

Sentiment Analysis Using LSTM

PROJECT REPORT

Created by

Abdullah Raheel (18K-0170)

Mohit Kumar (18K-1160)

As part of the course

Deep Learning for Perception (CS-452)

Work presented to

Dr. Jawwad Shamsi

To The Department of Computer Science Faculty of the Department of Computer Science National

University of Computing and Emerging Sciences

## 

Contents

[Abstract 3](#_Toc73422370)

[1. Introduction 3](#_Toc73422371)

[a. Why sentiment analysis is important? 4](#_Toc73422372)

[2. Design 5](#_Toc73422373)

[a. Architecture of Proposed Network Used 5](#_Toc73422374)

[3. Methodology 6](#_Toc73422375)

[a. Raw test 6](#_Toc73422376)

[b. Tokenization 7](#_Toc73422377)

[c. Embedding 8](#_Toc73422378)

[d. Bi-Directional LSTM 9](#_Toc73422379)

[e. Softmax 9](#_Toc73422380)

[4. Results 10](#_Toc73422381)

[a. Output 10](#_Toc73422382)

[b. Summary 10](#_Toc73422383)

[c. Accuracy 11](#_Toc73422384)

[**i.** **Confusion Matrix** 13](#_Toc73422385)

[**ii.** **Classification Report** 13](#_Toc73422386)

[5. Conclusion 14](#_Toc73422387)

[6. References 14](#_Toc73422388)

**Abstract:**

Analyzing the big textual information manually is tougher and time-consuming. Sentiment analysis is an automated process that uses computing (AI) to classify emotions from the text. Sentiment analysis is widely used for getting insights from social media comments, survey responses, and merchandise reviews to create data-driven decisions. Sentiment analysis systems are accustomed to adding up to the unstructured text by automating business processes and saving hours of manual processing. In recent years, Deep Learning (DL) has garnered increasing attention within the industry and academic world for its high performance in various domains. Today, Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) are the foremost popular types of DL architectures used. We do sentiment analysis on text dataset by using Bi-Directional Long Short-Term Memory (LSTM). Recently, thanks to their ability to handle large amounts of knowledge, neural networks have achieved a good success on sentiment classification.

# Introduction

Sentiment analysis is that the computerized process of the higher cognitive process to an opinion a couple of given subjects from a transcription. in an exceedingly present generation, we create quite 1.5 quintillion bytes of information daily, sentiment analysis has become a key tool for creating a sense of that data. it absolutely was utilized by the businesses to induce key insights and automate every kind of process for their business development. Sentiment Analysis is also called opinion mining. Sentiment analysis isn't only a sentiment mining but also contextual mining of text which identifies and extracts subjective information in source material and helping a business to know the social sentiment of their service, brand or product while monitoring online conversations. Sentiment Analysis is that the most used text classification tool that analyses an incoming message and classifies the essential opinion. Sentiment analysis will be applied at different levels of scope like Document-level sentiment analysis obtains the sentiment of an entire document or paragraph. Sentence level sentiment analysis obtains the results of one sentence. Sub-sentence level sentiment analysis obtains the results of sub-expressions within a sentence.

## Why sentiment analysis is important?

It’s estimated that 80% of the world’s data is unstructured and not organized during a pre-defined manner. Most of this comes from text data, like reviews, emails, chats, social media, surveys and articles. These texts are usually difficult and time-consuming to investigate and understand. The sentiment analysis system authorizes company to create sense of this huge amount of unstructured text by automating business processes, saving hours of manual processing and getting actionable insights. Recurrent Neural Networks (RNNs) are one of the most prevalent architectures because of the ability to handle variable-length texts. Humans can't analyze from scratch every second. Any human can understand each word based on his understanding of previous words. He doesn’t throw everything away and start thinking from scratch again. His thoughts have persistence. Traditional neural networks can’t do this, and it seems like a speed process is coming. For example, imagine a human want to classify what kind of event is happening at every point in a movie. It’s not clear how a traditional neural network could use its reasoning about previous events in the film to inform later ones. Recurrent neural network addresses can face this type of issues. They are networks with multiple loops in them, allowing information to continue. Though RNNs are capable of modeling long sequential data theoretically they fail to represent long sequences in real time applications [1]. Recently, LSTM is most popular to deal with sentiment classification. LSTM is proposed by Hoch Reiter and Schmid Huber in 1997 and was refined and popularized by many people in the following work. They work tremendously well on large different types of problems and are now widely used. LSTMs are explicitly designed to ignore the long-term dependency problem [2]. Remembering information for a long time is practically their default behavior, not something they struggle to learn. All recurrent neural networks have the form of a chain of repeating modules of the neural networks. In the level of RNNs, this repeating module having a very simple structure, such as a single tanh layer. Our model uses relatively small dataset of personal emotional comments scrapped through internet.

# Design

Long short-term memory (LSTM) is a synthetic recurrent neural network (RNN) architecture employed in the sphere of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. LSTM networks are well-suited to classifying, processing, and making predictions supported statistic data since there may be lags of unknown duration between important events in a very statistic. LSTMs were developed to accommodate the exploding and vanishing gradient problems that may be encountered when training traditional RNNs. Relative insensitivity to gap length is a bonus of LSTM over RNNs, hidden Markov models, and other sequence learning methods in numerous applications. There are several architectures of LSTM units. a typical architecture consists of a cell (the memory a part of the LSTM unit) and three "regulators", usually called gates, of the flow of knowledge inside the LSTM unit: an input gate, an output gate and a forget gate. Some variations of the LSTM unit don't have one or more of those gates or even produce other gates. as an example, gated recurrent units (GRUs) don't have an output gate.

LSTM with a forget gate The compact forms of the equations for the forward pass of an LSTM unit with a forget gate are:

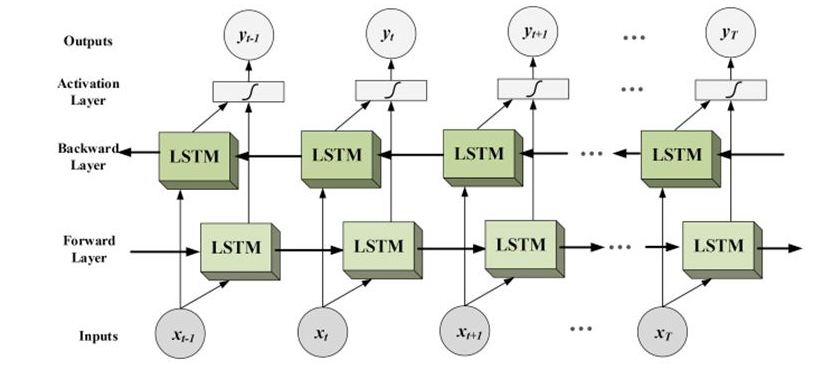
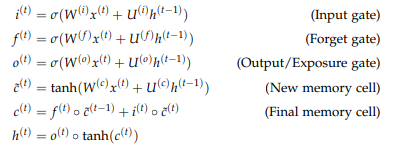


Figure Bi-Directional LSTM



## Architecture of Proposed Network Used

## 

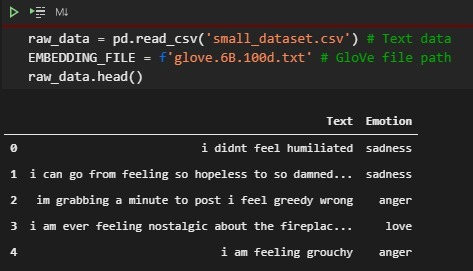
Figure (System Architecture Design)



# Methodology

## Raw test

A Dataset of emotional internet comments has been used to train and validate our model. Total of 21458 comments are labelled into 6 classes for example Sad, Anger, Love, Surprise, Fear and Happy. Dataset is uneven in nature. Comments are unevenly distributed among these 6 classes. Dataset is locally stored in machine and then loaded through pandas data frame to preprocess the data. i.e., We convert the text into lower case, remove the punctuations and separate the individual comments and store them in individual list elements.



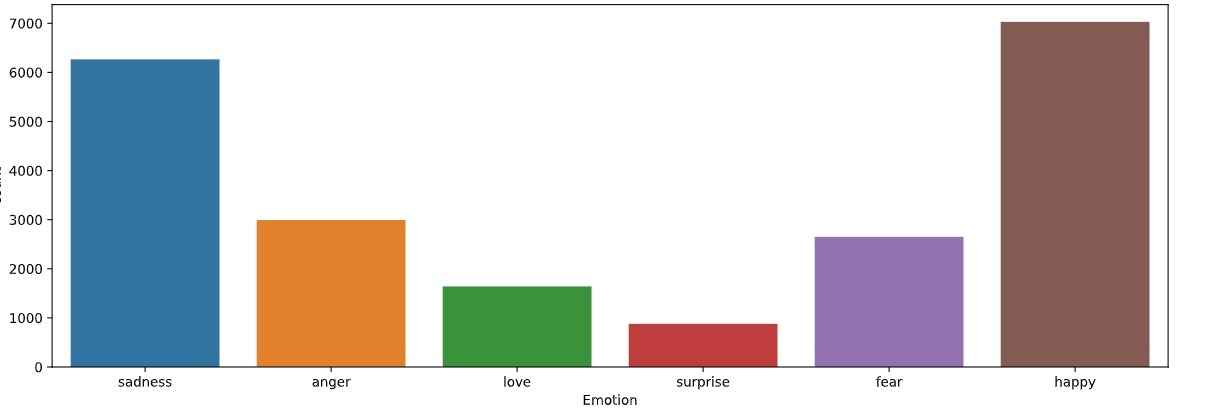
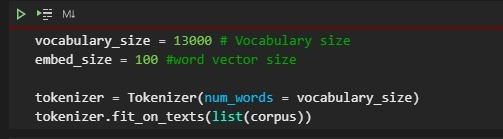
Figure 3 Sample of Used Dataset

Figure Distribution of Dataset

## Tokenization

Tokenization is that the process of tokenizing or splitting a string, text into an inventory of tokens. One can consider token as parts sort of a word could be a token in a very sentence, and a sentence could be a token in every paragraph.  
In our model natural language toolkit stopwords is used to remove the common words that don’t hold any meaning e.g., the, a, he, have etc. words are stemmed and corpus of only useful words is created and stored in csv format. This corpus of words is then Tokenized into integer format to be further encoded and fed into neural network.



## Embedding

Figure Text is Tokenized.

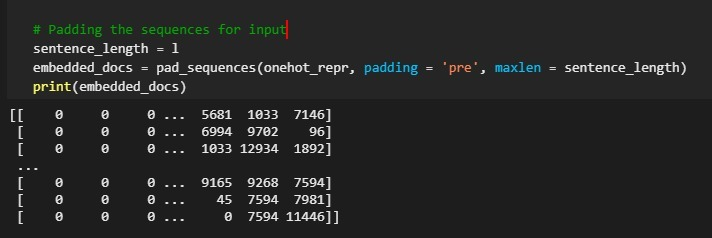
Word Embedding emerged from the field of Natural Language Processing (NLP) which is an intersection of Computer Science, Artificial Intelligence, Machine Learning and computational linguistics. Word embedding is a text mining technique of establishing relationship between words in textual data (Corpus), Given the corpus of tokenized integers indices, embedding layer is used to create dense vectors. Embedding Layer is then used to embed classification labels so that data can be effectively fed to CNN. For our model we have used glove a pre-trained word embedding layer. Output vector is further represented into one hot encoding representation. Furthermore, Padding is applied to one hot representation of output vector in order to provide the input vector of fixed shape to neural network.

Figure Padded One hot representation of output vector.

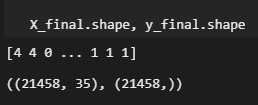


Figure Final Shape of data to be fed to Model.

## Bi-Directional LSTM

A Bidirectional LSTM is a sequence processing model that consists of two LSTMs: one taking the input in a forward direction, and the other in a backwards direction. BI-LSTMs effectively increase the amount of information available to the network, improving the context available to the algorithm (e.g., knowing what words immediately follow and precede a word in a sentence).

An individual LSTM works in following way

* + 1. Take input the current input, the previous hidden state and the previous internal cell state.
    2. Calculate the values of the four different gates by following the below steps: -
* For each gate, calculate the parameterized vectors for the current input and the previous hidden state by element-wise multiplication with the concerned vector with the respective weights for each gate.
* Apply the respective activation function for each gate element-wise on the parameterized vectors. Below given is the list of the gates with the activation function to be applied for the gate.
  + 1. Calculate the current internal cell state by first calculating the element-wise multiplication vector of the input gate and the input modulation gate, then calculate the element-wise multiplication vector of the forget gate and the previous internal cell state and then adding the two vectors



* + 1. Calculate the current hidden state by first taking the element-wise hyperbolic tangent of the current internal cell state vector and then performing element wise multiplication with the output gate.



Just like Recurrent Neural Networks, an LSTM network also generates an output at each time step and this output is used to train the network using gradient descent. The only main difference between the Back-Propagation algorithms of Recurrent Neural Networks and Long Short-Term Memory Networks is related to the mathematics of the algorithm.

## Softmax

Softmax function is used for Multilabel Classification models. Softmax layers are good at determining multi class probabilities. After working of data through bi-directional LSTM output returned through LSTM is feeded to Softmax function. Softmax Function returns the probability of all the classes. Our target class will have High probability.

# Results

## Output

Given an input Text our model returns sentiment class.

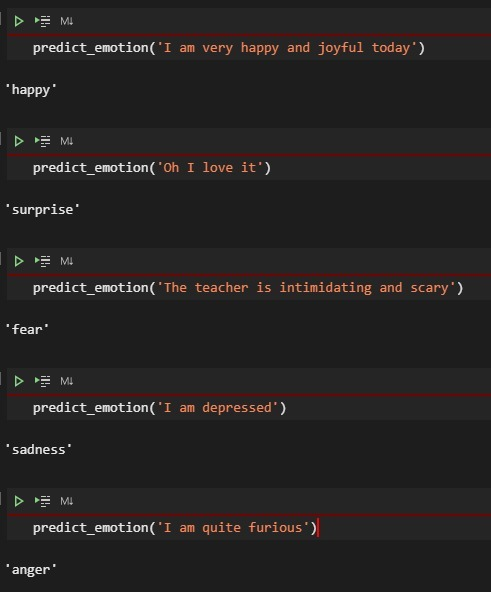


Figure 8 Sample Output generated by model

## Summary

We have trained a Sequential Keras model having embedding layers to which embedded vectors are provided as output, Output of embedded layer is provided to BI-LSTM. Activation function of relu along with L1 regularization is applied on output of BI-LSTM. Finally, Softmax function is applied for Prediction

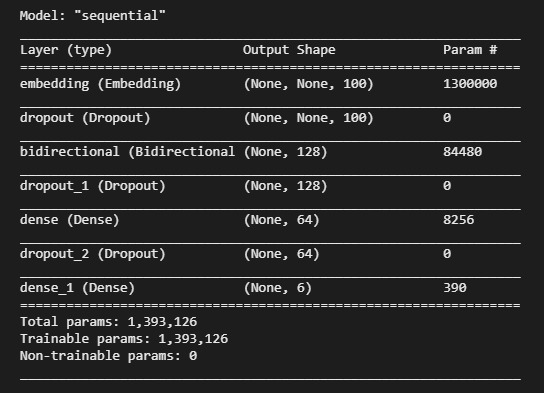
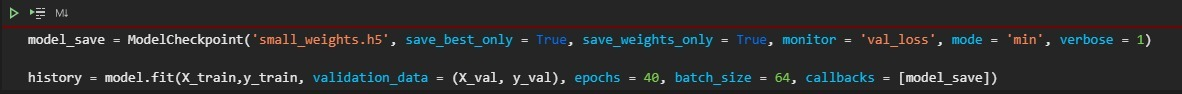


Figure 9 Model Summary

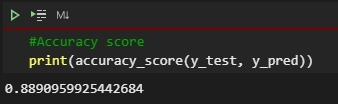
Model is trained Monitoring value of Loss Function using metrics of accuracy and Adam optimizer with learning rate of 0.001, Loss Function of sparse categorical cross entropy is applied on model.



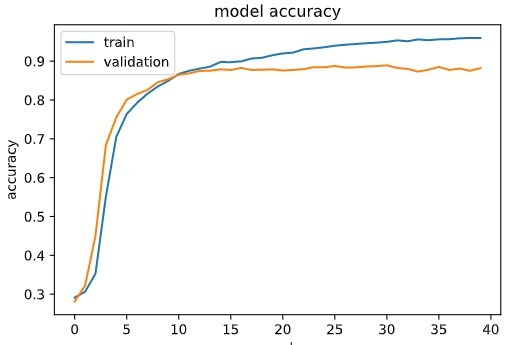
## Accuracy

Accuracy of model is calculated by comparing predicted values with real values.

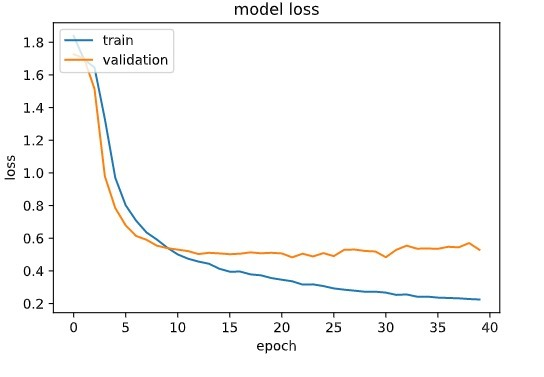
Our model gives the accuracy of 88.9%



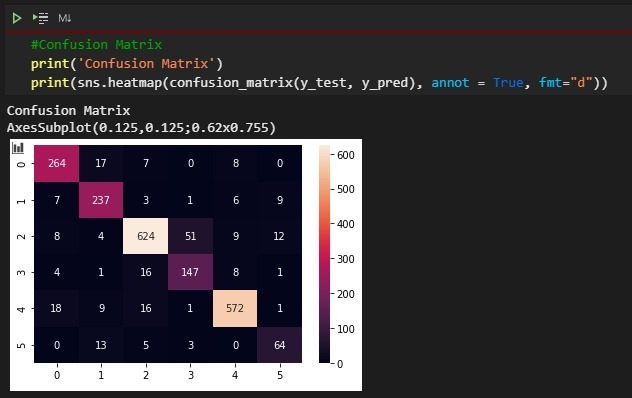
**Model Accuracy Graph**



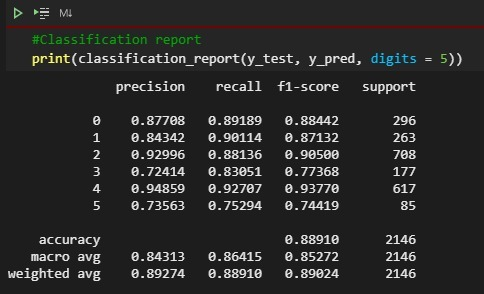
**Model Loss Graph**



### **Confusion Matrix**



### **Classification Report**

Classification Report tells the accuracy of prediction of each individual class and how the model accuracy is evaluated.

# Conclusion

In this paper, we have proposed a sentiment classification approach based on BI-LSTM for text data. Users from all over the world express and publicly share their opinions on different topics. Manual analysis of large amounts of such data is very difficult, so a reasonable need for their computer processing has emerged. DL methods such as BI-LSTM show better performance of sentiment classification with 89% accuracy when there are more amounts of training data.

In future we are planning to extend this study to a larger extent where different embedding models can be considered on large variety of the datasets.

# References

[1] <https://doi.org/10.1007/s12088-011-0245-8>

[2] Hoch Reiter S, Schmidhuber, J. (1997). Long short-term memory. In: Neural Computation 9(8): 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>