# Artificial Intelligence and Machine Learning

# (6CS012)

## Deep Learning RNN

Student Id : 2227097

Student Name : Radhika Neupane

Group: L6CG10

Module Leader : Siman Giri

Lecturer : Siman Giri

Tutor : Sunita Parajuli

Submitted on: 14 May 2024

## Title

TripAdvisor Sentimental Analysis using RNN

## Abstract

This report has the overall process done for the sentiment analysis on hotel reviews for tripadvisor by creating models with multiple layers. We also have the process of building a model along with training it. Lastly it is evaluated and prediction is made based on it. Additionally this report has the feasibility of using the RNN on “Trip Advisor Hotel Reviews” for sentiment analysis done on **2227097_RadhikaNeupane_Code_TextClassification**.

# 1. Introduction

This report presents sentiment analysis for the text data in  Recurrent Neural Network (RNN). In this sentiment analysis we have identified the positive and negative analysis with the reviews of the customers and understand the users sentiments towards the services of the hotel.

The primary objective of this analysis is sentiment classification model capability for accurately classifying the sentiment.

The model architecture we have used is a very common architecture called sequential model with four layers i.e., Embedding Layer, LSTM layer, Dense layer and output layer.

In this report we have detailed description of the text data used for training and evaluation along with the model development and model evaluation. Finally we have discussed the results and found the summary of sentiment analysis.

# 2. Methodology

For the methodology we have provided the detailed process of preparing the data for analysis and the training for the sentiment classification model.

## 2.1. Data Pre-Processing

In this section we have provided the process of cleaning and preparing the model along with pre processing techniques.

We have removed the unwanted characters along with texts, urls, emoji and so on since these kinds of things introduced noise for our sentiment analysis. We have also normalized the text by stopping the stop words and all. Stemming and Lemmatization are also used in order to reduce the base form of words. Likewise we have word clouds for the visualization in order to explore the most frequent words in our data.

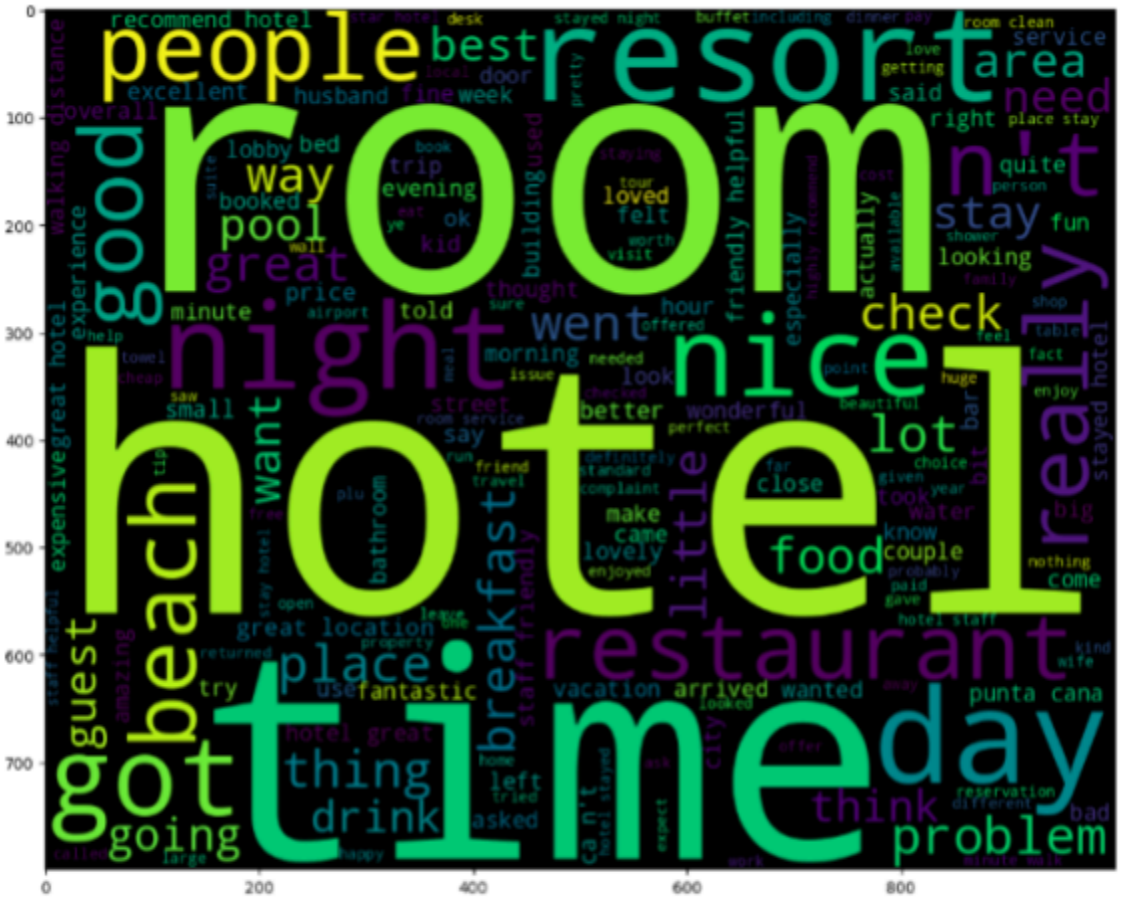



Fig 1: Word Cloud

The text data are also splitted into tokens and the sequences are padded to ensure the compatibility with the models.

## 2.2. Model Building

In this section we will be discussing the process of building the model along with the training process and its evaluation.

### 2.2.1. Model Summary:

We have four typical layers for the sequential model using the network called LSTM (Long Short-Term Memory). We have the embedding size, 100 which determines the dimensionality of the word vectors where each of the words will be represented by the

100 dimensional vector. Then we have an LSTM layer with 256 of the hidden units. Similarly we have a dense layer where we have three fully connected dense layers with the ReLU(Rectified Linear Unit). We have the decreasing number of i.e., 128, 64 and 32 which transform the retracted features progressively. Finally we have the output layer with one unit and activation function sigmoid.With the help of activation function we get the output of a probability score between 0 and 1 which represents the sentiment for the review that is either positive or negative.

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, 100, 100)          6663500

 lstm_3 (LSTM)               (None, 256)               365568

 dense_12 (Dense)            (None, 128)               32896

 dense_13 (Dense)            (None, 64)                8256

 dense_14 (Dense)            (None, 32)                2080

 dense_15 (Dense)            (None, 1)                 33

=================================================================
Total params: 7072333 (26.98 MB)
Trainable params: 7072333 (26.98 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig 2 : Model Summary

2.2.2. Training of the models

For the training purpose we have used binary cross-entropy which is one of the common choices when it comes to any kind of binary classification of the tasks in the sentiment analysis. As we are well known that the binary represents 0 and 1 so for this model prediction we also have 0 and 1. The model is trained with the adam optimizer since it is one of the most popular and most effective optimizers used in deep learning in order to find the final and optimal solutions. We have 10 epochs for our model where

the entire dataset is passed in the model for the learning which means that the model will iterate for the 10 times in the training data.

```
Epoch 1/10
257/257 [==============================] - 336s 1s/step - loss: 0.6703 - accuracy: 0.5720 - val_loss: 0.6850 - val_accuracy: 0.5633 - lr: 0.0010
Epoch 2/10
257/257 [==============================] - 305s 1s/step - loss: 0.6875 - accuracy: 0.5569 - val_loss: 0.6847 - val_accuracy: 0.5633 - lr: 0.0010
Epoch 3/10
257/257 [==============================] - 308s 1s/step - loss: 0.6866 - accuracy: 0.5567 - val_loss: 0.6820 - val_accuracy: 0.5633 - lr: 0.0010
Epoch 4/10
257/257 [==============================] - 299s 1s/step - loss: 0.6697 - accuracy: 0.5636 - val_loss: 0.6533 - val_accuracy: 0.5860 - lr: 0.0010
Epoch 5/10
257/257 [==============================] - 305s 1s/step - loss: 0.5865 - accuracy: 0.6991 - val_loss: 0.5709 - val_accuracy: 0.7033 - lr: 0.0010
Epoch 6/10
257/257 [==============================] - 305s 1s/step - loss: 0.5000 - accuracy: 0.7677 - val_loss: 0.5223 - val_accuracy: 0.7568 - lr: 0.0010
Epoch 7/10
257/257 [==============================] - 308s 1s/step - loss: 0.4115 - accuracy: 0.8225 - val_loss: 0.6705 - val_accuracy: 0.7365 - lr: 0.0010
Epoch 8/10
257/257 [==============================] - 309s 1s/step - loss: 0.3216 - accuracy: 0.8700 - val_loss: 0.6209 - val_accuracy: 0.7631 - lr: 0.0010
Epoch 9/10
257/257 [==============================] - 318s 1s/step - loss: 0.2831 - accuracy: 0.8859 - val_loss: 0.6345 - val_accuracy: 0.7553 - lr: 0.0010
Epoch 10/10
257/257 [==============================] - 312s 1s/step - loss: 0.1978 - accuracy: 0.9255 - val_loss: 0.7381 - val_accuracy: 0.7448 - lr: 0.0010
<keras.src.callbacks.History at 0x7f6eb2dba8c8>
```

Fig 3 : Training Model

Figure of training and validation loss

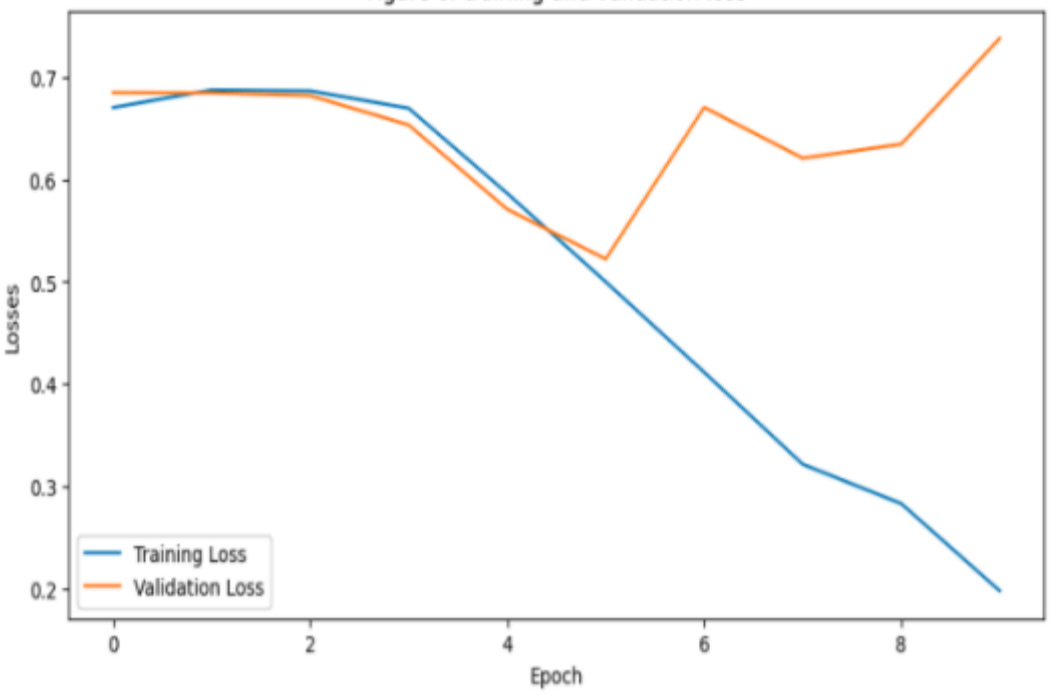
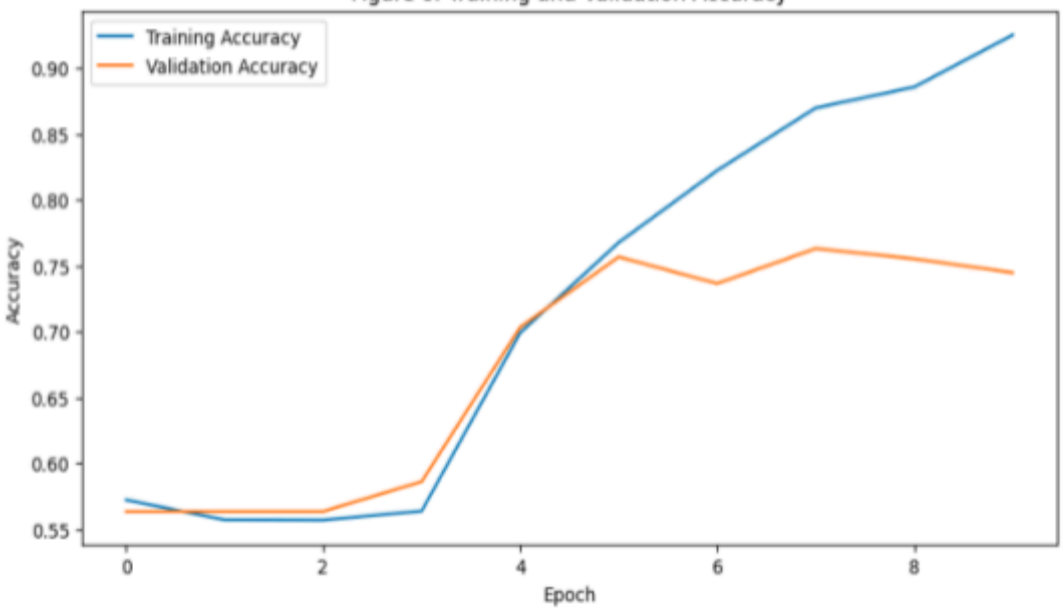



Figure of Training and Validation Accuracy



2.2.3. Evaluation of the model

Looking at the loss we have a value around 0.7380 and the accuracy is 74.48%. Now from the confusion of true positive,true negative,false positive and false negative we have true positive 1774 and true negative 1279 which means our model prediction is pretty good enough. The false positive is 535 and false negative is 511 which shows the model incorrectly.

```
129/129 [==============================] - 19s 147ms/step - loss: 0.7381 - accuracy: 0.7448
Validation Loss: 0.7380915284156799
Validation Accuracy: 74.48%
129/129 [==============================] - 24s 186ms/step
Accuracy = 74.48%
```

Fig 4 : Loss and Accuracy

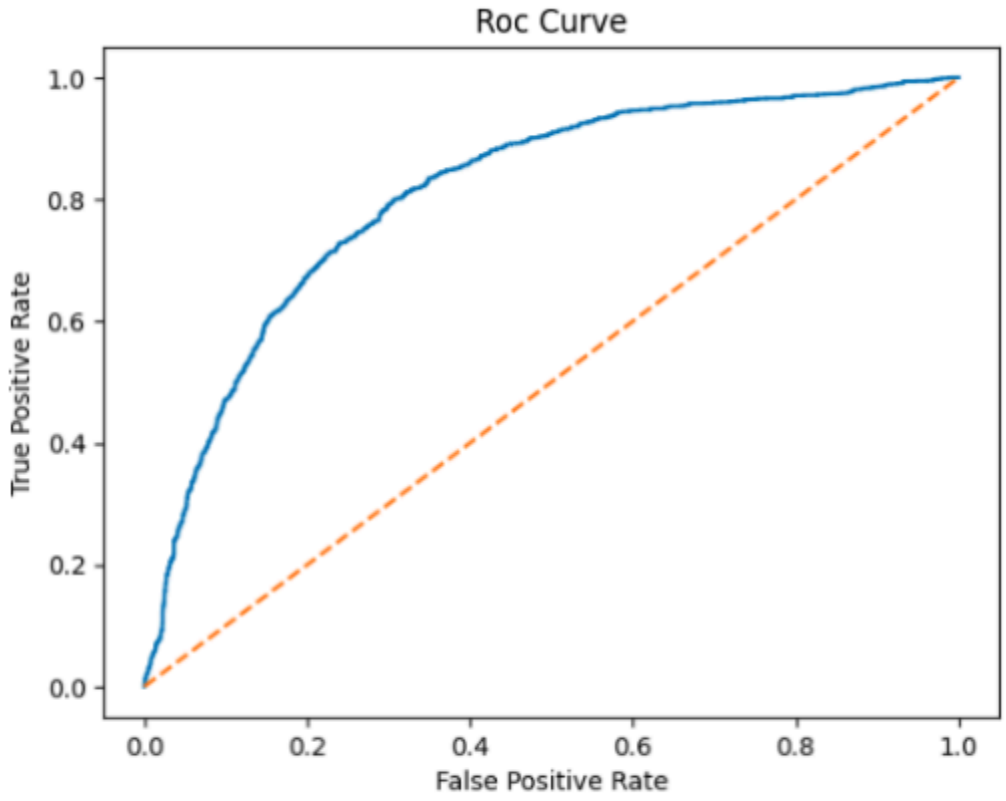



Fig 5 : Roc Curve

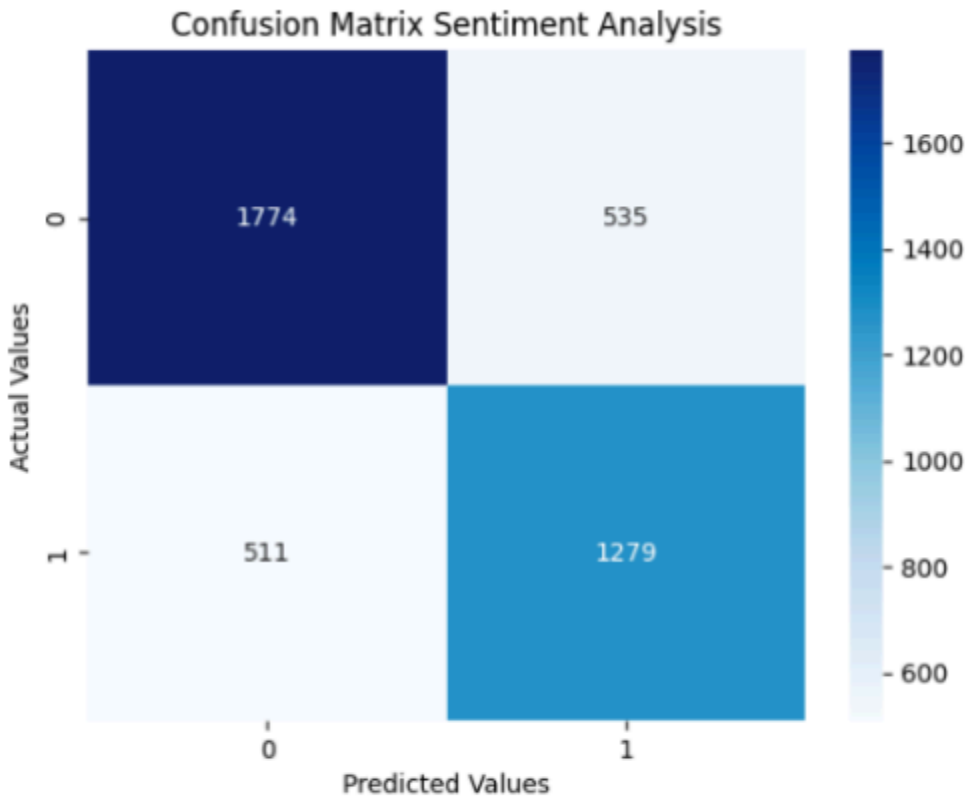



Fig 6 : Confusion Matrix

Here is the prediction for our input text and its prediction for the data.

```
1/1 [==============================] - 0s 66ms/step
Input Text: very bad hotel bad environment tooo
Binary Prediction: Negative

Input Text: nice hotel
Binary Prediction: Positive
```

Fig 7 : Prediction of input text

## 3. Final Discussions

In conclusion, around 74.48% accuracy in the training and the validation accuracy shows us that the model is somehow working well. The AUC-ROC score shows us the metric in order to distinguish the positive and negative sentiment for the model. The confusion matrix also shows us the specific classification errors(e.g., false positives or false negatives).