Tiwari, K. (2019). Automation in Social Networking Comments with the Help of Robust fastText and CNN. Proceedings of 1st International Conference on Innovations in Information and Communication Technology, ICIICT 2019. https://doi.org/10.1109/ICIICT1.2019.8741503

[12]. Shang, L., Zhang, D. Y., Wang, M., & Wang, D. (2019). VulnerCheck: A Content-Agnostic Detector for Online Hatred-Vulnerable Videos. Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019. https://doi.org/10.1109/BigData47090.2019.9006329

[13]. Ibrahim, M., Torki, M., & El-Makky, N. (2019). Imbalanced Toxic Comments Classification Using Data Augmentation and Deep Learning. Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018. https://doi.org/10.1109/ICMLA.2018.00141

[14]. Chandra, N., Khatri, S. K., & Som, S. (2018). Anti social comment classification based on kNN algorithm. 2017 6th International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2017. https://doi.org/10.1109/ICRITO.2017.8342450

[15]. Takeda, M., Kobayashi, N., Kitagawa, F., & Shiina, H. (2016). Classification of comments by tree kernels using the hierarchy of wikipedia for tree structures. Proceedings - 2016 5th IIAI International Congress on Advanced Applied Informatics, IIAI-AAI 2016. https://doi.org/10.1109/IIAI-AAI.2016.62

Cite this article as :

Monika Dandotiya, Dr. Rajni Ranjan Singh Makwana, Nidhi Dandotiya, "An Intense Study of Machine Learning Research Approach to Identify Toxic Comments", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 8 Issue 4, pp. 71-81, July-August 2022. Available at doi : https://doi.org/10.32628/CSEIT228391
Journal URL : https://ijsrcseit.com/CSEIT228391