Toxicity Detection on Bengali Social Media Comments using Supervised Models

Preprint · November 2019

DOI: 10.13140/RG.2.2.22214.01608

2 authors:

Nayan Banik
Comilla University
8 PUBLICATIONS **17** CITATIONS
SEE PROFILE

Md. Hasan Hafizur Rahman
Comilla University
11 PUBLICATIONS **26** CITATIONS
SEE PROFILE

Toxicity Detection on Bengali Social Media Comments using Supervised Models

Nayan Banik
Department of Computer Science & Engineering
Green University of Bangladesh
Dhaka - 1207, Bangladesh
Email: cse.nayan@gmail.com

Md. Hasan Hafizur Rahman
Department of Computer Science & Engineering
Comilla University
Comilla - 3506, Bangladesh
Email: hhr@cou.ac.bd

***Abstract*—Social media playing an indispensable role in our daily life providing a public platform to share opinions including threats, spam and vulgar words often referred to as toxic comments. This type of expression depicts the anti-social behavior of the commentators which may hamper the online atmosphere. Filtering such toxic comments by handcrafting rules is cumbersome because they are unstructured and often include misspelled obscene words. Automated machine learning-based models to classify such toxic comments constitute a part of Sentiment Analysis and they are extensively used for the English language; showing promising results than statistical models. Though Bengali is a widely spoken language around the globe, little research works have been done to detect toxic comments in this language. Hence in this scholarly manuscript, we provide a comparative analysis of five supervised learning models (Naive Bayes, Support Vector Machines, Logistic Regression, Convolutional Neural Network, and Long Short Term Memory) to detect toxic Bengali comments from an annotated publicly available dataset. As our research finding, we demonstrate that both the deep learning-based models have outperformed other classifiers by 10% margin where Convolutional Neural Network achieved the highest accuracy of 95.30%.**

Keywords*—*Text Classification*, *Machine Learning*, *Natural Language Processing

I. Introduction

Social media provides a place for common people to share their opinions, feelings, and reactions on diverse topics. This public platform becomes a day-to-day habit for minors to age-old people who spend a myriad amount of time to socialize with their fellow peers. But often this online atmosphere creates disputable topics ranging from political propaganda, religious insanity and random hoax. Divided parties on such phenomena exchange hate comments including threats, vulgar words to attack each other personally. Such obscene words referred to toxic comments are harmful to safe user experience on the platform and hence need to be filtered out [1]. Considering the social media as an information hub, excluding such toxic comments for analysis is an open challenge to the human as well as automated comment filter.

According to [2], Bengali stands sixth as the most spoken language in the world considering 228 million native speakers and this count is increasing rapidly due to its significance in demographic and political purposes. Moreover, Bengali as a south Asian language is the national language of Bangladesh and it is also used partially in many regions of India. The increasing number of Bengali social media users post numerous statuses, comments, graphics, etc and they are instantly available for others to react. Generally, this often results in text with toxic comments and needs to be filtered out.

Manual toxic comment filtration with offensive word lists and complex rules is not an easy task for inflectional language like Bengali. The unstructured nature of social media text also makes it an arduous task with a poor acceptability score. Handwritten rules using manual linguistic features were hard previously but cheap computing resources have changed this scenario with automated systems [3]. Extracting notable features from textual data using computational mechanisms is a part of Natural Language Processing (NLP) which requires annotated corpus to convey information relating to many applications including toxic comment detection.

Researchers have tried statistical machine learning models to detect sentence toxicity. But such models require frequency-based feature engineering or probabilistic phenomena and hence do not scale well with unstructured social media comments[4]. To tackle these limitations, deep learning-based models have proven their effectiveness by capturing low-level features and combining them into layer-wise abstractions[5]. It is shown in different works that such models have outperformed other supervised machine learning models by a great margin when applied to English text. In this scholarly work, we have investigated that claim for the Bengali toxic comment detection task comparing five classifiers named Naive Bayes (NB), Support Vector Machines (SVM), Logistic Regression (LR), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), especially Long Short Term Memory ($LSTM$). Our experimental work demonstrates that deep learning based models have better accuracy on the noisy nature of Bengali toxic comments.

Further organization of this paper starts with a brief overview of related works with their notable achievements and limitations in **Section II**. **Section III** describes our applied models and architectures. In **Section IV**, we provide the implementation details of our proposed models along with the comparative analysis and associated metrics. The paper concludes with the conclusion and future references in **Section V**.

978-1-7281-6309-3/19/$31.00

II. Related Works

Toxic comment detection becomes a research topic due to its variational nature from the linguistic perspectives as well as the commentator's overview. Researchers in [6] described the inherent challenges in detecting toxic comments and they had proposed that the ensemble method of combining several classifiers works well when the comments have the variational vocabulary. The inherent unaddressed complexities of toxic comment detection and possible solutions to overcome them systematically is proposed in [7]. Considering the effect of toxic tweets, the researchers in [8] experimented with CNN and showed that toxicity can be revealed over time and the inherent knowledge can be extracted. Relating to that, researchers in [9] experimented with Youtube comments to detect toxicity on specific channels contents. They applied Latent Dirichlet Allocation to find out the topics on which the toxic comments were posted. To detect abusive text in Bangla Facebook comments from different pages, authors in [10] proposed several classifiers and claimed that SVM with linear kernel performs better when Term Frequency - Inverse Documnet Frequency ($TF - IDF$) vectorizer is used. Relating to that, authors in [11] proposed a root level algorithm to detect abusive comments from specific Facebook pages with a manually collected dataset. The work lacks comparisons with traditional classifiers and the small dataset is also a limiting factor of this research. Researchers in [12], applied six classifiers to detect abusive comments from a manually collected dataset. Here the authors collected data from Youtube, Facebook and Prothom-Alo pages and pre-processed them. Their experimental results showed that deep learning-based model outperforms other models on a great margin but the use of small dataset having only 4700 comments is a limiting factor of this work. Comprehensive study on Bengali Sentiment Analysis (SA) is provided in [13], where the authors have demonstrated text-based research works in this field with their approaches, dataset, performances and drawbacks.

III. Methodology

The proposed models to find toxic comments is described in this section. Initially, the acquired annotated dataset statistics are given. To clean the noisy unstructured data, preprocessing is applied. Then we prepare our data compatible to be fed into the model which structure is provided in the last part of this section.

A. Data Acquisition

Almost every supervised classification task like toxicity detection, a labeled dataset must be needed to train as well as to test the performance of the classifier. Manual preparation of a dataset requires resources including human effort, knowledge and a lot of time. To minimize that exhaustive search for a manual dataset, we have experimented with a human-annotated public domain dataset for our work available at Github[14]. The dataset contains five tags for each Bengali social media comment named toxic, threat, obscene, insult, racism. But the number of tagged comments for all columns except toxic is considerably small and hence we have only experimented with the toxic columns as our label for classification. The value 0 indicates non-toxic comments and 1 signifies toxic. The detailed statistics for the dataset is given in Table I.

TABLE I
Dataset Statistics

Total Comments	10219
Toxic Comments	4255
Non-Toxic Comments	5964
Longest Comment length (in words)	528
Smallest Comment Length (in words)	1
Average Comment Length (in words)	12
Unique Words	23600

B. Preprocessing

The nature of social media comments is usually not structured or follow any specific standards. For abusive and slang comments, these scenarios are more complicated as the commentators express their waves of anger and depressions through intentional misspellings and repetitive use of adjacent characters. Though there are some limitations to the comment size, they are also platform dependent on which the comments are posted. Some other major problems on social media comments include spelling errors, useless punctuation, emojis, and random duplication. In any text processing manual approach, several preprocessing tasks are performed to clean the noisy data. But considering the nature of toxic comments, we only perform punctuation and emoticons removal before doing the tokenization as they do not convey any meaning for toxicity. Sometimes the long version of words or intentional use of repetition may convey the attitude of the commentators and hence we do not perform any stemming or lemmatization on our dataset. Moreover, from the dataset statistics, the unique word count is not so large. So our feature space is also within the scope for the classifier to check their performances in time.

C. Representation of Word Embeddings

For any text-based model depending on a deep neural network requires each word in a corpus to be represented as a vector of fixed length which is known as word embeddings. $Word2Vec$ is such an algorithm that can be trained on a corpus to extract the embeddings by learning the similarities of word meanings[15]. In our work, we first build a D size vocabulary from our dataset. Then each sentence in the dataset is transformed into D length one-hot encoded vector. These vectors are then fed into a neural network of a single hidden layer containing m nodes. The hidden layer has a linear activation function and the output layer has softmax activation function containing D nodes. Here each node represents the likelihood of putting that word in the sentence. Upon training this neural network, we build a $D \times m$ matrix; where D is the length of input sentence and m is the number of hidden layer nodes.

D. Model Architecture

Our approach to toxic comment detection utilizes two popular deep learning based architectures named Long Short Term Memory ($LSTM$) and Convolutional Neural Network (CNN). The architectures of the models are shown in Figure 1 and 2.For the performance analysis, we have also implemented baseline models with Naive Bayes (NB), Support Vector Machines (SVM) and Logistic Regression (LR).

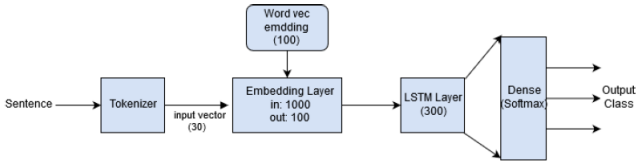

Fig. 1. LSTM Architecture

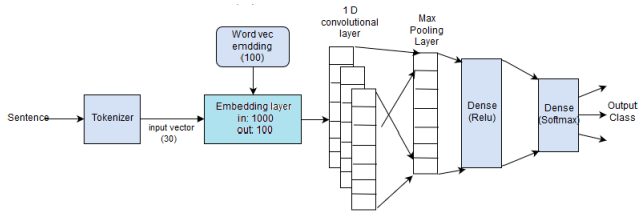

Fig. 2. CNN Architecture

1) LSTM: The preprocessed sentences in our dataset are first passed to the tokenizer to calculate the one-hot encoded vector of length 50 because the social media comments are short in nature. To limit the computational time for training the network and to simplify the model, we consider the 1000 most occurring words in the obtained vocabulary after tokenization. We also set a boundary on length 30 and hence comments with longer than 30 words are truncated and the shorter are padded with zeroes. These modified vectors are then fed to the embedding layers and the weights are adjusted. We set the output layer dimension as 100 so that each word is represented with 100 dimensions. Then the 50 words are fed to the LSTM layer before adding the dense layer with softmax activation function.

2) CNN: In our CNN architecture, we have used 1D convolutional layer right after the embedding layer as described in the previous section for $LSTM$. The convolutional layer is set to use 100 filters as the features are limited. In the next layer, we have used global max-pooling layer which extracts the max value from them. Here the dimension of the output vector is the same as the applied filters. Before the last dense layer having softmax activation, we pass our data through a dense layer with ReLu activation function.

3) Baseline Models: To compare the performances of our deep learning-based models, we have applied three classifiers named NB, SVM, LR as our baseline models. The baselines are used with their standard default parameters for the simplicity of evaluation. Moreover, we do not apply any cross-validation to improve baselines accuracies. The dataset is trained on training data and test on the testing data. No separate validation data is used for parameter tuning. The specifics of the baselines are set the default to the libraries used for the experiment.

IV. Experimental Evaluation

The performance of our proposed models for toxicity detection is discussed in this section. We also compare the performance with the three baselines.

A. Experimental Libraries

In order to train the network and test the performance, we have used several freely available python based machine learning resources. We have used *TensorFlow*[1]; an open source library for numerical computation and large-scale machine learning including deep learning models and algorithms. *Theano*[2] is another python library that allows users to compute and optimize complex mathematical computations relating high-dimensional arrays. *Scikit-Learn*[3] is also a python library which has clean, uniform, and streamlined API and provides solid implementations of a range of machine learning algorithms.

B. Results Analysis

From several evaluation metrics, we have considered accuracy for our binary toxicity detection task. The performance of baselines and our proposed deep learning-based models are shown in Table II and the corresponding loss-accuracy curves are visualized in Fig. 3 and Fig. 4.

TABLE II
Performance Statistics

Classifier	**Accuracy**
Naive Bayes	81.80
Support Vector Machines	84.73
Logistic Regression	85.22
LSTM	94.13
CNN	**95.30**

Here from the Table II, we can see that both of our deep learning-based models have outperformed all the three baselines by roughly 10% margin and the best accuracy achieved in CNN classifier as 95.30%. The baselines use word frequency as the mechanism to select features for the classification. This type of feature engineering results in poor performances as all the baselines utilize $TF-IDF$ as their feature extractor. On the other hand, the neural-network-based models utilize word embeddings as their feature extractor. The word embeddings are trained weights which play a crucial role to learn the low-level abstractions and gradually improve itself through gradient-based backpropagation technique.

From the accuracy and loss curves of $LSTM$ in Fig. 3 we can see though the training accuracy reached on the maximum

[1]https://www.tensorflow.org/
[2]http://deeplearning.net/software/theano/
[3]http://scikit-learn.org/stable/

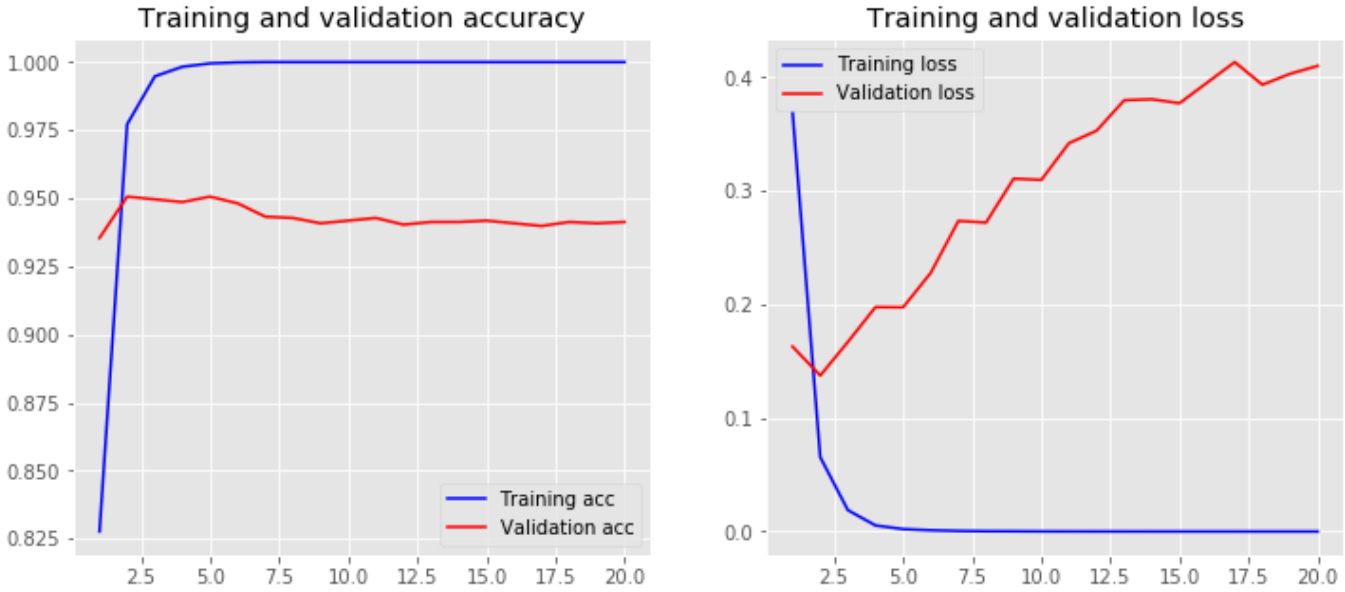

Fig. 3. Accuracy and Loss trends of LSTM Model

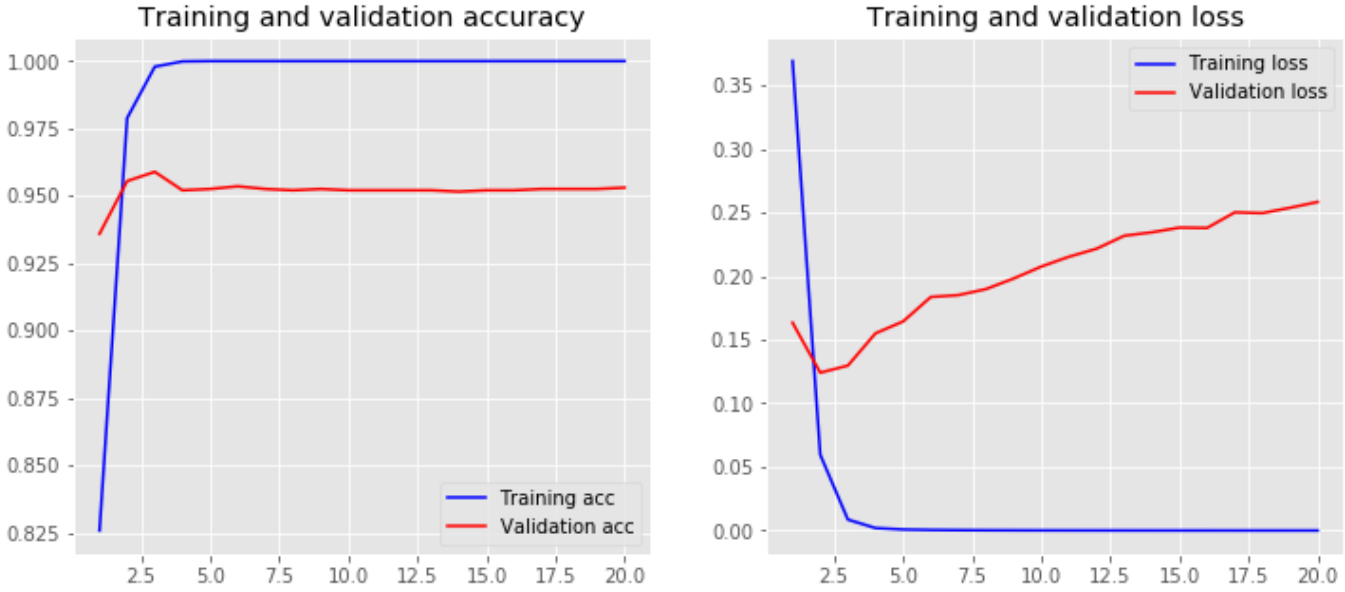

Fig. 4. Accuracy and Loss trends of CNN Model

level early in the training phase but the validation accuracy came to the stable state a little late as the netwok converges. The loss curve, on the other hand, depicts the overfitting on the data and it gradually increases for validation data on every epoch. The reason of this nature is quite straightforward as the toxic comments are having many out of vocabulary words unseen during the training phase. As a result, we can claim that the accuracy of $LSTM$ network for this specific dataset is losing confidence in every epoch and the model is degrading.

Contrary to $LSTM$ network, accuracy and loss curves of CNN in Fig. 3 showes steady growth in accuracy as well as better generalizations in loss trends. From our experiment, we have seen that the optimal use of Global Average Pooling Layer instead of Max Pooling Layer results in better convergence during the training phase. Though the loss curve for validation data is slowly increasing, we can claim that our CNN model has better confidence than the $LSTM$ model and hence the performance justifies for a marginal increase with baselines.

V. Conclusion

Comment filtration is an important task for any promising social media to provide safe atmosphere to its users.

Filtering toxic comments in Bengali is currently an ongoing research topic as the number of Bengali speaking users in social media is increasing. Previously, traditional machine learning approaches were used to detect toxicity but since their performance are not scaling well with the large dataset, deep learning based models emerged. In this scholary work, we have demonstrated Bengali toxic comment detection system using two deep learning models named Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM). Both of our models outperformed three baselines (Naive Bayes, Support Vector Machines and Logistic Regression) by 10% margin while CNN achieved the highest accuracy of 95.30%.

ACKNOWLEDGEMENT

This work has been financially supported by Green University of Bangladesh Research Fund.

REFERENCES

[1] T. Cooper, C. Stavros, and A. R. Dobele, "Domains of influence: exploring negative sentiment in social media," *Journal of Product & Brand Management*, 2019.

[2] "What are the 10 most spoken languages in the world? — babbel," https://www.babbel.com/en/magazine/the-10-most-spoken-languages-in-the-world/, (Accessed on 09/30/2019).

[3] A. Goyal, V. Gupta, and M. Kumar, "Recent named entity recognition and classification techniques: A systematic review," *Computer Science Review*, vol. 29, pp. 21–43, 2018.

[4] N. Banik and M. H. H. Rahman, "Evaluation of naïve bayes and support vector machines on bangla textual movie reviews," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*. IEEE, 2018, pp. 1–6.

[5] ——, "Gru based named entity recognition system for bangla online newspapers," in *2018 International Conference on Innovation in Engineering and Technology (ICIET)*. IEEE, 2018, pp. 1–6.

[6] B. van Aken, J. Risch, R. Krestel, and A. Löser, "Challenges for toxic comment classification: An in-depth error analysis," in *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, 2018, pp. 33–42.

[7] B. Vidgen, A. Harris, D. Nguyen, R. Tromble, S. Hale, and H. Margetts, "Challenges and frontiers in abusive content detection." Association for Computational Linguistics, 2019.

[8] S. V. Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis, and V. P. Plagianakos, "Convolutional neural networks for twitter text toxicity analysis," in *INNS Big Data and Deep Learning conference*. Springer, 2019, pp. 370–379.

[9] A. Obadimu, E. Mead, M. N. Hussain, and N. Agarwal, "Identifying toxicity within youtube video comment," in *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*. Springer, 2019, pp. 214–223.

[10] S. C. Eshan and M. S. Hasan, "An application of machine learning to detect abusive bengali text," in *2017 20th International Conference of Computer and Information Technology (ICCIT)*. IEEE, 2017, pp. 1–6.

[11] M. G. Hussain, T. Al Mahmud, and W. Akthar, "An approach to detect abusive bangla text," in *2018 International Conference on Innovation in Engineering and Technology (ICIET)*. IEEE, 2018, pp. 1–5.

[12] E. A. Emon, S. Rahman, J. Banarjee, A. K. Das, and T. Mittra, "A deep learning approach to detect abusive bengali text," in *2019 7th International Conference on Smart Computing Communications (ICSCC)*, June 2019, pp. 1–5.

[13] N. Banik, M. H. H. Rahman, S. Chakraborty, H. Seddiqui, and M. A. Azim, "Survey on text-based sentiment analysis of bengali language."

[14] "Bangla-abusive-comment-dataset," https://github.com/aimansnigdha/Bangla-Abusive-Comment-Dataset, (Accessed on 09/30/2019).

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.