A Generative Approach of Sentiment Analysis based on Deep Learning

Shahriar Hasan Chowdhury
170104030@aust.edu

Tasmiyah Binta Islam
170104033@aust.edu

Sadia Tasnim
170104037@aust.edu

Sabiha Benta Sayed
170104051@aust.edu

***Abstract*—Sentiment analysis on social networks has become an influential means of knowing about the opinions of users and it has a broad range of implementation. Nevertheless, the efficiency and accuracy of sentiment analysis are being obstructed by the challenges confronted in natural language processing. Recently, it has been manifested that deep learning models are a favorable solution to the challenges of NLP. For the project, the IMDB dataset has been taken for training the deep learning models for sentiment analysis. The dataset is divided into 50% positive and 50% negative reviews and using some deep learning models for text classification like RNN, LSTM, Bidirectional LSTM, and also by using CNN, an acceptable outcome is conveyed in this project.**

I. Introduction

Sentiment analysis is generally a process where there is a determination about a collection of writing being positive, negative, or neutral. It studies the subjective information in an expression like opinions, emotions, or attitudes towards any topic. For example, "I appreciate the way you talk", is a positive attitude towards a person. The main purpose of sentiment analysis is to evaluate the expressive direction of the reviews correctly.

The improvement and growth of various websites and social networks in user-generated content have escorted an expanding power of different social networks to give opinions about their outcomes or services among others. So, this turned the opinions of others into a valuable asset as it can affect any brand reputation, customer experience, or market strategy. Sentiment analysis can help the data analysts to scale and monitor these situations and work on them. Also, there are many approaches regarding this like, movie reviews, customer services, product review analysis, etc. With the help of sentiment analysis, the opinions or emotions can be measured without reading a huge stack of comments or texts in various areas at once. To analyze this massive task, there would be a need for natural language processing which can classify the sentiments and opinions of a text.

Though sentiment analysis is important, it can be challenging in many ways. For example, human tones can be difficult to detect verbally, so to figure out that in written words is more difficult. Also, there is polarity, sarcasm, negations, and so on which can lead the machine to confusion in detecting the true context of the implications of such responses. Then on social media platforms, people tend to express their emotions using various emojis. As NLP is language-specific, the special characters of the emojis tend to get removed in the text processing part, which does not give a proper insight for analyzing the actual result. Our job is to overcome these challenges and build a model for sentiment analysis using deep learning algorithms.

II. Related Works

The purpose of this study is to review different approaches and methods in sentiment analysis that can be taken as a reference in future empirical studies. We have focused on key aspects of research, such as technical challenges, datasets, the methods proposed in each study, and their application domains. Summaries of some research papers that we read are given here in the below section.

The objective of this paper [1] was to develop a model for sentiment analysis and to compare the performance of the algorithms. The authors used three deep learning networks (CNN, RNN, LSTM) to classify movie reviews from the IMDB dataset into positive and negative categories. The dataset contained 50000 reviews from the online movie database, from which 25,000 were used for training and another 25,000 were for testing. To ensure better fairness, negative and optimistic comments each was accounted by 50%. They found that the Convolutional neural network gave a good performance for processing a sequence of data, achieving an accuracy of 88.22% with a loss of 0.3.

The accuracy here is quite good with the CNN model, but it could be improved by increasing the training data. They used 50% data for training from 50000 reviews which limit the data for training. Also, for preprocessing they only removed punctuation, but more preprocessing could be done here to improve the performance.

The following paper [2] described a deep learning system for sentiment analysis of tweets. The authors worked on a new model which initialized the parameter weights of the CNN. They made a comparison between the results of their approach and the system. They used the ReLU activation function and a simple max-pooling in their model training and used the word2vec tool in the word embedding process. For the dataset, they used 50M tweets and got an accuracy of 84.79 by which they ranked 1st in Semeval-2015 for phrase and tweet-level subtasks.

The authors of the following paper [3] explored the predictive power of historical news sentiments based on financial market performance to forecast financial news sentiments. They used the pre-trained word vector of GloVE based on Wikipedia 2014 for word embedding. By training RNN with LSTM units, they showed that the model predicts the positive news as positive and the negative news as negative, on average. They also stated that beyond 200k iterations, the accuracy of the random selection case fluctuates around 95%. But the accuracy of the hierarchical selection case fluctuates around 97.5%. This proved that the hierarchical selection case performs better in this case.

In the next paper [4], the author suggested a hybrid model which is CNN-SVM for sentiment analysis. They used two datasets for training(Twitter Hindi, movie review Hindi) and got maximum accuracy of 62.52% and 65.96%. As the dataset size was small, the accuracy was not enough. As deep learning models are data-driven models, increasing the dataset might improve the accuracy of this process.

III. Project Objective

The focus of our project is the deep learning approach, which is a subset of machine learning. It has a huge variety of processes to recognize patterns within natural language. For our project, we planned a workflow that helped us to go through a proper way to complete our proposed target.
The flow chart is given below.

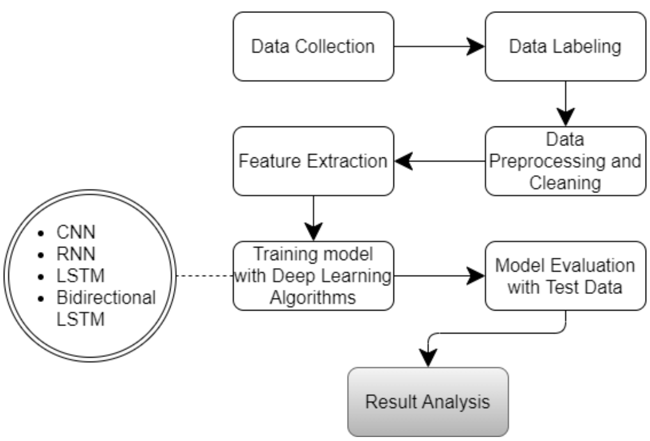

Fig. 1. Work Flow Diagram

After training the model, the model should be able to classify between the positive and negative sentiments. Here is a dummy input and output given below for understanding our proposed system,

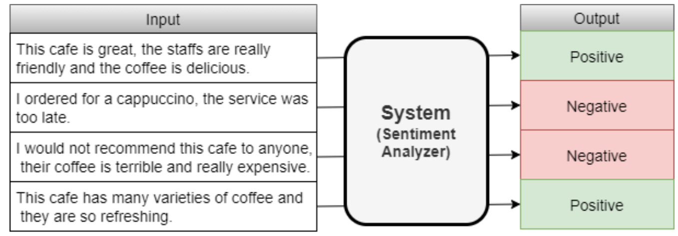

Fig. 2. Dummy input output

IV. Methodology

After reviewing the methods of sentiment analysis, we came to decide the models which we will be using to classify the sentiments from the reviews. We chose to use Convolutional Neural Network, Recurrent Neural network, Long Short-Term Memory, and Bidirectional LSTM. These models could provide high accuracy working with the text classification dataset comparing to others, which are explained in the following,

A. Convolutional Neural Network (CNN)

A convolutional neural network is a special type of feed-forward neural network originally employed in areas such as computer vision, recommender systems, and natural language processing. Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are convolutional layer, pooling layer, and fully-connected (FC) layer [5]. The first layer is the convolutional layer which can be followed by additional convolutional layers or pooling layers, and the final layer is the FC layer. The earlier layers prioritize simple features, while with the increasing number of layers, the complexity increases. The data progresses through the layers and learns for its intended task at last.

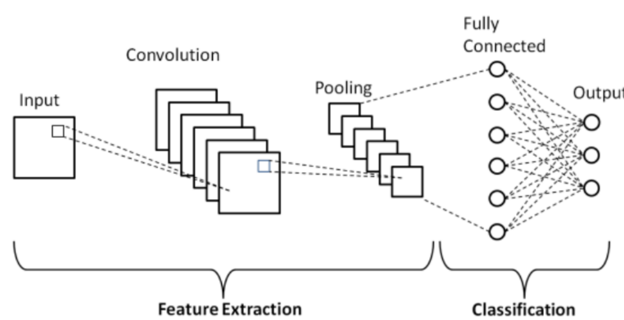

Fig. 3. Schematic diagram of a basic CNN architecture [6]

B. Recurrent Neural Network (RNN)

RNN (Recurrent Neural network) is a type of Neural Network. Here output from the previous step will be used for the current step as input. Traditionally in neural networks, all input and outputs are independent of each other. But when we need to remember the previous word to predict the next word of a sentence, it is a different case. RNN can solve this problem by using a hidden layer. The hidden state is the most important state for RNN, which can remember some information about a sequence. To remember what has been calculated RNN has a memory. So, this can remove the complexity of parameters, unlike other neural networks.

RNN is the first algorithm that proceeds to remember the input given as containing an internal memory so that it is ideally suited for machine learning problems involving sequential data.

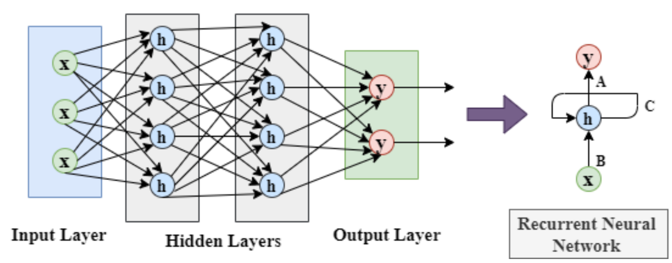

Fig. 4. Simple Recurrent Neural Network

Here, the nodes x, h, y in the separate layers of the neural network are compacted to generate a single layer that is of recurrent neural networks. A, B, and C are the parameters of the network. So far, this is the conversion into a Recurrent Neural Network from a feed-forward neural network.

C. Long Short-Term Memory (LSTM)

Long short-term memory networks are a special kind of recurrent neural network that are competent in order dependence learning of prediction problems that are sequenced. This network is kind of a complex area in deep learning. LSTM has a chain-like structure but the repeating model is quite different in structure as it does not have a single neural network layer but four interacting layers which can communicate extraordinarily.

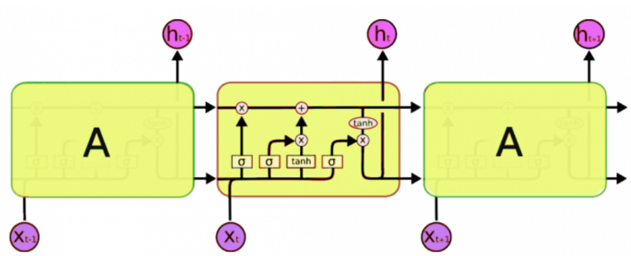

Fig. 5. Architecture of LSTM Network [8]

LSTM network is a composite version of the various memory blocks that are showed in the picture which are called cells. Here, the two states, the cell state, and the hidden state are transmitted to the next cell. For remembering things, the memory blocks are present and these memory manipulations are done across three major mechanisms which are called gates. The gates are Forget gate, Input gate, and Output gate.

- Forget gate: This gate is in charge of removing information from the cell state. As per requirement, the information that is no longer needed for the LSTM to acknowledge things or the information that is no longer of much importance is removed through a multiplication of a filter.
- Input gate: This gate works for adding information to the cell state. After the processing for the addition of information, it is ensured that the cell state consists of the information that is important and not dispensable.
- Output gate: The pieces of information that run across the cell state, all of the information are not fit to be the output at a specific time. This output gate makes sure that the requirements for output are ready as called for by the filter which is built on the input and hidden state values and that is applied in the cell state vector.

To sum up, LSTM has the required properties which make it able to be an improvement over recurrent neural networks.

D. Bidirectional Long Short-Term Memory

Bidirectional LSTM is a sequence processing model consisting of two LSTMs, one is for taking the input in a forward direction and another is for the backward direction. It can enhance the performance of a model on sequence classification problems. It trains two LSTMs instead of one on the input sequence. By this, additional context is added to the network and results are faster.

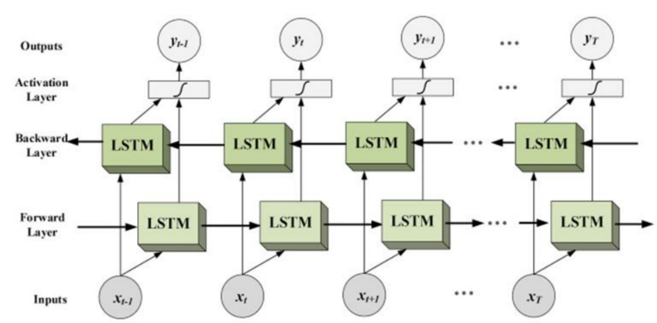

Fig. 6. Architecture of Bi-LSTM Network [9]

Using this bidirectional, inputs will be run in two ways, one from past to future and one from future to past. It is a different approach than LSTM because, in the case of LSTM, it runs backward to preserve information from the future. But in BiLSTM, by using the two hidden states combined it is possible to preserve information from both past and future at any point.

V. Experiments

A. Dataset

- **Statistics of the dataset**

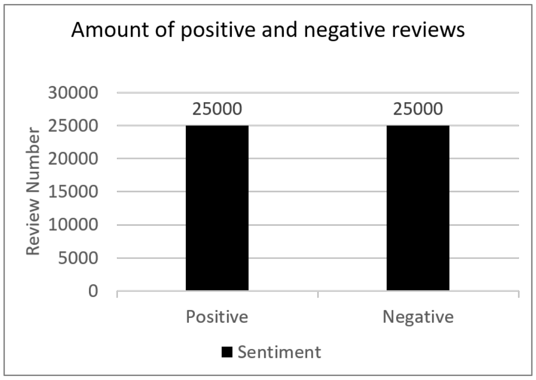

Fig. 7. Statistics of reviews

This is a graphical representation of our dataset where the classification type and quantity is indicated.

- **Samples of dataset**
In the given table, we have made a scenario of our dataset with the classes and labels to get a proper understanding of the collection type of dataset.

TABLE I
Dataset Preview

Serial	**Review**	**Sentiment**
4	I thought this was a wonderful way to spend time ... great comedy to go see with friends.	positive
9	This show was an amazing, fresh & innovative idea ... awful. I can't believe it's still on the air.	negative
16	This a fantastic movie of three prisoners ... I recommand this movie to everybody. Greetings Bart	positive
19	This movie made it into one of my top 10 most awful movies. ... Save your money.	negative
49966	I saw this last week during Bruce Campbell's book ... I highly recommend it.	positive
50000	I'm going to have to disagree ... technological progress can bring about.	negative

- **Train-Test Separation in Dataset**

TABLE II
Train Test Split

	Train Data	**Test Data**
Positive	12500	12500
Negative	12500	12500

We followed the way of splitting the dataset according to the paper we followed, though we divided the train and test data once to 70% and 30% respectively for CNN in between the experiment to better the performance.

B. Evaluation Metric

We have processed the dataset by removing stopwords, punctuations, tags, mentions, links, etc. Also, we used lemmatization for finding the root form of a word. Following the paper, we also measured the performance metrics for CNN, RNN, and LSTM and tried to better the performance. Also, we used the Bi-LSTM model to train the model and got a result for it.
Here is a table given below, where the accuracy, precision, recall, f1-score for the models we got is displayed.

TABLE III
Model Performance

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
CNN	88.27%	88.56%	87.81%	88.19%
RNN	79.88%	80.52%	78.35%	79.43%
LSTM	85.54%	84.46%	87.09%	85.76%
Bi-LSTM	86.59%	83.91%	90.69%	87.17%

We also prepared the confusion matrix for each model we trained in our system, which is displayed below,

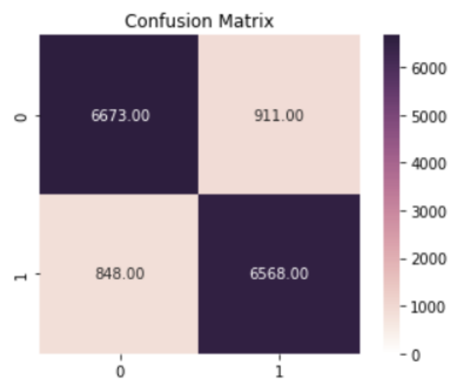

Fig. 8. Confusion Matrix for CNN

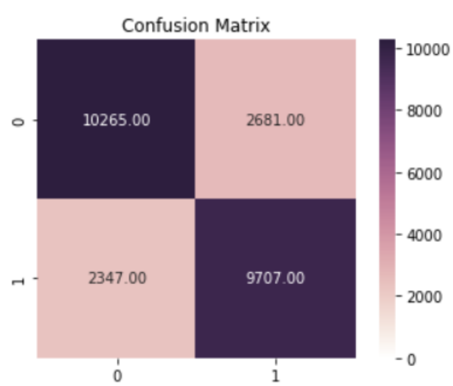

Fig. 9. Confusion Matrix for RNN

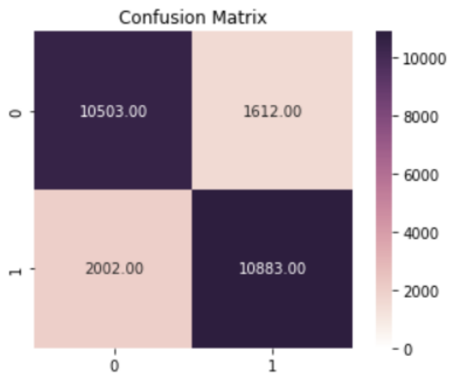

Fig. 10. Confusion Matrix for LSTM

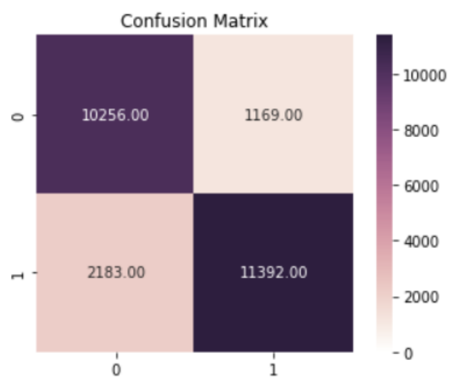

Fig. 11. Confusion Matrix for Bi-LSTM

As we followed a paper to build this system, the performance results of the models varied and by making an accuracy comparison table, an overview is given here,

TABLE IV
ACCURACY COMPARISON BETWEEN PAPER AND TRAINED MODELS

Model	Paper	Generated
CNN	88.22%	88.27%
RNN	68.84%	79.88%
LSTM	85.32%	85.54%

As we trained the model, we got the results rather almost the same as the paper or better than the paper. From the given tables, we can see the performance percentages for each model. We got better performance in some cases because we used more techniques than the paper for data preprocessing. Generally, we trained and tested the model with 50000 reviews, keeping 50% to train and 50% to test, but the results could be improved, if we increased the training data, for example, 80% for training and 20% for testing.

C. Result Analysis

- Accuracy Comparison among the models:

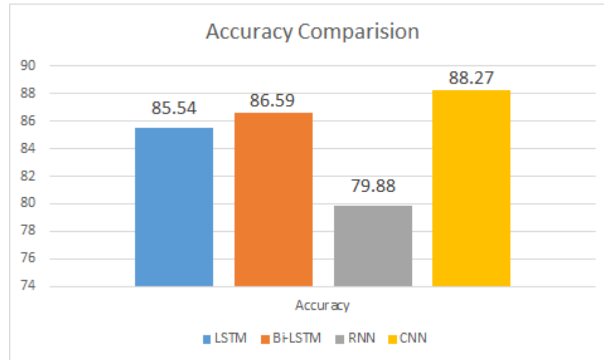

Fig. 12. Graphical representation of accuracy

Here, we can see CNN has the highest accuracy which is 88.27% and RNN has the lowest accuracy which is 79.88%.

- Accuracy curve:

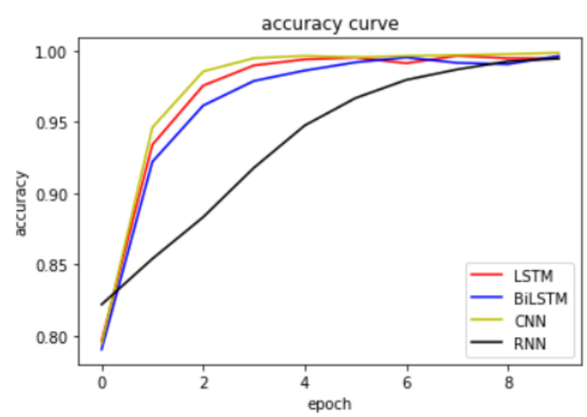

Fig. 13. Accuracy vs Epoch

- Loss curve:

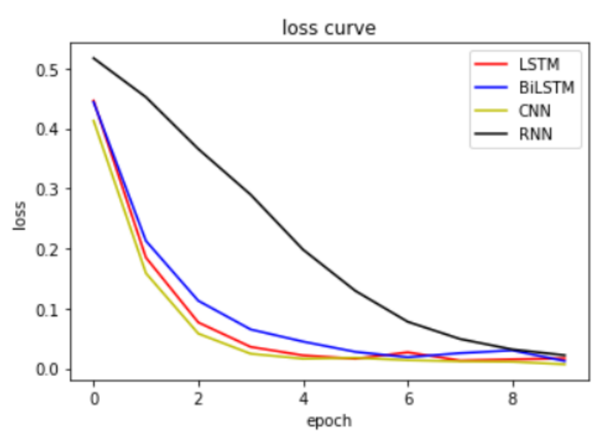

Fig. 14. Loss vs Epoch

From the above graphical representations, we can see that CNN has the highest accuracy of 88.27% where the loss was 0.0024. We did 20 epochs for each model in our experiments and from the curves above we can see that after almost 4 epochs we get the highest accuracies for LSTM, BiLSTM, and CNN.

- Ablation Experiment:
  We got the best performance using convolutional neural network, but we had to change some hyperparameters and others while we were training the model to get the accuracy percentage. Here is a table given below as an ablation experiment for CNN,

TABLE V
EXPERIMENT ON CNN

	Exp1	**Exp2**	**Exp3**	**Exp4**	**Exp5**
Sequence Length	2000	4000	4000	4000	4000
Hidden Layer	2	2	2	2	2
Hidden Nodes	250	250	250	250	250
Batch size	128	128	128	128	128
Epochs	20	20	20	50	50
Activation Function	tanh	tanh	tanh	tanh	tanh
Stopword removal	Yes	Yes	No	No	No
Train:Test	50:50	50:50	50:50	50:50	70:30
Accuracy	86.38%	86.83%	87.58%	87.44%	88.27%
Loss	0.0029	0.0044	0.0041	0.0026	0.0024

VI. Conclusion and Future Work

Deep learning algorithms have completely changed the field of speech and image recognition as they tend to give better performance in lengthy data than general machine learning algorithms. In this project, we have used 4 deep learning algorithms (CNN, RNN, LSTM, and Bi-directional LSTM) to classify movie reviews using IMDB dataset into positive and negative sentiments. Performing this experiment we found that CNN works best on this particular dataset. It gives 88.27% accuracy with a loss of 0.0024. Dividing the dataset into a 7:3 ratio with preprocessed data helped to increase the accuracy.

In our project, we have used only the count vectorizer for feature extraction, we could use the TF-IDF vectorizer to get better accuracy in the future. For future, we could also split the dataset into 8:2 ratio as we have seen in CNN, the 7:3 ratio gave better accuracy than the 5:5 ratio. So, if we train the model with an increased train-set, the model could give better performance.

References

[1] X. Ouyang, P. Zhou, C. H. Li and L. Liu, ”Sentiment Analysis Using Convolutional Neural Network,” 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, 2015, pp. 2359-2364, doi: 10.1109/CIT/IUCC/DASC/PICOM.2015.349.

[2] Severyn, Aliaksei, and Alessandro Moschitti. ”Twitter sentiment analysis with deep convolutional neural networks.” Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. 2015.

[3] Souma, Wataru, Irena Vodenska, and Hideaki Aoyama. ”Enhanced news sentiment analysis using deep learning methods.” Journal of Computational Social Science 2.1 (2019): 33-46.

[4] Akhtar, Md Shad, et al. ”A hybrid deep learning architecture for sentiment analysis.” Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. 2016.

[5] What are Convolutional Neural Networks? https://www.ibm.com/cloud/learn/convolutional-neural-networks/. Accessed: October 02, 2021.

[6] Neural Networks Methods for image classification - BIT Blog. https://blog.bitsathy.ac.in/neural-networks-methods-for-image-classification/. Accessed: October 02, 2021.

[7] Bi-directional RNN & Basics of LSTM and GRU. https://medium.com/analytics-vidhya/bi-directional-rnn-basics-of-lstm-and-gru-e114aa4779bb/. Accessed: October 02, 2021

[8] Long Short Term Memory — Architecture Of LSTM. https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep learning-introduction-to-lstm/. Accessed: October 02, 2021.

[9] Deep Dive into Bidirectional LSTM. https://www.i2tutorials.com/deep-dive-into-bidirectional-lstm/. Accessed: October 02, 2021.