2021 International Conference on
Artificial Intelligence (ICAI)
Islamabad, Pakistan, April 05-07, 2021

A Novel Preprocessing Technique for Toxic Comment Classification

1st Muhammad Husnain
Department of Computer Science
National University of Computer and Emerging Science
Islamabad, Chiniot-Faisalabad Campus, Chiniot 35400, Pakistan
m.husnain@nu.edu.pk

2nd Adnan Khalid
College of Software Engineering
Northeastern University
Shenyang, China
1828034@stu.neu.edu.cn

3rd Numan Shafi
Department of Computer Science
National University of Computer and Emerging Science
Islamabad, Chiniot-Faisalabad Campus, Chiniot 35400, Pakistan
numan.shafi@nu.edu.pk

2021 International Conference on Artificial Intelligence (ICAI) | 978-1-6654-3293-1/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICAI52203.2021.9445252

***Abstract*—The threat of online abuse and harassment is increasing day by day in the cyber community. To tackle this problem, many platforms have devised policies. But these policies require prior identification of the content that is inappropriate and offensive. Furthermore, the data contains various aspects of negativity, for example, a particular piece of comment can express, disgust, disbelief, and threat at the same time. It points that even the negativity/toxicity exhibited in a comment can have various facets. Hence, the challenge is to identify what exactly is exhibited in comments so that respective policies can be formulated and applied to penalize the offender. This study makes use of two approaches to identify these underlying toxicities in the comments. The first approach is to train separate classifiers against each facet of the toxicity in comments. The second approach deals with the problem as a multi-label classification problem. Different machine learning approaches including logistic regression, Naïve Bayes, and decision tree classification are employed to carry out this study. The dataset is taken from Kaggle and 10-fold cross-validation is used to report the robustness of the model. The study uses a novel preprocessing scheme that transforms the multi-label classification problem into the multi-class classification problem. The preprocessing strategy has shown a significant improvement in the accuracies when employed for simple classification models encouraging its use for other sophisticated models as well. Experimental results show that in both the binary classification and the multi-classification, logistic regression turns out to be a better performer. This indicates the potential use of the preprocessing for the neural classification models.**

Keywords—*Text Classification, Text Mining, Toxic Comment Classification, Multilabel Classification, Word Embedding's, Mining Toxic Comments*

I. Introduction

The use of social media and the internet has mainly influenced daily life that the human civilization is affected drastically. Many online social platforms and e-commerce are widely used to discuss, emphasize or criticize each other's opinions through short comments [1]. Text classification is an essential portion in various applications, for example, information filtering, topic categorization, web seeking, and sentimental analysis [3]. In the text classification, a set of documents and classes are given to the defined function f that will assign a value from a set of classes to each document [21]. With the advancement of technology and its growth, there is a huge volume of data is generated in the form of comments. Unlike normal text, these comments are normally noisier, less topic-focused, and much shorter such that the lots of sentences are comprised of only a few words [2]. This type of information requires text mining perfectly and reasonably.

With the increase in online social discussions, an increase in cyber abuse and trolls is also significantly increased. As per a study conducted in 2017 by Pew Research Center, it is reported that 41% of the American internet users are being or been harassed online and 61% have at least witnessed these kinds of behaviors happening on others [5]. To deal with this situation of cyberbullying, many social platforms have their own devised policies. For example, Wikipedia policy states that one should not make personal attacks on others. If anyone is found guilty, respective accounts of such users are blocked in addition to removing personal attacks [6]. These policies can be made into action after identification of harassment. Therefore, it is of utmost importance to identify harassment in discussions and social media. A commonly used medium to bully others in the textual format is in form of comments.

Comments can carry negativity in varied respects. These comments tend to carry disgust and negative sentiments on various levels. Some comments are made just to show one's disbelief or distrust about the views. Others employ abusive language and derogatory remarks to register their opinion [4]. Therefore, analysis of comments to find out the number of negative sentiments carried within is an important task.

Hence, the primary purpose of this paper is to identify the level of negativity and toxicity in online comments. This analysis can help in filtering those who harass others. Thus, it would eventually help in implementing the policies and punishing those who do not respect these policies that can further be used to reduce the toxicity level in discussions [7]. Hence, this study presents a methodology to classify online comments with respect to their toxicity level.

The rest of this paper is structured as follows: Section II briefly describes the background and presents relevant literature. Section III describes the methodology opted to carry out the task. Section IV describes acquired results in the light of conducted experiments. Section V represents the conclusion and future directions for this work followed by the bibliography.

II. Background

In the last years, many researchers have dedicated their efforts to the analysis of comments. A Pioneer study that focused on comment abuse classification is presented in [7]. the study used the term frequency-inverse document frequency (TF-IDF) to the features and then applied support vector machine (SVM) algorithm. For abusive comment detection purposes, Google Jigsaw published a paper named "Ex Machina: Personal Attacks Seen at Scale" [11]. This paper proposed and discussed some of the data-mining algorithms to automatically detect toxic comments. For training, a dataset was taken from Wikipedia's Detox project [11]. This paper also talks about evaluation methods for classifiers that played an important role in the development of "Perspective API". This Perspective API is also proposed by google aiming at accommodating healthier conversations [13]. The authors of the paper [11] tested logistic regression and multi-layer perceptron on the top for prediction purposes. For feature selection, n-gram characters or word technique has been exploited which is based on bag of word model. This paper also describes data labels in terms of one-hot vector and empirical distributions. In rest of the paper is focused on determining relations or co-relations between comment class with individual (or commentator) identity and commenting patterns.

Because abuse classification is still a relatively new research topic in NLP and there are legal and privacy issues with making this information public, only a few datasets have been accumulated especially for this problem in general. However, we have studied different datasets used by others in the same field and have found things in this dimension that can be useful for making our data set more precise.

For example, the authors of [8] were successful in accumulating only a little more than 2510 comments with manual labels. Approximately, half of them were positive (i.e., with no personal abuses) while the other half of the dataset was abusive or negative. Similarly, [9] could get only about seven hundred text messages from a well-known telecommunication company. Kaggle released a dataset, as one of their competitions titled" Detecting Insults in Social Commentary" [10]. And in conjunction with Google's Perspective API released in February 2017, the Wikipedia Detox project released a publicly available dataset containing comments scraped from Wikipedia's talk page diffs. The dataset also included Crowd flower-sourced labels for comments based on aggression, toxicity, and personal attacks.

Recently in February 2018, a paper titled "Convolutional Neural Networks for Toxic Comment Classification" introduced a new model for toxic comment classification based on convolutional neural network (CNN) [14]. Instead of using the traditional bag-of-words model, this paper uses CNN over document term matrix (DTM) and compares the results with other well-known classification models such as k-Nearest Neighbors (KNN), support vector machine (SVM), Linear Discriminant Analysis (LDA), and Naïve Bayes (NB). The dataset used by the authors of this paper was the same Kaggle toxic comment challenge dataset. The use of CNN made it possible to achieve accuracy greater than previously achieved by others (i.e., approx. 91.2%) [14].

It can be noted that most of the researchers of this field have generated their own datasets. Manual generation of the datasets is not bad, but they all have some constraints on dataset size. So, we decided to use a dataset provided by Kaggle.com that is crawled from Wikipedia talk pages which is the same used by [14]. This dataset contains more than 150 thousand examples for training and is the largest among all other available datasets previously.

III. Research Methodology

In this section, the methodology that is employed to carry out this research has been briefly described. Fig.1 shows the overall workflow.

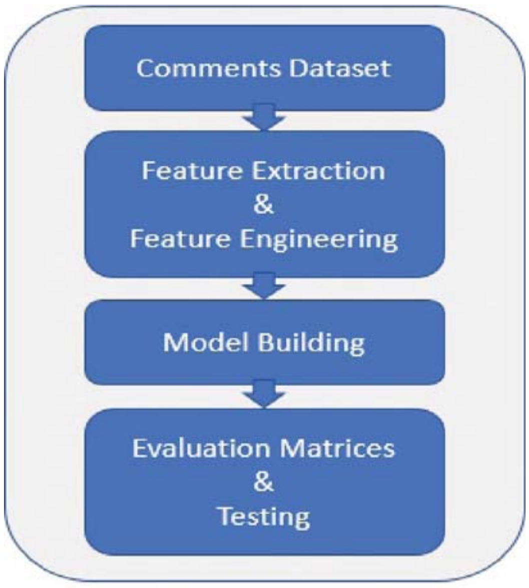
Comments Dataset
Feature Extraction & Feature Engineering
Model Building
Evaluation Matrices & Testing

Fig. 1. Overall workflow

A. *Dataset Specifications and problem Context*

Dataset for this research has 1,59,571 comment observations in total. For each observation total of 6 binary class labels are possible. These class labels are Toxic, Severe-Toxic, Obscene, Threat, Insult, and Identity-Hate. One comment observation can belong to multiple classes such as a comment can belong to toxic, severe-toxic, and insult classes at the same time [16]. The detailed specifications of the dataset are described in Fig.2.

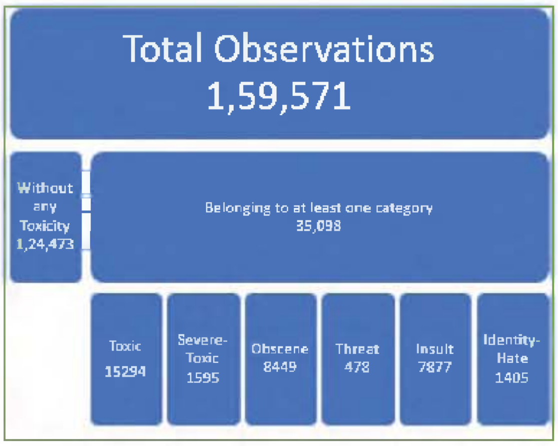

Fig. 2. Detailed Dataset Specifications

Now the problem context is quite clear. It is important to note that problem is multi-label classification instead of the multi-class problem.

B. Feature Extraction and Feature Engineering

After the dataset acquisition, the next important step is to find features that are to be used for classification purposes as an input to the model. Finding the appropriate features for any data mining problem is a very critical task. This process often includes a lot of trial and error, trying to identify what patterns in the data are the true representative of each class. Finding features is often considered an iterative process and in our case the process was exploratory.

First, to extract features, data was pre-processed. The pre-processing has been performed by exploiting multiple morphological techniques including tokenization, stop-words removal, and stemming. Tokenization refers to breaking down the sentences into individual tokens, where a token is a single word entity in a sentence. Stop-words refer to those words that are abundant in the dataset but do not give very useful information that includes conjunctions, pronouns, helping verbs, etc. such as is, am, are, he, she, it. After removing such words, the overall dataset is greatly reduced. Further, by applying stemming, the size of data is further minimized as stemming refers to mapping the word into the base word that is not necessarily a valid word.

After this pre-processing of data, feature extraction was started out by trying different lengths of the n-grams. N-grams refers to the words of length N. So, one-word n-gram carries a single token, a bigram would comprise of two tokens and so-on. In this study, 1-gram, 2-grams, and 3-grams are studied and analyzed for feature analysis. After analysis, it was found that 2-grams are better.

So, by means of applying case-folding and stemming on the original dataset, the dataset was normalized. Later, TF-IDF scores against each token were calculated. The top 10,000 features carrying the highest TF-IDF scores were selected. Thus, based on these features, a comment was represented in the form of a ten thousand-dimensional word-vector. So, a new dataset was made which consisted of 1,59,571 word-vectors for each comment. Hence, these word vectors against each comment were later used for training and evaluation of machine learning algorithms.

In addition to these features, through closely analyzing the dataset, it was found that a lot of the comments didn't follow the grammar rules. In other words, many sentences were written without a proper sentence structure. In addition to that, the sentences in the dataset also contained an unreasonable number of punctuations. Sometimes, these punctuation marks are too many and in some other sentences, they are too few.

To develop the various models, simply bag-of-words representation is used. Previous studies have shown that syntactic and morphological features do not provide more accuracy than n-gram features [12] in the detection of abusive comments or phrases in comments in an online discussion. Therefore, in this study, n-gram features are being used.

C. Model Building

After feature extraction and engineering, the next step is to construct the machine learning model. This work has treated the problem of identifying personal attacks in two ways. The first way is to treat toxic comment classification as a binary text classification problem that will involve training separate models against each class. The second approach is to deal with this problem as a multi-labeled classification problem.

Binary Classification:

In the case of binary classification, each target class has its own classifier. Hence, separate classifiers are trained against each target class. The approaches used to build these classifiers include 'logistic regression', 'Naïve Bayes', and 'classification and regression tree (CART)' classifier. Logistic Regression is a very nice option when the dependent variable is binary '0' or '1'.

Multi-labeled Classification:

First of all, we added one more class column in the dataset which we called 'Multi-label'. This class attribute is simply a string generated by concatenating binary 0 or 1 values of each type of toxicity. For example, for a comment observation, if values of toxicities are 1,0,0,1,0,0 for label columns 'toxic', 'severe-toxic', 'obscene', 'threat', 'insult', and 'identity-hate' respectively then 'Multi-label' value for this particular observation is '100100'.

When treating the problem as multi-labeled, sixty-four (64) different class combinations are possible. As, one comment can carry multiple labels at a time, and further, each toxicity class can further have two values only. Therefore, the total possible combinations can be $2^6 = 64$ as there 6 class columns. All these combinations are not valid due to the problem nature. Out of these 64, only 33 are valid combinations (i.e., 32 combinations with 'toxic' value equal to '1' plus 1 combination of the form '000000'). This is because, if a comment is not 'toxic', this simply implies that it does not have any type of toxicities. Out of 64, 31 combinations are those which have value for label column 'toxic' as '0' but, values equal to '1' for one or more label columns. The approaches used to build this classifier include logistic regression, decision tree classifier, and multinomial Naïve Bayes classifier. Here important to note is that logistic regression is a binary classification model, but the standard sci-kit-learn library allows One-verses-Rest (OvR) scheme for multi-class classification if the value of the 'multi-class' option is set to 'ovr' [15]. Therefore, we have used logistic regression for this multi-class problem.

D. *Evaluation Matrices & Testing Strategies*

There exist a variety of measure to evaluate the performance of a classifier. This study has employed various metrics including training-accuracies and testing-accuracies (or train-test-split). Furthermore, K-folds cross-validation is used to evaluate and report the results.

Training and testing accuracies are initially necessary to determine the over-fitting or under-fitting of the model. While cross-validation confirms the true accuracy of the model. To get the training accuracy of a model, simply a training dataset is used as a testing set without any change. For testing-accuracy, the given dataset is petitioned into two parts 1) Training set 2) Testing set. the train-test split used in this study is 80-20 that is 80% of data is used for the training set whereas the remaining 20% is used for testing.

K-folds cross-validation technique is used to remove the bias of data. As it involves the inclusion of every instance as a training and testing set. This technique makes K folds of equal sizes. One of the K folds is used as a testing set and all other K-1 sets as the training set. This procedure is repeated for each of the folds and the accuracy is stored. In the end, accuracies obtained at all the folds are averaged and the result is returned. The widely-used value for K-folds is 10, hence, in this study, 10-fold cross-validation is being performed.

All these accuracies and average accuracies are being calculated against three models out of various machine learning models that are used to solve the classification problem. These models are Logistic Regression, Naive Bayes, and Decision Tree Classifier.

E. *Tools and Technologies*

All implantation is carried out in the Python programing language. Related libraries for stemming, tokenizing, training, testing, and results visualizing are employed that include Snowball Stemmer of NLTK library [19], ScikitLearn [17], Numpy, SciPy [18], Pandas and matplot [20], etc.

IV. Results

One of the objectives of this study is to develop models that can help in identifying toxicity levels in comments that appear in an online discussion. This task is carried out using two approaches. This section highlights the results achieved against both approaches. Furthermore, results using various classifiers are also described.

A. *Binary Classification of Comments*

As discussed earlier when treated as a binary classification problem each type of toxicity is classified separately. Results from different models are reported below.

Train Test Split Accuracies:

Fig.3 illustrates that Naïve Bayes and CART are performing almost equally. While logistic regression is a better performer.

In part (a) of Fig.3, train-test-split accuracies of logistic regression are depicted for every value of C. Here C is a positive float constant which is inverse of regularization. Lower values of C indicate high regularization. Results are an average of ten splits for each value of C. These ten splits have different random states for each split. This result shows that a high when the value of C is 2, then LR is performing at its peak. For higher values of C, performance going down. This indicates that as regularization decreases the performance of LR also decreases.

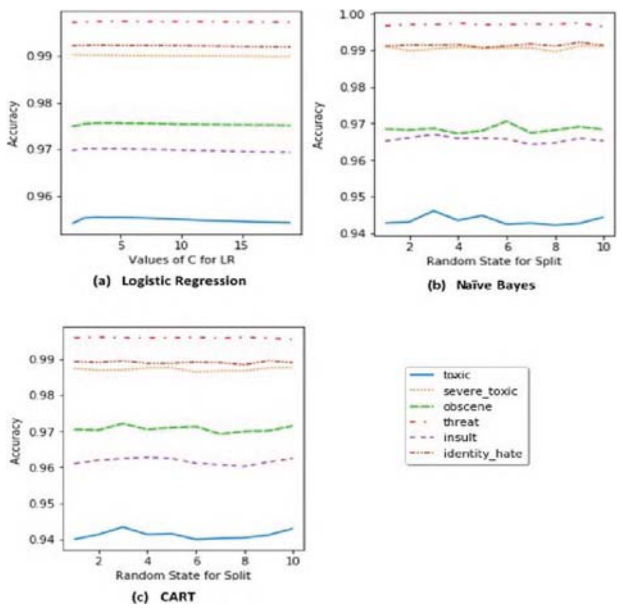

Fig. 3. Train Test Split Accuracies

Part (b) of fig.3 shows testing accuracies of the Naïve Bayes classifier on testing sets that are obtained using different random states for splitting the dataset into training and testing sets. Here, the random state is a seed value used by the pseudorandom number generator of the splitting algorithm in sci-kit learn. If it is set to some constant value then for each execution of the split function, always same training and testing set is made.

Part (c) of fig.3 displays train-test-split accuracies of Classification and Regression Tree (CART) classifier for different random states of train-test split function. By nature, CART tends to overfit on datasets with thousands of features. To avoid this overfitting, we have used 'selectKbest' method of the 'feature selection' class of sci-kit learn. The selectKbest method takes the dataset with labels and returns features with K best scores according to the scoring function. The scoring function which is used in this study is chi2 pronounced as 'Chi-Square'. The Chi-square test calculates that how much a variable is independent of the class attribute. If the independence of a feature with the class attribute is high then its importance is very low. In the picture, results are shown after applying selectKbest with *k=30* and *score_func = chi2*.

Cross-Validations:

Table1 shows our results for 10-fold cross-validation on different classification models. While Fig.4 depicts a boxplot Comparison of Logistic Regression and Naïve Bays. Box plots of model accuracies are useful to show robustness towards the split state of the splitting algorithm. It shows variations in the accuracies during cross-validation and median accuracy. Box plot usually "five number summaries" of data. This summary includes minimum, 1st (lower corner of the box) quartile, median (shown in Orange line), 3rd quartile (higher corner of the box), and maximum. It can also be used to depict suspected outliers in the data which are depicted by small circles on the box plot. They are usually the data points that are *1.5×IQR* or more above than 3rd quartile or *1.5×IQR* or more below the 1st quartile. Here IQR is Inter Quartile Range which is the difference between the 3rd quartile to the 1st quartile.

Table 1: Cross-Validation Accuracies

Class Labels	**Logistic Regression**	**Naïve Bayes**	**CART**
Toxic	95.39%	94.43%	93.71%
Severe Toxic	99.02%	99.05%	98.72%
Obscene	97.52%	96.87%	96.93%
Threat	99.71%	99.69%	99.64%
Insult	97.00%	96.54%	96.05%
Identity Hate	99.21%	99.12	98.97%

From the results depicted and illustrated in Table 1 and fig.4, it is clear that Logistic regression outperforms the other classifiers except 'Severe Toxic'. This is because, for each type of toxicity, logistic regression is giving better median accuracies. In only 'Severe Toxic' toxicity Naïve Bayes is performing better than logistic regression. Little circles in fig.4 top left shows that both LR and NB have at least one suspected outlier [exceptionally low] accuracy during cross-validation which is less than 1.5×IQR of the box plot. In all types of toxicities, CART is not performing well. In 'Obscene' toxicity CART is performing better than Naïve Bayes but not better than logistic regression.

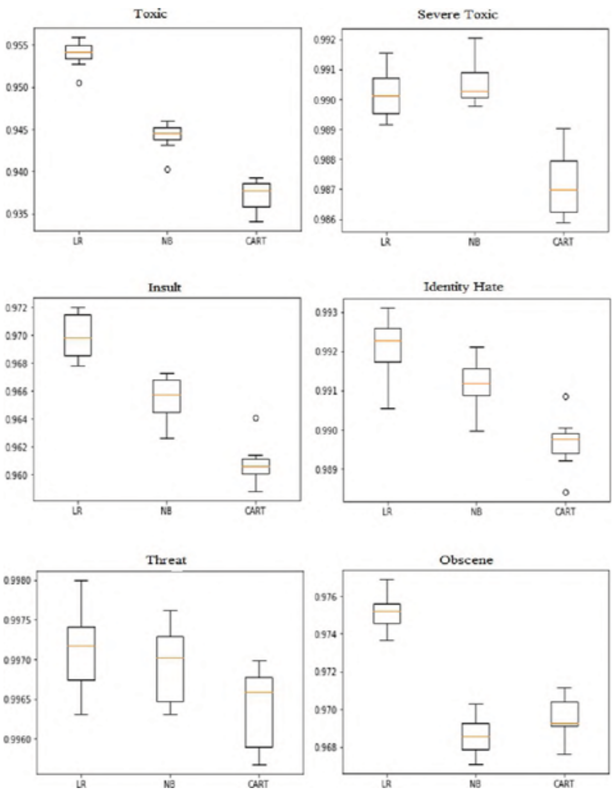

Fig. 4. Cross-Validation box plot comparisons of Naive Bayes and Logistic Regression for each kind of toxicity.

B. *Multi-Labeled Classification*

Tackling the toxic comment classification problem as multi-labeled transforms this problem from a binary classification problem to a multi-class classification problem. Now the total number of class labels is extended to 33. Because there is a total of 64 combinations of 6 binary variables. After keen observation, we were revealed by the fact that 31 combinations out of 64 are those that have 'toxic' value equal to zero but some toxicity type equal to one. Which is logically incorrect. So, these 31 are removed from class labels.

Training Accuracies:

Table 2 shows training accuracies of logistic regression (LR), Naïve Bayes (NB), and classification & regression tree (CART). From Table 2 one can think that the Decision Tree Classifier (i.e., Classification and Regression Tree) is best performing for this problem but we will see in the cross-validation section that the decision tree has been overfitting the dataset.

Table 2: Multi-labeled Training Accuracies

Classifier	**Training Accuracy**
Logistic Regression	93.3%
Multinomial Naïve Bayes	90.9%
Decision Tree Classifier	89.4%

Cross-Validation Accuracies:

Accuracies of the above three classifiers are cross-validated with 10-fold cross-validation. The following results in table 3 lead us to the conclusion that Decision Tree Classifier (i.e., Classification and Regression Tree) overfits over training set while giving poor performance over the test set. Logistic Regression is again best performing here.

Table 3: Multi-labeled Cross-Validation Accuracies

Classifier	**Cross-Validation Accuracy**
Logistic Regression	91.48%
Naïve Bayes	90.78%
CART	89.54%

Fig. 5 also depicts the same results in a Box plot for an in-depth understanding of accuracies of classifiers.

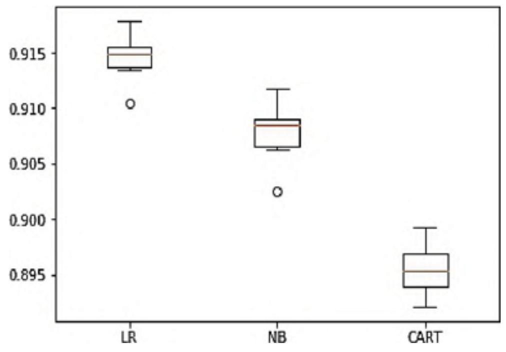

Figure 3. Box plot of Cross-Validations

From fig. 5, we can clearly get the idea that Logistic Regression is the best performer here also. Furthermore, results show that in the case of logistic regression accuracy never dropped below 91.4% except for one suspected partition. When here it comes to comparing the LR results with NB we can clearly see that in all the partitions of cross-validation LR is at least equal or better than the highest accuracy achieved by the NB classifier. Here, CART is not performing so well as others, but it is more robust to different partitions of the dataset used during10-fold cross-validation for training and testing.

V. CONCLUSION

Due to the increase in online discussions after the advent of social media, there has been a huge increase in cyber harassment. Various social discussion portals and social networking sites have devised policies to penalize the offenders. However, identification of the type and level of negativity/toxicity contained within a comment presents a huge challenge. Therefore, the study has proposed two approaches to deal with this problem. Firstly, a dataset downloaded from the Kaggle competition is analyzed. After thorough data analysis, key-features that are significant indicators of toxic comments are identified. Later, two different approaches are applied to perform the classification that includes binary classification against each facet of toxicity and multi-label classification. In addition, various machine learning approaches are also applied against each classification scheme. 10-fold cross-validation is applied to perform the evaluation. In the light of experiments, Logistic Regression tends to outperform various other machine learning approaches including Naīve Bayes and Decision Tree Classifier. The promising results of LR after applying the novel preprocessing strategy suggest its applicability to the neural models.

In the future, the work can be extended by employing the recurrent neural network (RNN) model. Moreover, community-bias analysis can also be incorporated into the models. This will allow the models to train according to the community standards of toxicity.

REFERENCES

[1] G. Song, Y. Ye, X. Du, X. Huang, and S. Bie, "Short Text Classification: A Survey." Journal of Multimedia, vol. 9, pp.635-643, 2014.

[2] Y. Rui, C. Xian-bin, L. Kai, "Dynamic Assembly Classification Algorithm for Short Text," ACTA ELECTRONICA SINICA, vol. 37(5), pp. 1019-1024, 2009.

[3] C. Aggarwal and Z. ChengXiang, "A survey of text classification algorithms," Mining text data, Springer, pp. 163– 222, 2012.

[4] Maeve Duggan, "Online harassment," Pew Research Center, 2014.

[5] "Wikipedia: No personal attacks.," Wikipedia, https://en.wikipedia.org/wiki/Wikipedia:No_personal_attacks.

[6] M. Duggan, "Online Harassment 2017," Pew Research Center: Internet, Science & Tech, Jul 11, 2017.

[7] "Harassment consultation 2015–Meta," Available: https://meta.wikimedia.org/wiki/Harassment_consultation_2015. [Accessed: May 28, 2019].

[8] A. Maus, "SVM approach to forum and comment moderation," Cl. Proj. CS, 2009.

[9] A. Mosquera, L. Aouad, S. Grzonkowski, and D. Morss, "On Detecting Messaging Abuse in Short Text Messages using Linguistic and Behavioral patterns," ArXiv Prepr. ArXiv14083934, 2014.

[10] P. Goyal and G. S. Kalra, "Peer-to-peer insult detection in online communities," IITK Unpubl., 2013.

[11] E. Wulczyn, N. Thain, and L. Dixon, "Ex machina: Personal attacks seen at scale," in Proceedings of the 26th International Conference on World Wide Web, pp. 1391–1399, 2017.

[12] L. Song, R. Y. Lau, and C. Yin, "Discriminative Topic Mining for Social Spam Detection.," PACIS, pp. 378, 2014.

[13] "Perspective," Available: https://www.perspectiveapi.com/#/. [Accessed: July 12, 2019].

[14] S. V. Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis, and V. P. Plagianakos, "Convolutional Neural Networks for Toxic Comment Classification," ArXiv180209957 Cs, Feb. 2018.

[15] "sklearn.linear_model.LogisticRegression — scikit-learn 0.19.2 documentation." Available: http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html. [Accessed: May 16, 2018].

[16] "Toxic Comment Classification Challenge," Available: https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge. [Accessed: May 28, 2018].

[17] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Mach. Learn. Res., 2011, vol. 12, pp. 2825–2830.

[18] E. Jones, T. Oliphant, P. Peterson, and others, SciPy: Open source scientific tools for Python, 2001.

[19] Bird, Steven, Edward Loper and Ewan Klein , Natural Language Processing with Python. O'Reilly Media Inc, 2009.

[20] J. D. Hunter, "Matplotlib: A 2D graphics environment," Comput. Sci. Eng., 2007, vol. 9(3), pp. 90–95.

[21] MR. Murty, JVR. Murthy, and P. Reddy. "Text Document Classification based-on Least Square Support Vector Machines with Singular Value Decomposition," International Journal of Computer Applications, vol. 27(7), pp. 21-26, 2011.