Proceedings of the SMART–2019, IEEE Conference ID: 46866
8th International Conference on System Modeling & Advancement in Research Trends, 22nd–23rd November, 2019
College of Computing Sciences & Information Technology, Teerthanker Mahaveer University, Moradabad, India

Bangla Toxic Comment Classification (Machine Learning and Deep Learning Approach)

A.N.M. JuBaer[1], Abu Sayem[2] and Md. Ashikur Rahman[3]
[1,2,3]*Dept. of CSE, Daffodil International University,*
Dhaka, Bangladesh
E-mail: [1]jubaer15-7850@diu.edu.bd, [2]abu15-7682@diu.edu.bd,
[3]ashikur15-7723@diu.edu.bd

***Abstract*—Toxic comment classification problem is a popular classification problem nowadays. There are many attempts in English but it's rare in Bangla language. We tried to build a classifier for Bangla language. We tried different approach to find the optimized classifier with better accuracy and optimized for log-loss, hamming-loss. As this is a multi-level problem, we used binary relevance methods for binary classifiers.**

Keywords: Bangla Toxic Comment, Machine Learning, Deep Learning, Binary Relevance, MultinomialNB, Classifier Chain, GausseanNB, Label Powerset, MLkNN, BP-MLL Neural Network

I. Introduction

Toxic comments are comments that irritate people and spread hates among community. So, to keep the environment clean, there needs a regulation over online conversation. We were first introduced with toxic comment classification problem at www.kaggle.com by Jigsaw/ Conversation AI [8]. They provided a huge dataset. Inspired by that, we decided to do the same with Bangla language. But the problem was the dataset. We build a dataset taking comments from Facebook pages posts. Our dataset has seven columns. One for feature and six for labels. The labels are representing the six different forms of toxic comments. The feature is comment's texts and the labels are toxic, severe toxic, obscene, threat, insult, identity hate.

We basically used Binary Relevance method for MultinomialNB, which is well known for classification with discrete features, SVM, GausseanNB, that is specially used when the features have continuous value, Classifier chain with MultinomialNB, which is a problem transformation method, that transform a multi-label classification problem in one or more single-label classification problems so that existing single-label classification algorithms such as SVM and Naïve Bayes can be used. We also used Label Powerset with MultinomialNB, another problem transformation method, MLknn(Multi-Label k-Nearest Neighbor) and at last BP-MLL(Backpropagation for Multi-Label Learning).

We first divide our feature and labels and converted labels as array, then from feature, removed Punctuation. Then tokenized our feature or comment text by CountVectorizer and removed Bangla stop words. We collected Bangla stop words from a GitHub Inc. repository [10]. Then we split our dataset into desired ratio and were good for implementation. After implementing all classifier, we visualized our results and compared between different classifier. The work flow is shown in the Figure 1.

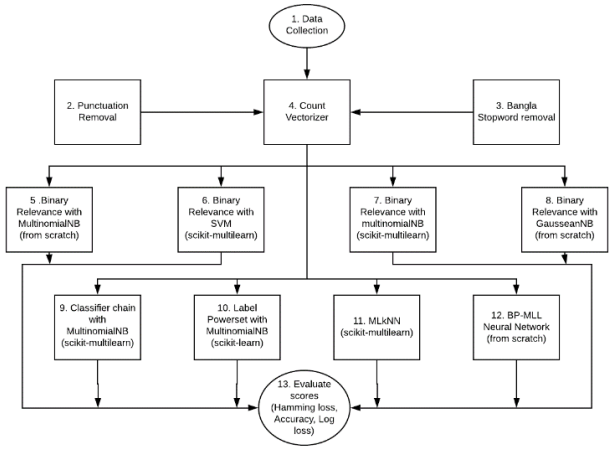

Fig. 1: Flow Chart of Work Procedure

II. Paper Review

Deep learning and shallow approaches do a good job in this regard. Betty van Aken et al. [1] showed this.

Dataset they used looks like Table 1 based on number of occurrences.

Table 1: Dataset of [1]

Class	Number of occurrences
Clean	201,081
Toxic	21,984
Obscene	12,140
Insult	11,304
Identity Hate	2,117
Severe Toxic	1,968
Threat	689

Comparison of precision, recall, F1-measure, and ROC AUC on Wikipedia dataset is shown in Table 2.

TABLE 2: RESULT OF WIKIPEDIA DATASET

Model	Wikipedia			
	P	R	F1	AUC
CNN(FastText)	0.730	0.860	0.776	0.981
LSTM(Glove)	0.740	0.840	0.777	0.980
Bidirectional LSTM (Glove)	0.740	0.840	0.777	0.981
Bidirectional GRU Attention (FastText)	0.740	0.870	0.783	0.983
Logistic Regression (char-ngrams)	0.740	0.840	0.776	0.975
Ensemble	0.740	0.880	0.791	0.983

There is an analysis on sarcasm detection of Debanjan Ghosh et al. [3], which is pretty similar to our work. They basically used SVM with discrete features (SVMbl), Long Short-Term Memory Networks. Their results are shown in Table 3.

TABLE 3: RESULT OF DIFFERENT CLASSIFIER ON SARCASM DETECTION

Experiment	P(S)	R(S)	F1(S)	P(NS)	R(NS)	F1(NS)
SVM^{r}_{bl}	65.55	66.67	66.10	66.10	64.96	65.52
SVM^{c+r}_{bl}	63.32	61.97	62.63	62.77	64.10	63.50
$LSTM^{r}$	67.90	66.23	67.10	67.02	**68.80**	67.93
$LSTM^{c}$ + $LSTM^{r}$	66.19	79.49	72.23	74.33	59.40	66.03
$LSTM^{conditional}$	**70.03**	76.92	**73.32**	74.41	67.10	**70.56**
$LSTM^{r}a_{s}$	69.45	70.94	70.19	70.30	68.80	69.45
$LSTM^{r}a_{s+}LSTM^{r}a_{s}$	66.90	**82.05**	**73.70**	**76.80**	59.40	66.99
$LSTM^{c}a_{w}$+$LSTM^{r}a_{w+s}$	65.90	74.35	69.88	70.59	61.53	65.75

Another work related to toxic comment classification problem is of Spiros V. Georgakopoulos et al. [4]. In this paper, they used CNN to classify toxic text. Their encoding model is shown in Figure 2.

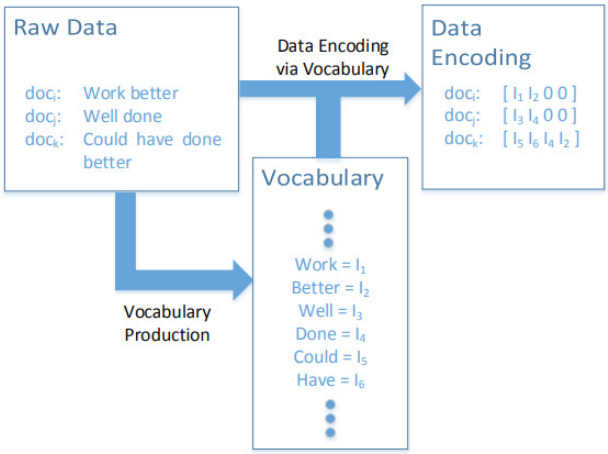

Fig. 2: Encoding Methodology

They mainly depended on Convolutional Neural Network for the toxic comment classification. The CNN work flow of them is shown in Figure 3.

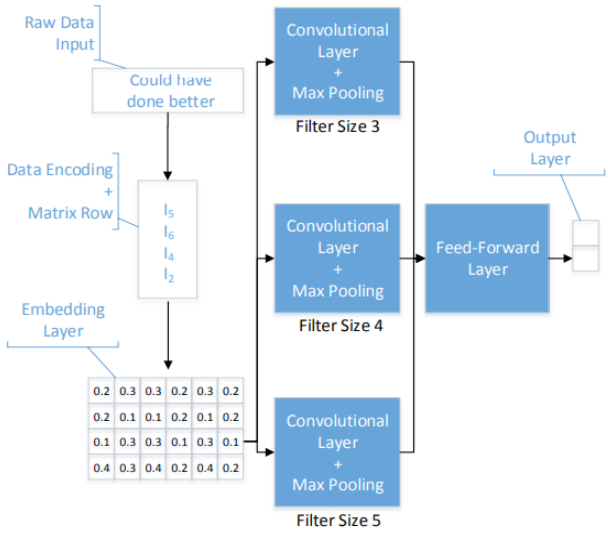

Fig. 3: CNN Work Flow

The result of various classifier of them [4] is shown in Table 4.

	Accuracy		Specificity		FDR	
	Mean	STD	Mean	Std	Mean	Std
CNN_{fix}	0.912	0.002	0.917	0.006	0.083	.007
CNN_{rand}	0.895	0.003	0.906	0.015	0.092	0.017
kNN	0.697	0.008	0.590	0.016	0.335	0.010
LDA	0.808	0.005	0.826	0.010	0.179	0.009
NB	0.719	0.005	0.776	0.012	0.250	0.010
SVM	0.811	0.007	0.841	0.012	0.167	0.012

III. DATASET

We took data from Facebook only, from different pages. We collect comments as data from pages. Then we labeled every single comment according to its meaning. Our labels are toxic, severe_toxic, obscene, threat, insult, identity_hate.

Sample data label is visualized in Table 5.

TABLE 5: DATA LABEL VISUALIZATION

	Toxic	Severe_Toxic	Obscene	Threat	Insult	Identity_Hate
0	0	0	0	0	0	0
1	1	0	0	0	0	0
2	0	0	0	0	0	0
3	1	0	0	0	1	0
4	0	0	0	0	0	0

Some sample examples from our dataset are shown below,

"আমার দখো একজন নরিহংকার প্লয়োর করকিটে থকেে বদিায় নতিে চলছেনে... যনিি ছয় খলেওে হাসনে আবার উইকটে পাইলওে হাসনে,,,, কমবশেি সবারই একজন খুব ভাল প্লয়োর || আজ তার আর্ন্তজাতকি ...# ODI এর শষে ম্যাচ ভাল থাকবনে লজিন্ডে।"

Example 1: 1st Sample Comment of Dataset

র'দশে উন্নয়নরে জয়োরে ভাসতছে"

Example 2: 2nd Sample Comment from Dataset

Example 3
।"কুত্তার বাচ্চা "

Example 3: 3rd Sample from Dataset

A. Dataset Creation Methodology

From Example 1, we can see that this comment is a non-toxic comment. Not only non-toxic, it is a clean comment. So, we put zeros to all labels.

Example 2 is a bit toxic, so we put one to toxic label and zeros to others.

But, the third one, Example 3 violating toxic, severe toxic, obscene rules at the same time. So, we put one to those labels and zero to others.

By following this technique, we got our own dataset to work on.

The comments in dataset based on length is shown in Figure 4.

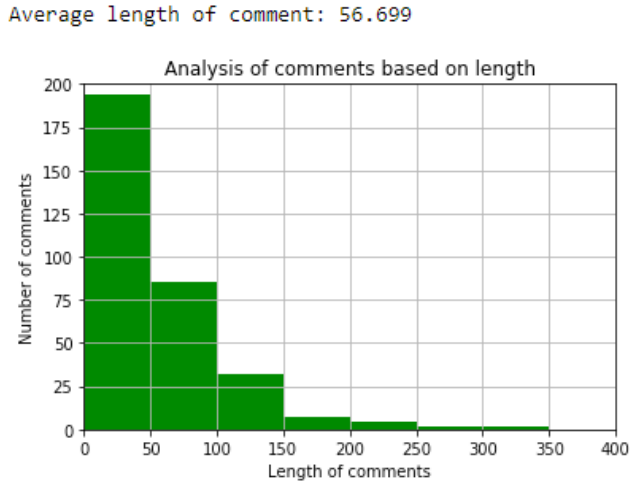

Fig. 4: Length based Comment Numbers

Histogram of comments based on length is shown Figure 5.

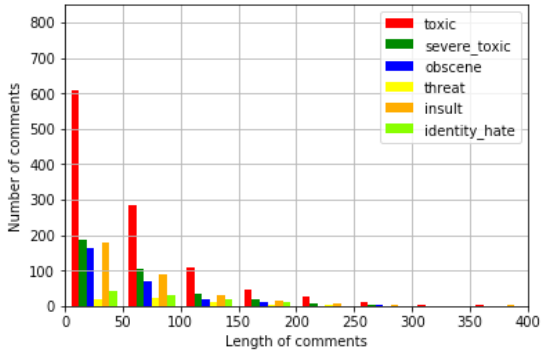

Fig. 5: Histogram of Comments (length based)

IV. Work Procedure

In this part we will talk about the methods we used in our work flow.

Binary Relevance is a problem transformation technique that allows single-label classifier to perform on a multi-label problem. Such one has a n-labeled problem. BR transforms it into n single labeled problem and give opportunity to perform well known single labeled classifier on them.

A. Binary Relevance Method with MultinomialNB Classifier

MultinomialNB is one of the three types of Naïve Bayes classifier. It is perfect specially for classifying with discrete features. It works well with integer value.

The basic of MultinomialNB is equation 1:

$$P(t|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct}} \quad (1)$$

MultinomialNB takes tokens of a document given. It concerns about frequency of term or token.

However, it also works with fractional value, like from tf-idef. We did BR MukltinomialNB from scratch. Accuracy of our BR MultinomialNB classifier is 52.30%.

B. Binary Relevance with SVM Classifier

After converting our multi-labeled problem into multiple single labeled problem, we fed them to Support Vector Machine. It is a supervised learning method which is constraint followed and conduct optimized. Means it takes into account the separate hyperplanes and pick up line that maximize the separating margin. Hyperplane is always n-1 dimensional if the model is n dimensional. Any hyperplane can be written mathematically like equation 2.

$$\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \ldots + \beta_n * x_n = 0 \quad (2)$$

We implemented it from sklearn. We got accuracy of 30.76%. This doesn't satisfy us.

C. BR Method with MultinomialNB (from scikit-learn)

Again, we did the same MultinomialNB, but this time from sklearn. Accuracy we got is 52.30%, that is same as before.

D. BR Method with GausseanNB

Gaussian method is another one of the three Naïve Bayes based classifier. It implements the Gaussian Naïve Bayes algorithm for classification. It depends on the Bayes Theorem.

The bayes theorem is shown in equation 3.

$$P(X|Y) = \frac{\mathrm{P(Y|X)\ (P(X))}}{\mathrm{P(Y)}} \quad (3)$$

Where X is the hypothesis and Y is the evidence.

We got accuracy of 49.23% from BR Method with GaussianNB.

E. Classifier Chain with MultinomialNB Classifier

Classifier chain is another problem transformation method. We used MultinomialNB on single-labeled problem provided by classifier chain. Accuracy is 52.30%.

F. Label Powerset with MultinomialNB Classifier

Label powerset is a problem transformation approach that transforms a multi-label to a multi-class problem. After that we implement MultinomialNB on that. 58.46% is the highest accuracy we got implementing Label Powerset with MultinomialNB classifier.

G. MLkNN with k =2

It is a kNN classification method adapted for multi-label classification. MLkNN builds uses k-NN, finds nearest examples to a test class and uses Bayesian inference to select assigned labels. We got 58.46% for accuracy.

H. BP-MLL Neural Networks

This is Back propagation Multi Label Learning. We used keras for this purpose. There are two layers of filter 4 and 8 in this Neural Network. Accuracy we got using BP-MLL is 60.00%, highest among others.

V. Results

Implementing different classifiers, we can see that the accuracy of BP-MLL Neural network is the highest one. This can be seen from the Figure 6, which is showing the accuracy of different classifier in bar chart.

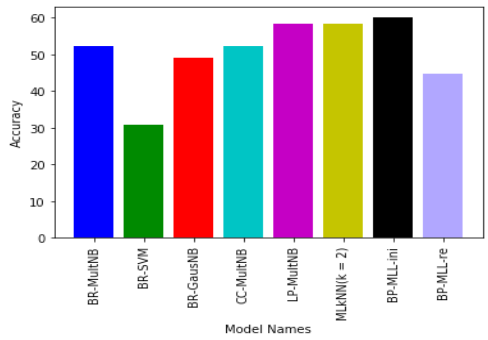

Fig. 6: Accuracy Bar Chart of Different Classifier

From Figure 7, we can see that the hamming loss of BP-MLL Neural Network is low. That means, it gives the most accurate prediction among other models.

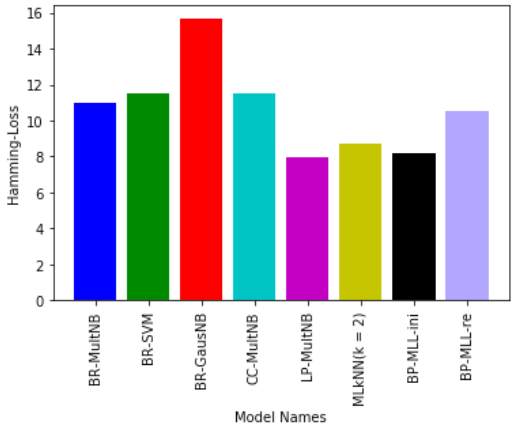

Fig. 7: Hamming-loss Bar Chart of Classifiers

Again, from Figure 8, the Log-Loss is low for BP-MLL Neural Network. That means, it is most related with the actual results. The diversity from actual results is low here.

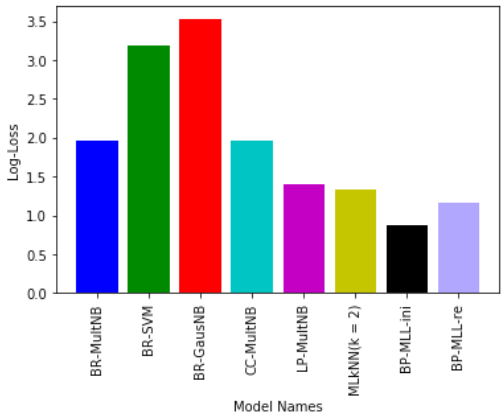

Fig. 8: Log-loss Bar Chart of Different Classifiers

So, based on the whole discussion, it is clear that BP-MLL Neural Network works well in Bangla Toxic Comment Classification problem. Any system, implemented by BP-MLL Neural Network will give better result. So, Bangla toxic comment in various community can be detected by implementing BP-MLL Neural Network and a system can be developed which will omit the detected toxic comment to make the in-community relationship better.

VI. Future Scope

The accuracy can be improved in future. The stemming could make a dramatic change in the accuracy here. Important thing is, as this is the first attempt to such work with Bangla Language, there is a dataset already created by us for future research. So, people who are interested in work with Bangla NLP, can use the dataset.. For furthermore work, this dataset can be a part of bigger thing too.

VII. Conclusion

In this work we showed different approaches to classify Bangla toxic comments. Among them some worked fine. The Neural Network worked best among all of classifier. In Bangla language, the NLP related work is very few. To prevent the hate spreading in online Bangla community, this classifier can perform a vital role. Any system implementing the BP-MLL Neural Network, can perform good in detecting toxic comments in Bangla online conversation. People who are toxic in comments, can be warned even at extreme, can be banned from community. As a result, the community environment with each another will be clean.

References

[1] Challenges for Toxic Comment Classification: An In-Depth Error Analysis by Betty van Aken, Julian Risch, Ralf Krestel, Alexander Löser, Beuth University of Applied Sciences Berlin, Germany, 2018

[2] Deep Learning for User Comment Moderation John Pavlopoulos, Prodromos MAlakasiotis, Ion Androutsopoulos 2017

[3] The Role of Conversation Context for Sarcasm Detection in Online Interactions Debanjan Ghosh, Alexander Richard Fabbri, Smaranda Muresan, USA, 2017

[4] Convolutional Neural Networks for Toxic Comment Classification by Spiros V. Georgakopoulos, Sotiris K. Tasoulis, Aristidis G. Vrahatis, Vassilis P. Plagianakos, University of Thessaly Lamia, Greece, 2018

[5] Multi-label Text Classification Based on Sequence Model by Wenshi Chen , Xinhui Liu , Dongyu Guo2 , and Mingyu Lu, Dalian Maritime University, Dalian, China, 2019

[6] Text Representation in Multi-label Classification: Two New Input Representations by Rodrigo Alfaro and Hector Allende, Chile, 2011

[7] Sentiment Analysis in Twitter using Machine Learning Techniques by Neethu M S, Rajasree R, College of Engineering Trivandrum, 695016, India, 2013

[8] @misc{kaggle_2017, title={Toxic Comment Classification Challenge}, url={https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge}, journal={Kaggle}, year={2017}, month={Dec}}

[9] @misc{baghel_2018, title={Toxic Comment Classification}, url={https://medium.com/@nupurbaghel/toxic-comment-classification-f6e075c3487a}, journal={Medium}, publisher={Medium}, author={Baghel, Nupur}, year={2018}, month={Jun}}

[10] @misc{stopwords-iso, title={stopwords-iso/stopwords-bn}, url={https://github.com/stopwords-iso/stopwords-bn/blob/master/package.json}, journal={GitHub}, author={Stopwords-Iso}}