Analyzing Bangla Text Sentiment through Leveraging Limited-Resource Convolutional Neural Network

Joyanta Jyoti Mondal*, Priya Saha†
*†School of Data and Sciences, BRAC University, Dhaka, Bangladesh
*joyanta.jyoti.mondal@g.bracu.ac.bd, †priya.saha@g.bracu.ac.bd

***Abstract*—Sentiment Analysis is widely used in recent days to get point of view of mass people in terms of their social perspective or political perspective or user experience perspective with regards to any product they buy. With a significant growth of user number who uses Bangla as their primary language and user interactions using Bangla through prominent advances towards the access of web, it is gaining more focus from an academic and a commercial point of view. It is one of the cores and well-researched areas in Natural Language Processing (NLP) in languages like English. Currently, it is a more prominent research field of Bangla Natural Language Processing (BNLP) system as the limelight has been driven towards research works regarding sentiment analysis for this language in very recent years. However, one of the major challenges for low-resource languages, such as Bangla, is a lack of resources. Also a lack of comparable results from research studies has become a significant limitation for BNLP. Mainly, sentiment analysis is an automated text mining approach that determines the emotion of a given text. A given text can be classified into many certain emotions using sentiment analysis. In this paper, we propose method of a low-cost Deep Learning Framework based on Convolutional Neural Network (CNN) that analyzes sentiments from texts written in Bangla. The classifier model obtains a classification accuracy of 82.75%, which is almost 2% better and a F1-Score of 73% which is also 30% better than the available state-of-the art Bangla sentiment classifier having a low amount of parameters in our proposed approach.**

***Index Terms*—Text Sentiment Analysis, Convolutional Neural Network, Deep Learning, Bangla Natural Language Processing, Machine Learning.**

I. Introduction

Sentiment Analysis (SA), also referred as opinion mining, is growing popularity as a region of research due to its many pragmatic and empirical applications connected to people's emotions. It contributes in the identification and exploration of people's feelings, point of views, behavior, attitudes, and emotions about entities such as persons, organizations, goods, services, recent or earlier topics, and their qualities using written or spoken language, [1]. As publicly and privately accessible material on the internet is constantly expanding, a large amount of texts expressing viewpoints are available in various blogs and other pieces in online journals, social networking sites such as Facebook, Twitter, and product review sites. And, with the present pace of machine learning advancement, the work of text processing and analysis is evolving day by day in terms of application in numerous commercial domains such as product analysis, market analysis, and social media monitoring.

Bangla is one of the most broadly spoken languages in the world, with around 250 million native speakers worldwide. Numerous studies on sentiment analysis for English and other languages such as Chinese, Hindi, Urdu, and Arabic are conducted earlier [2]–[4]. Despite being one the most frequently spoken language in the world, Bangla is a low-resource language. Recently, we see the emergence of large amounts of digital information in Bangla, such as Facebook status updates,review comments, which allow us to study sentiment in Bangla. As more Bangla-speaking individuals are getting engaged in internet activities, analyzing sentiment in Bangla is becoming an important problem. However, such data is critical in understanding market trends and consumer preferences, in movie reviews, and so on. However, manually assessing this massive quantity of data and determining individuals' interests is time-consuming and, in some situations, impossible. In these cases, automated sentiment analysis can be quite useful in determining people's interests and opinions.

There is a limited number of works that evaluate sentiment in Bangla language. They do, however, share some difficulties. For example, we can get a high accuracy rate in such systems if a certain trait appears in the texts. Furthermore, the approach they offer is unsuitable for extracting emotion from complicated texts. We are focused to address the challenges mentioned above for Bangla sentiment analysis. We provide a framework for sentiment analysis that does not require any special dependencies. For instance, the existence of any certain characteristic in the texts. Currently, we divide emotion into three categories: positive, negative, and neutral.

For example, the following sentence is a positive word about food.

"যাইহোক, খাবারগুলো বেশ ভালো।"

Based on our study, we make the following set of contributions in the research:

- We present a low-cost Deep Learning architecture to extract sentiment from a sentence that shows convincing performance than the other popular machine learning models.
- We perform experiments in which we implement our model on accessible dataset and compare it to existing machine learning methods. The comparison demonstrates

that our proposed approach outperforms available alternative options.

The following is the paper's framework. We begin with a review of some of the recent research works related to Text Sentiment Analysis problems in Bangla in Section II. Following that, Section III illustrates the approach we introduce. Then, in Section IV, we show experiments implementing the proposed architecture and discuss the evaluation of our experiment. In Section V, we conclude the paper and demonstrate possible future implications.

II. Related Work

A. Machine Learning Methods

In terms of BNLP, there have been some initiatives which focus on solving this problem using Machine Learning Approaches.

Tuhin et al. [5] proposes a Naive Bayes based approach to solve this problem. Moreover, Mahtab et al. [6] focuses on coming up with a SVM based approach of only one aspect of Bangladesh Cricket and the reaction of Bangladeshis towards them.

B. Deep Learning Methods

From the perspective of BNLP, deep neural network based aprroaches are being deployed in recent times with a remarkable success.

Long Short Term Memory (LSTM) [7], a Recurrent Neural Network (RNN) special variant with feedback connections, is being applied in several BNLP models. Hassan et al. [8] present a Deep Recurrent Neural Network based approach which is mainly a special varient of LSTM to extract the sentiment from Bangla and Romanized Bangla comments. Also, Wahid et al. [9] present a LSTM based architecture to get the sentiment from Bangla comments related to Bangladesh Cricket Team. Moreover, Ahmed et al. [10] present another approach with LSTM which results 94% accuracy.

Convolutional Neural Network (CNN) is a type of neural network that uses convolutional layers to apply weights and biases to various features of input data. CNN-based models also have been presented to perform well in this job. Rahman et al. [11] present a CNN based approach to extract the sentiment from Bangla comments.

III. Research Methodology

In this section, we propose our idea for a CNN architecture to analyze a given text (in Bangla) and implement it in live web services. We begin by creating a variant of the typical CNN framework by modifying some critical parameters and functions.

A. Convolutional Neural Network (CNN)

A CNN is a type of artificial neural network that is especially developed to analyze vector input and is used in image recognition and processing.

CNNs are sophisticated AI architectures that employ deep learning to do both generative and descriptive tasks, frequently utilizing machine vision, which includes image and video recognition, recommendation systems, and even complicated tasks like natural language processing (NLP).

A neural network is a special architecture that mimics the activity of neurons in the brain. Traditional neural networks are not optimized for image processing, thus they have to be input images in low-resolution chunks. The "neurons" of a CNN are more like those of the frontal lobe, the part of the brain responsible for processing visual information in humans and other animals. The layers of neurons are organized in such a way that they span the whole visual field, eliminating the partial image processing issue that affects regular neural networks.

A CNN uses a mechanism similar to that of a multilayer perceptron, which is tuned for a faster processing rate. A CNN is made up of numerous convolutional layers, pooling layers, dropout layers, fully connected (FC/Dense) layers, and normalization layers, as well as an input layer, an output layer, and a hidden layer.

B. Proposed Methodology

Our primary objective is to have a minimal number of parameters of the model. A lower number of parameters in a model results in increased accuracy and computation speed.

1) Data Preprocessing: Data preprocessing plays a vital role on text analysis to make the model understand the data. Text data contains a lot of noise and as a result, its a challenge to clean the texts. Data pre-processing reduces the size of the input text documents significantly and is done by various steps:

- Removing Special Characters: We remove the special characters since they occasionally cause confusion, and we believe that in these types of Bengali datasets, special characters would add to classification difficulty.
- Removing Punctuations: One of the most common and widely used preprocessing techniques is punctuation removal. In Bengali, even the full-stop "." corresponds to the "|" symbol. As a result, we eliminate all punctuation.

The phrase is subsequently tokenized and transformed into a series of tokens before being fed into the model, as shown in Figure 1. Tokenization is done so that embeddings can be generated afterwards. A tokenzier primarily allows vectorization of a text corpus by converting each text into either a series of integers (each integer being the index of a token in a dictionary) or a vector where the coefficient for each token can be binary, based on word count, or based on TF-IDF.

- TF-IDF - The dataset should be in numerical form for training statistical algorithms employing machine learning. To make these statistical methods operate, we first transform the sentences into numbers. This technique contains two terms: TF (Term Frequency) relates to the number of times a word occurs in a document, and IDF (Inverse Document Frequency) refers to how significant the word is in the document [12]. TF-IDF is a popular method for weighing terms in NLP tasks because it assigns a value to a term based on its significance in a document scaled by its relevance across all documents in the corpus, mathematically eliminating naturally occurring words in the language and selecting words that are more descriptive of the given text.

The equation for TF and IDF given below:

$$\text{TF(t)} = \frac{Number\_of\_times\_term\_t\_appears\_in\_a\_document}{Total\_number\_of\_terms\_in\_the\_document} \quad (1)$$

$$\text{IDF(t)} = \log_e \frac{Total\_number\_of\_documents}{Number\_of\_documents\_with\_term\_t\_in\_it} \quad (2)$$

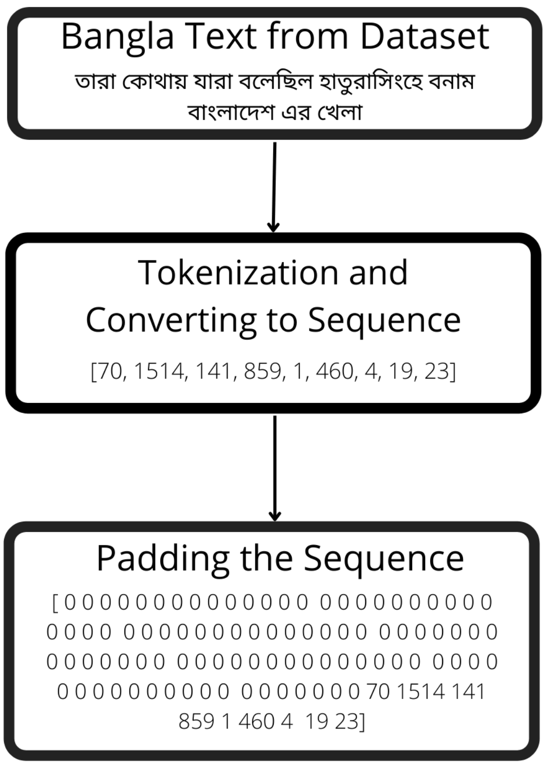

Fig. 1: Text Preprocessing

Our CNN model starts with creating embedding from a padded/processed text of the dataset. The main advantage of word embedding is that words that share a similar context can be represented close to each other in the vector space. Thus, vectors carry a sense of semantic of a word. We use three Conv1D layers with 3 kernel size. Also, to avoid overfitting in the model, we use 10% dropout in the first and 15% dropout in third Conv1D layer. We choose ReLU as the activation function because its gradient is not saturated, which considerably accelerates the evolving of stochastic gradient descent (SGD) compared to the other activation functions such as the Tanh / Sigmoid function. Finally, we convert the values to an one-dimensional array then we begin adding fully connected (FC) layers to the CNN, starting with 128 nodes, and finally 64 nodes. In the output layer, we use the Sigmoid activation function as a network classifier. Sigmoid is mainly a non-linear function. This essentially means - the output can be non linear as well which will help us to get the polarity of texts of complicated sentences.

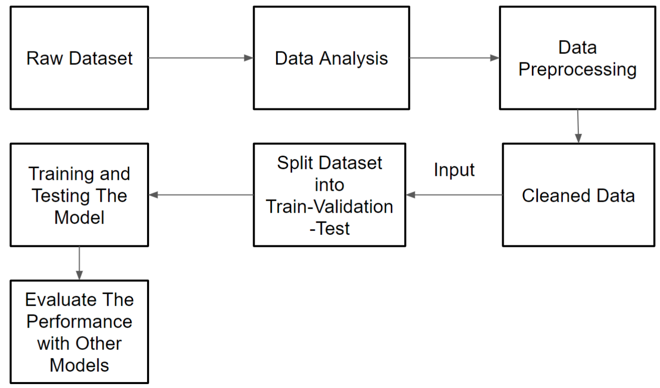

Fig. 2: Workflow of Our Approach

Table I shows the optimum parameters applied in different datasets for the proposed Deep CNN model.

Parameter	Value
No. of Training Epochs	30
Size of Mini Batch	4
Learning Rate	0.00001
Training Set Size	2,145
Validation Set Size	238
Test Set Size	596

TABLE I: Hyper Parameters of Our Proposed Deep CNN Model

After experimenting with various tweaks in the model, we come up with our Deep CNN architecture. This architecture provides the best performance while minimizing the number of parameters and therefore less computing resources.

IV. Experimental Evaluation

Following the development of our suggested architecture, we test our model on against some machine learning approaches. Our primary objective is to create a model that is less in size and as much as more accurate possible than those distinct designs.

A. Dataset

With respect to BNLP, a few datasets have been introduced by researchers.

The dataset we use in this work is presented by Rahman et al.[1] [13]. The dataset focuses mainly 3 aspects: positive,

[1] https://github.com/AtikDU/Bangla_ABSA_Datasets

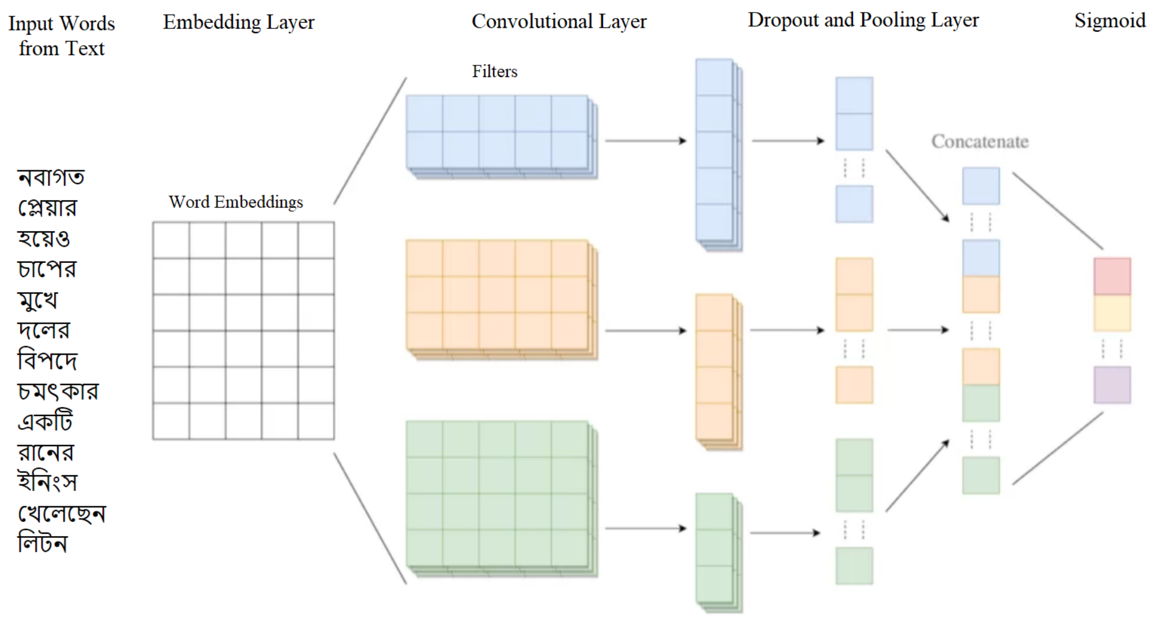

Fig. 3: General CNN Architecture for Text Sentiment Analysis

Layers	Output Shape	Parameters
Embedding Layer	(None, 100, 100)	238400
1D Convolutional Layer	(None, 98, 64)	19264
Batch Normalization	(None, 98, 64)	256
1D Convolutional Layer	(None, 96, 96)	18528
Batch Normalization	(None, 96, 96)	384
1D Convolutional Layer	(None, 94, 128)	36992
Batch Normalization	(None, 94, 128)	512
Flatten	(None, 12032)	0
Dense	(None, 128)	1540224
Batch Normalization	(None, 128)	512
Flatten	(None, 128)	0
Dense	(None, 64)	8256
Batch Normalization	(None, 64)	256
Dense	(None, 3)	195
Total Parameters:		1,863,779
Trainable Parameters:		1,862,819
Non-trainable Parameters:		960

TABLE II: Output Shape and Parameter Size of Each Layer of The Proposed Model

negative and, neutral. The dataset is of a cricket dataset which consists of 2900 comments, collected from online news sites. They collect the data of cricket dataset from two facebook pages of two renowned news platforms in Bangladesh: BBC Bangla News [2] and Prothom Alo Newspaper [3].

We use 80% of dataset for training and 20% for testing. Among the total training data, we divide 15% data for validation.

[2]https://www.facebook.com/BBCBengaliService
[3]https://www.facebook.com/DailyProthomAlo

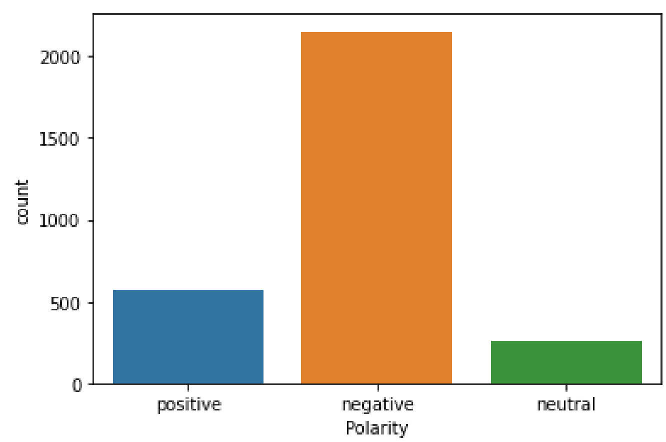

Fig. 4: Polarity Amount of The Dataset

B. Experimental Setup

Tensorflow, Keras, Pillow, and OpenCV Python libraries are used to make the training and testing protocols for this Deep CNN model. The models are trained and evaluated on PC with an NVIDIA RTX 2070 with 7.5 TeraFLOPs of performance.

C. Performance Evaluation

In order to evaluate the performance of the proposed methods, usually these performance metrics are used to evaluate the prediction results: Accuracy, Loss, Precision, F1-Score and Recall. The accuracy, precision, recall, and loss measures are used in this study as the metrics.

The accuracy is defined as the ratio of the model's correct predictions to the total number of predictions made.

The Precision is used to determine a proportion of correct identifications. Precision is calculated by dividing the number of true positive outcomes (TP) by the number of predicted positive outcomes (TP + FP). The precision ranges from 0 to 1, and it is determined using the following equation:

$$\text{Precision } = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{3}$$

The recalls are used to figure out what percentage of true positives is appropriately detected. Recall is calculated by dividing the number of true positives (TP) by the total amount of data (TP + FN) is the important factor. The recall is calculated using the following equation:

$$\text{Recall } = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{4}$$

Similarly, the F1 Score is a popular metric for assessing the effectiveness of machine learning algorithms. It is calculated as the arithmetic mean of accuracy and recall. It has a value between 0 and 1. The number of occurrences properly identified by the learning models is reflected in the F1 scores. The following equation is used to determine the F1 Score:

$$\mathrm{F1} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{5}$$

The experimental findings on the dataset using the proposed CNN model and other traditional techniques are shown in Table III. When compared to previous techniques, our model performs much better in terms of Accuracy, Precision, and Recall. The recall rate implies that CNN has a greater learning rate than other techniques. It is also obvious from Table III that the accuracy and recall rate differ in the majority of situations. As a result, we compute the F1 score, which is the harmonic mean of accuracy and recall. Because precision and recall are critical in this situation, we can observe that the proposed CNN has the greatest F-1 score, 73%, when compared to other approaches. It is evident that the scores for RF, LR, and SVM are comparatively better in the dataset. KNN and NB perform poorly in comparison to other algorithms. Moreover, for the dataset, the proposed CNN model has a substantial level of accuracy. In the dataset, we get 82.75% accuracy, whereas classification using Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbour (KNN), and previously implemented CNN achieve only 19%, 25%, 22%, and 81%, respectively.

Model	Accuracy	Precision	Recall	F1
SVM	19%	0.71	0.22	0.34
RF	24%	0.60	0.27	0.37
KNN	21%	0.45	0.21	0.35
NB [14]	-	0.23	0.27	0.18
CNN [15]	81%	0.54	0.48	0.51
Proposed Approach	**82.75%**	**0.78**	**0.69**	**0.73**

TABLE III: Experiment Result using The Dataset [15]

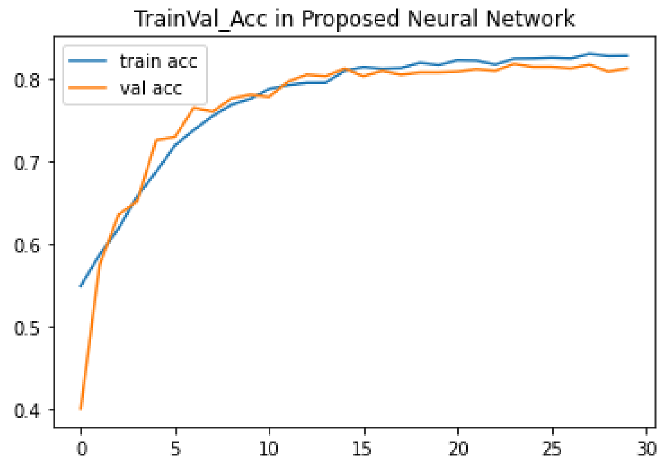

Fig. 5: Accuracy

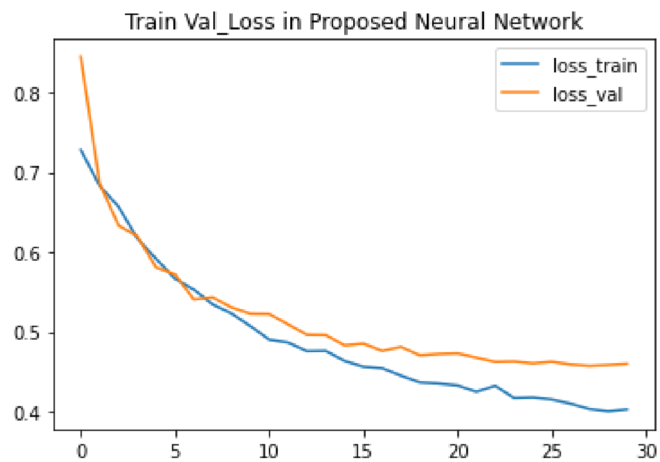

Fig. 6: Loss

D. Discussion

As shown in Figure 4, the number of negative comments is substantially larger than the number of positive and neutral phrases, resulting in a worse model performance. Because, different people think differently and communicate their opinions from various perspectives. As a result, too much diversity of opinion; imbalance of types of data in a dataset can likewise be a cause of poor performance. Because, this imbalance can create a biasness of the model for a certain type of data.

V. Conclusion and Future Work

There has been a push in recent years to increase NLP work in Bangla. In this study we present a model for opinion extraction based on low-cost CNN architecture with a bit of preprocessing. We use traditional steps to clean data and achieve better results comparing other state-of-the-art approaches. More detailed annotated dataset like SemEval for Bengali language can lead to impressive results.

Sentiment analysis is becoming increasingly used for identifying spam reviews/comments and detecting fraudulent apps.

As a result, broadening research to include additional non-English languages and aspects can lead to a more accurate knowledge of users and their assessments, which can aid in making better business decisions and increasing cyber security.

We want to use Aspect Based Sentiment Analysis (ABSA) in the future, which is more sophisticated than typical sentiment analysis. Aspect-based sentiment analysis improves opinion mining by allowing users to categorize data by characteristic (aspect) and subsequently identify attributable opinions (sentiment). It automates time-consuming tasks, works in real time, is easy to scale, and gives an unbiased experience for a truly customer-centric experience. In order to achieve the goals of Aspect Based Sentiment Analysis, we will work on a specific dataset and technique that correlates sentiments with the associated aspects.

References

[1] B. Liu, "Sentiment analysis and opinion mining," *Synth. lect. hum. lang. technol.*, vol. 5, no. 1, pp. 1–167, 2012.

[2] O. Oueslati, E. Cambria, M. B. HajHmida, and H. Ounelli, "A review of sentiment analysis research in arabic language," *Future Generation Computer Systems*, vol. 112, pp. 408–430, 2020.

[3] C. Nanda, M. Dua, and G. Nanda, "Sentiment analysis of movie reviews in hindi language using machine learning," in *2018 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 2018, pp. 1069–1072.

[4] D. A. Pereira, "A survey of sentiment analysis in the portuguese language," *Artificial Intelligence Review*, vol. 54, no. 2, pp. 1087–1115, 2021.

[5] R. A. Tuhin, B. K. Paul, F. Nawrine, M. Akter, and A. K. Das, "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)*. IEEE, 2019.

[6] S. Arafin Mahtab, N. Islam, and M. Mahfuzur Rahaman, "Sentiment analysis on bangladesh cricket with support vector machine," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*. IEEE, 2018.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] A. Hassan, M. R. Amin, A. K. Al Azad, and N. Mohammed, "Sentiment analysis on bangla and romanized bangla text using deep recurrent models," in *2016 International Workshop on Computational Intelligence (IWCI)*. IEEE, 2016, pp. 51–56.

[9] M. F. Wahid, M. J. Hasan, and M. S. Alom, "Cricket sentiment analysis from bangla text using recurrent neural network with long short term memory model," in *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)*. IEEE, 2019, pp. 1–4.

[10] A. Ahmed and M. A. Yousuf, "Sentiment analysis on bangla text using long short-term memory (lstm) recurrent neural network," in *Proceedings of International Conference on Trends in Computational and Cognitive Engineering*. Springer, 2021, pp. 181–192.

[11] M. H. Alam, M.-M. Rahoman, and M. A. K. Azad, "Sentiment analysis for bangla sentences using convolutional neural network," in *2017 20th International Conference of Computer and Information Technology (ICCIT)*. IEEE, 2017, pp. 1–6.

[12] A. Panchal, "NLP — text summarization using NLTK: TF-IDF algorithm," https://towardsdatascience.com/text-summarization-using-tf-idf-e64a0644ace3, Jun. 2019, accessed: 2022-5-14.

[13] M. Rahman and E. Kumar Dey, "Datasets for aspect-based sentiment analysis in bangla and its baseline evaluation," *Data (Basel)*, vol. 3, no. 2, p. 15, 2018.

[14] S. Haque, T. Rahman, A. K. Shakir, M. Arman, K. B. B. Biplob, F. A. Himu, D. Das, M. S. Islam *et al.*, "Aspect based sentiment analysis in bangla dataset based on aspect term extraction," in *International Conference on Cyber Security and Computer Science*. Springer, 2020, pp. 403–413.

[15] M. A. Rahman and E. K. Dey, "Aspect extraction from bangla reviews using convolutional neural network," in *Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*. IEEE, 2018, pp. 262–267.