Improving the Performance in Sentiment Analysis

Sumanto Kar
Student
Fr. Conceicao Rodrigues College of Engineering
Mumbai

J. Scriptu Rajan
Student
Fr. Conceicao Rodrigues College of Engineering
Mumbai

Sebastian Dmello
Student
Fr. Conceicao Rodrigues College of Engineering
Mumbai

Sapna Prabhu, PhD
Professor
Dept. of Electronics and Computer Science
Fr. Conceicao Rodrigues College of Engineering
Mumbai

ABSTRACT

Sentiment Analysis is the contextual mining of text which determines whether the given piece of words is positive, negative, or neutral. The main objective of this work is to make a system that rates movies based on the user's comments made on the movie. The system analyses the data in order to check for user sentiments associated with each comment and gathers all the comments made on a particular movie. It then calculates an average rating in order to score it. The system model checks for sentimental keywords and predicts user sentiment associated with it. Also, the system works on the sarcastic comments in order to find whether the comment is positive or not. Various Python libraries and Django Web Server has been used for the pre-processing of data.

Keywords

Deep Learning, LSTM Model, Movie Review System, Sentiment Analysis, Sarcasm Classifier, Bagging Algorithm.

1. INTRODUCTION

The explosion of the Internet has led to a lot of information all around the world. The information is normally stored in the database. A huge amount of data is available related to any field of study whether it be science, technology, sports, medical, movies, dramas, etc. Today data is available just at the fingertips. Just in few seconds, one can get information about any part of the world. This data can be used in many ways like for education, for analyzing the various conditions, comparing one item with the other, for manufacturing, for analyzing the market stock exchanges, for analyzing the climatic conditions and many other fields which one might not even have thought of.

In order to put a huge amount of data into the application for the well-being of the people, various fields have emerged in Computer Science and Information technology like Artificial Intelligence, Machine Learning, Deep Learning, Data Mining, Big Data, Fuzzy Logic, Neural Networks and many more. One of the most important applications is Sentiment Analysis. A lot of data available all around can be classified into facts and opinions which would help a computer algorithm to identify the sentiment of the people. Sentiment Analysis is widely being used in various areas like movies, dramas, sports, books, etc. to understand the sentiment of the people. The more the amount of data available, the output of the Sentiment analysis becomes more and more accurate. Various platforms like Facebook, YouTube, Google, Amazon, Flipkart, Alibaba, etc. apply Sentiment analysis in order to know the taste of the people and likewise show the results of the searches made by the people.

Nowadays as more and more movies are being released on OTT platforms, people must select the movie which they want to see out of many movies on the OTT platforms. So before seeing any movie people tend to see the reviews and the ratings of the movie so that they get an idea of what they are going to see and save their time by not seeing a badly rated movie. This makes the movie rating sites very important but if one has a site that will automatically give ratings by just using people’s reviews/comments made on a movie by doing sentiment analysis, this would be the best thing. So, as the world is rapidly moving in the technology field where everything is automated, it has been learned that the famous movie rating sites still rate movies based on few experts who see the movie and then give their ratings.

In this work, a model is being proposed which will analyze the reviews/comments made on a movie by the people and automatically give ratings based on the sentiment analysis done. This model is not only limited to movie rating, it can also be used anywhere where there is a requirement of understanding people or products through their comments or reviews. For example, it can be used in shopping sites to rate a product by doing sentiment analysis. It can be used to monitor social media, customer feedback etc.

Several online social networking services allow users to post and read messages to express their opinions related to products, politics, stock markets, entertainments etc. In addition, users usually post a lot of complex sentence structured messages, making it difficult to identify its sense for a machine as well as for humans. Therefore, Sentiment analysis as well as its application i.e., Sarcasm detection has become emerging trend in Data mining.

2. RELATED WORK

Most of the work or research on sentiment analysis have been done at the document level.

S.Chavan et al. in their paper made use of a machine learning algorithm i.e. Naive Bayes Multinomial (NBM) Method to build their classifier whose accuracy was 74.849 % [1]. Their focus was on the pre-processing of data which has the following steps. 1. Basic Operations and Cleaning which includes removing hash tags, symbols, etc. 2. Dictionary which will correct the incorrect spellings 3. Remove Stop words such as the, in, a, an, with, etc. and 4. Stemming i.e., joining similar words for example connect, connecting, connected, connections as single word connect.

A Jeyapriya et al. [2] discuss opinion-mining based on phrases which performs finer analysis and directly looks at the opinion in the online reviews given, which is then used to extract important aspects and generalized opinion over an item and to predict the orientation. It also makes use of Machine Learning for classification.

As Sentiment Analysis is used for marketing as well and various similar research purposes in order to find out the generalized and unbiased opinion of the various products or services offered the Paper “Sentiment Analysis of Social Issues” by Mostafa Karamibekret et al [3] suggested that sentiment analysis can also be applied to many social issues all around. Based on what they found, they proposed an approach to consider the role of the parts of speech like verbs, nouns, etc and found that verb as the most important term in expressing opinions regarding social issues. Their results have shown that considering parts of speech specially verbs are very necessary and they are considered as keywords for any kind of sentiment classification and it also improves the performance of sentiment analysis. The fact that the verb is the heart of a sentence is proved here as well.

Bing Liu et al. [4] has developed a system, named “Opinion Observer,” which basically focuses on the feature-based sentiment approach. He also discusses the various problems related to the sentiment-based analysis. The paper says that opinions can be expressed on various items, e.g., a product, an event, an individual, an organization, a service, or a topic. One uses the term object in order to denote the refer to the entity that has been commented or reviewed on.

Nasukawa et al. [5] use a sentence-level sentiment classification to find favorability using Natural Language Processing (NLP). They have used a parser to split the sentences into words and find out the subject of the sentence. Then they have parsed out the other parts of speech like verbs, adjectives, etc. They have developed three cases in order to carry out the analysis namely A) the system assigned either a positive or negative sentiment, B) the human assigned either a positive or negative sentiment and C) the system output based on the manual judgment.

Somasundaran et al. [6] have worked on the social application of sentiment analysis. There are various social debates based on the existence of God, healthcare debates, debates based on government etc. Their approach was first to classify a debate based on arguing opinion and the opinion target. Then they moved on to constructing an arguing lexicon and at last stance classification.

Sentiment analysis is a recent field of research based on the Natural Language Processing (NLP), which aims at detecting subjectivity in text and/or extracting and classifying opinions and sentiments.[7] Sentiment analysis is the use of various fields under natural language processing in order to identify, extract, quantify, and study various states and information and predict the sentiment of a person on a specific item or a thing. It can be applied to various fields anywhere and everywhere. For example, customer materials such as comments and survey charts, online and social media like Facebook and Amazon, various healthcare materials for applications, various electronic products like mobile phones, laptops, etc.

The various algorithms used in Sentiment analysis can be classified as namely Linear Regression, Naive Bayes, Support Vector Machines, various Recurrent Neural Nets (RNN) derivatives long short-term memory (LSTM) and Gated Recurrent Unit (GRU) [19]. The long short-term memory under the Recurrent Neural Networks is considered as the most powerful neural network in today's world.[20] Hence LSTM is been adopted in order to analyze our comments based on movie reviews in this paper.

LSTM units are used for sequence learning of numerical data points in successive order. It is a unit of Recurrent Neural Nets. RNN composed of LSTM units is generally referred to as “LSTM network”. They are one of the most advanced deep learning algorithms and can be easily used for the analysis of the comments and reviews [20]. The first step of the model is pre-processing of data. This process includes removing the symbols, hashtags etc. This is done using a built-in Python library called Natural Language Toolkit (NLTK) which is used for Natural Language Processing (NLP). Then the words will be tokenized in order to obtain a data set – namely, the training set. Then, this data set is involved in the learning step, which uses a Deep Learning algorithm i.e., LSTM which yields a trained classifier. After training the classifier, it has to be tested on a different data set – namely the test set.

Alessia et al. have also faced some challenges related to various special cases of applying sentiment analysis like irony and sarcasm [8]. The social posts, comments, reviews, etc. often have a lot of ambiguity in them. Some thoughts are expressed in sarcastic manner but not directly. For example, if someone comments as “good movie with nonsense acts”. It becomes very difficult to apply sentiment analysis on this as the commenter on one hand is saying that the movie is good and on the other hand that the movie has nonsense acts. In such cases predicting the review becomes much more difficult. Here a solution is demonstrated by integrating a sarcasm classifier with our LSTM sentiment classifier.

The machine is first being trained based on the sarcasm-based dataset where it learns the differences between the sarcastic and non- sarcastic sentences. Then this trained model is linked with our LSTM Classifier so now the comments will be classified as positive or negative along with sarcastic or not.

The front end of the project is a website where people can search for a movie and then give their review on that movie and our LSTM classifier model which is being trained will classify all the reviews/comments made on that movie and rate the movie by doing the sentiment analysis. The website is made using Django which is a high-level Python Web framework and the website is styled using Html and CSS. Django is a collection of many Python libraries and is simple to use yet it is the most powerful framework. Our website has a list of many movies from which the user can select which movie they want to review. The website is also made responsive so that it can be used on both PC’s and mobiles interface as well.

3. ALGORITHM

The algorithm used for the sentiment classifier is LSTM (long short-term memory). It is used for predicting the value between 0 to 1. It is a recurrent neural network so it makes predictions on past data (Refer to Fig. 1). The algorithm is basically divided into four different stages

1)Forget state: The first stage is the forget gate. Here the gate determines which information should be kept and which ones should be rejected just like forgetting state in the human brain. Information from the previous hidden state and the present input is passed through the sigmoid function. Values come out between 0 and 1. The closer to 0 implies that the system must forget whereas closer to 1 means to keep.

$$f_t = \sigma\left(X_t * U_f + H_{t+1} * W_f\right)\#(1)$$

2)Input state: In order to update the cell state, the input gate is present. It receives the previous hidden state and current input and passes it to a sigmoid function. The sigmoid function determines which values are going to be updated by transforming the values to be between 0 and 1. 0 means not important, and 1 means important. The hidden state and current input is passed into the tanh (hyperbolic tan) function to swing the values between -1 and 1 in order to assist regulate the network. Then the tanh output and the sigmoid output are being multiplied. The sigmoid output will decide which information is vital to keep from the output of the tanh.

$$\bar{C_t} = \tanh(X_t * U_c + H_{t-1} * W_c) \ \#(2)$$

$$I_t = \sigma(X_t * U_i + H_{t-1} * W_i) \#(3)$$

3)Cell state: A lot of information is required for the cell state. After the input state one will have enough information to calculate the cell state. The cell state here gets pointwise multiplied by the forgets vector. This represents a possibility of rejecting values within the cell state if it gets multiplied by values near 0. Then one uses the output from the input gate to do a pointwise addition which updates the cell state to the new values that the neural network finds relevant. That gives us our new cell state.

$$C_t = f_t * C_{t-1} + I_t * \bar{C_t} \#(4)$$

4)Output state: At last, the output is required. The output gate decides what next hidden states should be. The hidden state contains information on previous inputs. The hidden state is also used for making predictions. The previous hidden state is passed and then the current input into a sigmoid function. Then the newly modified cell state is passed here to the tanh function. Then the tanh output is multiplied with the sigmoid output to decide what information the hidden state should carry. The output is the hidden state. The new cell state and the new hidden state are then carried over to next time steps.

$$O_t = \sigma(X_t * U_0 + H_{t+1} * W_0) \#(5)$$

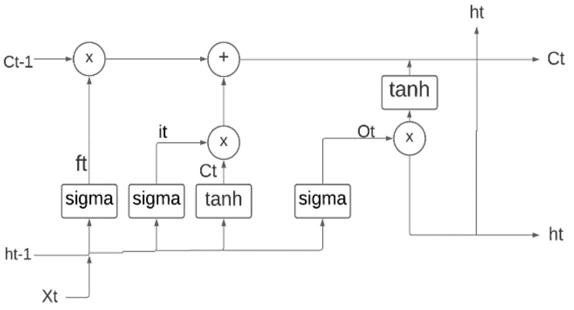


Fig 1. Representation of LSTM classifier

4. RESULTS

Here LSTM (Long Short-Term memory) is being used and sarcasm classifier for this project. So, for LSTM finding the accuracy of model, the data is split into training and testing. Training is done for 5 epochs and at every epoch a validation_split of 0.2 happens i.e., 20 % of training data is split for testing (Refer to Fig. 2).

The predicted values are compared with the actual values for finding the accuracy of both training and testing data. The accuracy of our model on the sarcastic data is 56% on training and 56.33% on testing.

For Sarcasm Classifier as well finding the accuracy of model, the data is split into training and testing. The training is done for 30 epochs. Here as well the predicted values with the actual values for finding the accuracy of both training and testing data are compared. The accuracy of our model is 95.74% on training and 85.39% on testing (Refer to Fig. 3).

The confusion matrix is computed on 100 testing comments, out of which 47 were (True positive), 40 were (True negative) meaning 87 of 100 comments were rightly predicted

Confusion matrix (Refer to Fig. 4) is a two-dimension table (actual and predicted) with two rows and two columns that reports the number of false positives, false negatives, true positives, and true negatives. For example, in true positives, true represents actual result is true and positive represents decision is true [9]. From the table of confusion, accuracy can be calculated as,

$$Accuracy = \frac{(True\ Positives\ + True\ Negatives)}{(True\ Positives\ +\ True\ Negatives \atop +\ False\ Positives\ +\ False\ Negatives)} \#(6)$$

The training data is divided into subsets of Sarcasm and LSTM data (Refer Fig. 5). The data is then used for learning the LSTM and Sarcasm Classifier (Refer Fig. 2 and Fig. 3). After passing training the two classifiers the models of the two classifiers are combined to get the new better predicted decision. (Refer Fig. 6 and Fig. 7).

The bagging algorithm proposed by Leo Breiman in 1994 for the combining of the two classifiers is being used. In the process of Bagging, the bootstrapped samples are first created. Next, each sample is passed through either a regression or classification algorithm. Finally, an average or a mean is taken over all the outputs predicted by the individual learners. This is where aggregation comes into the picture [10].

Mathematically, Bagging algorithm is represented by the formula given in Equation 7.

$$\widehat{f_{bag}} = \hat{f_1}(X) + \hat{f_2}(X) + \cdots + \hat{f_b}(X) \#(7)$$

The term on the LHS is the bagged prediction, and terms on the RHS are the individual learners.

By this it can be concluded that for sarcastic statements classified by LSTM along with the Sarcasm classifier was giving better results than using on LSTM Classifier.

Using the formula given in Equation 8, the accuracy of our new combined classifier is calculated as:

$$\begin{matrix} Accuracy\ of \\ Combined\ Classifier \end{matrix} = \frac{\left({Accuracy\ of \atop LSTM\ classifier} + {Accuracy\ of \atop Sarcasm\ classifier} \right)}{2} \#(8)$$

Therefore, the training accuracy of our ensembled classifier becomes:

$$(56\% + 95.74\%)/2 = 75.87\%$$

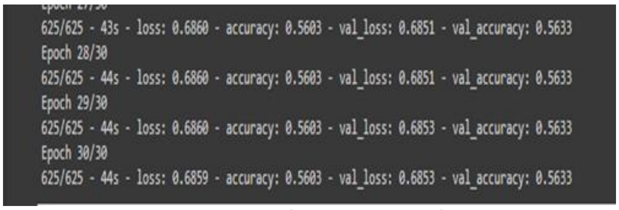
and the testing accuracy becomes:

$$(56.33\% + 85.39\%)/2 = 70.86\%$$

Table 1 shows a summary of the achieved accuracies using the various classifieds

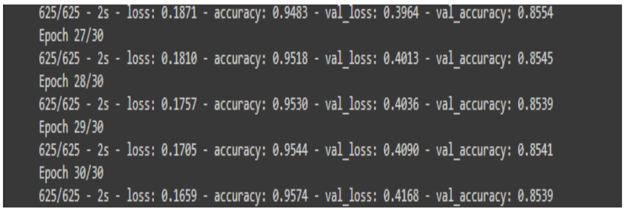
Thus, an improvement in the training accuracy and the testing accuracy can be seen by some percentage so it solves the problem of overfitting of Sarcasm Classifier and there is an overall rise in accuracy and perfection of the model (Refer to Fig. 6 and Fig. 7).

So, testing the two algorithms on sarcasm testing dataset is done. When tested on LSTM algorithms, the results where poor since the algorithm was not capable of recognizing sarcastic statements and Sarcasm algorithm did well with about 855 accuracies on testing data. So, after combining the two algorithms, a better accuracy of around 70% can be observed.



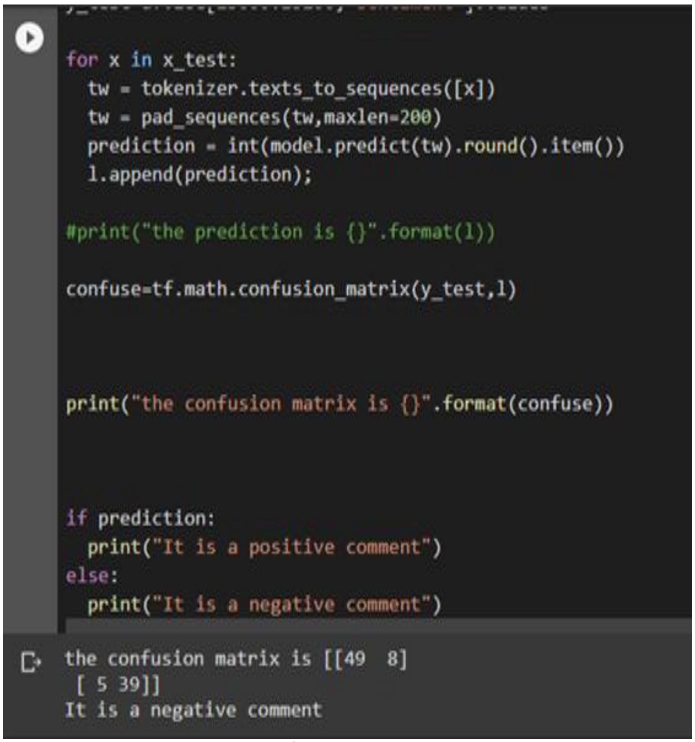
```
Epoch 27/30
625/625 - 43s - loss: 0.6860 - accuracy: 0.5603 - val_loss: 0.6851 - val_accuracy: 0.5633
Epoch 28/30
625/625 - 44s - loss: 0.6860 - accuracy: 0.5603 - val_loss: 0.6851 - val_accuracy: 0.5633
Epoch 29/30
625/625 - 44s - loss: 0.6860 - accuracy: 0.5603 - val_loss: 0.6853 - val_accuracy: 0.5633
Epoch 30/30
625/625 - 44s - loss: 0.6859 - accuracy: 0.5603 - val_loss: 0.6853 - val_accuracy: 0.5633
```

Fig 2. Epochs of LSTM Classifier



```
625/625 - 2s - loss: 0.1871 - accuracy: 0.9483 - val_loss: 0.3964 - val_accuracy: 0.8554
Epoch 27/30
625/625 - 2s - loss: 0.1810 - accuracy: 0.9518 - val_loss: 0.4013 - val_accuracy: 0.8545
Epoch 28/30
625/625 - 2s - loss: 0.1757 - accuracy: 0.9530 - val_loss: 0.4036 - val_accuracy: 0.8539
Epoch 29/30
625/625 - 2s - loss: 0.1705 - accuracy: 0.9544 - val_loss: 0.4090 - val_accuracy: 0.8541
Epoch 30/30
625/625 - 2s - loss: 0.1659 - accuracy: 0.9574 - val_loss: 0.4168 - val_accuracy: 0.8539
```

Fig 3. Epochs of Sarcasm Classifier



```
for x in x_test:
  tw = tokenizer.texts_to_sequences([x])
  tw = pad_sequences(tw,maxlen=200)
  prediction = int(model.predict(tw).round().item())
  l.append(prediction);

#print("the prediction is {}".format(l))

confuse=tf.math.confusion_matrix(y_test,l)


print("the confusion matrix is {}".format(confuse))


if prediction:
  print("It is a positive comment")
else:
  print("It is a negative comment")
```

```
the confusion matrix is [[49  8]
 [ 5 39]]
It is a negative comment
```

Fig 4. Confusion Matrix

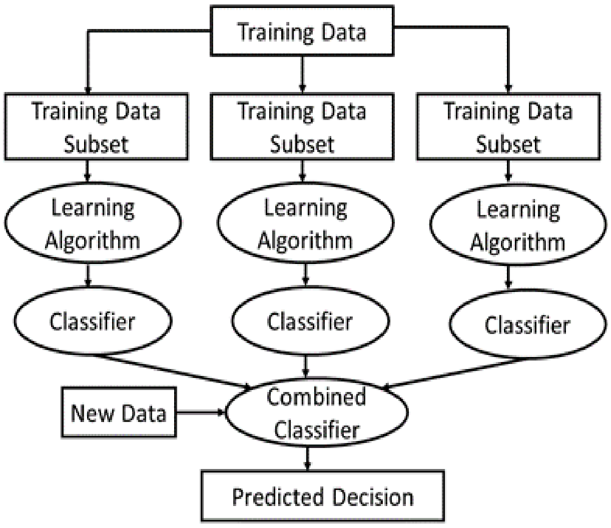
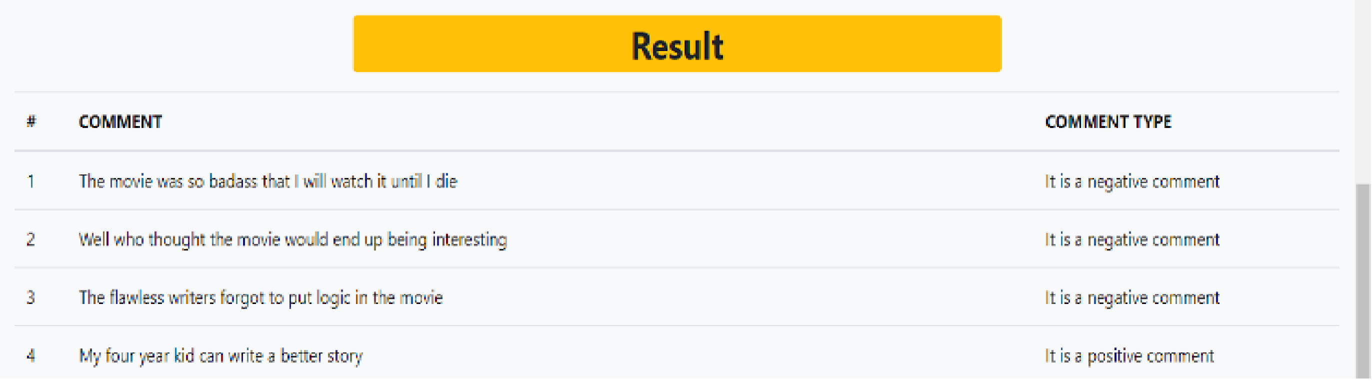


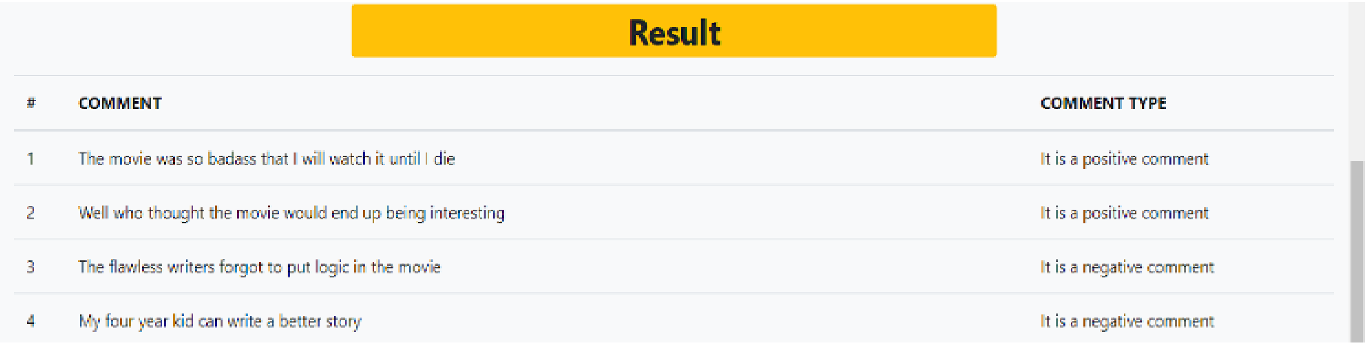
Fig 5. Process of Combining LSTM and Sarcasm Classifier



Result

#	COMMENT	COMMENT TYPE
1	The movie was so badass that I will watch it until I die	It is a negative comment
2	Well who thought the movie would end up being interesting	It is a negative comment
3	The flawless writers forgot to put logic in the movie	It is a negative comment
4	My four year kid can write a better story	It is a positive comment

Fig 6. Sarcastic Statements classified by LSTM classifier



Result

#	COMMENT	COMMENT TYPE
1	The movie was so badass that I will watch it until I die	It is a positive comment
2	Well who thought the movie would end up being interesting	It is a positive comment
3	The flawless writers forgot to put logic in the movie	It is a negative comment
4	My four year kid can write a better story	It is a negative comment

Fig 7. Sarcastic Statements classified by LSTM classifier along with Sarcasm Classifier

Table 1. Output of various classifiers on sarcastic data

Classifier	Accuracy
LSTM	56.33%
Sarcasm	85.39%
(LSTM + Sarcasm)	70.86%

5. CONCLUSION AND FUTURE SCOPE

One often encounters many online movie rating sites, where the admin manually rates the movie, depending upon the comments, ratings and reviews of the users or a few experts rate the movie according to them.

Here successfully a system is being implemented that can classify the comment as positive or negative by doing the sentiment analysis by the LSTM Classifier Several classifiers like Naive - Bayes, SVM classifiers, etc., were tested but LSTM Classifier had the maximum accuracy i.e., 70.86.

Successfully in a website was also created and the model was implemented on the website and then the movies were rated based on user's comments/reviews made on the movie.

6. ACKNOWLEDGMENTS

We express our sincere thanks to our Guide Dr. Sapna Prabhu, Head of Electronics and Computer Science Department for expert suggestion and support during every stage of this work. We also express our deep sense of gratitude to our Principal, Dr. Srija Unnikrishnan. In the end our special thanks to all faculty in ECS department for providing support in every aspect.

7. REFERENCES

[1] Sagar et al Sentiment Analysis of Movie Ratings System. International conference on computing and virtualization (ICCCV-17)

[2] A.Jeyapriya et al Extracting aspects and mining opinions in product reviews using supervised learning algorithms. In the proceedings of 2nd International Conference on Electronics and Communication Systems (ICECS), 2015

[3] Mostafa Karamibekret et al Sentiment Analysis of Social Issues. In the Proceedings of International Conference on Social Informatics (Social Informatics), 2012

[4] B. Liu. Sentiment analysis and subjectivity. Handbook of Natural Language Processing, 2010.

[5] T. Nasukawa and J. Yi. Sentiment analysis: capturing favorability using natural language processing. In Proceedings of the 2nd international conference on Knowledge capture, pages 70–77. ACM, 2003.

[6] S. Somasundaran and J. Wiebe. Recognizing stances in ideological online debates. In Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, pages 116–124. ACM, 2010.

[7] Wikipedia: The free encyclopedia, 2004. https://en.wikipedia.org/wiki/Sentiment_analysis

[8] Alessia D'Andrea and Fernando Ferri. Approaches, Tools and Applications for Sentiment Analysis Implementation. In proceedings of International Journal of Computer Applications, 2009

[9] M. Balamurugan and S. Kannan, Analyse the performance of Ensemble Classifiers using Sampling Techniques. In proceedings of ICTACT Journal on Soft Computing, July 2016

[10] Paperspace Blog, 2021. https://blog.paperspace.com/bagging-ensemble-methods/

[11] S. R. Das and M. Y. Chen. Yahoo! for amazon: Sentiment extraction from small talk on the web. Management Science, 53(9):1375–1388, 2007.

[12] C. Fellbaum. Wordnet: An electronic lexical database.

[13] V. P. H. Binali and W. Chen. A state-of-the art opinion mining and its application domains. In IEEE International Conference on Industrial Technology, pages 1–6, February 2009.

[14] F. Inc. The free dictionary, 2012.

[15] W. F. Inc. Wikipedia: The free encyclopedia, 2004. https://en.wikipedia.org/wiki/Text_mining

[16] C. M. Kristina Toutanova, Dan Klein and Y. Singer. Feature-rich part of-speech tagging with a cyclic dependency network. In HLT-NAACL, pages 252–259. ACM, 2003.

[17] S. L. Kushal Dave and D. M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In Proceedings of the 12th international conference on World Wide Web, pages 519–528. ACM, 2003.

[18] B. Liu. Sentiment analysis and subjectivity. Handbook of Natural Language Processing, 2010.

[19] S. C.-O. Michael Gamon, Anthony Aue and E. Ringger. Pulse: Mining customer opinions from free text. In

Proceedings of the 6th International Symposium on Intelligent Data Analysis, pages 121–132, 2005.

[20] T. Nasukawa and J. Yi. Sentiment analysis: capturing favorability using natural language processing. In Proceedings of the 2nd international conference on Knowledge capture, pages 70–77. ACM, 2003.

[21] Anju Joshi et al Aspect Level Opinion Mining on Customer Reviews using Support Vector Machine. International Journal of Advanced Research in Computer and Communication Engineering ISO 3297:2007 Certified Vol. 6, Issue 7, July 2017

[22] The App Solutions https://theappsolutions.com/blog/development/sentiment-analysis/

[23] Hyejung Chung and Kyung-shik Shin Genetic Algorithm-Optimized Long Short-Term Memory Network for Stock Market Prediction. MDPI Journal on Sustainability

[24] B. Pang and L. Lee. Opinion Mining and Sentiment Analysis. The Essense of Knowledge, 2008.

[25] B. Pang, L. Lee, and S. Vaithyanathan. Sentiment classification using machine learning techniques. In EMNLP02 Proceedings, pages 79–86. Association for Computational Linguistics, 2002.

[26] S. Shandilya and S. Jain. Automatic opinion extraction from web documents. In Proceeding of International Conference on Computer and Automation Engineering, pages 351–355, March 2009.

[27] S. Somasundaran and J. Wiebe. Recognizing stances in ideological online debates. In Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, pages 116–124. ACM, 2010.

[28] P. D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th Annual Meeting, pages 417–424. Association for Computational Linguistics (ACL), 2002.

[29] B. L. Xiaowen Ding and P. S. Yu. A holistic lexicon-based approach to opinion mining. In Proceedings of the international conference on Web Search and web Data Mining, pages 231–240. ACM, 2008.