Establishing a Formal Benchmarking Process for Sentiment Analysis for the Bangla Language

A K M Shahariar Azad Rabby [](https://orcid.org/0000-0003-3994-3105)1, Aminul Islam1 and Fuad Rahman2

1 Apurba Technologies, Dhaka, Bangladesh

2 Apurba Technologies, Sunnyvale, CA, USA  
rabby[@apurbatech.com](mailto:@apurbatech.com), aminul@apurbatech.com, [fuad@apurbatech.com](mailto:fuad@apurbatech.com)

**Abstract.** Tracking sentiments is a critical task in many natural language processing applications. A lot of work has been done on many leading languages in the world, such as English. However, in many languages such as Bangla, sentiment analysis is still in early development. Most of the research on this topic suffers from three key issues: (a) the lack of standardized publicly available datasets, (b) the subjectivity of the reported results, which generally manifests as a lack of agreement on core sentiment categorizations, and finally, (c) the lack of an established framework where these efforts can be compared to a formal benchmark. Thus, this seems to be an opportune moment to establish a benchmark for sentiment analysis in Bangla. With that goal in mind, this paper presents benchmark results of ten different sentiment analysis solutions on three publicly available Bangla sentiment analysis corpora. As part of the benchmarking process, we have optimized these algorithms for the task at hand. Finally, we establish and present sixteen different evaluation matrices for benchmarking these algorithms. We hope that this paper will jumpstart an open and transparent benchmarking process, one that we plan to update every two years, to help validating newer and novel algorithms that will be reported in this area in future.

**Keywords:** Sentiment Analysis, NLP, Bangla Sentiment Corpus, Annotation, Benchmarking.

1. Introduction

The explosion of information technology, especially the use of social media, has resulted in a vast amount of content that is thrown at human beings at any given moment. A lot of this content is tied to social, political, and economic interests, publishers of all of which have a vested interest in tracking whether the audience likes the content or not. For instance, data-driven trend analysis is an essential part of modern politics and advertising. Less dramatic, but equally critical applications of sentiment analysis are customer reviews on online shopping sites or opinion mining on newspapers to gauge public sentiment on national security issues, just to name a few.

Bangla is spoken as the first language by almost 200 million people worldwide, 160 million of whom hold Bangladeshi citizenship. But Natural Language Processing (NLP) development of the Bangla language is in very early stages, and there is not yet enough labeled data to work with for the language. Because of the scarcity of labeled data and standardized corpora, little work has been reported in this space.

Recently, a sentiment analysis corpus of about 10,000 sentences was made public by Apurba Technologies [1]. We searched and located two additional, albeit smaller, open-sourced datasets in this space [2]. We built ten different sentiment analysis algorithms using Machine Learning (ML), statistical modeling, and other methods. This paper benchmarks these 10 algorithms on the above-mentioned 3 annotated corpora.

The paper is arranged as follows. We begin by reviewing the existing state of the art of sentiment analysis in Bangla—which as stated already is not very rich—but the principal issue that becomes crystal clear is that whatever efforts have been reported on this topic, it is absolutely impossible to compare them since they use different datasets and almost always the datasets reported are not available to other researchers. As a natural segue from this topic, we then present how we combined all the possible sources of sentiment corpora available publicly and built a large dataset. We then move to designing 14 different matrices that form the benchmarking framework. We then describe 10 different sentiment analysis algorithms that have been reported in the literature. Although this list is not exhaustive in any sense, it does cover the majority of the work ever reported in this space. We not only implemented these algorithms, we also fine-tuned the parameters for optimizing each of these solutions. Finally, these 10 algorithms were benchmarked by the 14 different matrices identified earlier. The paper ends with a discussion on the reported work.

1. Brief Background

There are three classification levels in sentiment analysis: document-level, sentence-level, and aspect-level. In the document level, overall sentiment is assessed based on the complete text. The sentence-level analysis aims to classify sentiment expressed in each sentence. The first step is to identify whether the sentence is subjective or objective. If the sentence is subjective, sentence-level analysis will determine whether the sentence expresses positive or negative opinions [3]. In aspect-based sentiment analysis, sentiments are assessed on aspects or points of view of a topic, especially with multi-clausal sentences. For the rest of this paper, we will exclusively focus on sentence-level sentiment analysis.

Machine learning techniques for sentiment analysis are getting better, especially for vector representation models, where some of these models can extract semantics that helps to understand the intent of the messages [4]. Many machine learning and deep learning techniques have been reported for identifying and classifying sentiment polarity in a document or sentence. Existing research demonstrates that that Long Short-Term Memory networks (LSTMs) are capable of learning the context and inherent meaning of a word and provide more accurate results for sentiments [5]. Classification algorithms such as Random Forest, Decision Tree Classifier, and the k-nearest neighbors (KNN) algorithm, are suitable for classification based on feature sets. Naive Bayes works based on Bayes’ theorem of a probability distribution. Convolutional Neural Networks (CNNs), a commonly used tool in deep learning, works well for sentiment analysis as its standard architecture can map the sentences of variable length into

**Table 1.** Bangla Sentiment analysis - Previous work

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Year | 2017 | 2014 | 2017 | 2016 | 2018 | 2017 |
| Acc | 75.5% | SVM 88%  MaxEnt 88% | Lr:75.91%  SVM: 79.56% Tree:76.64% | 78% | 83.79% | (MSE)  0.0529 |
| Availability | Not publicly available | Not publicly available | Not publicly available | Not publicly available | Not publicly available | Not publicly available |
| Size | 15,000  Comments | 1,300 tweets | 15,325 headlines | 10,000 Bangla  text samples | 1,899,094 Sentences  23,506,262 Words, 394,297 que Words | NA |
| Dataset | Self-collected comments data | Bangla Tweets | Self-collected news head-line data. | Self-collected | Bangla Web Crawl  Bangla Sentiment Dataset | Bangla tweets using Twitter APIs. |
| Method | word2vec and Sentiment extraction of words | Support Vector Machine  (SVM) and Maximum Entropy (MaxEnt). | Support Vector Machine, Logistic Regression, etc. | LSTM, using two types of loss functions – binary cross-entropy and categorical cross-entropy | Word embedding methods Word2vec Skip-Gram and Continuous Bag of Words with an addition Word to Index model for SA in Bangla language | Fuzzy rules to represent semantic rules that are simple but greatly influence the actual polarity of the sentences |
| Author | Md. Al- Amin, Md. Saiful Islam, Shapan Das Uzzal | Shaika Chow-dhury, Wasifa Chowdhury | Mohammad Samman Hoss-ain, Israt Jahan Jui, Afia Zahin Suzana | Asif Hassan, Mohammad Rashedul Amin, Abul Kalam Al Azad, Nabeel Mohammed | Sakhawat Hosain Sumit, Md. Zakir Hossan, Tareq Al Muntasir and Tanvir Sourov | Md. Asimuzzaman, Pinku Deb Nath, Farah Hossain, Asif Hossain, Rashedur M. Rahman |
| Paper Title | Sentiment Analysis of Bengali Comments with Word2Vec and Sentiment Information of Words [7] | Performing Sentiment Analysis in Bangla Microblog Posts [8] | Sentiment Analysis for Bengali Newspaper Headlines [9] | Sentiment Analysis on Bangla and Romanized Bangla Text (BRBT) using Deep Recurrent models. [10] | Exploring Word Embedding for Bangla Sentiment Analysis [11] | Sentiment Analysis of Bangla Microblogs Using Adaptive Neuro Fuzzy System [12] |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Year | 2019 | 2019 | 2019 | 2017 | 2016 | 2018 |
| Acc | Above 90% | 84.4% | 87% | 99.87% | 88.54% | 65.97% three, 54.24% five labels |
| Availability | Not publicly available | Not publicly available | Not publicly available | Not publicly available | Not publicly available | Not publicly available |
| Size | 7,500 Bangla sentences | 9337 post | 1050 Bangla texts | 850 Bangla comments from different sources | 68356 translated reviews | 15689 YouTube comment |
| Dataset | Self-collected | Dataset from Hasaan, Asif, et al. | Self-collected | Self-collected | Generated from Amazon's Watches English dataset. | Self-collected YouTube comment |
| Method | Naïve Bayes Classification Algorithm and Topical approach to extract the emotion. | Long Short-term Memory (LSTM) Neural Networks for analyzing negative sentences in Bangla. | Random Forest Classifier to classify sentiments. | The model is generated by a neural network variance called Convolutional Neural Network | Mutual Information (MI) for the feature selection process and also used Multinomial Naive Bayes (MNB) for the classification | Deep learning based modelsto classify a Bangla sentence with a three-class |
| Author | Rashedul Amin Tuhin, Bechitra Kumar Paul, Faria Nawrine, Mahbuba Akter, Amit Kumar Das | Abdul Hasib Uddin; Sumit Kumar Dam; Abu Shamim Mohammad ArifChakrabarty | Nusrath Tabassum; Muhammad Ibrahim Khan | Md. Habibul Alam ; Md-Mizanur Rahoman ; Md. Abul Kalam Azad | Animesh Kumar Paul; Pintu Chandra Shill | Nafis Irtiza Tripto ; Mohammed Eunus Ali |
| Paper Title | An Automated System of Sentiment Analysis from Bangla Text using Supervised Learning Techniques [13] | Extracting Severe Negative Sentence Pattern from Bangla Data via Long Short-term Memory Neural Network [14] | Design an Empirical Framework for Sentiment Analysis from Bangla Text using Machine Learning [15] | Sentiment analysis for Bangla sentences using convolutional neural network [16] | Sentiment mining from Bangla data using mutual information [17] | Detecting Multilabel Sentiment and Emotions from Bangla YouTube Comments [18] |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Year | 2016 | 2018 | 2019 | 2018 | 2016 | 2019 |
| Acc | 83% | 89.271% | 70% | 80% | 73% | 80.48% |
| Availability | Not publicly available | Not publicly available | Not publicly available | Not publicly available | Not publicly available | Not publicly available |
| Size | 1500 short Bangla comment | 9,500 comments | 201 Comments | 45,000 | 9000 words | 1000 restaurant reviews |
| Dataset | Collected from various social sites | Collected from different source | Collected from YouTube | Collected from Facebook using Facebook graph api | Collected from Facebook Group | Self-collected |
| Method | Used Tf.Idf to come out a better solution and give more accurate result by extracting different feature | One vector containing more than one words using N-gram | A backtracking algorithm used, where the heart of this approach is a sentiment lexicon | Represent Bangla sentence based on characters and extract information from the characters using an RNN | Naïve Bayes and Dictionary Based Approach used to Lexicon Based Sentiment Analysis | Multinomial Na ̈ıve Bayes used for sentiment analysis. |
| Author | Muhammad Mahmudun Nabi, Md. Altaf, Sabir Ismail | SM Abu Taher; Kazi Afsana Akhter ; K.M. Azharul Hasan | Tapasy Rabeya ; Narayan Ranjan Chakraborty ; Sanjida Ferdous ; Manoranjan Dash ; Ahmed Al Marouf | Mohammad Salman Haydar ; Mustakim Al Helal ; Syed Akhter Hossain | Sanjida Akter; Muhammad Tareq Aziz | Omar Sharif; Mohammed Moshiul Hoque; Eftekhar Hossain |
| Paper Title | Detecting Sentiment from Bangla Text using Machine Learning Technique and Feature Analysis [19] | N-Gram Based Sentiment Mining for Bangla Text Using Support Vector Machine [20] | Sentiment Analysis of Bangla Song Review- A Lexicon Based Backtracking Approach [21] | Sentiment Extraction from Bangla Text: A Character Level Supervised Recurrent Neural Network Approach [ 22] | Sentiment analysis on the Facebook group using lexicon-based approach [23] | Sentiment Analysis of Bengali Texts on Online Restaurant Reviews Using Multinomial Naïve Bayes [24] |

sentences of fixed size scattered vectors [6].[[1]](#footnote-1)

Table 1 shows the state of the art of Bangla sentiment analysis research.

One observation that is painfully plain in this table is that all of the authors of these papers spent valuable time in building and annotating their own datasets. What is even more alarming is that none of these datasets were then made publicly available. This has made it impossible to compare the validity and relative strengths or weaknesses for any of these solutions, making the task of establishing a benchmark framework impossible.

1. Dataset

In this research, we used three different datasets. The first dataset is our own, that we previously published [1], representing the largest open-access sentiment analysis dataset for Bangla, with 9,630 samples. The second is the ABSA Sports dataset [2], with 2,979 samples. The third and final dataset [2] is the ABSA Restaurant dataset, with 2,059 samples. All datasets have three sentiment categorizations: positive, negative, and neutral. For simplicity, we excluded all of the neutral data from our datasets. After eliminating the neutral samples, the Apurba, ABSA Sports, and ABSA Restaurant datasets have 7,293, 2,718, and 1,808 positive and negative samples, respectively.

The proposed benchmarking system has four stages: data collection, data pre-processing, training, and evaluation.

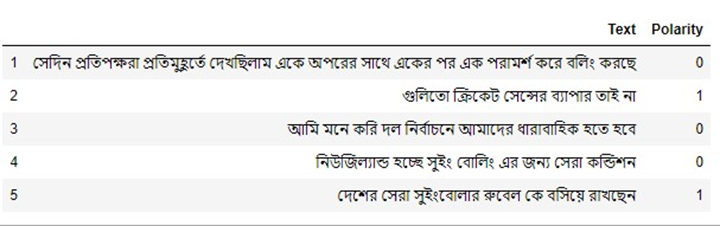
* 1. Dataset Collection

The Apurba Dataset was collected from a popular online news portal “Prothom Alo” (প্রথম আলো), tagged manually and checked twice for validation. Also, the dataset is open-source for all types of non-commercial usage, intended for educational and research use. The other two datasets can easily be obtained from GitHub. We also merged these three datasets and made a mixed dataset.

* 1. Data Pre-Processing

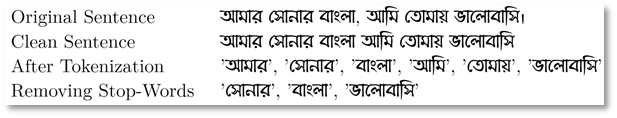
Data cannot be used as-is in most machine learning algorithms—it needs to be processed before anything else can be done.

In this research, we took the text and annotated sentiment values. We excluded the neutral samples and represent the positive class with 0 and the negative level with 1. We removed all unnecessary characters, including punctuation, URL, extra white space, emoticons, symbols, pictographs, transport and maps symbol, iOS flags, digits, and 123 other characters, and so forth. After all these steps, the preprocessed dataset looks as shown in Fig. 1.



**Fig. 1.** Processed Dataset Sample

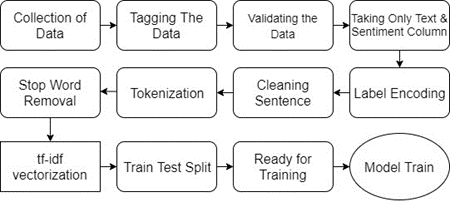
Tokenization is a task of separating the given sentence sequence each word, which are then known as tokens. Tokenizers accomplish this task by locating word boundaries. The ending point of a word and the beginning of the next word are our word boundaries. We tokenize each sentence based on white space. The next step is removing stop-words, which are commonly used words (such as "a" or “and”) which our algorithm ignores. Fig. 2 shows a typical example of these steps.



**Fig. 2.** Pre-processing steps

We then prepare a “term frequency-inverse document frequency” vectorization, commonly known as tf-idf, that creates a sparse matrix. The sparse matrix contains a vector representation of our data. The tf-idf output is used as a weighting factor to measure how important a word is in a document in a collection of given corpus.

Then we split our data into two portions, 80% is for training purposes and 20% for test the model performance. Fig. 3 shows the flowchart of these pre-processing steps.



**Fig. 3.** Flowchart of the pre-processing steps

1. Benchmarking Indices

Sensitivity analysis is a model that determines how target variables are affected based on changes in other variables known as input variables. This model, also referred to as what-if or simulation analysis, is a way to predict the outcome of a decision given a certain range of variables. By creating a given set of variables, an analyst can determine how changes in one variable affect the outcome. We have used a set of universally standardized indices for validating the algorithms including Confusion Matrix (CM), True Positive Rate (TPR), True Negative Rate (TNR), False Negative Rate (FNR), False Positive Rate (FPR), Positive Predictive Value (PPV), Negative Predictive Value (NPV), False Discovery Rate (FDR), False Omission Rate (FOR), Accuracy (ACC), F1 Score, R2 Score, Receiver Operating Characteristic (ROC), and Area Under the Curve (AUC) [25][26][27][28][29].

1. Sentiment Analysis Algorithms

We used ten different algorithms, which are: Multinomial Naive Bayes, Bernoulli Naive Bayes, Logistic Regression, Decision Tree Classifier, K-Nearest Neighbors Classifier (KNN), Support Vector Machine (SVM), Ada-Boost Classifier, Extreme Gradient Boosting (XGBoost) and long short-term memory (LSTM). LSTM achieves the best performance among them. We used K-fold cross-validation and Grid Search to find the best parameters for all of our algorithms.

* 1. Multinomial Naive Bayes

Multinomial Naive Bayes estimates the conditional probability of a particular word given a class as the relative frequency of term *t* in samples belonging to class *c*. Multinomial Naive Bayes simply assumes a multinomial distribution for all the pairs, which seems to be a reasonable assumption in some cases, especially for word counts in documents.

* 1. Bernoulli Naive Bayes

The Bernoulli Naive Bayes classifier assumes that all our features are binary—that they take only two values. This is similar to the Multinomial Naive Bayes, but the predictors are Boolean variables. The parameters that we use to predict the class variable take up only values, yes or no, for example, if a word occurs in the text or not.

* 1. Logistic Regression

Logistic Regression is the primary form of statistical method to find a binary dependent variable. In this technique, models try to find the probability of each class. Logistic Regression is a ML classification algorithm that used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as either 1 (yes, success, etc.) or 0 (no, failure, etc.). In other words, the logistic regression model predicts P (Y = 1) as a function of X.

* 1. Random Forest

A forest usually consists of lots of trees; in a random forest, a large number of individual decision trees operated like ensemble. Every decision tree gives their vote to a particular class, and the class that gets the most votes is selected for model prediction.

* 1. Decision Tree Classifier

A decision tree is the purest form of the classification algorithm. A decision tree contains nodes, edges, and leaf nodes for classifications. Decision trees consist of: (a) nodes to test for the value of a particular attribute, (b) edges/branches to correspond to the outcome of a test and connect to the next node or leaf, and (c) leaf nodes which are terminal nodes that predict the outcome (such as class labels or class distribution).

* 1. KNN Classifier

In the field of AI, the k-nearest neighbors’ algorithm is a non-parametric technique used for classifications. It is easy to implement, but the major problem is that it becomes slow as the amount of data increases.

* 1. SVM Classifier

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm builds an optimal hyperplane that separates new examples into constituent classes. In two-dimensional space, this hyperplane is a line dividing a plane into two parts wherein each class lies on either side.

* 1. Ada-Boost Classifier

The general idea behind boosting methods is to train predictors sequentially, each trying to correct its predecessor. The basic concept behind Ada-boost is to set the weights of classifiers and to train the data samples in each iteration such that it ensures accurate predictions, even for unusual observations.

* 1. XGBoost

XGBoost is a decision-tree-based ensemble ML algorithm that uses a gradient boosting framework. XGBoost Gradients are fantastic models because they can increase accuracy over a traditional statistical or conditional model and can apply themselves quite well to the two primary types of targets.

* 1. LSTM

Long Short-Term Memory (LSTM) networks are a modified version of recurrent neural networks that enables the memory storage of past data. RNN's vanishing gradient problem is solved here. LSTM is ideal for classifying, analyzing, and forecasting time series owing to uncertain time lags.

1. Performance
   1. Multinomial Naive Bayes

We found that if the alpha value set to 0.9, Multinomial Naive Bayes gets a maximum of 76.65 % accuracy. Table 2 shows the performance of Multinomial Naive Bayes. And Table 3 shows the sensitivity analysis for this algorithm.

**Table 2.** Multinomial Naive Bayes performance

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | CM | ACC | ROC AUC |
| Apurba | [342, 264]  [195, 658] | 68.54% | 73.05% |
| ABSA Sports | [[ 38, 72]  [ 55, 379]] | 76.65% | 67.93% |
| ABSA Restaurant | [225, 37]  [ 52, 48] | 75.41% | 72.64% |
| All Dataset | [ 566, 466]  [ 271, 1061] | 68.82% | 73.05% |

**Table 3.** Sensitivity Analysis of Multinomial Naive Bayes

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | TPR | TNR | FNR | FPR | PPV | NPV | FDR | FOR | F1 |
| Apurba | 77.14 | 56.44 | 22.86 | 43.56 | 71.37 | 63.69 | 28.63 | 36.31 | 74.14 |
| ABSA Sports | 87.33 | 34.55 | 12.67 | 65.45 | 84.04 | 40.86 | 15.96 | 59.14 | 85.65 |
| ABSA Restaurant | 48.0 | 85.88 | 52.0 | 14.12 | 56.47 | 81.23 | 43.53 | 18.77 | 51.89 |
| All Dataset | 79.65 | 54.84 | 20.35 | 45.16 | 69.48 | 67.62 | 30.52 | 32.38 | 74.22 |

* 1. Bernoulli Naive Bayes

For all datasets, we found the alpha value of 0.8 got the best performance. Table 4 shows the performance, and Table 5 shows the sensitivity analysis for Bernoulli Naive Bayes.

**Table 4.** Bernoulli Naive Bayes performance

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | CM | ACC | ROC AUC |
| Apurba | [342, 264]  [195, 658] | 69.16% | 73.27% |
| ABSA Sports | [ 23, 87]  [ 20, 414] | 80.33% | 70.50% |
| ABSA Restaurant | [225, 37]  [ 52, 48] | 71.82% | 73.64% |
| All Dataset | [ 566, 466]  [271,1061] | 67.98% | 73.54% |

**Table 5.** Sensitivity Analysis of Bernoulli Naive Bayes

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | TPR | TNR | FNR | FPR | PPV | NPV | FDR | FOR | F1 |
| Apurba | 78.19 | 56.44 | 21.81 | 43.56 | 71.64 | 64.77 | 28.36 | 35.23 | 74.78 |
| ABSA Sports | 92.86 | 23.64 | 7.14 | 76.36 | 82.75 | 45.61 | 17.25 | 54.39 | 87.51 |
| ABSA Restaurant | 25.0 | 89.69 | 75.0 | 10.31 | 48.08 | 75.81 | 51.92 | 24.19 | 32.89 |
| All Dataset | 80.56 | 51.74 | 19.44 | 48.26 | 68.3 | 67.34 | 31.7 | 32.66 | 73.92 |

* 1. Logistic Regression

Table 6 shows the performance, and Table 7 shows the sensitivity analysis for Logistic Regression.

**Table 6.** Logistic Regression performance

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | CM | ACC | ROC AUC |
| Apurba | [338, 268]  [203, 650] | 67.72% | 72.51% |
| ABSA Sports | [ 23, 87]  [ 20, 414] | 80.33% | 70.50% |
| ABSA Restaurant | [237, 25]  [ 66, 34] | 74.86% | 75.39% |
| All Dataset | [ 566, 466]  [ 276, 1056] | 68.61% | 74.30% |

**Table 7.** Sensitivity Analysis of Logistic Regression

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | TPR | TNR | FNR | FPR | PPV | NPV | FDR | FOR | F1 |
| Apurba | 76.2 | 55.78 | 23.8 | 44.22 | 70.81 | 62.48 | 29.19 | 37.52 | 73.4 |
| ABSA Sports | 95.39 | 20.91 | 4.61 | 79.09 | 82.63 | 53.49 | 17.37 | 46.51 | 88.56 |
| ABSA Restaurant | 34.0 | 90.46 | 66.0 | 9.54 | 57.63 | 78.22 | 42.37 | 21.78 | 42.77 |
| All Dataset | 79.28 | 54.84 | 20.72 | 45.16 | 69.38 | 67.22 | 30.62 | 32.78 | 74.0 |

* 1. Random Forest

Table 8 shows the performance, and Table 9 shows the sensitivity analysis for the Random Forest model.

**Table 8.** Random Forest performance

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Dataset | CM | ACC | ROC AUC | F1 | Precision | Recall |
| Apurba | [340, 266]  [309, 544] | 60.59% | 65.56% | 65.42% | 67.16% | 63.77% |
| ABSA Sports | [ 47, 63]  [ 41, 393] | 80.88% | 73.30 | 88.31% | 86.18% | 90.55% |
| ABSA Restaurant | [240, 22]  [ 75, 25] | 73.20% | 70.00% | 34.01% | 53.19% | 25% |
| All Dataset | [629, 403]  [387, 945] | 66.58% | 71.36% | 70.52% | 70.10% | 70.94% |

**Table 9.** Sensitivity Analysis of Random Forest

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | TPR | TNR | FNR | FPR | PPV | NPV | FDR | FOR | F1 |
| Apurba | 64.71 | 59.08 | 35.29 | 40.92 | 69.0 | 54.32 | 31.0 | 45.68 | 66.79 |
| ABSA Sports | 88.71 | 43.64 | 11.29 | 56.36 | 86.13 | 49.48 | 13.87 | 50.52 | 87.4 |
| ABSA Restaurant | 28.0 | 91.98 | 72.0 | 8.02 | 57.14 | 77.0 | 42.86 | 23.0 | 37.58 |
| All Dataset | 68.77 | 62.02 | 31.23 | 37.98 | 70.03 | 60.61 | 29.97 | 39.39 | 69.39 |

* 1. Decision Tree Classifier

Table 10 shows the performance, and Table 11 shows the sensitivity analysis of the Decision Tree Classifier.

**Table 10.** Decision Tree performance

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Dataset | CM | ACC | ROC AUC | F1 | Precision | Recall |
| Apurba | [316, 290]  [341, 512] | 56.75% | 57.11% | 61.87% | 63.84% | 60.02% |
| ABSA Sports | [ 49, 61]  [ 73, 361] | 75.37% | 65.88% | 84.34% | 85.55% | 83.18% |
| ABSA Restaurant | [216, 46]  [ 55, 45] | 72.10% | 65.13% | 47.12% | 49.45% | 45% |
| All Dataset | [601, 431]  [492, 840] | 60.96% | 60.99% | 64.54% | 66.09% | 63.06% |

**Table 11.** Sensitivity Analysis of Decision Tree

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | TPR | TNR | FNR | FPR | PPV | NPV | FDR | FOR | F1 |
| Apurba | 58.85 | 55.61 | 41.15 | 44.39 | 65.11 | 48.98 | 34.89 | 51.02 | 61.82 |
| ABSA Sports | 83.18 | 47.27 | 16.82 | 52.73 | 86.16 | 41.6 | 13.84 | 58.4 | 84.64 |
| ABSA Restaurant | 41.0 | 82.06 | 59.0 | 17.94 | 46.59 | 78.47 | 53.41 | 21.53 | 43.62 |
| All Dataset | 63.21 | 60.95 | 36.79 | 39.05 | 67.63 | 56.21 | 32.37 | 43.79 | 65.35 |

* 1. K-NN Classifier

Table 12 shows the performance, and Table 13 shows the sensitivity analysis of KNN.

**Table 12.** K-NN Classifier performance

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | CM | ACC | ROC AUC |
| Apurba | [293, 313]  [308, 545] | 57.44% | 57.42% |
| ABSA Sports | [ 25, 85]  [ 29, 405] | 79.04% | 66.31% |
| ABSA Restaurant | [236, 26]  [ 77, 23] | 71.55% | 63.69% |
| All Dataset | [500, 532]  [368, 964] | 61.92% | 63.10% |

**Table 13.** Sensitivity Analysis of KNN

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | TPR | TNR | FNR | FPR | PPV | NPV | FDR | FOR | F1 |
| Apurba | 63.89 | 48.35 | 36.11 | 51.65 | 63.52 | 48.75 | 36.48 | 51.25 | 63.71 |
| ABSA Sports | 93.32 | 22.73 | 6.68 | 77.27 | 82.65 | 46.3 | 17.35 | 53.7 | 87.66 |
| ABSA Restaurant | 23.0 | 90.08 | 77.0 | 9.92 | 46.94 | 75.4 | 53.06 | 24.6 | 30.87 |
| All Dataset | 72.37 | 48.45 | 27.63 | 51.55 | 64.44 | 57.6 | 35.56 | 42.4 | 68.18 |

* 1. SVM Classifier

Table 15 shows the performance, and Table 14 shows the sensitivity analysis of the SVM.

**Table 14.** SVM performance

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | CM | ACC | ROC AUC |
| Apurba | [293, 313]  [308, 545] | 66.83% | 72.24% |
| ABSA Sports | [ 25, 85]  [ 29, 405] | 70.77% | 69.37% |
| ABSA Restaurant | [236, 26]  [ 77, 23] | 69.89% | 72.87% |
| All Dataset | [500, 532]  [368, 964] | 67.94% | 73.95% |

**Table 15.** Sensitivity Analysis of SVM

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | TPR | TNR | FNR | FPR | PPV | NPV | FDR | FOR | F1 |
| Apurba | 69.75 | 62.71 | 30.25 | 37.29 | 72.47 | 59.56 | 27.53 | 40.44 | 71.09 |
| ABSA Sports | 75.81 | 50.91 | 24.19 | 49.09 | 85.9 | 34.78 | 14.1 | 65.22 | 80.54 |
| ABSA Restaurant | 62.0 | 72.9 | 38.0 | 27.1 | 46.62 | 83.41 | 53.38 | 16.59 | 53.22 |
| All Dataset | 70.35 | 64.83 | 29.65 | 35.17 | 72.08 | 62.88 | 27.92 | 37.12 | 71.2 |

* 1. Ada-Boost Classifier

We got the best accuracy for Ada-Boost if the number of the estimator set to 50. Table 16 shows the performance, and Table 17 shows the sensitivity analysis of the Ada-Boost Classifier.

**Table 16.** ADA Boost performance

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | CM | ACC | ROC AUC |
| Apurba | [293, 313]  [308, 545] | 64.22% | 65.92% |
| ABSA Sports | [ 25, 85]  [ 29, 405] | 79.42% | 66.74% |
| ABSA Restaurant | [236, 26]  [ 77, 23] | 73.20% | 69.38% |
| All Dataset | [500, 532]  [368, 964] | 65.44% | 70.44% |

**Table 17.** Sensitivity Analysis of ADA Boost

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | TPR | TNR | FNR | FPR | PPV | NPV | FDR | FOR | F1 |
| Apurba | 82.77 | 38.12 | 17.23 | 61.88 | 65.31 | 61.11 | 34.69 | 38.89 | 73.01 |
| ABSA Sports | 96.77 | 11.82 | 3.23 | 88.18 | 81.24 | 48.15 | 18.76 | 51.85 | 88.33 |
| ABSA Restaurant | 18.0 | 93.89 | 82.0 | 6.11 | 52.94 | 75.0 | 47.06 | 25.0 | 26.87 |
| All Dataset | 82.88 | 42.93 | 17.12 | 57.07 | 65.21 | 66.02 | 34.79 | 33.98 | 72.99 |

* 1. XGBoost

Table 18 shows the performance, and Table 19 shows the sensitivity analysis of XGBoost.

**Table 18.** XGBoost performance

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | CM | ACC | ROC AUC |
| Apurba | [291, 315]  [140, 713] | 68.81% | 6580 |
| ABSA Sports | [ 15, 95]  [ 16, 418] | 79.60% | 54.97% |
| ABSA Restaurant | [244, 18]  [ 67, 33] | 76.52% | 63.06% |
| All Dataset | [ 490, 542]  [ 185, 1147] | 69.25% | 66.80% |

**Table 19.** Sensitivity Analysis of XGBoost

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | TPR | TNR | FNR | FPR | PPV | NPV | FDR | FOR | F1 |
| Apurba | 83.59 | 48.02 | 16.41 | 51.98 | 69.36 | 67.52 | 30.64 | 32.48 | 75.81 |
| ABSA Sports | 96.31 | 13.64 | 3.69 | 86.36 | 81.48 | 48.39 | 18.52 | 51.61 | 88.28 |
| ABSA Restaurant | 33.0 | 93.13 | 67.0 | 6.87 | 64.71 | 78.46 | 35.29 | 21.54 | 43.71 |
| All Dataset | 86.11 | 47.48 | 13.89 | 52.52 | 67.91 | 72.59 | 32.09 | 27.41 | 75.94 |

* 1. LSTM

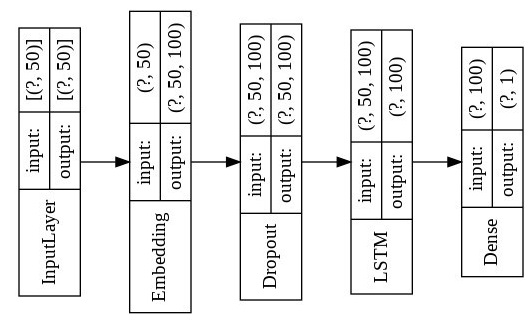
In word2vec [32], vector representations help to get a closer relationship among the words. Deep learning models such as LSTMs can remember important information across long stretches of sequences [33]. For semantic understanding or ‘meaning’ that based on context, it is important to get the actual sentiment of a sentence [4]. Hence LSTM model with word2vec has been implemented to get the results over the newly published corpora. Here are the implementation details:

* Word Embedding using vord2vec
* Window size: 2
* Minimum word count frequency is 4 (ignored lower than 4)
* The dimensionality of the word vectors: 100
* Embedding layer dropout: 50
* LSTM layer dropout: 20
* Recurrent dropout: 20
* The dimensionality of the output space 100
* Activation function: Sigmoid
* Optimizer: Adam
* Loss function: Binary cross-entropy
* Number of Epoch: 10
* Batch Size: 100

Table 20 shows the performance, and Table 21 shows the sensitivity analysis of the datasets. For the ABSA dataset, it doesn’t work well for the lack of enough data in both classes. So, the model was biased for those two ABSA datasets. Fig 4 is showing the proposed LSTM model.

**Table 20.** LSTM performance

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | CM | ACC | ROC AUC |
| Apurba | [361, 245]  [175, 678] | 69.52% | 69.53% |
| ABSA Sports | [ 0, 110]  [ 0, 434] | 79.77% | 50% |
| ABSA Restaurant | [262, 0]  [100, 0] | 72.38% | 50% |
| All Dataset | [ 579, 453]  [ 181, 1151] | 73.18% | 71.26% |



**Fig. 4.** Proposed LSTM Architecture

**Table 21.** SenSSsitivity Analysis of LSTM

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | TPR | TNR | FNR | FPR | PPV | NPV | FDR | FOR | F1 |
| Apurba | 79.25 | 60.56 | 20.75 | 39.44 | 73.88 | 67.46 | 26.12 | 32.54 | 76.47 |
| ABSA Sports | 100 | 0 | 0 | 100 | 79.78 | - | 20.22 | - | - |
| ABSA Restaurant | 0 | 100 | 100 | 0 | - | 72.38 | - | 27.62 | - |
| All Dataset | 82.81 | 62.5 | 17.19 | 37.5 | 74.03 | 73.80 | 25.97 | 26.20 | 78.17 |

1. Discussion

In this section, we will benchmark the ten algorithms. Table 22 shows the comparison of all the algorithms on all the datasets.

The algorithms are sorted based on their performance on the merged dataset. According to this evaluation, LSTM performs the best, followed by XGBoost and Multinomial Naive Bayes and so forth.

**Table 22.** Benchmark comparison - 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Acc Apurba | Acc Sports | Acc Restaurant | Acc All Data |
| LSTM | 69.52% | 79.77% | 72.38% | 73.18% |
| XGBoost | 68.81% | 79.60% | 76.52% | 69.25% |
| Multinomial Naive Bayes | 68.54% | 76.65% | 75.42% | 68.82% |
| Logistic Regression | 67.72% | 80.33% | 74.86% | 68.61% |
| Bernoulli Naive Bayes | 69.16% | 80.33% | 71.82% | 67.98% |
| SVM | 66.83% | 70.77% | 69.89% | 67.94% |
| Random Forest | 60.59% | 80.88% | 73.20% | 66.58% |
| ADA Boost | 64.22% | 79.42% | 73.20% | 65.44% |
| K-NN Classifier | 57.44% | 79.04% | 71.55% | 61.92% |
| Decision Tree Classifier | 56.75% | 75.37% | 72.10% | 60.96% |

Note that although LSTM performs best on the combined dataset, it was beaten by Random Forest on the Sports and by XGBoost on the Restaurant datasets, respectively, as noted by the highlighted cells in Table 22. Another point to note is that Bernoulli Naive Bayes is twice in the second-best position: on the Apurba and the Sports datasets, as indicated by the gray cells in Table 22.

To rank these algorithms based on how consistent they are, we start by assigning 1, 2, … 10 positions for each dataset, and then adding up their ranks on each dataset. The algorithm with the smallest sum can be ranked as most consistent, assuming the degree of difficulty of each dataset is the same, which, admittedly, we cannot know for sure. But it still gives us a ‘sense’ of how they perform over a range of different problem domains. Table 23 shows this revised ranking. This indicates that LSTM and XGBoost are tied in the first place, followed by another tie between Multinomial Naive Bayes and Logistic Regression. Decision Tree Classifier is again at the bottom of this table.

**Table 23.** Benchmark comparison - 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Algorithm | Accuracy  Apurba | Accuracy  Sports | Accuracy  Restaurant | Accuracy  All Data | Sum of Rankings | Overall Ranking |
| LSTM | 1 | 3 | 5 | 1 | 10 | 1st |
| XGBoost | 3 | 4 | 1 | 2 | 10 | 1st |
| Multinomial Naive Bayes | 4 | 7 | 2 | 3 | 16 | 2nd |
| Logistic Regression | 5 | 2 | 3 | 4 | 14 | 2nd |
| Bernoulli Naive Bayes | 2 | 2 | 7 | 5 | 16 | 3rd |
| SVM | 6 | 9 | 9 | 6 | 30 | 6th |
| Random Forest | 8 | 1 | 4 | 7 | 20 | 4th |
| ADA Boost | 7 | 5 | 4 | 8 | 24 | 5th |
| K-NN Classifier | 9 | 6 | 8 | 9 | 32 | 7th |
| Decision Tree Classifier | 10 | 8 | 6 | 10 | 34 | 8th |

Since LSTM seems to be leading the ranking on both tables, we should take a closer look at this algorithm. LSTM is a deep learning algorithm. Therefore, it has a different way of learning from data. The other six models are classification algorithms using various types of features. As described earlier, LSTM learns the context or semantic meaning from word2vec, but the rest of the models work on the frequency of a given word from encoded vector representation. As the dataset contains only about 12,000 records, this is not enough for getting consistent and accurate output, especially for LSTM, as it is learning the context or semantic lexicon. It needs more data to perform better. We have tested the LSTM model by parameter tuning, input shuffling, and changing the input size. We found that it sometimes provides very different outputs for small changes in the value of the parameters.

1. Conclusion and Future Work

This paper presents a detailed benchmarking of ten sentiment-analysis algorithms on three publicly available Bangla datasets. One of the core issues that we face in Bangla natural language processing research is the unavailability of standard datasets. In other languages, such as English or Chinese, this is not a concern. The absence of a standard, publicly available dataset means that every researcher has to first collect and label the data before any training can take place. And since each new algorithm is evaluated on a different dataset, it is also virtually impossible to compare the different approaches in terms of their accuracy and quality. We hope that this paper will alleviate those problems to some degree. Since we have fine-tuned the algorithms for these particular datasets, researchers in the future can improve on these algorithms by comparing their performance against these benchmarked datasets, which will aid in the overall improvement in the development of NLP tools for Bangla.

One of the essential factors in sentiment analysis that has not been addressed in this paper is multi-aspect sentence evaluation. In a sentence, there might be multiple clauses, and different clauses may have different sentiments. For example, examine the following quote: “Sakib’s batting was good, but he did not bowl well.” Here, we need to take the sentiment based the aspects of batting and bowling. The same goes for customer reviews: a product may be bad or good from different perspectives. So, a future task would be to extend these benchmarking models for aspect-based sentiment analysis. For sentiment analysis, there are some smarter and more complicated models, such as CNN-LSTM, where the dimensional approach can provide more fine-grained sentiment analysis [14]. We decided not to include those models since we wanted to start the benchmarking with the fundamental, commonly used, algorithms, especially within the nascent Bangla NLP domain. In the next iteration of this research, we plan to include some of these more advanced models. Finally, the size of the datasets used in this benchmarking is still minimal. We hope that other researchers will come forward and fill this gap by publicly offering larger labeled datasets for Bangla sentiment analysis.

# References

1. Rahman, F., Khan, H., Hossain, Z., Begum, M., Mahanaz, S., Islam, A., & Islam, A. (2020). An Annotated Bangla Sentiment Analysis Corpus. 2019 International Conference on Bangla Speech and Language Processing (ICBSLP).
2. Rahman, M., & Kumar Dey, E. (2018). Datasets for aspect-based sentiment analysis in Bangla and its baseline evaluation. *Data*, *3*(2), 15.
3. W. Medhat, A. Hassan, and H. Korashy, “Sentiment analysis algorithms and applications: A survey,” 2014.
4. LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, *3361*(10), 1995.
5. M. Le, M. Postma, J. Urbani, and P. Vossen, “A deep dive into word sense disambiguation with LSTM,” in *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 354–365.
6. “Sentiment analysis using deep learning techniques: A review,” *International Journal of Advanced Computer Science and Applications*.
7. Al-Amin, M., Islam, M. S., & Uzzal, S. D. (2017, February). Sentiment analysis of Bengali comments with word2vec and sentiment information of words. In *2017 International Conference on Electrical, Computer and Communication Engineering (ECCE)* (pp. 186-190). IEEE.
8. Chowdhury, S., & Chowdhury, W. (2014, May). Performing sentiment analysis in Bangla microblog posts. In *2014 International Conference on Informatics, Electronics & Vision (ICIEV)* (pp. 1-6). IEEE.
9. Hossain, M. S., Jui, I. J., & Suzana, A. Z. (2017). *Sentiment analysis for Bengali newspaper headlines* (Doctoral dissertation, BRAC University).
10. Hassan, A., Amin, M. R., Mohammed, N., & Azad, A. K. A. (2016). Sentiment analysis on Bangla and Romanized Bangla text (BRBT) using deep recurrent models. *arXiv:1610.00369*.
11. Sumit, S. H., Hossan, M. Z., Al Muntasir, T., & Sourov, T. (2018, September). Exploring word embedding for bangla sentiment analysis. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)* (pp. 1-5). IEEE.
12. Asimuzzaman, M., Nath, P. D., Hossain, F., Hossain, A., & Rahman, R. M..Sentiment analysis of Bangla microblogs using adaptive neuro fuzzy system. In *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery* (pp. 1631-1638).
13. Tuhin, R. A., Paul, B. K., Nawrine, F., Akter, M., & Das, A. K. (2019, February). An Automated System of Sentiment Analysis from Bangla Text using Supervised Learning Techniques. In *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)* (pp. 360-364). IEEE.
14. Uddin, A. H., Dam, S. K., & Arif, A. S. M. (2019, December). Extracting Severe Negative Sentence Pattern from Bangla Data via Long Short-term Memory Neural Network. In *2019 4th International Conference on Electrical Information and Communication Technology (EICT)* (pp. 1-6). IEEE.
15. Tabassum, N., & Khan, M. I. (2019, February). Design an Empirical Framework for Sentiment Analysis from Bangla Text using Machine Learning. In *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)* (pp. 1-5). IEEE.
16. Alam, M. H., Rahoman, M. M., & Azad, M. A. K. (2017, December). Sentiment analysis for Bangla sentences using convolutional neural network. In *2017 20th International Conference of Computer and Information Technology (ICCIT)* (pp. 1-6). IEEE.
17. Paul, A. K., & Shill, P. C. (2016, December). Sentiment mining from Bangla data using mutual information. In *2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)* (pp. 1-4). IEEE.
18. Tripto, N. I., & Ali, M. E. (2018, September). Detecting multilabel sentiment and emotions from Bangla YouTube comments. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)* (pp. 1-6). IEEE.
19. Al-Amin, M., Islam, M. S., & Uzzal, S. D. (2017, February). Sentiment analysis of Bengali comments with word2vec and sentiment information of words. In *2017 International Conference on Electrical, Computer and Communication Engineering (ECCE)* (pp. 186-190). IEEE.
20. Taher, S. A., Akhter, K. A., & Hasan, K. A. (2018, September). N-gram based sentiment mining for Bangla text using support vector machine. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)* (pp. 1-5). IEEE.
21. Rabeya, T., Chakraborty, N. R., Ferdous, S., Dash, M., & Al Marouf, A. (2019, February). Sentiment Analysis of Bangla Song Review-A Lexicon Based Backtracking Approach. In *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)* (pp. 1-7). IEEE.
22. Haydar, M. S., Al Helal, M., & Hossain, S. A. (2018, February). Sentiment Extraction from Bangla Text: A Character Level Supervised Recurrent Neural Network Approach. In *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)* (pp. 1-4). IEEE.
23. Akter, S., & Aziz, M. T. (2016, September). Sentiment analysis on Facebook group using lexicon-based approach. In *2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)* (pp. 1-4). IEEE.
24. Sharif, O., Hoque, M. M., & Hossain, E. (2019, May). Sentiment Analysis of Bengali Texts on Online Restaurant Reviews Using Multinomial Naïve Bayes. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)* (pp. 1-6). IEEE.
25. Fawcett, Tom (2006). [*"An Introduction to ROC Analysis"*](http://people.inf.elte.hu/kiss/11dwhdm/roc.pdf) *(PDF)*. Pattern Recognition Letters. **27** (8): 861–874. [*doi*](https://en.m.wikipedia.org/wiki/Doi_(identifier)):[*10.1016/j.patrec.2005.10.010*](https://doi.org/10.1016%2Fj.patrec.2005.10.010)
26. Powers, David M W (2011). [*"Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation"*](http://www.flinders.edu.au/science_engineering/fms/School-CSEM/publications/tech_reps-research_artfcts/TRRA_2007.pdf) *(PDF)*. Journal of Machine Learning Technologies. **2** (1): 37–63.
27. Ting, Kai Ming (2011). [*Encyclopedia of machine learning*](https://link.springer.com/referencework/10.1007%2F978-0-387-30164-8). Springer [*ISBN*](https://en.m.wikipedia.org/wiki/ISBN_(identifier))[*978-0-387-30164-8*](https://en.m.wikipedia.org/wiki/Special:BookSources/978-0-387-30164-8).
28. Brooks, Harold; Brown, Barb; Ebert, Beth; Ferro, Chris; Jolliffe, Ian; Koh, Tieh-Yong; Roebber, Paul; Stephenson, David (2015-01-26). [*"WWRP/WGNE Joint Working Group on Forecast Verification Research"*](https://www.cawcr.gov.au/projects/verification/). Collaboration for Australian Weather and Climate Research. World Meteorological Organisation*. Retrieved 2019-07-17*.
29. Chicco D, Jurman G (January 2020). [*"The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation"*](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6941312). BMC Genomics. **21** (6). [*doi*](https://en.m.wikipedia.org/wiki/Doi_(identifier)):[*10.1186/s12864-019-6413-7*](https://doi.org/10.1186%2Fs12864-019-6413-7). [*PMC*](https://en.m.wikipedia.org/wiki/PMC_(identifier)) [*6941312*](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6941312). [*PMID*](https://en.m.wikipedia.org/wiki/PMID_(identifier)) [*31898477*](https://pubmed.ncbi.nlm.nih.gov/31898477).
30. J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” *CoRR*, vol. abs/1810.04805, 2018.
31. M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in *Proc. of NAACL*, 2018.
32. T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *CoRR*, vol. abs/1301.3781, 2013.
33. S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

1. Recently lots of pre-trained language models like BERT [30], ELMo [31], XLNet have been reported to achieve promising results on several NLP tasks including sentiment analysis. However, these models are mainly targeted to the English language, not Bangla. [↑](#footnote-ref-1)