Sentiment Analysis for Bangla Sentences using Convolutional Neural Network

Md. Habibul Alam
Semantics Lab.
&
Department of Computer Science & Engineering
Begum Rokeya University, Rangpur, Bangladesh.
Email: habibul@semanticslab.net

Md-Mizanur Rahoman
Semantics Lab.
&
Department of Computer Science & Engineering
Begum Rokeya University, Rangpur, Bangladesh.
Email: mizan@semanticslab.net

Md. Abul Kalam Azad
Department of Computer Science & Engineering
Begum Rokeya University, Rangpur, Bangladesh.
Email: mizan@semanticslab.net

***Abstract*—Sentiment analysis, also known as opinion mining or emotion analysis, is a process to determine emotional reaction of people towards an interaction or an event. An opinion may be positive, negative or neutral depends on individuals judgment or evaluation towards a topic or an event. Usually sentiments can be varied in cultures and languages. On sentiment analysis, an extensive amount of research works can be seen for well-resourced languages like English, Japanese etc. However, such works are relatively less observed for low-resourced language like Bangla. In this paper, we propose a framework that analyzes sentiments from texts written in Bangla. In our proposal, we use Bangla comments and generate a classification model. The model is generated by a neural network variance called Convolutional Neural Network. The classifier model obtains a classification accuracy of 99.87%, which is 6.87% better than the available state-of-the art Bangla sentiment classifier.**

***Index Terms*—Sentiment Analysis, Bangla, Convolutional Neural Network**

1. Introduction

Sentiment can be defined as the opinion of people towards a specific interaction [1]. Sentiment Analysis is the process of understanding people's opinion. Well resourced languages like - English, Japanese [2], [3] enjoy great progress in Sentiment Analysis(SA) while low resourced languages like Bangla, Swahili etc. stay behind due to lack of resources. For example, resources like annotated datasets, language parser etc. are available in well resourced languages while low resourced languages lack in such resources.
On the other hand, Bangla being a widely spoken language in the world, still it belongs to low resourced language. More than 200 million people use it as a native language in Bangladesh and in some states of India. Recently we see availability of huge digital contents such as review comments, facebook status written in Bangla which give an opportunity to analyze sentiment in Bangla. As now more and more Bangla speaking people are getting involved in online activities thus such analysis is very much needed in analyzing market trends and consumer interests, in movie reviews etc. However, analyzing this vast amount of data and figuring out individuals interest in a manual way is tedious and in some cases simply intractable. Thus in these scenarios sentiment analysis can play a vital role to figure out people interests and opinion.
There are some works to analyze sentiment in Bangla language are being seen. However, they have some common issues. For example, in such approaches we can have a high accuracy rate if some specific feature appeared in the sentences. Moreover, their proposed methodology is not suitable for analyzing sentiment from complex sentences.
In this work, we are motivated to handle the above mentioned problems for Bangla sentiment analysis. We proposed a sentiment analysis framework that does not need any specific dependency. For example, presence of any specific feature in the sentences. Currently, we classify two categories of sentiment i.e. Positive and Negative.
In recent times, we observed that Neural Network shows an excellent performance in classification tasks. Our proposed solution uses Convolutional Neural Network which is a variant of Artificial Neural Network. By using the proposed framework we obtained an accuracy of 99.87% which is the highest accuracy in analyzing Bangla sentiment by our observation.
The remainder of this paper is organized as follows. Section 2 describes Related Work in SA field, Section 3 describes Background Studies. In Section 4 we describe the proposed framework in details. In Section 5 we describe experiments implementing the proposal and discuss our results. Finally, Section 6 concludes our study.

2. Related Work

With the advancement of social media such as Facebook, twitter which come with a huge amount of public opinion or sentiment mostly in the form of text (status, comments), which give a boost in Sentiment Analysis. We went through several research works to classify sentiment in well-resourced language like English as well as relatively low-resourced language like Bangla to know various developments on sentiment analysis. Below we tried to categorize these works based on languages.

2.1. Sentiment Analysis in English Language

As we have observed, the term Sentiment Analysis appeared for the first time in Nasukawa and Yi [4] work. In their work, they tried to analyze sentiment by properly identifying the semantic relationships between the sentiment expressions and the subject. By applying semantic analysis with a syntactic parser and sentiment lexicon, they obtained an accuracy between 75-95%, depending on the data. However, their proposed approach was basically word based which is little bit obsolete.
Another paper by Prabowo and Thelwall where they combined rule-based classification, supervised learning and machine learning into a new combined method [5]. Then they tested the method on movie reviews, product reviews and MySpace comments. However, there can be thousand of rules and they can start to interact in complex way which will be difficult to maintain.

2.2. Sentiment Analysis in Bangla Language

Due to the lack of standard resources, works on analyzing sentiment have not gained that much success in low-resourced language Bangla. Yet some researchers tried to classify sentiments from Bangla text corpus. In order to gain update on those work, we went through some of them. For example, Chowdhury and Chowdhury did an work to analyze sentiment from bangla tweets [6]. They used Support Vector Machines (SVM) and Maximum Entropy (MaxEnt) on a combination of various features set and compared the performance of these two machine learning algorithms. As a consequence, they obtained a high accuracy of 93% using SVM against a feature Unigram+Emoticons (Emotion Symbols). However, their work have some dependency issues. For example, for only some specific features (like emotion symbols) their system performs well while in real life it is not necessary that these features appear in every sentences. Moreover, their proposed system is not suitable for analyzing complex sentences.

On the other hand, another work was done by Hassan et.al., on not just standardized Bangla, but Banglish (Bangla words mixed with English words) and Romanized Bangla using Deep Recurrent Model more specifically Long Short Term Memory [7]. However, their proposed system's accuracy is not good enough (78%).

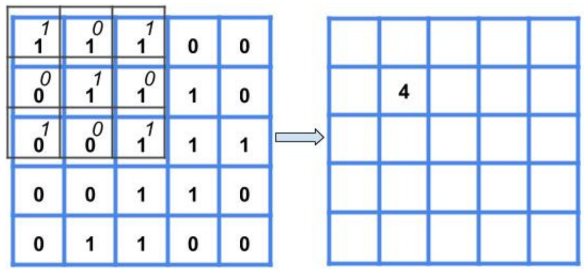


Figure 1. Convolution with 3×3 filter

3. Background Studies

Before describing our proposed framework, we will go through some background studies which will be helpful to readers to understand our proposed methodology. We will focus on a variant of Artificial Neural Network [8] that is Convolutional Neural Network (CNN).

CNN is primarily used for computer vision. It gives a breakthrough in image classification. Observing CNNs performance in classification tasks, now a days it is also applied in the field of Natural Language Processing [9] and Signal Processing.
In machine learning, we can think of convolution as a window function sliding over a matrix [10]. Filters with different shapes are applied to convolutional layer for a specific number of times which generate different features. These features ultimately used for classification. Figure-1 can be considered as the graphical representation of convolution in a matrix. Thus we can say that a CNN is basically several layers of convolutions locally connected. That is the output of one layer become the input of another [11]. After that an activation function may be applied to the output of each layer. The output of a convolutional layer for a specific filter size can be calculated by the formula

$$(W - F + 2P)/S + 1$$

Here W is input volume size, F is filter size, S is stride that means how much the filter shifts each time, P is padding.
On the other hand, pooling is a technique that is used to reduce the output dimensionality keeping the most salient feature. Sometimes these features are enough for classification. Pooling may be applied to the output of convolution layer. The output from pooling layer may then be fed to a fully connected layer which implements regularization technique to prevent the network from over-fitting and loss-function to calculate loss. A loss function calculates the deviation of the obtained label from the original label. To minimize the loss CNN adjust weights by optimization technique in back-propagation. Mathematically the optimization process can be expressed as

$$W = W_i - \eta \frac{dL}{dW}$$

Here W defines weight, W_i defines initial weight and learning rate η defines how bigger steps to be taken to

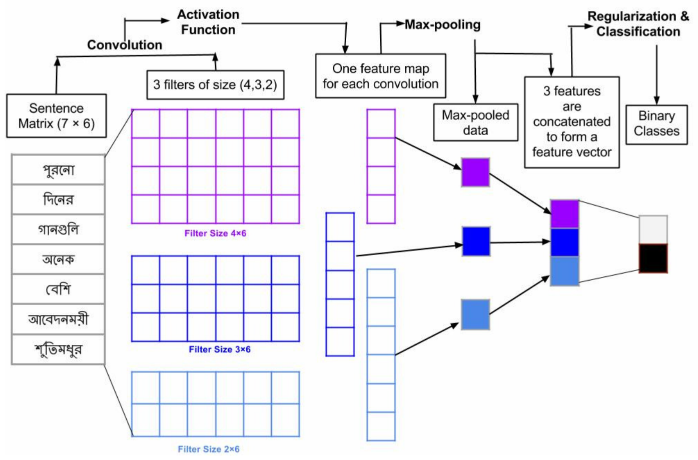


Figure 2. Convolutional Neural Network for NLP

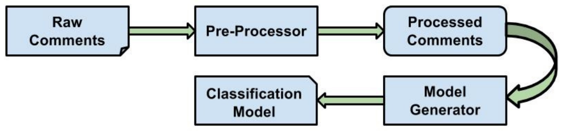


Figure 3. Workflow of proposed system

update weights.

Now, when CNN applied to Natural Language Processing (NLP), the main difference with image is that the sentences will be represented as a 2-D matrix where each row will represent a word. The convolution process will be then applied on the matrix. Here the filters will slide over full rows of the matrix that is covering full word [12]. After that, a pooling technique may be applied on the generated feature maps with an activation function. Pooling will result a feature vector for individual feature map. Then all the feature vectors will be concatenated into one big feature vector. At the end, the big feature vector will be fed into a fully-connected layer for regularization and classification. Figure-2 illustrates the implementation of CNN for NLP [12].

4. Proposed Framework

Our proposed system consists of two processes, i) pre-processor and ii) model generator. Figure-3 shows workflow of the proposed system. Here, the rectangles denote processors and the remaining blocks either input or output. Below we will describe each process in details.

4.1. Pre-Processor

In pre-processor, we take raw comments and produces training data for next process. To generate the training data, pre-processor uses three different sub-processes, i) Data Cleanser ii) Vocabulary Creator iii) Matrix Generator. Figure-4 illustrates the Pre-Processor. Below we will

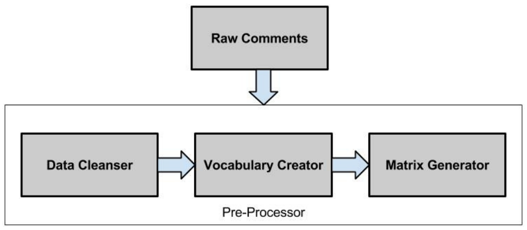


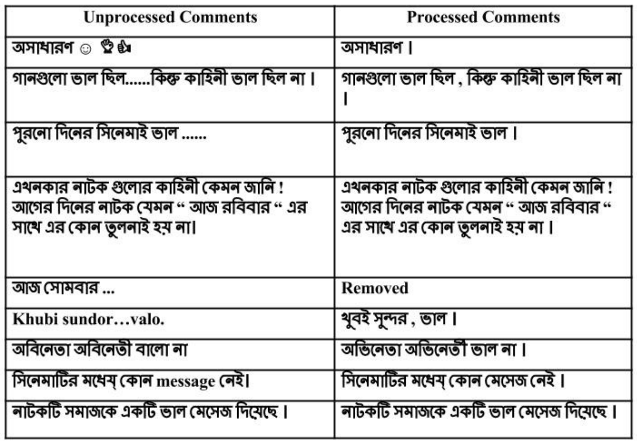
Figure 4. Sub-processes of Pre-Processor

describe each of these sub-processes one after another to show how our data will be prepared for training.

4.1.1. Data Cleanser. Data Cleanser process filters comments by removing unnecessary things. The primary collection of comments were noisy and contained URLs, emoticons, dots, slangs and many unnecessary things. Therefore, we removed the URLs as they do not express any sentiment. Moreover, emotion symbols were removed and sequence of dots between two words in a sentence which do not represent any sentiment were replaced by comma (,). In addition, we appended Dari (|) which is equivalent of English Full Stop (.) at the end of every sentences. Many comments contain multiple sentences, thus we placed a comment in one line to indicate them as the parts of a single comment. Furthermore, words between double quotation were treated as a single word. However, there were comments in raw file which express no sentiment, which can be defined as neutral comments. As in this work we only deal with negative and positive sentiments, we removed those comments. Further more, there were comments written in Banglish (Bangla written in English). We wrote them down in proper Bangla words. On the other hand, as almost all the punctuation marks (i.e. ?,!,/) and their role are same in English and Bangla Language, we do not remove them from Bangla sentences.
As many foreign words are regularly used in Bangla Language, therefore we do not translate those words which are very common in Bangla Language. In addition, we removed slangs and corrected the misspelled Bangla words. Figure-5 illustrates the comments before and after processing where the first column denotes unprocessed comments and the second column denoted processed comments.

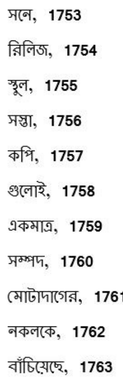
4.1.2. Vocabulary Creator. As our Model Generator can not take string as input, we use Vocabulary Creator to generate vocabulary from Bangla sentences. Vocabulary Creator maps the words of Bangla comments from the dataset with numbers. Matrix Generator later uses these vector representations to generate input matrices. Figure-6 illustrates a portion of the vocabulary created by the Vocabulary Creator. In this figure the words and their corresponding index numbers are separated by comma.

4.1.3. Matrix Generator. Matrix Generator generates input matrices which are then fed to Model Generator process. At first matrix generator creates low dimensional vector



Unprocessed Comments	Processed Comments
অসাধারণ ☺ ✋👍	অসাধারণ।
গানগুলো ভাল ছিল......কিন্তু কাহিনী ভাল ছিল না।	গানগুলো ভাল ছিল , কিন্তু কাহিনী ভাল ছিল না ।
পুরনো দিনের সিনেমাই ভাল	পুরনো দিনের সিনেমাই ভাল ।
এখনকার নাটক গুলোর কাহিনী কেমন জানি ! আগের দিনের নাটক যেমন “ আজ রবিবার “ এর সাথে এর কোন তুলনাই হয় না।	এখনকার নাটক গুলোর কাহিনী কেমন জানি ! আগের দিনের নাটক যেমন “ আজ রবিবার “ এর সাথে এর কোন তুলনাই হয় না ।
আজ সোমবার ...	Removed
Khubi sundor…valo.	খুবই সুন্দর , ভাল ।
অবিনেতা অবিনেত্রী বালো না	অভিনেতা অভিনেত্রী ভাল না ।
সিনেমাটির মধ্যেয্ কোন message নেই।	সিনেমাটির মধ্যেয্ কোন মেসেজ নেই ।
নাটকটি সমাজকে একটি ভাল মেসেজ দিয়েছে ।	নাটকটি সমাজকে একটি ভাল মেসেজ দিয়েছে ।

Figure 5. Comments before and after the Data Cleansing process



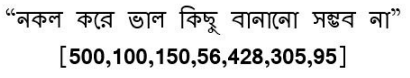
সনে,	1753
রিলিজ,	1754
স্কুল,	1755
সস্তা,	1756
কপি,	1757
গুলোই,	1758
একমাত্র,	1759
সম্পদ,	1760
মোটাদাগের,	1761
নকলকে,	1762
বাঁচিয়েছে,	1763

Figure 6. Graphical representation of a portion of the vocabulary

representation of the sentences using word indexes from vocabulary. Each vector representation is then padded to maximum sentence length in comment sentences with a specific token <0> to make each sentences to equal length. Padding to same length helps to feed data in neural network efficiently. After that matrix generator generates data batches using these vector representations.

$$[Batch\ Size, Sequence\ Length]$$

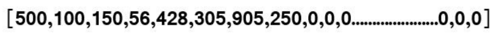
Figure-7 illustrates the vector representation of a sentence. Figure-8 illustrates the vector representation of a sentence after padding with <0>. Figure-9 shows graphical representation of all comments.



“নকল করে ভাল কিছু বানানো সম্ভব না”

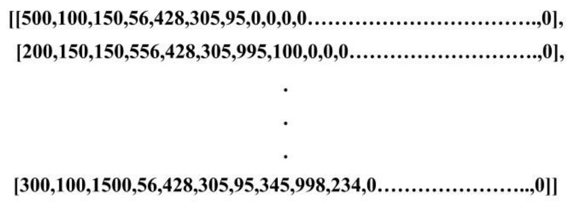
[500,100,150,56,428,305,95]

Figure 7. Vector representation of the sentences



[500,100,150,56,428,305,905,250,0,0,0....................0,0,0]

Figure 8. Vector representation of the sentences after padding



```
[[500,100,150,56,428,305,95,0,0,0,0…………………………….,0],
 [200,150,150,556,428,305,995,100,0,0,0……………………….,0],
 .
 .
 .
 [300,100,1500,56,428,305,95,345,998,234,0…………………..,0]]
```

Figure 9. Batch of training data

4.2. Model Generator

The model generator generates classification model which will be used for sentiment classification. The model generator process will be described for different settings.

4.2.1. Convolution and Max-Pooling. In this setting convolution operations are performed on the input matrices to generate feature maps. The filters slide over the matrix without padding the edges that is narrow convolution. Activation function decides whether a neuron would activate or not. We apply Sigmoid as an activation function on the feature maps as it needs less computation to compute gradient and it performs better for binary classification tasks. The mathematical notation of Sigmoid

$$f(z) = \frac{1}{1 + e^{-z}}$$

Where z is the output of a neuron. Pooling is a technique that results fixed sized output matrix and reduced the output dimensionality but keeps the important feature. Therefore, we applied max pooling on the feature maps. The output actually a feature vector. However, each convolution produces feature maps of different shapes. We perform max-pooling on them which generates feature vector for individual feature map. We then concatenate the feature vectors into a big feature vector.

4.2.2. Fully Connected Layer. In fully-connected layer each neuron locally connected to all the neurons of the next level. The max-pooled data is then fed to fully connected layer to calculate probability distribution and loss and accuracy. In this layer we apply regularization technique on neurons, Softmax-Classifier to calculate probabilities and Cross-Entropy to calculate loss.

Regularization Regularization prevents the network from overfitting. We apply Dropout technique to regularize our network. Dropout is suitable for regularization in mini-batches where batch size is relatively small. Dropout creates a different network by randomly disabling neurons. As a result, the whole procedure is looked like a combination of many different networks each trained with a single sample. During test time the whole network is used but with average weights [13]. We find the highest accuracy for dropout keep probability of 35% during training.

Loss and Prediction We applied Softmax-Classifier to convert the scores to probabilities. Softmax-Classifier maps a vector to output probabilities in binary classification. Moreover, softmax algorithm is suitable for classification problems that are linearly separable. Another advantage of softmax classifier is that it consists of a simple model and thus relatively fast to train and predict [14]. Softmax function calculates probabilities by the following mathematical notation

$$P_i = \frac{exp(y_i)}{\sum\limits_{c=1}^{C} exp(y_c)}$$

Where $i, c \in \{1, ..., C\}$ range over classes, P_i= class probabilities, y_i, y_c = values for a single instance.

We use cross-entropy as a loss function to calculate the error of our network model as it performs well for binary classification. When the output is probability distribution, cross-entropy calculates the distance between the obtained label from the original label. Further more, cross-entropy allow errors to update weights even when nodes derivatives are asymptotically close to 0. The mathematical notation for cross-entropy function

$$C = -\frac{1}{n} \sum_x [y \ln(a) + (1 - y) \ln(1 - a)]$$

Where, $a = \sigma(z)$; is output of the neuron and n = total number of items of training data and x, y = corresponding desired output.

Optimization Optimization is the process of minimizing the gradient by adjusting the weights in back-propagation. We use Adam Optimizer to optimize the gradient in our work. Adam Optimizer is computationally efficient, requires less memory, invariant in rescaling the gradients diagonally [15]. It is also efficient to deal with problem like noisy and/or sparse gradients.

5. Experiments and Evaluation

Models trained with a good dataset with sufficient amount of data are an approximation of the true model. Thus we create a dataset to do experiment with our proposed framework. We experiment our proposed methodology with different hyperparameters and observe the results. Then we test our dataset with other classification algorithms and available Bangla sentiment classifiers to compare their performance with our obtained highest accuracy.

5.1. Dataset

As we know Deep Neural Networks are data intensive and it performs better if trained with a dataset with large amount of data. As dataset in Bangla language is publicly not available, we collect about 850 Bangla comments from different sources and process them to make them suitable to train the network. Among the comments, 500 comments are positive and the rests are negative. But as the amount of comments is not sufficient to train our proposed network, we just copy-paste the comments repeatedly to increase the size of our dataset. We increase the number of comments in our dataset approximately to 120000. Where 60000 comments are positive and 60000 comments are negative. We divide the positive and negative comments into separate files. To evaluate the model at training period we use 10-fold cross-validation on the training data.

TABLE 1. ACCURACY AGAINST DIFFERENT HYPER-PARAMETERS

Batch Size Probability	Dropout Keep Probability	Filter Sizes	Filters Number Per Size	Accuracy
64	.35	3,4,5	128	90.13%
75	.35	3,4,5	128	99.87%
100	.35	3,4,5	128	68.85%

5.2. Experiments and Results

In order to get the best performance from our proposed system we first test the system against different hyperparameters and compare the obtained accuracy. We change the batch size but keep the filter sizes and number of filters per size constant. In addition, we keep the learning rate .0001 unchanged.

Table-1 shows the accuracy against different hyperparameters where column 1 represents batch sizes, column 2 represents dropout keep probability, column 3 and 4 represent filter sizes and number of filters per size respectively and column 5 represents the accuracy obtained against these hyperparameters. From Table-1 as we can see, the highest accuracy is obtained at batch size 75. We can also see that if the batch size is decreased or increased then the accuracy is decreasing.

After finding best hyperparameters for our system, we compare the obtained accuracy with other classification algorithms, as well as other available Bangla sentiment classifiers. In comparison we used same dataset that we described in section 5.1. For example, we conduct an experiment with a Bangla Sentiwordnet-based system. In comparison with our system, the Bangla Sentiwordnet-based system classifies sentiment with less accuracy. Furthermore, we experiment other available Bangla Sentiment classifier with our dataset. One available work on Bangla Sentiment classification by Chowdhury and Chowdhury where they use Support Vector Machine(SVM) to analyze sentiment from Bangla text corpus [6]. They polarize sentiment into two categories i.e, positive and negative. As our proposed framework also analyzes same categories of sentiment therefore we experiment their proposed methodology with our dataset for the feature Unigram+Bigram and observe the accuracy. In such a case our obtained accuracy is also better than them. Moreover we also compare our obtained accuracy with the accuracy of some other available Bangla sentiment classifier. In that case the dataset of those classifiers are different. One of them is the research work carried out by Hassan et.al., where they categorize sentiment either in positive or in negative from Bangla and Romanized Bangla text [7]. However, as their dataset is not publicly available. This is the reason, we just

TABLE 2. Comparison of accuracy of our proposed framework and other available Bangla sentiment classifiers and classification algorithms.

Algorithm	**Accuracy**
Support Vector Machine (Unigram+Emoticons) [6]	93%
Support Vector Machine (Unigram+Bigram) [6]	69%
Deep Recurent Network (LSTM) [7]	78%
Support Vector Machine (Unigram+Bigram) (With our processed data)	93.13%
Bangla Sentiwordnet (With our processed data)	68.85%
CNN With Single Channel	**99.87%**

compare our obtained accuracy with them.
Table-2 shows the comparison. The first column shows the systems/algorithms name while the second column shows the accuracy. As we can see from Table-2, the accuracy (99.87%) of our proposed system is better than other classification algorithms and available Bangla sentiment classifiers. Moreover, our proposed framework does not need any specific feature to classify sentences.

We use Bangla comments mainly from Bangla TV media. Our proposed framework used Convolution Neural Network which treats data as a whole. In CNN-based algorithm it extracts required features for classification automatically. Therefore the proposed framework is not dependent on a particular domain such as movie review or product review. So we can consider that our system will perform equally with any Bangla dataset that consists of large amount of data.
We propose a framework that implements CNN to train a model for sentiment classification from Bangla comments. To get the best performance from the model, we change the hyper-parameters during training and observe it's performance. We obtain a highest accuracy of 99.87% and compare it with other classification algorithms and available Bangla sentiment classifiers which results our accuracy better than other's.

6. Conclusion

In this paper, we propose a framework that implements Convolution Neural Network to train a model that can classify sentiment from Bangla comments. Currently, we classify two types of sentiment i.e. Positive and Negative. As no Bangla dataset is publicly available, thus for the training purpose we generate a small dataset of Bangla sentences to train and test the model. We collect Bangla comments from different sources, filter the comments and copy-paste them repeatedly to increase the number of comments. After pre-processing, we use the comments to train the network. We obtain an training accuracy of 99.87% and compare the accuracy with other available Bangla sentiment classifier. We find our obtained accuracy is the highest in analyzing Bangla sentiment by our observation.
Moreover, as our system currently can classify two categories of sentiment, thus in our furture work we would try to classify all three categories of sentiment i.e Positive, Negative and Neutral. In addition, we would try to increase the amount of comments in our dataset to train our model more effectively.

References

[1] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093 – 1113, 2014.

[2] H. Kanayama and T. Nasukawa, "Fully automatic lexicon expansion for domain-oriented sentiment analysis," in *the Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pp. 355–363, 2006.

[3] Y. Suzuki, H. Takamura, and M. Okumura, "Application of semi-supervised learning to evaluative expression classification," in *the Proceedings of the 7th International Conference of Computational Linguistics and Intelligent Text Processing*, pp. 502–513, 2006.

[4] T. Nasukawa and J. Yi, "Sentiment analysis: capturing favorability using natural language processing," in *the Proceedings of the 2nd International Conference on Knowledge Capture*, pp. 70–77, 2003.

[5] R. Prabowo and M. Thelwall, "Sentiment analysis: A combined approach," *Journal of Informetrics*, vol. 3, no. 2, pp. 143–157, 2009.

[6] S. Chowdhury and W. Chowdhury, "Performing sentiment analysis in bangla microblog posts," *2014 International Conference on Informatics, Electronics & Vision (ICIEV)*, vol. 00, pp. 1–6, 2014.

[7] A. Hassan, N. Mohammed, and A. K. A. Azad, "Sentiment analysis on bangla and romanized bangla text (BRBT) using deep recurrent models," *Computing Research Repository (CoRR)*, vol. abs/1610.00369, 2016. [Online]. Available: http://arxiv.org/abs/1610.00369

[8] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Institute of Electrical and Electronics Engineers(IEEE) Computer*, vol. 29, no. 3, pp. 31–44, 1996.

[9] Y. Kim, "Convolutional neural networks for sentence classification," in*the Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1746–1751, 2014.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *the Proceedings of the Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems*, pp. 1106–1114, 2012.

[11] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *the Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 1237–1242, 2011.

[12] Y. Zhang and B. C. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *Computing Research Repository (CoRR)*, vol. abs/1510.03820, 2015. [Online]. Available: http://arxiv.org/abs/1510.03820

[13] L. J. Ba and B. Frey, "Adaptive dropout for training deep neural networks," in *the Proceedings of the 26th International Conference on Neural Information Processing Systems*, pp. 3084–3092, 2013.

[14] K. Duan, S. S. Keerthi, W. Chu, S. K. Shevade, and A. N. Poo, "Multi-category classification by soft-max combination of binary classifiers," in *the Proceedings of the Multiple Classifier Systems, 4th International Workshop, (MCS)*, pp. 125–134, 2003.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computing Research Repository (CoRR)*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980