# Attention based Convolutional Neural network for Bangla Sentiment analysis

Sadia Sharmin and Danial Chakma
*Computer Science and Engineering Department*
*Bangladesh University of Engineering and Technology*
Dhaka, Bangladesh
sadiasharmin@cse.buet.ac.bd, danial08cse@gmail.com

June 1, 2021

## Abstract

With accelerated evolution of the internet in the form of websites, social networks, micro-blog and online portals, huge amount of reviews, opinions, recommendations, ratings, and feedback are generated by writers or users. This users generated sentiment content can be about books, people, hotels, products, research, events, etc. These sentiments become very beneficial for businesses, governments, and individuals. While this content is meant to be useful, a bulk of this writer generated content require using the text mining techniques and sentiment analysis. But there are several challenges facing the sentiment analysis and evaluation process. These challenges become obstacles in analyzing the accurate meaning of sentiments and detecting the suitable sentiment polarity specifically in Bangla language. Sentiment analysis is the practice of applying natural language processing and text analysis techniques to identify and extract subjective information from text. This paper presents how attention mechanism could be incorporated effectively and efficiently in analyzing Bangla Sentiment or Opinion.

**Keywords:** Sentiment Analysis(SA), Polarity Prediction, Attention, CNN, NLP, Deep Learning(DL).

# 1 Introduction

Sentiment Analysis or Opinion Mining is a Natural Language Processing and Information Extraction task that aims to obtain writers feelings expressed in positive or negative comments, questions and requests, by analyzing a large numbers of documents. Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall tonality of a document. In recent years, the exponential increase in the Internet usage and exchange of public opinion is the driving force behind Sentiment Analysis today. The Web is a huge repository of structured and unstructured data. The analysis of this data to extract latent public opinion and sentiment is a challenging task. Liu et al. (2009) [1] defines a sentiment or opinion as a quintuple-

> <$o_j$, $f_{jk}$, $so_{ijkl}$, $h_i$, $t_l$>, where $o_j$ is a target object, $f_{jk}$ is a feature of the object $o_j$, $so_{ijkl}$ is the sentiment value of the opinion of the opinion holder $h_i$ on feature $f_{jk}$ of object $o_j$ at time $t_l$, $so_{ijkl}$ is +ve, -ve, or neutral, or a more granular rating, $h_i$ is an opinion holder, $t_l$ is the time when the opinion is expressed.

The analysis of sentiments may be document based where the sentiment in the entire document is summarized as positive, negative or objective. It can be sentence based where individual sentiment bearing sentences in the text are classified. It can also be phrase based where the phrases in a sentence are classified according to polarity. Sentiment Analysis identifies the phrases in a text that bears some sentiment. The author may express about some objective facts or subjective opinions. It is necessary to distinguish between the two. SA finds the subject towards whom the sentiment is directed. A text may contain many entities but it is necessary to find the entity towards which the sentiment is directed. It identifies the polarity and degree of the sentiment. Sentiments are classified as objective (facts), positive (denotes a state of happiness, bliss or satisfaction on part of the writer) or negative (denotes a state of sorrow, dejection or disappointment on part of the writer). The sentiments can further be given a score based on their degree of positivity, negativity or objectivity.

*Novelty of our approach*: Researcher until now have performed sentiment analysis using Bangla text in several ways and approaches including SVM, MaxEnt, LSTM, CNN etc. which are discussed in sec. **??**. Attention techniques which is introduced in early 2016, developed and thoroughly investigated in NLP tasks involving English Language. However, to our knowledge, it is not yet investigated or employed in analyzing Bangla sentiment. This is the main difference between others approach and our approach in Bangla sentiment analysis task. This paper present an empirical study about the effectiveness of attention mechanism by incorporating it into convolutional neural network.

*Our contribution* in this paper is threefold: (a) Investigating the effectiveness of attention mechanism in detecting Bangla sentiment from Bangla social media data. (b) Minimization of false detection rate or enhancing prediction accuracy by empirical search of hyper-parameters and feature engineering. (c) Automated generation of synthetic data from existing real dataset using Generative Adversarial Network(GANs) and test the performance of our fine tuned model against these synthetic data. In addition, this is the first application of attention mechanism in Bangla sentiment analysis research papers which automatically deserves further investigation whether attention mechanism is actually capable to capture feature dependency among long Bangla sentences.

The rest of the paper is organized as follows. Sec. **??** discusses some prominent applications of SA, Section 1.2 describes challenges in SA, Section **??** describes our model and approach, Section **??** describes about the dataset and feature extraction process, Section **??** present experimental result and performance analysis, and finally Section **??** provides concluding remarks and future works.

## 1.1 Applications of Sentiment Analysis

The process of conveying information from person to person is called Word of Mouth(WOM) and plays a major role in customer buying decisions. In commercial situations, WOM involves consumers sharing attitudes, opinions, or reactions about businesses, products, or services with other people. WOM communication functions based on social networking and trust. People rely on families, friends, and others in their social network. Research also indicates that people appear to trust seemingly disinterested opinions from people outside their immediate social network, such as online reviews. This is where Sentiment Analysis comes into play. Growing availability of opinion rich resources like online review sites, blogs, social networking sites have made this “decision-making process” easier for us. With explosion of Web 2.0 platforms consumers have a soapbox of unprecedented reach and power by which they can share opinions. Major companies have realized these consumer voices affect in shaping voices of other consumers. Sentiment Analysis thus finds its use in Consumer Market for Product reviews, Marketing for knowing consumer attitudes and trends, Social Media for finding general opinion about recent hot topics in town, Movie to find whether a recently released movie is a hit. Pang-Lee et al. (2002) [3] broadly classifies the applications into the following categories.

- Applications to Review-Related Websites

  Movie Reviews, Product Reviews etc.

- Applications as a Sub-Component Technology

  Detecting antagonistic, heated language in mails, spam detection, context sensitive information detection etc.

- Applications in Business and Government Intelligence

  Knowing Consumer attitudes and trends

- Applications across Different Domains

  Knowing public opinions for political leaders or their notions about rules and regulations in place etc.

## 1.2 Challenges

Sentiment Analysis approaches aim to extract positive and negative sentiment bearing words from a text and classify the text as positive, negative or else objective if it cannot find any sentiment bearing words. In this respect, it can be thought of as a text categorization task. In text classification there are many classes corresponding to different topics whereas in Sentiment Analysis we have only 3 broad classes. Thus it seems Sentiment Analysis is easier than text classification which is not quite the case. The following general challenges not only apply for English languages but also other languages such as Bangla.

### 1.2.1 Implicit Sentiment and Sarcasm

A sentence may have an implicit sentiment even without the presence of any sentiment bearing words. Consider the following examples: (a) How can anyone sit through this movie? (b) One should question the stability of mind of the writer who wrote this book. Both sentences do not explicitly carry any negative sentiment bearing words although both are negative sentences. Thus identifying semantics is more important in SA than syntax detection.

### 1.2.2 Domain Dependency

There are many words whose polarity changes from domain to domain. Consider the following examples. (a) The story was unpredictable. (b) The steering of the car is unpredictable. (c) Go read the book. In the first example, the sentiment conveyed is positive whereas the sentiment conveyed in the second is negative. The third example has a positive sentiment in the book domain but a negative sentiment in the movie domain (where the director is being asked to go and read the book).

### 1.2.3 Thwarted Expectations

Sometimes the author deliberately sets up context only to refute it at the end. Consider the following example:

> This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it cant hold up.

In spite of the presence of words that are positive in orientation the overall sentiment is negative because of the crucial last sentence, whereas in traditional text classification this would have been classified as positive as term frequency is more important there than term presence.

### 1.2.4 Pragmatics

It is important to detect the pragmatics of user opinion which may change the sentiment thoroughly. Consider the following examples. (a) I just finished watching Barca DESTROY Ac Milan. (b) That final completely destroyed me. Capitalization in words can be used with subtlety to denote sentiment. The first example denotes a positive sentiment whereas the second denotes a negative sentiment. There are many other ways of expressing pragmatism.

### 1.2.5 Subjectivity Detection

This is to differentiate between opinionated and non-opinionated text. It is used to enhance the performance of the system by including a subjectivity detection module to filter out objective facts. But this is often difficult to do. Consider the following examples: (a) I hate love stories. (b) I do not like the movie “I hate stories”. The first example presents an objective fact whereas the second example depicts the opinion about a particular movie.

### 1.2.6 Entity Identification

A text or sentence may have multiple entities. It is extremely important to find out the entity towards which the opinion is directed. Consider the following examples. (a) Samsung is better than Nokia. (b) Bangladesh defeated Srilanka in Cricket. The examples are positive for Samsung and Bangladesh but negative for Nokia and Srilanka.

### 1.2.7 Negation

Handling negation is a challenging task in SA. Negation can be expressed in subtle ways even without the explicit use of any negative word. A method often followed in handling negation explicitly in sentences like “I do not like the movie”, is to reverse the polarity of all the words appearing after the negation operator (like not). But this does not work for “I do not like the acting but I like the direction”. So, we need to consider the scope of negation as well, which extends only till “but”. So the thing that can be done is to change polarity of all words appearing after a negation word till another negation word appears. But still there can be problems. For example, in the sentence “Not only did I like the acting, but also the direction” the polarity is not reversed after “not” due to the presence of “only”. So this type of combinations of “not” with other words like “only” has to be kept in mind while designing the algorithm.

# 2 Related Works

Sentiment classification [2] is classical problem associated with determining the attitude and effective reaction of a person to an event, document and/or media item. A typical setting for training sentiment classification models involves feature extraction on a labeled corpus, followed by a classifier training [3]. Such an approach has seen fair amount of success in several sentiment classification tasks such as twitter sentiment classification [4], movie reviews [5] and product reviews [6]. As the problem of sentiment classification expand to new arenas, such as new modes of expressions on social media, different languages and even different cultures, large amounts of labeled data and technical resources may not be available and also syntactical structure and semantic meaning of different language may differ from language to language.

Despite challenges, Sentiment analysis or Opinion mining have become major point of focus in Natural language processing and have gained extensive attention and popularity in research community. This is proven in the paper [8], with over 7,000 articles written on the subject. In recent period, deep learning applications have exhibited impressive performance across wide range of NLP tasks including Machine Translation(MT), Text to speech. Using deep learning methods, a large portion of works have involved learning word vector representations [9] and performing Convolutional Neural Network (CNN) based feature extractor for classification [10–12]. Shirnai explored the efficacy of different deep learning architectures for semantic analysis of movie reviews and achieved 46.4% accuracy in Stanford Sentiment Treebank dataset using CNN+word2vec model for five labels sentiment classification [23]. Nowadays, besides sentiment mining, the emotional aspects in texts also attract the attention of many research areas in NLP and different researchers focus on identifying emotions. Chaffar & Inkpen adopted a supervised machine learning approach to recognize Ekmans [14] six basic emotions using different feature sets. Bhowmick et al. [15] used an ensemble based multi-label classification technique called random k-label set (RAKEL) for emotion analysis.

Although the problem of classifying sentiment in recent year have seen fair amount of success in English language using deep learning methods, but it is not widely studied in Asian languages such as Bangla, Hindi, Chinese etc. We have found limited resources about sentiment analysis in Bangla. Among these works, Amitava Das in [26] constructed sentiment lexicon for Bangla language from English SentiWordNet [7], Shaika et al. [27] performed sentiment analysis using SVM and MxtEnt for Bangla micro-blog posts, Azharul et al. [28] reviewed sentiment analysis among some Asian languages including Bangla. Recently, researchers have expressed their interest in Bengali text and there are many publications based on sentiment and emotion analysis, theme detection, topic wise opinion summarization with data resources from various Bengali corpus [16]. Different Machine learning techniques like SVM with maximum entropy [27], Naive bayes (NB) [24], Multinomial Naive Bayes (MNB) with mutual information as feature representation [25] has been used to classify Bangla sentences into positive or negative in various Bangla domain texts like Micrblog posts, comments from blog, translated review dataset. Word2vec word co-occurrence score combined with the sentiment polarity score of the words was proposed by Amin et al. [17] and obtained 75.5% accuracy in each of two classes. Hasan et al. [30] has performed sentiment analysis on Bangla and Romanized bangla text using a Long Short Term Memory (LSTM) with binary and categorical cross entropy loss and achieved 70% accuracy for two class.

Much of the emotion analysis task in Bangla texts have been carried out by Das and Bandyopadhyay in words, phrases and sentence levels in Bangla corpora and blogs [20, 21]. Consequently, they developed WordNet Affect lists in Bangla from the affect wordlists already available in English. For identifying emotions from sentence, they used a Conditional Random Field (CRF) based classifier for recognizing six basic emotion tags for different words of a sentence.

# 3 Our Proposed Architecture and Method

We adopted attention based Convolutional neural network to analyze Bangla text sentiment. Our model utilizes the sparsity of data matrix which is the representation of all textual comments bearing positive, negative or neutral sentiment. In general, classification or prediction task in natural language processing represent a contextual document or comment in some numeric form based on word frequency or word presence. Representation of a document or a comment like this usually result in sparse dataset which basically means most elements in a document or comment are zeros except some element that represent that document or comments. Also, working with large volume of textual data tends to increase vocabulary size which present us huge challenge in containing data matrix dimension. This problem is called curse of dimensionality in Data mining field. Therefore, restricting data dimension or feature dimension is also utmost priority. Data set constructed from huge volume of vocabulary or features are sparse in nature. In those cases, application of attention in neural network is typically useful for sparse dataset not only for faster training but also for obtaining good generalization of neural network for future unknown data. With this in mind, our proposed architecture consist of convolutional layer for feature extraction followed by one attention layer and then couple of feed forward layer for classification of sentiment bearing Bangla comments. Our proposed model divided into four basic parts:

(a) Feature Extraction layer

(b) Intermediate Normalization Layer

(c) Attention Layer

(d) Feed forward Classification layer.

**Feature Extraction Layer** Feature extraction is one of the important aspects in machine learning based detection or prediction system. Convolutional neural network have shown promising result in extracting image features in Computer visions and many other fields. Inspired with that, We have used three convolutional layer successively one after another for both feature extraction and input dimension and/or timestep reduction. In first layer four one dimensional convolution (Conv1D keras layer) is used each having 8 filter with kernel size 2, 3, 4, and 5 respectively having stride of 5, then these four Conv1D layer is concatenated. Having stride size of 5 will reduce timestep (input feature dimension) to a factor of 5, in our case, to 1840/5 = 368. Next, second layer is constructed with four Conv1D each having 5 kernel of size 1, 2, 3, and 4 respectively strided with 5, then these four Conv1D is concatenated again to form one single keras tensor. Similarly, following third layer is formed but with each having 3 filter and stride of 2. In each convolution layer we have used Exponential Linear Unit (ELU) as activation function which is known to be faster and effective in training neural network. Each convolutional layer consist of filters(also called number of kernel) each of specific size(also called kernel size or sliding window size) and each kernel is slided or strided by a specific integer amount along the input dimension. First convolutional layer consist of 78 filters or kernels each of size 32 (also called kernel size or sliding window size) and each kernel is slided or strided by 2 along the input dimension. With the defined kernel size and input size, the layer will output a matrix of shape or size x by y, x is input-dim/stride and y is number of filters, each column of which holds weight of single filter. In our case, each learn-able filter in the first layer will contains 368 trainable weight. In general, each filter is sufficient to train one feature of input data tuple.

**Intermediate Normalization Layer** After feature extraction layer, a Maxpooling layer followed by one Batch Normalization and one ELUs layer is introduced which acted as intermediate bridge between Feature extraction layer and Attention layer. Batch Normalization [37] is crucial for deep neural network not only for network stabilization but also to solve overfitting and under-fitting problems during training deep network. At first, we were undecided whether to employ Maxpooling before Batch Normalization+Activation or not, but after some study we decided to go with Batch Norm+Activation+Maxpool.

**Attention Layer** Following the feature extraction layer, we introduced a attention layer which is critical component in our network architecture. Attention mechanism in neural network was born to solve long distance dependency problem in machine translation (MT) from source language to target language. The intuition behind attention came from how human pay visual attention to different region of object such as image or words in one sentence. Human visual attention usually focuses on a certain part with "high resolution or attention" while perceiving surrounding part in "low resolution or attention" and then adjust the focal point or do inference on the object accordingly. In a words, we do not need all aspect to recognize object only important one will suffice. Motivated from this, the attention mechanism was introduced in [31–34], via applying some weight on the input data. We used additive attention mechanism, our attention layer consist of 32 dimensional attention vector, local attention width of 16, and softmax activation function. Attention vector computation formula is given below:

$$h_t = \tanh(x_t^T W_t + x_t^T W_x + b_h) \tag{1}$$

$$e_t = h_t^T W_a + b_a \tag{2}$$

$$a_t = \frac{\exp(e_t)}{\sum_{i=1}^{N} \exp(e_t(i)) + \epsilon} \tag{3}$$

Where $a_t$ attention vector, $W_a, W_t, W_x$ are trainable weight matrices, $b_h, b_a$ are trainable biases and $N$ is input layer number in attention layer, $\epsilon =$

exp(−7) is added to avoid divide by zero run-time errors.

**Classification Layer** Classification Layer constitute of three consecutive dense layer one after another after having flatten previous layer output. With exception just before last dense layer in which a batch normalization layer is added to normalize previous two heavy dense layer output. Batch normalization proven to be useful for reduction of over-fitting during training. In each dense layer, ELU activation function is employed except the final layer where softmax is employed to produce a probability distribution over the target class. First two dense layer employed 64 and 32 neurons or units respectively whereas last layer witnessed 3 neurons or units as we had 3 different target classes positive, negative and neutral sentiment bearing comments in the data set. Activation function used:

$$f(x) = \begin{cases} \alpha(e^x - 1), & \text{if } x \leq 0. \\ x, & \text{if } x > 0. \end{cases} \tag{4}$$

where, $\alpha$ is a scaler parameter and usually assume values between [0, 1]. Softmax function is defined as:

$$f_i(\vec{x}) = \frac{e^{x_i}}{\sum_{j=1}^{J} e^{x_j}} \quad \text{for} \quad i = 1, 2, \ldots, J \tag{5}$$

where, $\vec{x}$ is a input vector to the softmax functon and J is number of input layer.

# 4 Dataset and Features Extraction

We have used dataset collected from the social media which are manually handcrafted and labelled. Among the social media, we have chosen BBC Bangla and Prothom Alo platform for the source of Benglai courpus. Some of the sample comments and reviews from the media are listed in figure 16. These are the user comments about a specific topics, these includes politics, sports etc. In sport topic, we categorized those user comments about specific aspect of a sport. For example, for the cricket, we have included user comments about bowling, batting, team management aspect of the cricket.

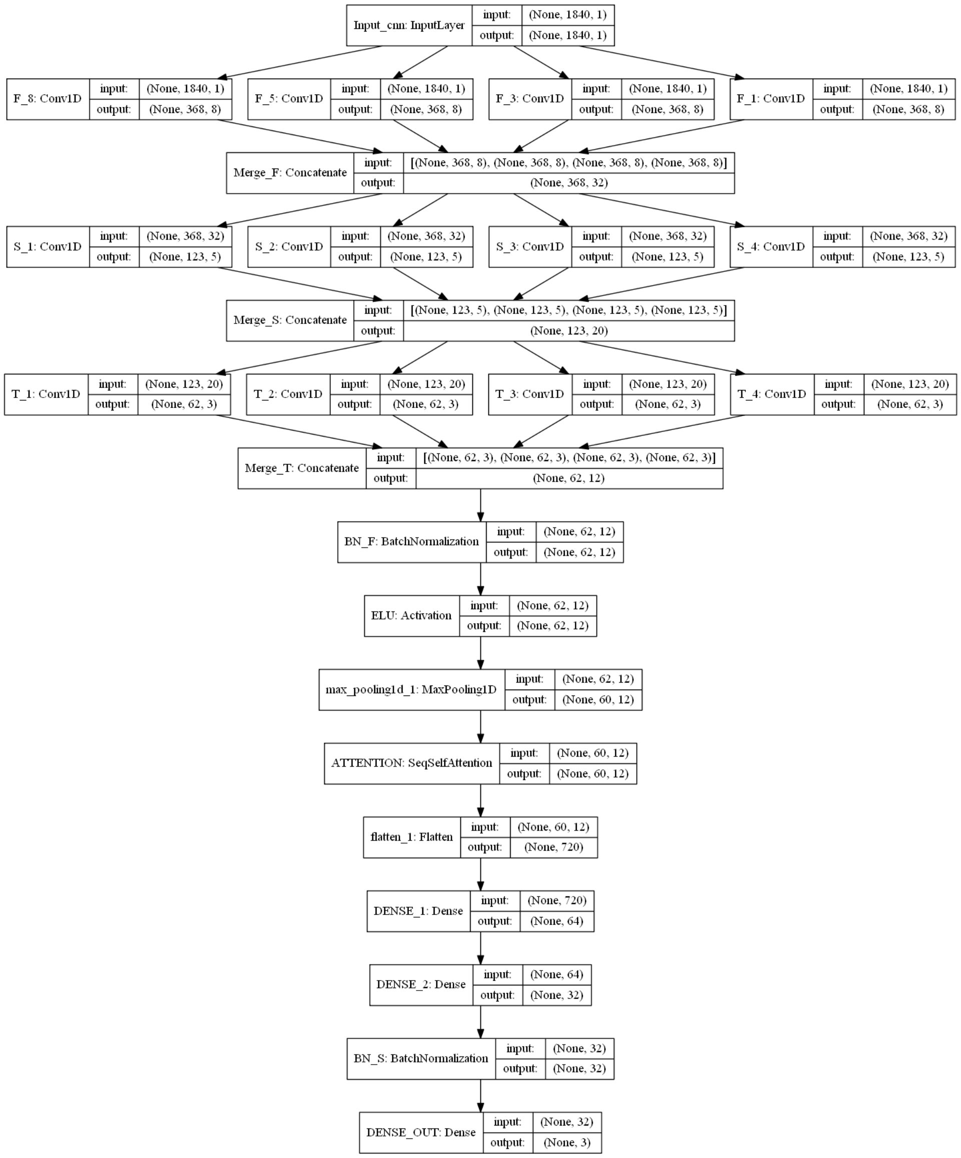


Figure 1: Our Proposed Neural network

Labeling of those reviews and comments were performed manually by examining the comment content. Our dataset includes in total 2979 reviews and comments, among these comments 566, 2152, and 261 were identified as positives, negatives and neutral comments respectively. Data distribution of these three classes are shown in figure 2. Dataset consists of 5316 unique words or vocabulary size after preprocessing of stop words and stemming. Stemming is a useful technique to find only root word after removing suffix and/or prefix from a specific word. In extracting feature, we have used one of Bag of words(BOW) techniques called term frequency and inverse document frequency(TF-IDF) to extract features from filtered comments. Although we pre-processed raw comments, we still had a large set of vocabulary or feature dimension to work with. So, we decided to apply feature reduction technique using PCA (principal component analysis) and reduced dimensionality to 1536 that kept around 92.34% variance among feature set. To decide which feature to keep or which should be discarded, we performed an experiment (PCA) with original dataset and plotted a line graph with y-axis as cummulative variance ratio and x-axis as number of component which is shown in figure 3. It is clear that if we want to ensure feature variance above 90, than

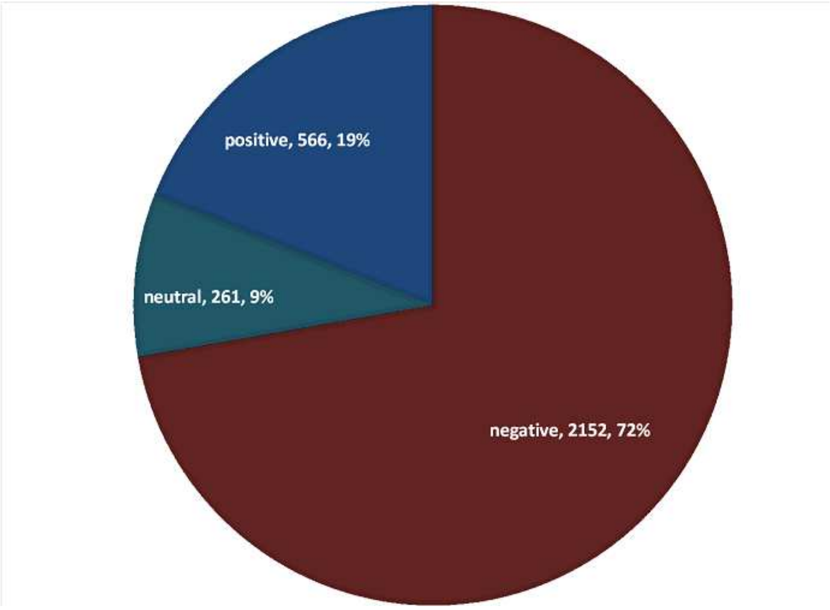


Figure 2: Proportion of class in Dataset

we need to keep at least 1500 features(or vocabulary size) among 5316 features. In figure 17, some of the most frequently occurring negative words are listed from our dataset. Finally, we checked for missing or null values in any attributes and replaced those with zero values. Then, dimension reduced dataset were transformed into the range of [-1,1] using the following min-max normalization formula for faster processing and/or training our neural network.

$$x_{std} = \frac{x - min(x)}{max(x) - min(x)} \tag{6}$$

$$x_{new} = x_{std} * (max - min) + min \tag{7}$$

where min and max are the new feature range.

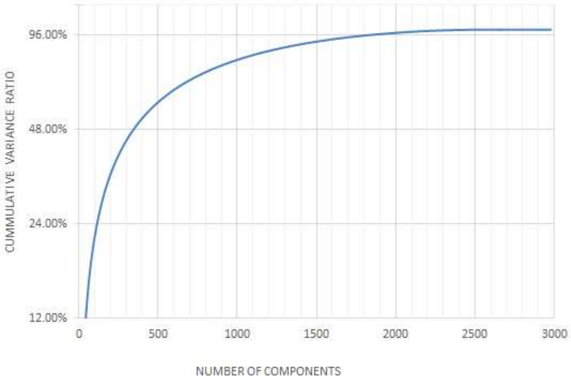


Figure 3: Cummulative variance ratio vs Component numbers

# 5 Experimental Result and analysis

We have performed experiment to test our model's rigidity with different settings. Following subsection will discuss about our model performance and sensitivity to different learning rate. In this experiment, we partitioned our dataset into training and validation subset which correspond to 75 and 25 percent respectively and trained our model with batch size 32 and 140 epoch. For optimization algorithm, we employed stochastic gradient descent algorithm using NADAM [35, 36] optimizer that optimizes loss function defined in equation 16 and 17. We have seen our model generate best performance with learning rate 0.002 and at the end of 140 epoch achieve 96.36% and 70% accuracy on training and validation set respectively.

## 5.1 Performance Metrics

To measure model performance, some evaluation metrics were used whose definitions and formulas are given below. True positive(tp): Number of actual positive instances that are predicted as positive. False negative(fn): Number of actual positive instances that are predicted as negative. False positive(fp): Number of actual negative instances that are predicted as positive. True negative(tn): Number of actual negative instances that are predicted as negative. N: Total number of instances or tuples. C: Number of different classes. Having defined above definition, we can now define accuracy, precision, recall, f-measure, micro and macro average measures.

$$Accuracy = \frac{\sum_c^C tp_c}{N}, \text{in general} \tag{8}$$

$$Accuracy = \frac{tp + tn}{N}, \text{for binary case} \tag{9}$$

$$Error(e) = 1 - Accuracy \tag{10}$$

$$precision = \frac{tp}{tp + fp} \tag{11}$$

$$recall = \frac{tp}{tp + fn} \tag{12}$$

$$F_{measure} = \frac{2 * precision * recall}{precision + recall} \tag{13}$$

The macro-averaged results can be computed as indicated by: $L = \{\lambda_j : j = 1 \ldots q\}$ is the set of all labels. Consider a binary evaluation measure B(tp, tn, fp, fn) that is calculated based on the number of true positives (tp), true negatives (tn),false positives (fp) and false negatives (fn). Let $tp_\lambda, fp_\lambda, tn_\lambda$ and $fn_\lambda$ be the number of true positives, false positives, true negatives and false negatives after binary evaluation for a label $\lambda$.

$$B_{macro} = \frac{1}{q} \sum_{\lambda=1}^{q} B(tp_\lambda, tn_\lambda, fp_\lambda, fn_\lambda) \tag{14}$$

A micro-averaged can be computed as follow:

$$B_{micro} = B\left(\sum_{\lambda=1}^{q} tp_\lambda, \sum_{\lambda=1}^{q} tn_\lambda, \sum_{\lambda=1}^{q} fp_\lambda, \sum_{\lambda=1}^{q} fn_\lambda\right) \tag{15}$$

For loss function, we have used categorical cross entropy to compute loss at the end of each epoch because our target(class) label were one hot encoded. The equation for computation of loss is given below:

$$\hat{y}_{norm} = \frac{\hat{y}}{\sum_{x \in \hat{y}} x} \tag{16}$$

$$\mathcal{L}(\hat{y}_{norm}, y_{target}) = -\sum_{x} y_{target}(x) * \log(\hat{y}_{norm}(x)) \tag{17}$$

where, $\hat{y}$ is the output vector of last layer, $\hat{y}_{norm}$ is the normalized vector of $\hat{y}$, $y_{target}$ is the true class label vector, x is the index of both equal dimensional vector $y_{target}$ and $\hat{y}_{norm}$, and finally $\mathcal{L}$ is the scalar loss value which is minimized by the optimizer.

## 5.2 Model Accuracy Evaluation

From figure 4, we observe that over epoch training accuracy increases for the noted learning rate [0.3, 0.1, 0.03, 0.01, 0.002, 0.001, 0.0003, 0.0001]. However, all the depicted learning rate(lr) settings, model do not perform equally. For instance, model achieve accuracy slightly above 0.7 with lr(0.3) while achieve accuracy slightly below 0.92 with lr(0.1, 0.03, 0.0001), for the others learning rate model achieves more or less same training accuracy at the end of 100 epoch. One important thing to notice here is, model performs best in accuracy when learning rate set to 0.002 and degrades performance if set either below or above this learning rate which confirms our claim in 5. Validation curves in figure 5 follows more or less similar trends. Notice that, for the best learning rate [0.001, 0.002, 0.003] within 20 epoch model learn around 80 percent about the distribution of training set while for validation set it achieves around 70 percent. From figure 6, we notice that validation accuracy curve follows training accuracy curve and on average validation accuracy is always below the training accuracy though we witness some overlapping and spike for both curves at the initial epoch, namely between [0, 21], perhaps this is due to the random initialization of weights and biases of the model in all layers. Overall, model achieves quite significant level of accuracy with limited model capacity(few layers and parameters).

From figure in 7, for learning rate [0.001, 0.002] the model do quite well to learn or distinguish target classes from feature set or data distribution, but performance degrades or slowly learn for others learning rate. For best learning rate group the model achieves overall training loss of around 0.1 at the end of epoch 100. For the second best learning rate group the model gain overall training loss of between 0.15-0.2 and for the third best group, it achieves training loss of around 0.25. Notice that for lr(=0.3), the model does not learn at all, loss curve goes upward instead of downward for almost all the epoch. Likewise, almost all Loss curves in figure 8 follows similar pattern for validation set also. To observe whether the model is over-fitting or under-fitting, we plotted training loss and validation loss together over epoch in figure 9. We witnessed that our model do not over-fit or under-fit for the future dataset significantly, which confirms effectiveness of batch normalization to reduce under-fitting or over-fitting. Therefore, our model do in-fact generalize towards ground truth quite efficiently for the future unseen data.

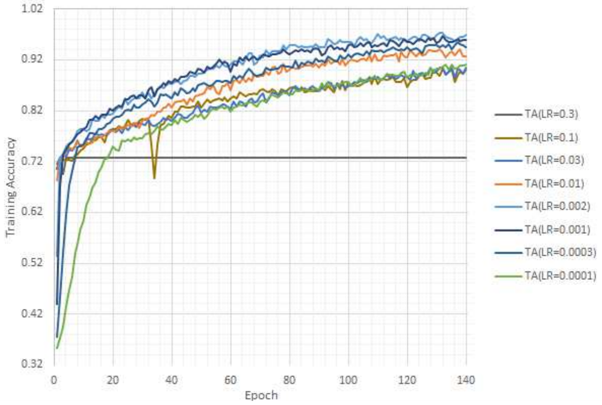


Figure 4: Training accuracy

## 5.3 Performance Comparison with LSTM

To compare performance of our base model(CNN), with other RNN based model we have chosen to replace third convolution layer with 8 unit of LSTM layer into our base model, keeping all other layer's parameters exactly the

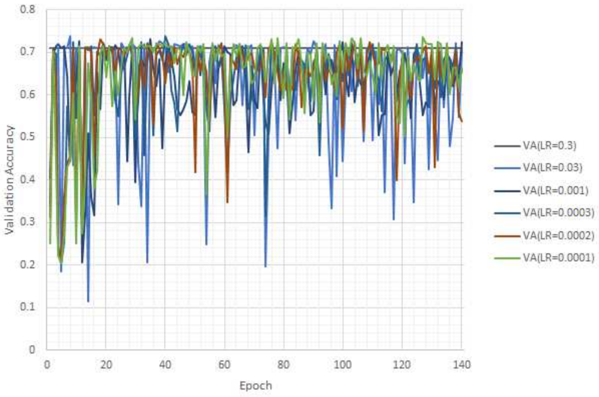


Figure 5: Validation accuracy

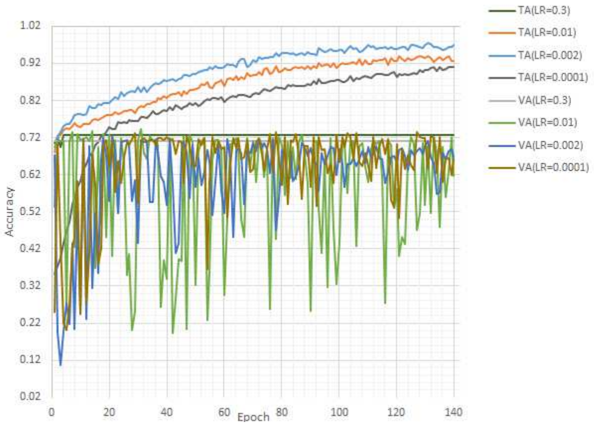


Figure 6: Training and Validation Accuracy

same. In fact, model capacity of CNN+LSTM is increased by amount of 15,924 in terms of training parameters. Because, our base model had a total of 51,740 parameters of which 51,652 are trainable and 88 are non-trainable. In contrast, CNN+LSTM model had 67,576 trainable and 80 non-trainable summing up 67,656 parameters in total. Increased model capacity of later, however, never reflected in performance in terms of training and validation accuracy, training and validation loss as clearly evident from figure 10, 11, 12, 13

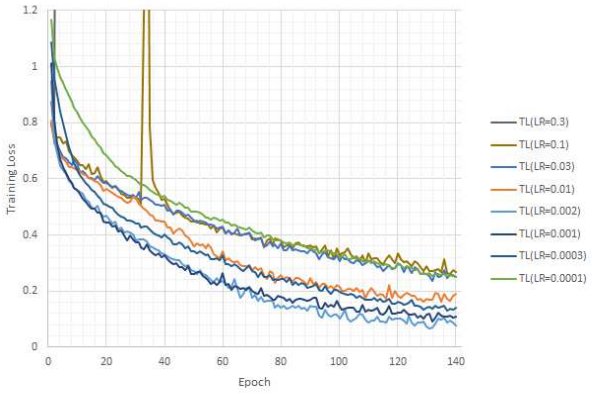


Figure 7: Training Loss

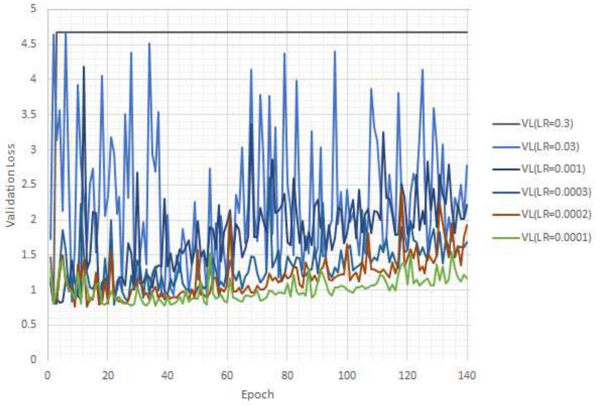


Figure 8: Validation Loss

## 5.4 Performance Evaluation with CM and ROC Curves

To further investigate and performance evaluation, we plotted confusion matrix and receiver operating curves(ROC) with validation data set. ROC curves are very useful technique or tool for performance evaluation of model having trained with unbalanced dataset and examining performance of specific classes of data. For instance, in medical diagnosis detecting cancer which is rare in nature from signature or symptom. In our case, measuring performance of model by evaluating validation dataset from each class label (positive, negative and neutral) perspective is justified. From figure in 14, we observe that our model achieved 90%, 35%, and 7% in predicting nega-

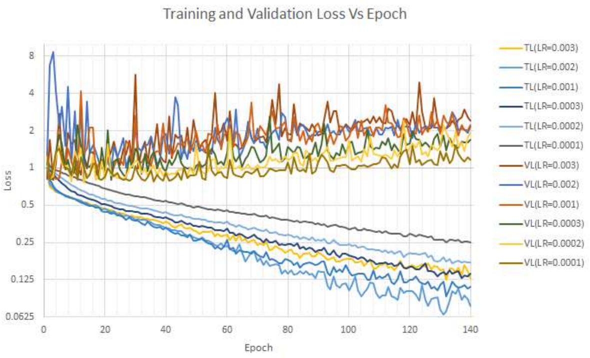


Figure 9: Training and Validation Loss

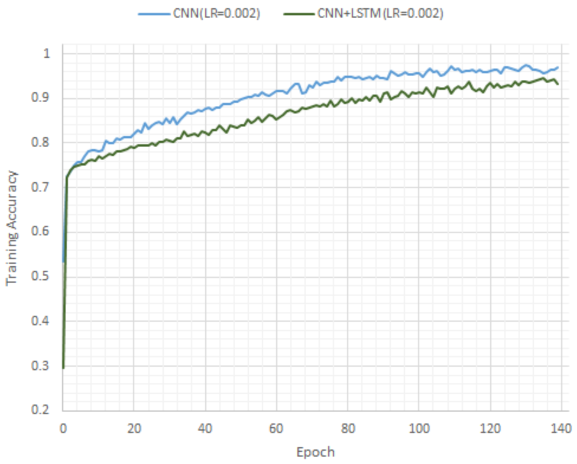


Figure 10: Training Accuracy Comparison

tive, positive and neutral sentiment containing comments respectively. So, our model quite convincingly detect negative sentiment bearing comments, but struggle to predict or correctly classify positive and neutral sentiment in comments. This is probably attributed to the words in sentences that have overlapping connotation and challenges mentioned in section 1.2. Confusion matrix(CM) in figure 15 not only confirms this findings but also tells us that considerable portion of positive and neutral sentiment bearing comments (61% and 76% respectively) are being predicted as negative and also 17% of neutral oriented comments are predicted as positive which is prohibiting our model to achieve overall greater performance (nearly 100%). Another

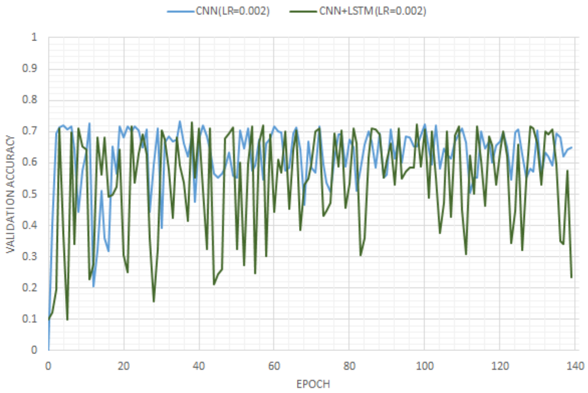


Figure 11: Validation Accuracy Comparison

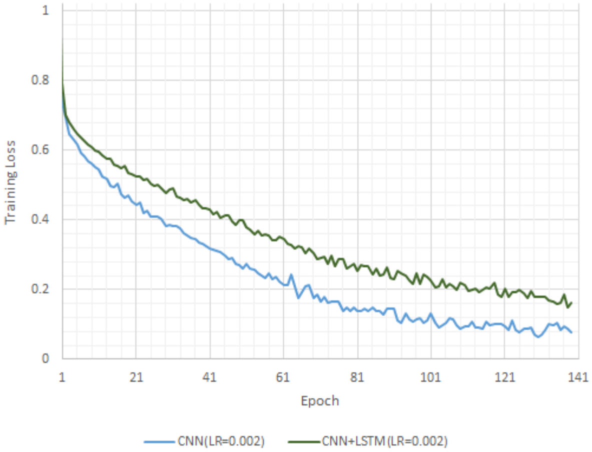


Figure 12: Training Loss Comparison

more important but concerning aspect about the prediction is some negative comments being predicted as neutral (2%) and positive (7%) which is quite disappointing as in sentiment prediction we are mostly concern with predicting negative sentiment correctly.

## 5.5 Cross Validation with StratifiedKFold

Final experiment we have performed to verify how our model performs with whole dataset in consideration. In the previous experiment, we have just

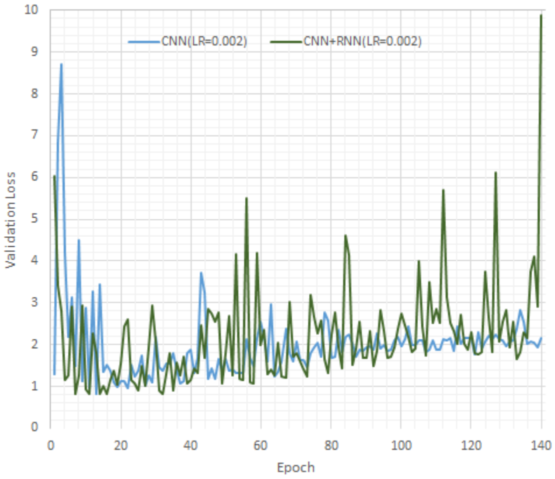


Figure 13: Validation Loss Comparison

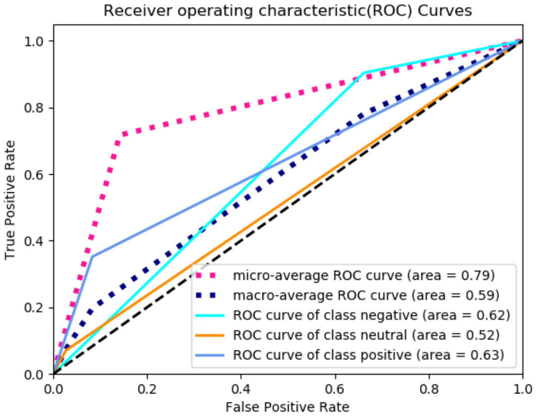


Figure 14: Receiver Operating Characteristic Curves

used two fixed independent set training and validation set for training and testing respectfully. In this case, we are going to perform stratified 10 folds cross validation which is similar to K-fold cross validation with one difference. K-fold do not consider class or group distribution while producing k number of folds but StratifiedKFold maintains equal percentage of samples for each class in each of the k-fold. In each of the 10 iteration, training set contains 9 folds and test set contains 1 fold, with 9 folds we train our model and at

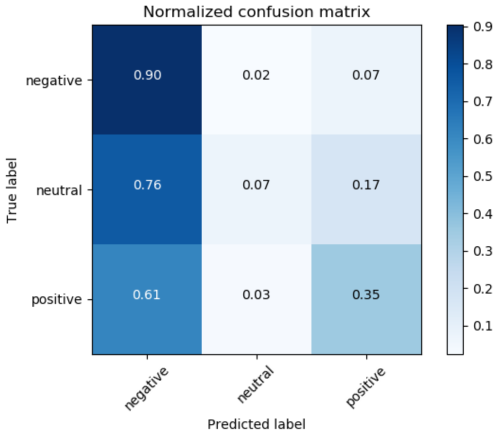


Figure 15: Normalized Confusion Matrix

the end of training we perform evaluation with remaining 1 fold. So, in a sense, we are testing all the dataset each time testing a unique fold against all the other 9 folds. Finally, taking the average of all 10 round scores as the final score. Our model achieved accuracy 66.06%(+/-3.44%), Precision 66.40%(+/-3.45%), Recall 65.66%(+/-3.51%), and F-measure 66.02%(+/-3.48%).

| **Round** | **Accuracy** | **Precision** | **Recall** | **F-measure** |
|---|---|---|---|---|
| 1 | 67.66 | 68.26 | 67.00 | 67.62 |
| 2 | 67.89 | 68.07 | 67.89 | 67.98 |
| 3 | 66.10 | 66.84 | 65.77 | 66.28 |
| 4 | 62.08 | 62.23 | 61.07 | 61.63 |
| 5 | 69.46 | 69.69 | 69.46 | 69.57 |
| 6 | 64.76 | 65.11 | 64.42 | 64.76 |
| 7 | 65.65 | 65.76 | 65.31 | 65.53 |
| 8 | 71.71 | 71.77 | 71.04 | 71.40 |
| 9 | 66.32 | 67.09 | 65.99 | 66.53 |
| 10 | 58.92 | 59.12 | 58.58 | 58.85 |
| Mean($\mu$) | **66.06** | **66.40** | **65.66** | **66.02** |
| SD($\sigma$) | **3.44** | **3.45** | **3.51** | **3.48** |

# 6 Conclusion and Future works

Sentiment Analysis have played a crucial role for the IT enabled business and service owners due to the automatic analysis of user reviews and opinions about goods and services. Furthermore, with sentiment analyzers in hand, it is now possible to understand the user activities and choices. Many works have already been done for English Languages. In contrast, work done in Bengala is not significant enough. This empirical research is a little step forward to fill the void. Despite being one of the most used languages in the world, Bengali lacks in both benchmark datasets and a well furnished model for sentiment analysis. Moreover, researchers usually do not publish their dataset. The dataset that was made for this research is clearly step ahead since it will be enriched and published for research purposes. Although our model achieve quite satisfactory performance in terms of performance metrics, it could still be improved to a higher performance than now if word sense ambiguity could be removed, and word sense semantic could be embedded in the scoring of word rather than term presence or term frequency. We will try to employ semantic meaning for each word and avoid word sense ambiguity as much as possible in the future. So, using attention mechanism in CNN based deep learning models, it is possible to achieve relatively better performance in Bangla sentiment analysis. In future it will be interesting to see the business applications or sentiment analyzers for Bengali text using attention in deep CNN models. In conjunction with attention based CNN(A-CNN), a mixture of others high performing techniques can also be applied to the task of sentiment classification and opinion mining.

# References

[1] Liu, Bing. Sentiment Analysis and Opinion Mining, 5th Text Analytics Summit, Boston,June 1-2, 2009.

[2] B. Pang and L. Lee. Opinion mining and sentiment analysis Foundations and Trends R in Information Retrieval, vol. 2, no.12, pp. 1135, 2008.

[3] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques in Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, 2002, pp.7986.

[4] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, Targetdependent twitter sentiment classification in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 2011, pp. 151160.

[5] A. Kennedy and D. Inkpen, Sentiment classification of movie reviews using contextual valence shifters Computational intelligence, vol. 22, no. 2, pp. 110125, 2006.

[6] H. Cui, V. Mittal, and M. Datar, Comparative experiments on sentiment classification for online product reviews in AAAI, vol. 6, 2006, pp. 12651270.

[7] Andrea Esuli and Fabrizio Sebastiani. SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining.

[8] R. Feldman, Techniques and applications for sentiment analysis Communications of the ACM, vol. 56, no. 4, pp. 8289, 2013.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, Efficient estimation of word representations in vector space arXiv preprint arXiv:1301.3781,2013.

[10] Y. Kim, Convolutional neural networks for sentence classification arXiv preprint arXiv:1408.5882, 2014.

[11] C. dos Santos and M. Gatti, Deep convolutional neural networks for sentiment analysis of short texts in Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, 2014, pp. 6978.

[12] Y. Zhang and B. Wallace, A sensitivity analysis of (and practitioners guide to) convolutional neural networks for sentence classification arXiv preprint arXiv:1510.03820, 2015.

[13] H. Shirani-Mehr, Applications of deep learning to sentiment analysis of movie reviews in Technical Report. Stanford University, 2014.

[14] P. Ekman, An argument for basic emotions Cognition & emotion, vol. 6, no. 3-4, pp. 169200, 1992.

[15] P. K. Bhowmick, Reader perspective emotion analysis in text through ensemble based multi-label classification framework, Computer and Information Science, vol. 2, no. 4, p. 64, 2009.

[16] V. K. Singh, Sentiment analysis research on bengali language texts, International Journal of Advanced Scientific Research Development (IJASRD), vol. 02, pp. 122127, 2015.

[17] M. Al-Amin, M. S. Islam, and S. D. Uzzal, Sentiment analysis of bengali comments with word2vec and sentiment information of words, in Electrical, Computer and Communication Engineering (ECCE), International Conference on. IEEE, 2017, pp. 186190.

[18] A. Hassan, M. R. Amin, N. Mohammed, and A. Azad, Sentiment analysis on bangla and romanized bangla text (brbt) using deep recurrent models, arXiv preprint arXiv:1610.00369, 2016.

[19] D. Das and S. Bandyopadhyay, Developing bengali wordnet affect for analyzing emotion, in International Conference on the Computer Processing of Oriental Languages, 2010, pp. 3540.

[20] D. Das, S. Roy, and S. Bandyopadhyay, Emotion tracking on blogs-a case study for bengali, in International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer, 2012, pp. 447456.

[21] R. E. Jack, O. G. Garrod, and P. G. Schyns, Dynamic facial expressions of emotion transmit an evolving hierarchy of signals over time, Current biology, vol. 24, no. 2, pp. 187192, 2014.

[22] T. Pranckeviius and V. Marcinkeviius, Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification Baltic Journal of Modern Computing, vol. 5, no. 2, p. 221, 2017.

[23] H. Shirani-Mehr, Applications of deep learning to sentiment analysis of movie reviews in Technical Report. Stanford University, 2014.

[24] M. S. Islam, M. A. Islam, M. A. Hossain, & J. J. Dey. Supervised approach of sentimentality extraction from bengali facebook status in

Computer and Information Technology (ICCIT), 2016 19th International Conference on. IEEE, 2016, pp. 383387.

[25] A. K. Paul and P. C. Shill, Sentiment mining from bangla data using mutual information, in Electrical, Computer & Telecommunication Engineering (ICECTE), International Conference on. IEEE, 2016, pp. 14.

[26] Amitava Das. SentiWordNet for Bangla.

[27] Shaika Chowdhury and Wasifa Chowdhury. Performing sentiment analysis in Bangla microblog posts.

[28] K. M. Azharul Hasan, Mosiur Rahman, Badiuzzaman. Sentiment detection from Bangla text using contextual valency analysis.

[29] Qurat Tul Ain, Mubashir Ali, Amna Riaz, Amna Noureen, Muhammad Kamran, Babar Hayat, & A. Rehman. Sentiment Analysis Using Deep Learning Techniques: A Review.

[30] A. Hassan, M. R. Amin, N. Mohammed, and A. K. A. Azad. Sentiment Analysis on Bangla and Romanized Bangla Text (BRBT) using Deep Recurrent models.

[31] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate.

[32] Kelvin Xu, Jimmy Lei Ba, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention.

[33] Minh-Thang, Luong Hieu, and Pham Christopher D.Manning. Effective Approaches to Attention-based Neural Machine Translation.

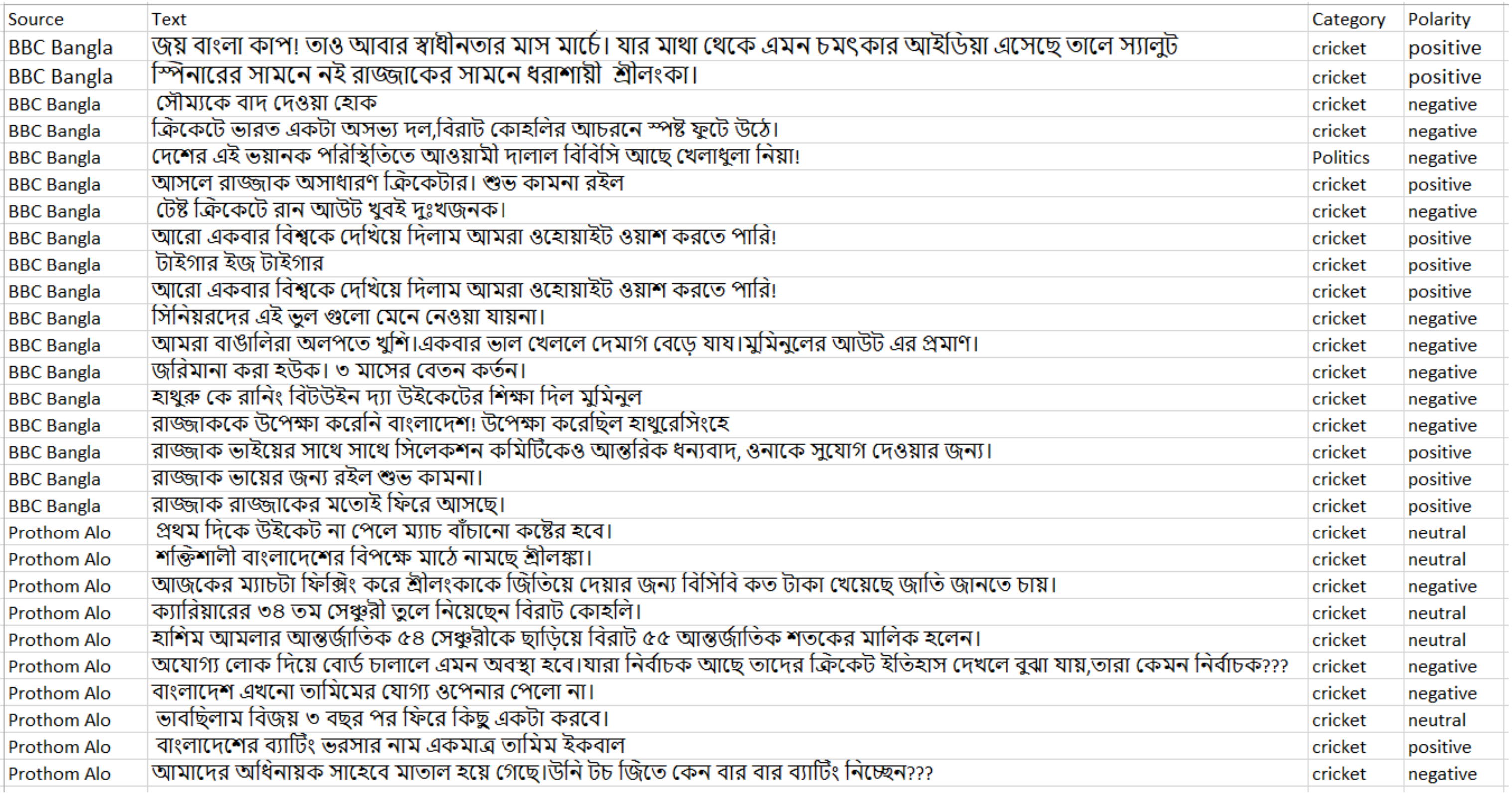
[34] Ashish Vaswani, Noam Shazeer, and Niki Parmar. Attention Is All You Need.

[35] Geoffrey Hinton, George Dahl, James Martens, and Ilya Sutskever. On the importance of initialization and momentum in deep learning.

[36] Timothy Dozat. Incorporating Nesterov Momentum into Adam.

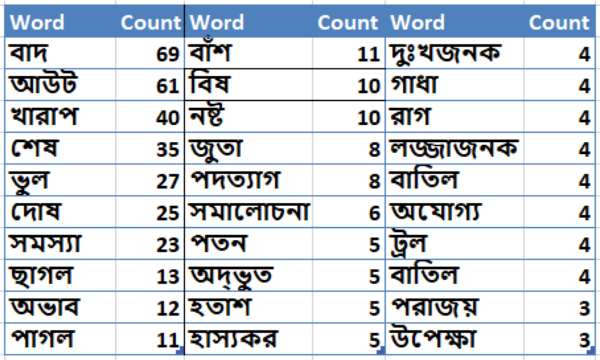
[37] Sergey Ioffe, and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv preprint arXiv:1502.03167, 2015.

# 7 Sample Reviews and Comments



| Source | Text | Category | Polarity |
|---|---|---|---|
| BBC Bangla | জয় বাংলা কাপ! তাও আবার স্বাধীনতার মাস মার্চে। যার মাথা থেকে এমন চমৎকার আইডিয়া এসেছে তালে স্যালুট | cricket | positive |
| BBC Bangla | স্পিনারের সামনে নই রাজ্জাকের সামনে ধরাশায়ী  শ্রীলংকা। | cricket | positive |
| BBC Bangla | সৌম্যকে বাদ দেওয়া হোক | cricket | negative |
| BBC Bangla | ক্রিকেটে ভারত একটা অসভ্য দল,বিরাট কোহলির আচরনে স্পষ্ট ফুটে উঠে। | cricket | negative |
| BBC Bangla | দেশের এই ভয়ানক পরিস্থিতিতে আওয়ামী দালাল বিবিসি আছে খেলাধুলা নিয়া! | Politics | negative |
| BBC Bangla | আসলে রাজ্জাক অসাধারণ ক্রিকেটার। শুভ কামনা রইল | cricket | positive |
| BBC Bangla | টেষ্ট ক্রিকেটে রান আউট খুবই দুঃখজনক। | cricket | negative |
| BBC Bangla | আরো একবার বিশ্বকে দেখিয়ে দিলাম আমরা ওহোয়াইট ওয়াশ করতে পারি! | cricket | positive |
| BBC Bangla | টাইগার ইজ টাইগার | cricket | positive |
| BBC Bangla | আরো একবার বিশ্বকে দেখিয়ে দিলাম আমরা ওহোয়াইট ওয়াশ করতে পারি! | cricket | positive |
| BBC Bangla | সিনিয়রদের এই ভুল গুলো মেনে নেওয়া যায়না। | cricket | negative |
| BBC Bangla | আমরা বাঙালিরা অলপতে খুশি।একবার ভাল খেললে দেমাগ বেড়ে যায।মুমিনুলের আউট এর প্রমাণ। | cricket | negative |
| BBC Bangla | জরিমানা করা হউক। ৩ মাসের বেতন কর্তন। | cricket | negative |
| BBC Bangla | হাথুরু কে রানিং বিটউইন দ্যা উইকেটের শিক্ষা দিল মুমিনুল | cricket | negative |
| BBC Bangla | রাজ্জাককে উপেক্ষা করেনি বাংলাদেশ! উপেক্ষা করেছিল হাথুরেসিংহে | cricket | negative |
| BBC Bangla | রাজ্জাক ভাইয়ের সাথে সাথে সিলেকশন কমিটিকেও আন্তরিক ধন্যবাদ, ওনাকে সুযোগ দেওয়ার জন্য। | cricket | positive |
| BBC Bangla | রাজ্জাক ভায়ের জন্য রইল শুভ কামনা। | cricket | positive |
| BBC Bangla | রাজ্জাক রাজ্জাকের মতোই ফিরে আসছে। | cricket | positive |
| Prothom Alo | প্রথম দিকে উইকেট না পেলে ম্যাচ বাঁচানো কষ্টের হবে। | cricket | neutral |
| Prothom Alo | শক্তিশালী বাংলাদেশের বিপক্ষে মাঠে নামছে শ্রীলঙ্কা। | cricket | neutral |
| Prothom Alo | আজকের ম্যাচটা ফিক্সিং করে শ্রীলংকাকে জিতিয়ে দেয়ার জন্য বিসিবি কত টাকা খেয়েছে জাতি জানতে চায়। | cricket | negative |
| Prothom Alo | ক্যারিয়ারের ৩৪ তম সেঞ্চুরী তুলে নিয়েছেন বিরাট কোহলি। | cricket | neutral |
| Prothom Alo | হাশিম আমলার আন্তর্জাতিক ৫৪ সেঞ্চুরীকে ছাড়িয়ে বিরাট ৫৫ আন্তর্জাতিক শতকের মালিক হলেন। | cricket | neutral |
| Prothom Alo | অযোগ্য লোক দিয়ে বোর্ড চালালে এমন অবস্থা হবে।যারা নির্বাচক আছে তাদের ক্রিকেট ইতিহাস দেখলে বুঝা যায়,তারা কেমন নির্বাচক??? | cricket | negative |
| Prothom Alo | বাংলাদেশ এখনো তামিমের যোগ্য ওপেনার পেলো না। | cricket | negative |
| Prothom Alo | ভাবছিলাম বিজয় ৩ বছর পর ফিরে কিছু একটা করবে। | cricket | neutral |
| Prothom Alo | বাংলাদেশের ব্যাটিং ভরসার নাম একমাত্র তামিম ইকবাল | cricket | positive |
| Prothom Alo | আমাদের অধিনায়ক সাহেবে মাতাল হয়ে গেছে।উনি টচ জিতে কেন বার বার ব্যাটিং নিচ্ছেন??? | cricket | negative |

Figure 16: Samples reviews and comments



| Word | Count | Word | Count | Word | Count |
|---|---|---|---|---|---|
| বাদ | 69 | বাঁশ | 11 | দুঃখজনক | 4 |
| আউট | 61 | বিষ | 10 | গাধা | 4 |
| খারাপ | 40 | নষ্ট | 10 | রাগ | 4 |
| শেষ | 35 | জুতা | 8 | লজ্জাজনক | 4 |
| ভুল | 27 | পদত্যাগ | 8 | বাতিল | 4 |
| দোষ | 25 | সমালোচনা | 6 | অযোগ্য | 4 |
| সমস্যা | 23 | পতন | 5 | ট্রল | 4 |
| ছাগল | 13 | অদ্ভুত | 5 | বাতিল | 4 |
| অভাব | 12 | হতাশ | 5 | পরাজয় | 3 |
| পাগল | 11 | হাস্যকর | 5 | উপেক্ষা | 3 |

Figure 17: Some most common negative words in dataset