

Movie Sentiment Analysis using Deep Learning - RNN

Nirsan Amal A¹, Vijayakumar A²

¹Student, ²Professor,

^{1,2}Jain Deemed-to-be-University, Department of Masters of Computer Application,
School of Computer Science and Information Technology, Bengaluru, Karnataka, India

ABSTRACT

Sentimental analysis or opinion mining is the process of obtaining sentiments about a given textual data using various methods of deep learning algorithms. The analysis is used to determine the polarity of the data as either positive or negative. This classification can help automate data representation in various sectors which has a public feedback structure. In this paper, we are going to perform sentiment analysis on the infamous IMDB database which consists of 50000 movie reviews, in which we perform training on 25000 instances and test it on 25000 to determine the performance of the model. The model uses a variant of RNN algorithm which is LSTM (Long Short Term Memory) which will help us make a model which will decide the polarity between 0 and 1. This approach has an accuracy of 88.04%

KEYWORDS: LSTM, Recurrent Neural Networks, Sentiment Analysis, Opinion Mining

How to cite this paper: Nirsan Amal A | Vijayakumar A "Movie Sentiment Analysis using Deep Learning - RNN" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-5 | Issue-4, June 2021, pp.891-896, URL: www.ijtsrd.com/papers/ijtsrd42414.pdf



Copyright © 2021 by author (s) and International Journal of Trend in Scientific Research and Development Journal. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0) (<http://creativecommons.org/licenses/by/4.0>)



I. INTRODUCTION

Sentiment Analysis is the process of obtaining sentiments from the sequence data. We as humans are subjective and biased with creating emotions based on situations and contexts. We are an emotional being, we need emotions to drive us on a daily basis. It affects every sphere of life. In this paper we are going to discuss how sentiment analysis will help us mine opinions about sequential data. Movies are an integral part of entertainment, we watch them for entertainment as well as artistic expressions used in the films. The performance of the films are usually measured by the collection in the box office and hence the film is concluded as either hit or flop. This method of analysis is quite flawed and requires much deeper analysis to truly understand the public opinion about the film. Hence Sentiment Analysis (SA) or Opinion Mining (OM) proves to be an effective tool of analysis on the overall performance of the movies based on the reviews which reflect the opinions of the people towards the film. For the analysis, we are going to train a model which will use the IMDB database which consists of review and sentiment parameters of about 50000 records in which 25000 will be used for training the model and the remaining 25000 will be used for testing purposes. We are going to use this model to predict the reviews based on the polarity between positive and negative. An index between 0 to 1 will be used to determine the sentiment in the input review. As for the algorithm, RNN algorithm is preferred as it is proven to be more accurate than traditional neural network algorithms. LSTM cells will be used as it can hold memory for a longer period. Hence a model with an higher precision and accuracy can be acquired.

II. EXISTING SYSTEM

The current system consists of review star system which lets the user input manually about the film between 1 to 5 stars which will determine the performance of the movie. This system is partially impaired as ratings doesn't involve the factors like the story, screenplay, and other specific details about the movie which can be analysed using the sentiment analysis system which will be used for the review system which will help the film makers about the specifics of the movie which went good and which didn't. So an overall sentiment can be analysed over a certain aspect of the movie. The existing sentiment analysis system has lower accuracy than the human accuracy benchmark of 80% hence more accurate and precise models are required in order to effectively collect review data and make an analysis of the film modularly. The existing system requires an automation system using deep learning model which will be more effective in movie review analysis. Reviews are generically either positive or negative, but there are exceptions where the review is both positive and negative. This exception is ambiguous and is hard to classify. The data on the votes on movies from 1920 to 2020 has a downward trend as the best movie performances are comparatively studied in order to find a solution to improve the quality of the movies delivered to the audience. This graph shows how much the sentiment has declined over a century.

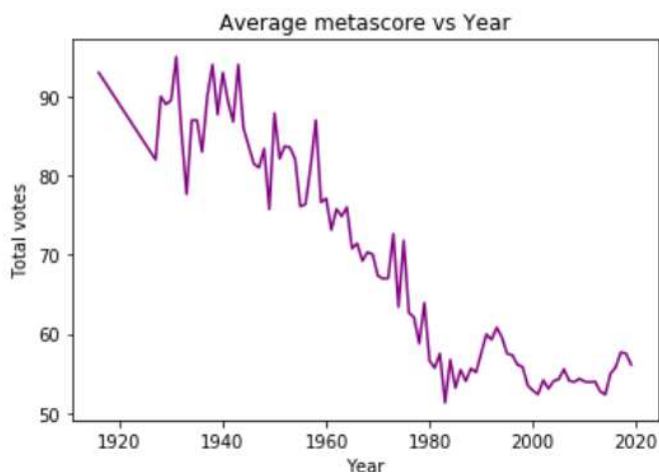


Fig.2. 1 Graphical Representation of best movies over a century

Hence a solution is required to improve the quality of the movies by providing deep analysis of the movies using reviews to the filmmakers so that they can recalibrate on their methods on creating a good movie. A generic sentiment analysis consists of various parameters including all humane emotions. This gives a overall view on the sentiments rather than a binary view of the sentiments, As to work on the simplifications of the model only two polarity of positive and negative is used in this paper.

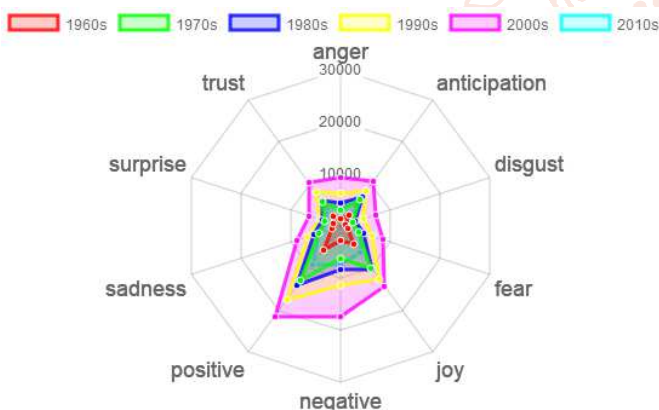


Fig.2.2 Sentiment analysis

A. Convolutional Neural Networks

Convolutional Neural Networks or CNN is one type of deep neural network. It is basically designed for visual imagery. This type of neural network is prominent in computer vision. It enables us to classify images better like a human brain. The ability of image classification is a strength of this neural networks. However it was used in Sentiment analysis aslo. It has achieved an accuracy of 85 percent. The CNN architecture consists of an input layer, hidden layer and an output layer. In convolution network the hidden layer are crucial as they perform convolutions. Generally this layer will perform dot product on the convolution kernel from the input layer. The kernel goes along the input matrix of the layer while generating a feature map.

A feature map, or activation map, is the output activations for a given filter and the definition is the same regardless of what layer we are on. This follows by other layers such as pooling layer, normalization layers, fully connected layers. This constitutes the architecture of a typical CNN. In CNN the results of one layer is passed onto the next layer. Before convolutions to take place the input layer takes in the input and transforms them into tensors which will have shapes : (number of inputs) x (input height) x (input width) x (input

channels). Then it goes under feature mapping process so that the network can process the given input. There are various parameters to the process such as weights, channels etc. All these parameters constitute to the architecture of the model. As a part of Neural Networks, it is designed in a way to imitate human brain connectivities. It imitates the process of how neurons communicate with each other. CNN is feed forward based and it is efficient in processing images, it used various methods to implement it on text classification as well using the maxpooling layers etc. The size of the output volume is determined by three hyper-parameters known as padding size, stride, depth. The depth of the volume is the dense neurons which are collectively packed in the same region as the input volume. Stride influences the column heights and weights. Padding helps in controlling the volume of output. While input volume taken as W, kernel field size K, stride taken as S and zero padding on the border taken as P The number of neurons that can fit in a given volume is

$$\frac{(W - K + 2P)}{S} \oplus 1$$

A parameter sharing scheme is used in convolutional layers to control the number of free parameters. It relies on the assumption that if a patch feature is useful to compute at some spatial position, then it should also be useful to compute at other positions. Denoting a single 2-dimensional slice of depth as a depth slice, the neurons in each depth slice are constrained to use the same weights and bias. Another important concept of CNNs is pooling, which is a form of non-linear down-sampling. There are several non-linear functions to implement pooling, where max pooling is the most common. It partitions the input image into a set of rectangles and, for each such sub-region, outputs the maximum. Intuitively, the exact location of a feature is less important than its rough location relative to other features. This is the idea behind the use of pooling in convolutional neural networks. The pooling layer serves to progressively reduce the spatial size of the representation, to reduce the number of parameters, memory footprint and amount of computation in the network, and hence to also control over fitting. This is known as down-sampling It is common to periodically insert a pooling layer between successive convolutional layers (each one typically followed by an activation function, such as a ReLU layer) in a CNN architecture. While pooling layers contribute to local translation invariance, they do not provide global translation invariance in a CNN, unless a form of global pooling is used All these combined influence the working of the neural network of CNN. CNN is based on the convolution kernel or filter which plays a prominent role in the entire network. In text classification, the text is passed to the CNN where it is passed over to the embedding layer of embedding matrix. GlobalMaxPooling1D layers are applied to each layer. All the outputs are then concatenated. A Dropout layer then Dense then Dropout and then Final Dense layer is applied. This is how a CNN can handle textual sequence data. Various filters are used on the textual data for mapping different vocabularies. This method has been experimented over sentiment analysis as well. But the disadvantages of CNN prevented accurate models and it is quite difficult to implement. It became quite tricky after a while. Hence a better and efficient algorithm is sought after. This brings us to modern RNN like LSTM which has back propagation feed which is quite effective as it can hold memory for a longer time. This proved quite useful in the field of sentiment analysis. CNNs use more hyper parameters than a standard

multilayer perceptron (MLP). While the usual rules for learning rates and regularization constants still apply, the following should be kept in mind when optimizing. Since feature map size decreases with depth, layers near the input layer tend to have fewer filters while higher layers can have more. To equalize computation at each layer, the product of feature values v_a with pixel position is kept roughly constant across layers. Preserving more information about the input would require keeping the total number of activations (number of feature maps times number of pixel positions) non-decreasing from one layer to the next. The number of feature maps directly controls the capacity and depends on the number of available examples and task complexity. Regularization is a process of introducing additional information to solve an ill-posed problem or to prevent over fitting. CNNs use various types of regularization.

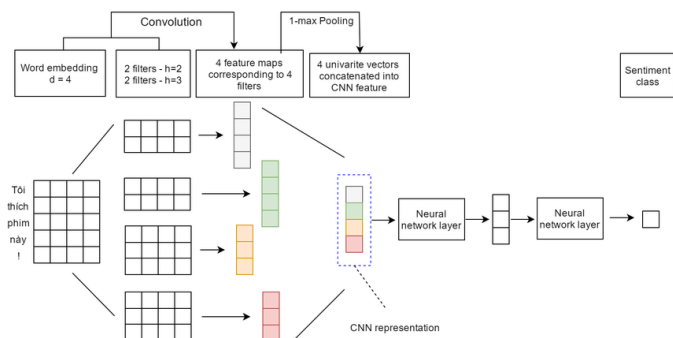


Fig2.3 CNN Architecture

The applications of CNN varies from field to field. They are basically bedrock of deep learning. They became the predecessors for modern deep learning algorithms. In image recognition systems CNNs are predominantly used. Using MNIST and NORB database it is stated that the prediction is really fast, similarly a CNN named AlexNet also performed equally compared with the above. The facial recognition is also one of the crucial area in which CNN can be used. It is proven that the error rate has been decreased while using CNNs. This laid a foundation for modern facial recognition systems. 97.6 % recognition rate has been confirmed on 10 subjects of nearly 5600 still images. The video quality is observed manually by the CNN so that the system had a less root mean square error. The benchmark for object classification and detection is the ImageNet Large Scale Image Recognition challenge which consists of millions of images and object classes. GoogleLeNet so far has the best performance of average precision of 0.439329, and the classification error is reduced to 0.06656. CNN used on ImageNet is close to human’s performance for detection. However there are problems which affect the performance of CNNs such as images distorted with filters which is a common phenomenon in modern images. In contrast, humans cannot accurately classify breeds of dogs and species of flowers as accurate as the CNN. Hence CNN has been better classifier of objects till date. Many layered CNN in 2015 has been used for face detection from almost every angles and proved to be efficient in detection of faces. A database of 200,000 images with various angles and orientations with another 2 million images without faces has been used to train the network. A batch of 128 images of over 50,000 iterations has been used. CNNs also play a vital role in the video domain. There are

comparitively less study made on the video classification than the image domain. One approach towards video classification is to have time and space as equal dimensions of the input. There is also another method in which we use use two CNNs i.e one for the spatial stream and another for the temporal stream. LSTM infused with the model to account for inter-clip dependencies is typically used to achieve this. The application of CNN has also evolved towards natural language processing as well. It is by this means a predominantly image classification and detection algorithm is introduced to the world of Natural Language Processing. It has been successful so far in regards to semantic parsing, search query retrieval, sentence modeling, classification, prediction and other traditional NLP tasks. It is also used in anomaly detection in videos as well. In order to achieve this, a CNN with 1D convolutions was used over time series in frequency domain with unsupervised model to detect anomalies in the time domain. It is also an useful tool on drug discovery as the model will be trained on identifying the reaction between biological proteins and molecules. This has helped in discovering potential treatment to a disease. A simple CNN was combined with Cox-Gompertz proportional hazards model and used to produce a proof-of-concept example of digital biomarkers of aging in the form of all-causes-mortality predictor. The application of CNN is very vast and researchers are using deep learning algorithms to further develop systems that will help us in every aspect of science and technology. In our experiment, the use of CNN for text classification has achieved an accuracy less than Recurrent Neural Networks or RNN. We have realised the CNN’s limitations on processing Natural Language Processing.

III. PROPOSED SYSTEM FOR MOVIE SENTIMENT ANALYSIS

RNN is the best approach towards modelling sequential data as it processes basic inputs with its internal memory states

$$ht = f(ht-1, xt) = \tanh(whhht-1 + wxh xt)$$

RNN can be used in the long sequent data theoretically they cannot effectively work with real time applications, this could be because of the inadequate gradient disadvantage. It uses back propagation through time (BPTT) For addressing the problems in RNN in real time applications LSTM or Long Short Term Memory is introduced, as it can effectively used to make models which has longer sequences at interval. Four gates are used in the data flow of LSTM. It uses different gates to see what proportion of the new information should be supplemental to the state cell (input(i)), the previous cell is forgotten (forget(f)), gate(g) and output(o) gate at the side of the cell (c) and hidden state (st) state. σ represents provision sigmoid.

Following formulas are the state values at each gate.

$$i = \sigma(xtUi + st-1Wi)$$

$$f = \sigma(xtUf + st-1Wf)$$

$$o = \sigma(xtUo + st-1Wo)$$

$$g = \tanh\sigma(xtUg + st-1Wg)$$

$$ct = ct-1of + goi$$

$$st = \tanh(ct)oo$$

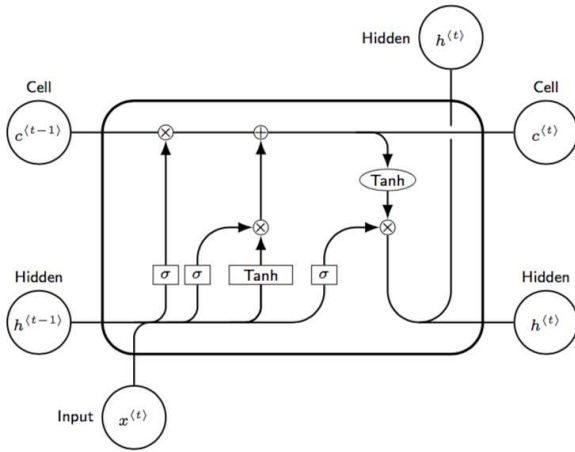


Fig 3.1 LSTM cell

When comparing with other neural networks such as CNN, GRU, LSTM proves to be more accurate than the above mentioned algorithms hence it has been decided for the model. Word embedding process is looked over by the word2vec embedding. Word2vec is basically a technique used in NLP (Natural Language Processing), which involves vectorizing the words by creating a neural network which will consist of word associations that are derived from a large set of textual data. As machines cannot words like humans, it is necessary to parse them into values which can be understood by machines. Hence word embeddings are used as the input to the model. Word2vec consists of a two layer neural link which will contain a vectore sapce created from the large set of textual dataset which will be used a linguistic model for the machine. It can be done with two model architecture for word distributions namely CROW (Continoious Bag of words) and Continious skip gram. In CROW the current word is predicted from the surrounding word corpus regardless of the order of the words. In continuous skip gram the surrounding words are predicted using the current word, in this method also the order is not an influencing factor. While CBOW is faster skip gram can be used for uncommon words. A tokenisor is also used in this model. The model will consist of 4 layers namely Embedding layer, 2 LSTM cell layer and a Dense Layer. This model will be trained with 15 epochs over IMDB dataset in order to achieve a precise model.

A. Word Tokenisation

Tokenization is the process of seperating textual data as tokens in order to conduct natural language processing efficiently. Without tokenisation it is hard for the machine to understand textual data. Most deep learning algorithms like RNN, GRU process the sequential data as tokens. This token level approach is essential for data processing. RNN recieves and processes tokens under a given time step. The tokenization outputs a vocabulary from the corpus. Vocabulary refers to the set of unique tokens. It can be constructed by considering each unique token in the corpus or by considering the top K Frequently Occurring Words. In this experiment the word tokenisation is used as the tokenization technique in which a pretrained word tokenization like word2vec is used. It consists of pretrained data that is obtained from conducting tokenization on a larger word corpus. This is not completely fool proof. The problem arises when an OOV occurs. The OOV refers to the Out of Vocabulary i.e which is not encountered during the training phase. The OOV will create a huge impact on the testing set as it is not aware of the new vocabulary used in the test data. This could even affect the accuracy of the

process. Hence a rich word tokenization is used in this model to prevent such scenarios. The character tokenization prevents the OOV problem yet it is not suitable for the above model as the length of input and output sentences increases a lot since we are using this method to characterise the entire sentence, This will prevent the model to bring meaningful sentences. Hence word tokenization i.e word2vec is used in order to be efficient in sentence vectorization.

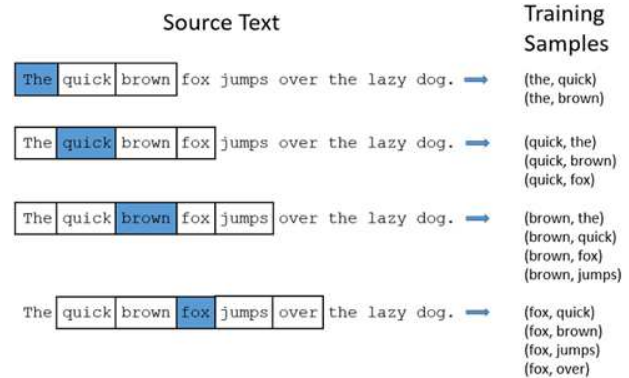


Fig.3.2 Architecture of CNN for text processing

The application of tokenization in deep learning is vast as it is primarily co-related with the natural language processing which serves as one of the bedrock for neural networks. Using this method the sequence data of text can be vectorized and hence it is compatible for the machines to process human language. Since machines communicate in vectors this method is suitable for the experiment.

IV. IMPLEMENTATION

The implementation includes the IMDB dataset which serves a benchmark for sentiment analysis, of about 50000 records in which a 50/50 split will be made for training and testing. The 25000 records of well reviewed data will be used for training on the model. Once the trained model is ready it is passed with the testing set to determine the performance of the model. The goal is to use this model to determine the polarity of the reviews as either positive or negative. We tend to compare models of planned models with existing models hence a comparison study has been carried out on existing models. The result states that LSTM with word2vec embedding offers higher performance than the existing different models. SVM offers the lowest performance when put next to other models, DNN offers slightly similar performance with LSTM models as well. Hence with all this comparisons, LSTM is decided to be the effective neural network algorithm that can be used for this move sentiment analysis. Before all this processes commences a word preprocessing must be carried out before loading it onto the model as the review data consists of unsanitised data which will hinder the training process. The data from IMDB could be web scraping result as it consists of various tags such as
</br>. Once the word preprocessing is done, the result data is passed to the word2vec embedding which will tokenise and vectorise the data as per the rules mentioned. Using this as an input to the model, the training fit starts with 15 epoch to achieve an accuracy of 88.04%. 0.001 is set as the regulation parameter to avoid overfitting. The model will be then loaded to a prediction program in which the user will input reviews to predict the polarity of the reviews as either positive or negative. The threshold value is set to 0.5. The value obtained above is stated as postive review and below the threshold is predicted as a negative review. A comparison study can be used to compare the performances.

TABLE.1 Performance of the model

Configuration of the model	Epochs	LSTM -units	Max_ review length	Accuracy
EMBEDDING LAYER	3	100	500	85.51 %
	3	100	1000	86.72 %
LSTM LAYER	3	100	500	87.44 %
	3	50	500	86.88 %
DENSE LAYER	3	200	500	86.65 %
	30	200	500	84.68 %
	10	100	500	84.96 %
	3	100	500	86.96 %
	10	100	500	86.87 %
	10	100	500	86.41 %
	50	100	500	88.46 %

This table shows the accuracy of models with different architectures. From the table, we are able to observe that with a hundred LSTM units and when the model is trained with fifty epochs we have a tendency to reach the most effective performance when put next to the opposite models. When the review length is about a thousand we are able to observe there's a dip in performance. Once the quantity of LSTM units is redoubled to 200 then additionally we are able to observe the deterioration in performance. This could result to overfitting drawback. This gives us a comparative study of performance of different classification models on the benchmark IMDB movie review dataset. This model is compared with logistic regression, SVM, MLP and CNN. Except CNN all the other models are unit shallow models and SVM is that the strong classification model compared to the other models. To compare the planned model with the present model a comparison study has been done used on totally different existing models. Table shows that the planned LSTM primarily based model with word2vec embedding offers higher performance compared to alternative models. Statistical regression is best than SVM. This could be thanks to the linear kernel used with the SVM model. It's a great deal evident that the planned model is giving higher performance when put next to the other models.

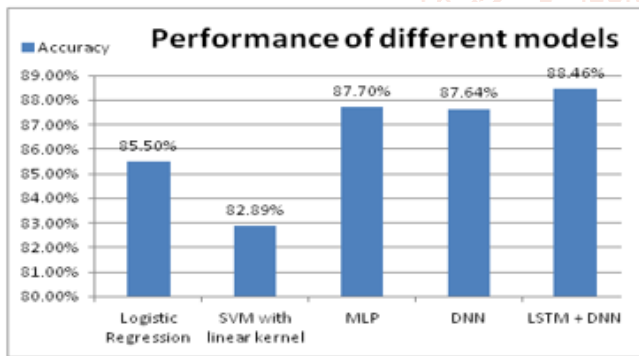


Fig.4.1. Performances of different models

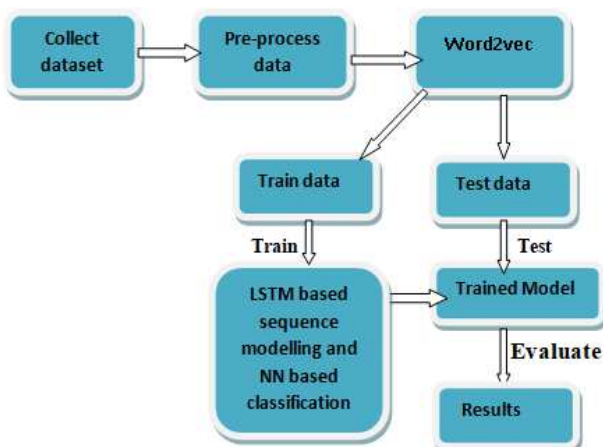


Fig.4.2 Architecture of LSTM model

V. RESULTS AND DISCUSSION

The result shows that the accuracy has reached 88 percentage and the loss is comparatively less considering previous attempts. The trained model is loaded to a prediction program in which the model is tested with real time reviews. The prediction program has a threshold value of 0.5. The values predicted for the review relates to the sentiments of the textual data. From this we can assume that the experiment is a success as the program can successfully predict the sentiments as either a positive or a negative review. The problem is that there is vanishing gradient problem which needs to be addressed so algorithms such as GRU can be utilised to avoid such issues.

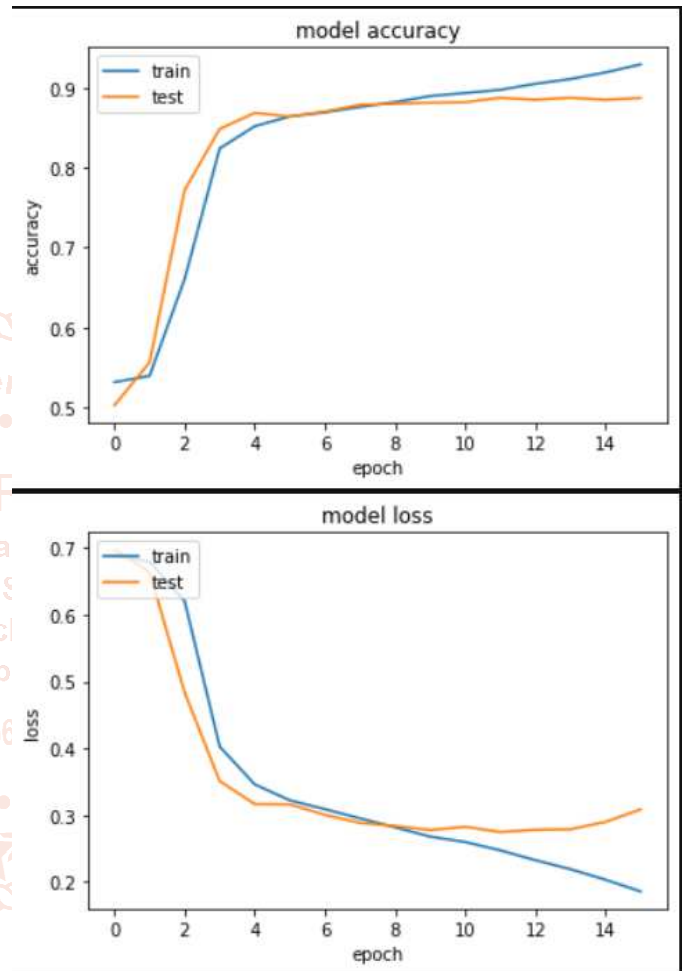


Fig.5.1 Accuracy and loss of the model

VI. CONCLUSION AND FUTURE ENHANCEMENT

This proposed system shows that LSTM is effective in the sentiment analysis process and the resulting model has achieved an accuracy of 88.04%. In future implementation, the model will be tested on other reliable datasets and better algorithms will be used to increase the accuracy of the model and hence a more precise sentiment analysis model can be achieved. There will be classifications on various aspects of a film such as screenplay, music, acting, comedy etc as the resulting prediction to further have a deep analysis of the review. This reviews serve as the most crucial feedback for the movie makers and hence this tool will be used to automate the feedback process in the future. With the big data combined this approach can be used in opinion mining of large scale datasets and hence a report can be generated on the diversified emotions of the people on a particular topic or trend. This will be an important role in digital marketing as it will give feedback on the performance of the products/services on the market. This can be used to

provide quality products/services as per the mass's needs and requirements in the current age. Sentiment analysis as a whole serves as powerful tool in every aspects of the commerce. So more and more deep learning techniques should be used in order to improve the existing models in the world. The applications of sentiment analysis is diverse and is a powerful way to understand public sentiments towards a subject.

REFERENCES

- [1] Zargar, Sakib. (2021). Introduction to Sequence Learning Models: RNN, LSTM, GRU. 10.13140/RG.2.2.36370.99522.
- [2] Shathik, Anvar & Karani, Krishna Prasad. (2020). A Literature Review on Application of Sentiment Analysis Using Machine Learning Techniques. 2581-7000. 10.5281/zenodo.3977576.
- [3] Martinelli, Samuele & Gonella, Gloria & Bertolino, Dario. (2020). Investigating the role of Word Embeddings in sentiment analysis. International Multidisciplinary Research Journal. 18-24. 10.25081/imrj.2020.v10.6476.
- [4] Okut, Hayrettin. (2021). Deep Learning for Subtyping and Prediction of Diseases: Long-Short Term Memory.
- [5] Camacho-Collados, José & Espinosa-Anke, Luis & Schockaert, Steven. (2019). Relational Word Embeddings. 3286-3296. 10.18653/v1/P19-1318. IOP Conference Series: Materials Science and Engineering.
- [6] Srinivas, Akana Chandra Mouli Venkata Satyanarayana, Divakar, Sirisha, Katikireddy Phani.(2021). Sentiment Analysis using Neural Network and LSTM. 1757-8981. <http://dx.doi.org/10.1088/1757-899X/1074/1/012007>
- [7] Papakyriakopoulos, Orestis & Medina Serrano, Juan Carlos & Hegelich, Simon & Marco, Fabienne. (2020). Bias in Word Embeddings. 10.1145/3351095.3372843.
- [8] Lison, P., Barnes, J., & Hubin, A. (2021). skweak: Weak Supervision Made Easy for NLP. ArXiv, abs/2104.09683.
- [9] [Rusch, Konstantin & Mishra, Siddhartha. (2020). Coupled Oscillatory Recurrent Neural Network (coRNN): An accurate and (gradient) stable architecture for learning long time dependencies.
- [10] Hoedt, P., Kratzert, F., Klotz, D., Halmich, C., Holzleitner, M., Nearing, G., Hochreiter, S., & Klambauer, G. (2021). MC-LSTM: Mass-Conserving LSTM. ArXiv, abs/2101.05186.

