

Research Article

Bidirectional LSTM-Based Sentiment Analysis for Assamese Text

Manashi Talukdar^{*} , Shikhar Kumar Sarma

Information Technology, Gauhati University, Guwahati, India

Abstract

With the enhanced exploration of the new generation of the web, people are free to state their opinion on any particular topic like product, services, organization and even on other people online in different social media platform and thus an innumerable amount of user generated contents are being created each moment. Hence, the need for mining this information has become the priority of the researcher so that they can identify the user's sentiments and guide other people in various fields. Sentiment analysis deals with analyzing the review, opinion, attitude and emotions of a person from a given set of text by categorizing those on the basis of polarity as positive, negative and neutral. In this paper, sentiment of the social media text in Assamese Language is being analyzed because most of the communication is done through regional language and as a researcher from this region it is utmost concern to mine this information. To analyze the sentiments from the manually prepared datasets, LSTM- deep learning algorithm is used and implemented it in Python environment and also overall performance is measured in terms of accuracy, precision, recall and f1-score.

Keywords

Sentiment Analysis, NLP, Deep Learning, LSTM

1. Introduction

The most popular communication platform world wide web contains lots of reviews, comments, opinions and other sentiments from various domains like product, research paper, books, daily text reviews on any personal or social events. Due to the availability and ease of operating, number of online users is being increased and hence huge amount of user generated contents are produced daily. Also as a human being it is obvious for us to be influenced by others' opinion in day to day life and that is why Sentiment Analysis has become utmost topics for the current research in the field computer science.

Here, mining of sentiments in Assamese language text on

social media is done as most of the communication is being done through regional languages like Assamese and this information is also needed to be analyzed for guiding others.

The main objective here is to prepare a datasets of Assamese language text for sentiment analysis from the social media and apply LSTM- deep learning algorithm there to find the results in terms of four different performance measures.

2. Literature Survey

Natural Language processing (NLP): NLP is a branch of

^{*}Corresponding author: mansitalukdar@gmail.com (Manashi Talukdar)

Received: 20 March 2024; Accepted: 8 April 2024; Published: 28 April 2024



Copyright: © The Author(s), 2024. Published by Science Publishing Group. This is an **Open Access** article, distributed under the terms of the Creative Commons Attribution 4.0 License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

Artificial intelligence that help computer to understand and process natural human languages. The purpose of NLP is to analyze and process huge amount of natural language data. It helps to resolve the ambiguity in the language and include required numeric structure to the data for many applications like speech recognition, text analytics, sentiment analysis etc.

Sentiments: Sentiments are the thoughts, attitudes, views depending on the person's perception, belief, and emotion. Any text information can contain either facts or opinion. Facts are just an objective expression whereas opinion bearing text holds subjective information and it can convey any kind of sentiments regarding any entities or events. For example: লকডাউন ভংগকাৰী লোকৰ বিৰুদ্ধে কঠোৰ ব্যৱস্থা গ্ৰহণ কৰিব

Here, a feature (কঠোৰ ব্যৱস্থা) of an entity (লোক) is identified as a negative opinion.

Sentiment analysis: Sentiment analysis is an application of natural language processing to extract sentiments from the given text by classifying those into subjective as positive, negative and neutral or as objective if it does not contain any sentiments.

Sentiment analysis is an NLP task to identify subjective opinion from a very large amount of text. Subjectivity means that the text has some opinion where as objectivity state that the text does not contain any opinion. For Example: Objective: ২৩ পৃষ্ঠাৰ বন্ধ / ছবি

Subjective: মেলবৰ্ণ টেবুত অষ্ট্ৰেলিয়াৰ বিৰুদ্ধে ভাৰতৰ ঐতিহাসিক জয়

The Subjective one can be categorized into three categories.....

Positive: বৰ সুন্দৰ আৰু বৰ অপৰূপ এই দৃশ্য

Negative: সময় পাৰ হৈ গৈ আছে

Neutral: ঘটনাস্থলীত উপস্থিত হৈ আৰক্ষীৰ তদন্ত আৰম্ভ

2.1. Types of Sentiment Analysis

The different types of sentiment analysis are...

Fine-grained Sentiment Analysis: Fine-grained Sentiment Analysis is mainly used to determine the polarity of an opinion. Sentiment polarity means a binary value to express either positive or negative. Higher specification like very positive, positive, neutral, negative, very negative may be there, depending on the use case considered.

Emotion detection: Emotion detection specifies particular emotions represented in the given text. To determine emotions expressed in text a combination of rule based and machine learning algorithms is a better option.

Aspect-based sentiment analysis: Aspect-based sentiment analysis determines an opinion regarding a particular element of the product. It is mostly used in product analytics to identify how the product is perceived and what is the strength and weakness of the product from the general customers' review.

Intent Analysis: Intent analysis finds out what are the main intentions of the text. It is generally used in customer support systems.

Fine-grained Sentiment Analysis is performed here that means determining the polarity of the opinion by simple positive, negative and neutral sentiment classification.

2.2. Sentiment Analysis Levels Types

Sentiment analysis can be performed at the following levels...

Document-level sentiment classification: Document-level sentiment classification used to identify whether the whole document represents a positive or a negative sentiment.

Sentence level sentiment classification: Sentence-level sentiment classification determines whether the sentence is a positive, negative, or a neutral sentence. It is highly used to obtain an accurate analysis of any specific object.

Entity and Aspect level sentiment classification: Aspect-level sentiment classification uses finer-grained analysis. It works on the principle that an opinion contains both sentiment - positive or negative and a target because opinion without target can not specify anything properly. Thus, it helps to find out what exactly people liked or disliked.

Word level sentiment classification: Word-level sentiment classification is used to evaluate each word and determine its sentiment. It is based on term frequency and total weight not on the polarity like sentence level sentiment classification.

In this work, sentiment analysis is applied on sentence level ie. it determines whether each sentence specify a positive, negative, or a neutral opinion.

2.3. Sentiment Analysis Approaches

Sentiment analysis approaches can be categorized as...

Rule-based or Lexicon based approach: Rule based systems implements sentiment analysis on a collection of manually crafted rules. Complex rule-based systems requires lots of time and both linguistics as well as topics knowledge is needed for that and also lots of analysis and testing is essential for modifying or adding new rules.

Automatic approach: This type of system depends on machine learning or deep learning techniques to learn from data and no need to invest lots of time to create any rules to get the predicated output.

Hybrid approach: This the combination of both rule based and automatic approaches.

In this project work, deep learning methods called Bidirectional LSTM is used.

2.4. Deep Learning

Deep learning means very large neural networks which are trained using huge amount of data. A Neural Network contains...

Input Layers: Input is given to the model here and the

number of neurons in this layer is equal to total number of features in the data.

Hidden Layer: Hidden layers may consist of many hidden layers based on the model and data size. The output from each layer is computed by using output of the previous layer, weights and biases of the layer followed by activation function.

Output Layer: Output of hidden layer is fed into function like sigmoid or softmax that converts the output of each class into probability score.

Then calculate the error using an error function like cross entropy, square loss error etc. then back propagate into the model by calculating the derivatives to minimize the loss.

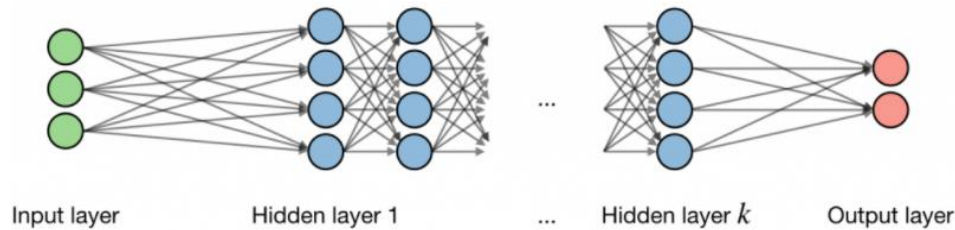


Figure 1. Working of Neural Network.

2.5. LSTM (Long-Short Term Memory)

LSTM is a special kind of Recurrent Neural Network. It is capable of learning long-term dependencies because LSTM consists of cells to hold information from initial to later time steps without getting vanished. It also contains three gates to control the flow of information...

- 1) Forget Gate: Forget gate select which information should forward or which to ignore using sigmoid function.
- 2) Input Gate: Input gate adds the new information by updating cell states using both tanh and sigmoid function.
- 3) Output Gate: Output gate generates the next hidden states and cell states are carried over the next time step using sigmoid function.

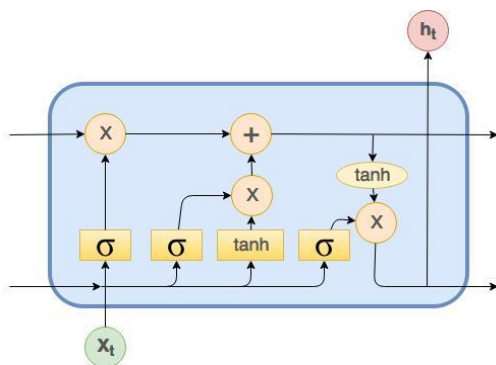


Figure 2. Single LSTM cell.

2.6. Bidirectional LSTM

LSTM is somehow lacking behind to consider post word information because the sentence is being read in forward way only. To solve this problem, a Bidirectional LSTM is

used. It is an improved version of LSTM. It tries to get information from both sides left to right and right to left and all other concept is same as LSTM.

Bidirectional LSTM uses two LSTMs whose outputs are stacked together. That means it uses two separate LSTM units to read the sentences- one for forward direction and other for backward direction. After each hidden states of LSTM processed their respective final word, these are joined. Thus, it can find more semantic information than LSTM. This improves the accuracy of models.

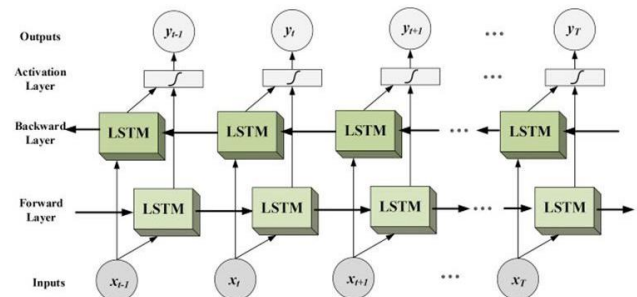


Figure 3. Bidirectional LSTM.

Related Study:

- 1) Sonali Rajesh Shah, Abhishek Kaushik, "Sentiment analysis on indian indigenous languages: a review on multilingual opinion mining"

They analyze, review and discuss the approaches, algorithms, challenges faced by the researchers while carrying out the SA on Indigenous languages. Their main aim is to understand the recent work that has been done in SA for indigenous languages and for this they studied 23 papers out of these 67% of the papers have used ML, DL and advanced DL algorithms and only 29% of researchers have used lexicon-based approach [8]. They stated that there is a need for

more SA work to be carried out at document level or aspect.

- 2) Bo Pang and Lillian Lee, Shivakumar Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques"

They consider the problem of classifying documents not by topic, but by overall sentiment, e.g., determining whether a review is positive or negative using movie reviews as data and found that standard machine learning techniques definitely outperform human-produced baselines [1].

- 3) Abhishek Bhagat, Akash Sharma, Sarat Kr. Chettri, "Machine Learning Based Sentiment Analysis for Text Messages"

They performed a sentiment analysis of text messages using supervised machine learning techniques [9]. Here, text data from product reviews, general tweets and movie reviews are taken into account to assess the polarity (positive or negative) of messages or tweets and used the classification algorithms namely SVM, Naïve Bayes and decision tree where they evaluated their models on the basis of metrics: accuracy, precision, recall, F1-score. They found that the results obtained from the Decision Tree and SVM have higher accuracy with most of the datasets and are considered to be good classifiers.

- 4) Rudy Prabowo, Mike Thelwall, "Sentiment Analysis: A Combined Approach"

They combines rule-based classification, supervised learning and machine learning into a new combined method and tested on movie reviews, product reviews and MySpace comments [2]. The results show that a hybrid classification can improve the classification effectiveness in terms of F1 score.

- 5) Betül AyKaraku, Muhammed Talo, Ibrahim Rıza Halla, Galip Aydın, "Evaluating deep learning models for sentiment classification"

They describe several deep learning models for a binary sentiment classification problem and compare the models in terms of accuracy and time performances [3]. They built several variants of CNN and LSTM by changing the number of layers, tuning the hyper-parameters, and combining models. Additionally, word embeddings were created by applying the word2vec algorithm with a skip-gram model on a large dataset composed of movie reviews. Experimental results have shown that the use of word embeddings with deep neural networks effectively yields performance improvements in terms of runtime and accuracy.

- 6) Dr. G. S. N. Murthy, Shanmukha Rao Allu, Bhargavi Andhavarapu, Mounika Bagadi, Mounika Belusonti, "Text based Sentiment Analysis using LSTM"

They proposed a sentiment classification approach based on LSTM for text data [4]. Reviews and social network posts are categories of textual documents that are most interesting for sentiment analysis. DL methods such as LSTM show better performance of sentiment classification with 85% accuracy when there are more amounts of training data.

- 7) Wisam Hazim Gwad Gwad, Imad Mahmood Ismael

- Ismael, Yasemin Gültepe, "Twitter Sentiment Analysis Classification in the Arabic Language using Long Short-Term Memory Neural Networks"

They have used the LSTM to analyze Arabic twitter user comments [6]. As a result of the proposed model, training and test, an average performance of 89.8% was achieved. The same Arabic dataset was tested with traditional machine learning algorithms and the proposed LSTM model achieved the highest performance.

- 8) S. Sachin Kumar (B), M. Anand Kumar, and K. P. Soman, "Sentiment Analysis of Tweets in Malayalam Using Long Short-Term Memory Units and Convolutional Neural Nets"

They present sentiment analysis of tweets in Malayalam language using CNN and LSTM [7]. The current work is first in its kind in this direction. In the experiment, the CNN is trained using four different filters taken and it presents an evaluation obtained via 10-fold cross-validation. They use four different LSTM cell parameters and three different activation functions such as ReLU, ELU and SELU. It is observed from the experiments that activation functions ELU and SELU improve the scores for CNN and LSTM.

- 9) Soe Yu Maw, May Aye Khine, "Aspect based Sentiment Analysis for travel and tourism in Myanmar Language using LSTM"

They use aspect based sentiment analysis in hotels and restaurants reviews [10]. They have collected reviews, status posts and comments from Facebook pages only for Myanmar language and applied Long Short-Term Memory. They stated that Bi-LSTM doesn't clearly classify the aspect term with context words on aspect based sentiment analysis. They also suggested a hybrid system combining both lexicon based approach and deep learning approach for this purpose.

- 10) Hanane Elfaik and El Habib Nfaoui, "Deep Bidirectional LSTM Network Learning-Based Sentiment Analysis for Arabic Text"

Here, an efficient Bidirectional LSTM Network (BiLSTM) is investigated to enhance Arabic Sentiment Analysis by applying Forward-Backward encapsulate contextual information from Arabic feature sequences [5]. The experimental results on six benchmark sentiment analysis datasets demonstrate that their model achieves significant improvements over the state-of-art deep learning models and the baseline traditional machine learning methods.

- 11) Subarno Pal, Dr. Soumadip Ghosh, Dr. Amitava Nag, "Sentiment Analysis in the light of LSTM Recurrent Neural Networks"

In this paper, they work with different types of LSTM architectures for sentiment analysis of movie reviews [11]. It has been showed that LSTM RNNs are more effective than Deep Neural Networks and conventional RNNs for sentiment analysis. Here, they explore different architectures associated with LSTM models to study their relative performance on sentiment analysis. A simple LSTM is first constructed and its performance is studied. On subsequent stages

the LSTM layer is stacked with one upon another that showed an increase in accuracy. Later the LSTM layers were made Bidirectional to convey data both way forward and backward in the network. They hereby show that a layered deep LSTM with bidirectional connections is having a better performance in terms of accuracy compared to the simpler versions of LSTM used.

3. Methodology

To perform sentiment analysis three methods are there- Rule based, Automatic and Hybrid. Among these, the automatic or machine or deep learning algorithms is implemented in this work and applied supervised learning there. The workflow of the system is as follows...

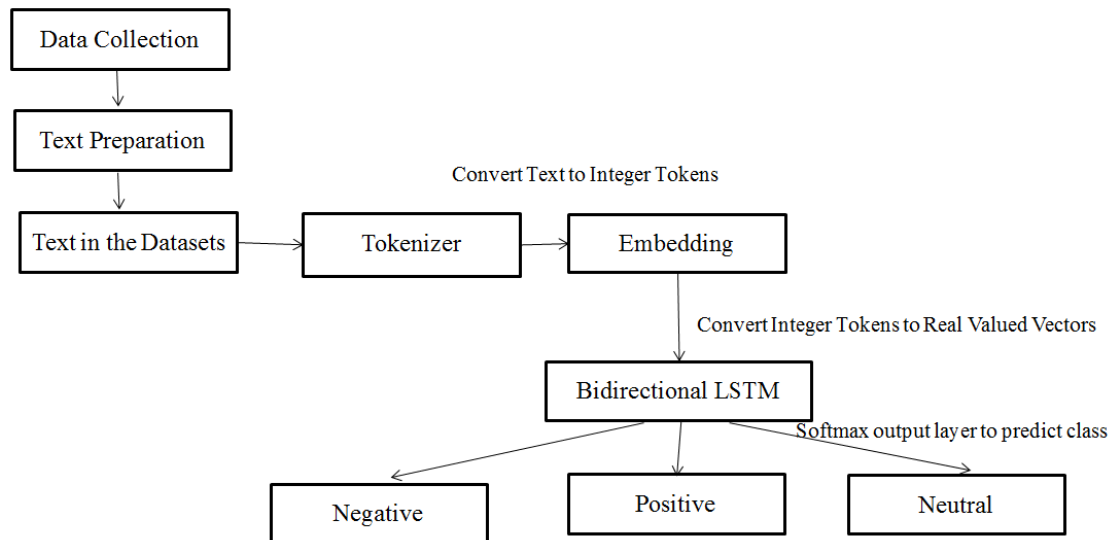


Figure 4. Flowchart of the methodology used.

Step 1: Data Collection

To analyze Assamese text, at first collected 1,00,900 sentences from different social media platform then manually label those sentences as positive, negative and neutral.

Step 2: Data Preparation

Since the datasets is gathered mainly from social media environment in the form of text; so, most of time noisy data is found there such as special characters, symbols, and hyperlinks, punctuations, tags etc. In this preprocessing step, the data in the datasets are filtered before analysis. It includes converting sentences to words, reducing words to its root word, removing noise etc. Then words are tokenized and determined the total number of sentences in the datasets also identifying total numbers of sentences in each of the categories considered- positive, negative and neutral. For this work, Tokenizer from Keras which is a Python Deep Learning library is used.

Step 3: Feature Extraction

As words in the datasets are discrete and categorical, for translating it to model understandable format, need to perform embedding here. And this mapping from text to real valued vectors is known as feature extraction. For this purpose, Keras's text pre-processing library, which convert each

sentence into a sequence of integers is applied. The sentence is mapped to a vector of size s - number of words in the sentence. To have same vector dimension in all sentences zero-padding strategy is used.

Step 4: Hereby divided the datasets in the ratio of 80 and 20 for training and testing purpose respectively.

Step 5: Selecting the Deep learning model

Then classification model is built to predict the classes. Here, Bidirectional LSTM model is used. Bidirectional LSTMs train two sides of the input sequence- left to right on the input sequence and then in reversed order. Thus this model provides one more context to the word to fit in the right context from words coming after and before. `embed_dim`, `lstm_out`, `batch_size`, `drouput_x` variables are considered as hyperparameters. A dropout layer is added here to avoid overfitting along with dense layer having softmax activation as categorical_crossentropy is considered by the model. At the output layer softmax is used to predict the classes.

Evaluation Metrics

To evaluate how efficiently the model is working, the performance evaluation metrics- accuracy, precision, recall, and F1 score is considered.

Table 1. Evaluation metrics.

Correct label		Positive	Negative
Predicted label	Positive	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

Accuracy:

Accuracy identifies the portion of the datasets that are predicted correctly.

$$\text{Accuracy} = (TP+TN) / (TP+FP+TN+FN)$$

Precision:

Precision states how many texts were predicted correctly out of the ones that were predicted as belonging to a given tag. That means it identifies the exactness of the model.

$$\text{Precision} = TP / (TP+FP)$$

Recall:

Recall states how many texts were predicted correctly out of the ones that should have been predicted as belonging to a given tag.

$$\text{Recall} = TP / (TP+FN)$$

F1 score:

The F1 score is the harmonic mean of precision and recall. It tells us how well the classifier performs if equal importance is given to precision and recall.

$$\text{F1 score} = (2 * \text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

4. Result Analysis and Discussion

4.1. Experimental Results

For implementation, the prepared dataset is being divided into two parts for training and testing as shown below.

(80739, 137) (80739, 3)
(20185, 137) (20185, 3)

Figure 5. Training and test data in the dataset.

From the dataset 21946 positive data, 9362 negative data, 49431 neutral data is used for training whereas 2341 negative data, 5486 positive data, and 12358 neutral data for testing.

After implementing the Bidirectional LSTM model on the Assamese dataset, the following result is found...

```
Epoch 1/30
1262/1262 [=====] - 87s 69ms/step - loss: 0.9020 - accuracy: 0.6142
Epoch 2/30
1262/1262 [=====] - 92s 73ms/step - loss: 0.8942 - accuracy: 0.6173
Epoch 3/30
1262/1262 [=====] - 87s 69ms/step - loss: 0.8886 - accuracy: 0.6179
Epoch 4/30
1262/1262 [=====] - 91s 73ms/step - loss: 0.8821 - accuracy: 0.6182
Epoch 5/30
1262/1262 [=====] - 87s 69ms/step - loss: 0.8739 - accuracy: 0.6211
Epoch 6/30
1262/1262 [=====] - 87s 69ms/step - loss: 0.8684 - accuracy: 0.6251
Epoch 7/30
1262/1262 [=====] - 87s 69ms/step - loss: 0.8627 - accuracy: 0.6271
Epoch 8/30
1262/1262 [=====] - 89s 70ms/step - loss: 0.8573 - accuracy: 0.6308
Epoch 9/30
1262/1262 [=====] - 88s 69ms/step - loss: 0.8544 - accuracy: 0.6322
```

Figure 6. Training accuracy and loss.

```

confusion matrix
[[ 100  2128  122]
 [    0 11613  601]
 [   49  5298 274]]

              precision    recall  f1-score   support

    0               0.67       0.04       0.08        2350
    1               0.61       0.95       0.74       12214
    2               0.27       0.05       0.08         5621

 accuracy                   0.59        20185
 macro avg               0.52       0.35       0.30        20185
 weighted avg            0.52       0.59       0.48        20185

```

Figure 7. Confusion matrix.

After plotting the accuracy and loss, the output is...

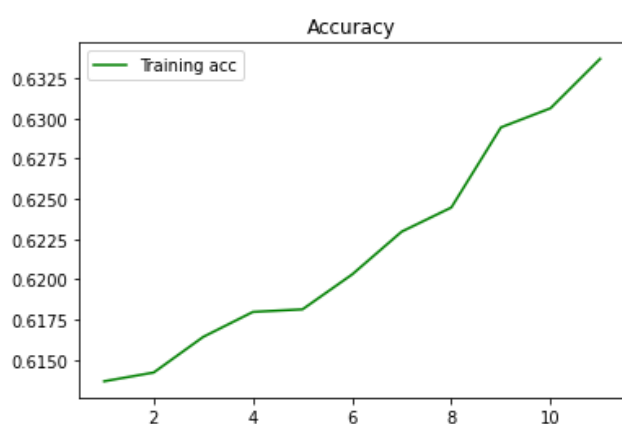


Figure 8. Training accuracy.

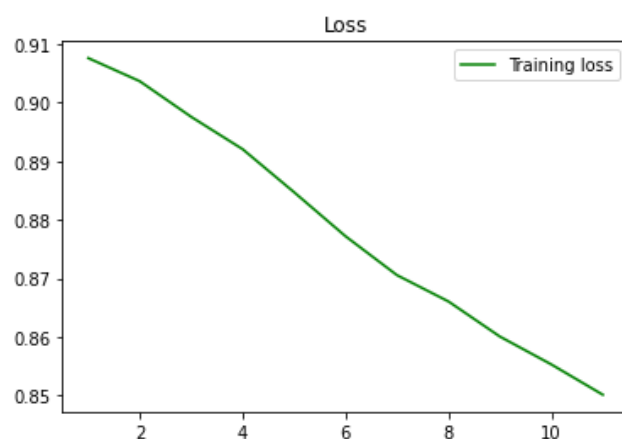


Figure 9. Training loss.

Finally measuring number of correct guesses by the model feeding new sentence to it...

[illegible]

Figure 10. Result of Test data 1.

```
twt = ["সমাপ্তবালভারে আমার আন্দোলনো অব্যাহত থাকিব"]

#vectorizing the tweet by the pre-fitted tokenizer instance
twt = tokenizer.texts_to_sequences(twt)
print(twt)

#padding the tweet to have exactly the same shape as `embedding_2` input
twt = pad_sequences(twt, maxlen=29, dtype='float32', value=0)
print(twt)

sentiment = model.predict(twt)[0] #(model.predict(twt) > 1.0).astype("int32")#
print(np.argmax(sentiment))
if(np.argmax(sentiment) == 0):
    print("negative")
elif (np.argmax(sentiment) == 1):
    print("positive")
elif (np.argmax(sentiment) == 2):
    print("neutral")
```

```
[[26, 134]]
[[ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  26.]]
0
negative
```

Figure 11. Result of Test data 2.

4.2. Result Analysis

After implementing the model on the training data considered and running epochs for 9 times, the following Accuracy and loss is found as shown in the table below.

Table 2. Training accuracy and loss.

Iterations	Accuracy	Loss
1	0.6142	0.9020
2	0.6173	0.8942
3	0.6179	0.8886
4	0.6182	0.8821
5	0.6211	0.8739
6	0.6251	0.8684
7	0.6271	0.8627
8	0.6308	0.8573
9	0.6322	0.8544

As a result, whenever the number of epochs increases, the values of accuracy is also increases whereas loss are decreasing with the increase in number of iterations.

Table 3. Result of the parameters.

	precision	recall	f1-score
0	0.67	0.04	0.08
1	0.61	0.95	0.74
2	0.27	0.05	0.08
Accuracy			0.59

From the confusion matrix generated, for negative sentences value of precision is high compared to other classes whereas for positive sentence both recall and f1-score is more. Also, it is clear that the overall accuracy of the model is 59%.

5. Conclusion and Future Works

5.1. Conclusion

In this work, a datasets of Assamese text from the social media platform is prepared and applied Bidirectional LSTM there to find best results in terms of different performance measures. It is found that the model is working on a moderate rate as compared to the results found in different papers considered for the related study. The overall accuracy of the model is 59%. But from the confusion matrix and the graph

taken, it is clear that the accuracy is increasing as well as loss is decreasing as the number of epochs increases. That means the overall working of the model on the training data is good. While testing, it is found that the given new sentence is categorized as it needs to be.

5.2. Future Works

To improve the accuracy of the model, up sampling may be applied or may manually balance the class label. Also, several variants of LSTM may be implemented by changing the number of layers or tuning the hyper-parameters to improve the overall working of the model.

Abbreviations

NLP: Natural Language Processing
 LSTM: Long-Short Term Memory
 RNN: Recurrent Neural Network
 CNN: Convolutional Neural Network
 TP: True Positive
 TN: True Negative
 FP: False Positive
 FN: False Negative

Conflicts of Interest

The authors declare no conflicts of interest.

References

- [1] Bo Pang and Lillian Lee, Shivakumar Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques", Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Philadelphia, July 2002, pp. 79-86.
- [2] Rudy Prabowo, Mike Thelwall, "Sentiment Analysis: A Combined Approach".
- [3] Betül AyKaraku, Muhammed Talo, IbrahimRızaHalla, GalipAydin, "Evaluating deep learning models for sentiment classification".
- [4] Dr. G. S. N. Murthy, Shanmukha Rao Allu, Bhargavi Andhavarapu, Mounika Bagadi, Mounika Belusonti, "Text based Sentiment Analysis using LSTM", International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181, May-2020.
- [5] Hanane Elfaik and El Habib Nfaoui, "Deep Bidirectional LSTM Network Learning-Based Sentiment Analysis for Arabic Text".
- [6] Wisam Hazım Gwad Gwad, Imad Mahmood Ismael Ismael, Yasemin Gültepe, "Twitter Sentiment Analysis Classification in the Arabic Language using Long Short-Term Memory Neural Networks", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958, Volume-9 Issue-3, February, 2020.
- [7] S. Sachin Kumar (B), M. Anand Kumar, and K. P. Soman, "Sentiment Analysis of Tweets in Malayalam Using Long Short-Term Memory Units and Convolutional Neural Nets", January 2017.
- [8] Sonali Rajesh Shah, Abhishek Kaushik, "SENTIMENT ANALYSIS ON INDIAN INDIGENOUS LANGUAGES: A REVIEW ON MULTILINGUAL OPINION MINING", November 26, 2019.
- [9] Abhishek Bhagat, Akash Sharma, Sarat Kr. Chettri, "Machine Learning Based Sentiment Analysis for Text Messages", IJCAT - International Journal of Computing and Technology, Volume 7, Issue 6, June 2020 ISSN (Online): 2348-6090.
- [10] Soe Yu Maw, May Aye Khine, "Aspect based Sentiment Analysis for travel and tourism in Myanmar Language using LSTM".
- [11] Subarno Pal, Dr. Soumadip Ghosh, Dr. Amitava Nag, "Sentiment Analysis in the light of LSTM Recurrent Neural Networks", International Conference on Information Technology and Applied Mathematics, 2017.