# Sentiment Analysis of IMDb Movie Reviews Using Long Short-Term Memory

**Conference Paper** · November 2020

DOI: 10.1109/ICCIS49240.2020.9257657

Saeed Mian Qaisar
Effat University

# Sentiment Analysis of IMDb Movie Reviews Using Long Short-Term Memory

Saeed Mian Qaisar
*College of Engineering, Effat University,* 21478, Jeddah, Saudi Arabia
sqaisar@effatuniversity.edu.sa

***Abstract*— The sentiment analysis is an emerging research area where vast amount of data are being analyzed, to generate useful insights in regards to a specific topic. It is an effective tool which can serve governments, corporations and even consumers. Text emotion recognizing lays a key role in this framework. Researchers in the fields of natural language processing (NLP) and machine learning (ML) have explored a variety of methods to implement the process with highest accuracy possible. In this paper the Long Short-Term Memory (LSTM) classifier is used for analyzing sentiments of the IMDb movie reviews. It is based on the Recurrent Neural Network (RNN) algorithm. The data is effectively preprocessed and partitioned to enhance the post classification performance. The classification performance is studied in terms of accuracy. Results show a best classification accuracy of 89.9%. It confirms the potential of integrating the designed solution in modern text based sentiments analyzers.**

Keywords- ***Sentiment Analysis, Machine Learning (ML), Natural Language Processing (NLP), Long Short-Term Memory (LSTM), Movie Reviews.***

## I. INTRODUCTION

The sentiment analysis is the process of using natural language processing and computational linguistics to extract, identify and categorize different opinions expressed in text format. It has attracted researchers from different disciplines, especially from computer science as it falls under interactive computation or human computer interaction (HCI). Sentiment analysis is mainly a classification problem that merges both domains natural language processing (NLP) and machine learning (ML) [1]. It has a wide range of applications such as opinion mining and business analytics. In addition it has a potential to be implemented for governmental purposes to prevent suicide incidents. There are generally two fundamental approaches for sentiment analysis: Lexicon based sentiment analysis and Machine learning based sentiment analysis. Lexicon approach depends on splitting text into tokens (tokenization), counting the number of occurrences for each word and looking up the subjectivity of each word from an existing lexicon. In machine learning approach a more complex yet add sophistication to the system, depends on training different classifiers with a data set referred to as training set. Followed by an evaluation step by testing the classification performance by using the testing data set [2]. Some other key approaches are the word vectors learning [3], Document to Vector [7], Ontology based approach [9] and voting ensemble classifier [11].

Although the algorithms and techniques have advanced in the last couple of years, there are still many challenges in the field which have not been resolved to this day. The main two limitations are: Keywords that hold many different meanings according to their context can lead to ambiguity, and incapability of classifying sentences which does not include clear emotional keywords may imply that sentence contains no emotions. Therefore, the devised system should accounts for these drawback and provide an accurate categorization of data since it will be deployed in highly sensitive applications.

In this paper, a Python based application is utilized to analyze the IMDb dataset. The dataset is tactfully divided into training and testing parts. The training part is used for preparing the LSTM classifier. Afterwards, the testing set is used to quantify the classification precision. Confusion matrix and accuracy results are outlined.

## II. MATERIALS AND METHODS

The proposed ML based model for text based sentiment analysis is presented in Fig.1. As illustrated the model consists of five main blocks along with few minor components integrated in the system.

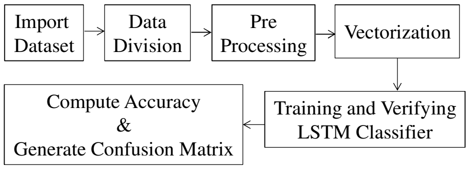

Import Dataset
Data Division
Pre Processing
Vectorization
Training and Verifying LSTM Classifier
Compute Accuracy & Generate Confusion Matrix

Fig. 1. The proposed system flowchart

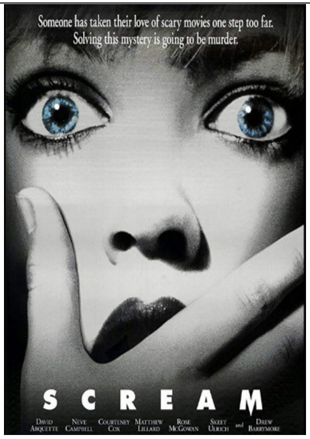
### A. Dataset

For this paper a data set containing 50k movie reviews from IMDb, created by Andrew Maas is utilized [4]. The data have already been splitted into 25k reviews for training purposes while the other 25k is intended for testing the classifier. In addition, both sets contain 12.5k positive and negative reviews. The reviews are classified into positive and negative in reference to the IMDb rating system. It allows viewers to rate on a scale from 1 to 10, and according to dataset creator anything ≤ 4 stars is labeled negative and ≥ 7 stars is marked

978-1-7281-5467-1/20/$31.00 ©2020 IEEE.

as positive. Reviews with ratings out of the above ranges are not included. There are at most 30 reviews for each movie. The average number of words per review is 234.76 with a standard deviation of 172.91 words. Collectively, the dataset contains 88585 different words. Glimpse of the dataset positive and negative reviews is shown in Table I.

TABLE I. GLIMPSE OF THE DATASET POSITIVE AND NEGATIVE REVIEWS ON THE SAME MOVIE

| Review | Movie | Sentiment |
|---|---|---|
| "Making a brilliant, original horror film is pretty hard these days. It was a great movie, and I suggest that you go see it." |   Someone has taken their love of scary movies one step too far.<br>Solving this mystery is going to be murder.<br>SCREAM<br>David Arquette Neve Campbell Courteney Cox Matthew Lillard Rose McGowan Skeet Ulrich and Drew Barrymore | Positive |
| "The opening ten minutes are admittedly great. Drew Barrymore reaching for the phone, only to find herself in conversation with a stalker is inspired. What a pity then about the rest of the film." | | Negative |

## B. Data Division

It is common to divide the dataset into training and testing vectors [1], [2]. The training vector is the set of data that trains the considered classifier. The validation vector is a subset of the training vector that does not necessarily train, but is used to give some insight on the classifier performance. Test data is to evaluate the model accuracy. The split of the training and validation, testing, or both can occur in many ways. However, there is a rule-of-thumb that training gets the most data. A recurrent ratio encountered in various eclectic ML settings is the 80-20 split which gives 80% to the training and 20% to the testing. This ratio finds interesting roots in what is referred to as the Pareto Principle or the law of the vital few and arises in finance and economic theories [5]. In this paper 10k reviews are considered. In order to avoid any biasness an equal representation of positive and negative reviews is employed. 80% of this data is used for the training purpose and the 20% is used for the testing purpose. To evade the biasness in classification results 10-fold cross-validation is utilized in this study.

## C. Preprocessing

Data that are not well cleaned and organized might lead to false identifications. Hence, data preprocessing is a crucial task in the data mining process. It refers to cleaning up the data from useless information that will not help in the training process and might cause confusion during the classification process. For the IMDb dataset, several data preprocessing steps are utilized. Firstly, all the symbols such as "?", "!" are removed. Secondly, all letters in the datset are converted into lowercase letters. Thirdly, all hybrid links are removed from the text. Fourthly, stop words such as, me, you, and we are evicted. Finally, stemming techniques are applied on the text to present the word in its original form after removing prefixes and suffixes.

## D. ***Vectroization***

Vectorization or text embedding is the process of extracting features from text and pass it as an input to the classifier. Python is a general purpose object-oriented high level programming language [3]. Due to the extensive libraries and frameworks dedicated to ML which facilitates the development process and save time, it is considered and used in this paper [3].

Each movie review is encoded "vectorized" into a numeric value. It is achieved by utilizing the *genism*, a Python library for topic modeling and NLP [3], [6]. Compared to other techniques, *Doc2Vec* model proved to deliver high accuracy results with lower computational cost [7]. A denotation of *Doc2Vec* parameters is outlined in Table II.

TABLE II. AN ILLUSTRATION OF DOC2VEC HYPERPARAMETERS [7]

| Hyper Parameter | Definition |
|---|---|
| vector_size=10 | Dimension of the feature vectors. |
| Window=2 | Left/Right context window size. |
| min_count=1 | Ignore words with a total frequency less than 1. |
| workers=4 | Use 4 worker threads to train the model results in a faster training for a multicore machine. |

## E. Long Short-Term Memory (LSTM) Classifier

It is based on the Recurrent Neural Network (RNN) algorithm. RNN has memory but it falls short when data has longer dependencies. In contrast, LSTM uses loops with the addition of gates to maintain a level of relevancy [8], and keeps pertinent data from vanishing in very long sequences. It elegantly addresses the vanishing gradient problem intrinsic in the RNN model using linear memory units, and certain gates which control the flow of information. LSTM mitigates the short term memory abound in RNNs and indeed solves a number of problems with long-range pathological temporal dependencies.

The LSTM is employed with adaptive moment estimation (Adam) optimizer. It is an adaptive learning rate optimization algorithm that is particularly designed for training the deep neural networks. The algorithms leverages the power of adaptive learning rates methods to find individual learning rates for each parameter. It uses estimations of first and second moments of gradient to adapt the learning rate for each weight of the neural network [9]. Three layers are employed. In this case, the first layer employs 50 nodes to present the intended words. The second layer is the LSTM with 101 units of memory and the final layer creates 2 outputs corresponding to the considered classes.

### F. *Classification Accuracy*

The classification accuracy is used to measure that how well the devised model is able to automatically identify the data. It is the percentage of labels that have been correctly classified. he mathematical formulation for accuracy is given in Equation (2). Where, TP, TN, FP, and FN respectively denote the true positives, true negatives, false positives, and false negatives in the predicted labels.

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \times 100\% \qquad (2)$$

The principle of confusion matrix formation is depicted with the help of Table III.

TABLE III. CONFUSION MATRIX PRINCIPLE

| | Positive | Negative |
|---|---|---|
| Positive | TP | FP |
| Negative | TN | FN |

## III. RESULTS AND DISCUSSION

In order to illustrate the classification results of four movie reviews examples are summarized in Table IV. As seen that 4 out of 4 sentiments turned out to be correct and valid.

TABLE IV. ACTUAL SENTIMENT AND PREDICTED SENTIMENT OF EXTERNAL DATA

| **Sentiment** | **Actual (Positive = 1) (Negative = 0)** | **Predicted** |
|---|---|---|
| *I love the movie* | 1 | 1 |
| *It was entertaining* | 1 | 1 |
| *It was a waste of time* | 0 | 0 |
| *It was uninteresting* | 0 | 0 |

While dealing with the IMDb intended database portion, the confusion matrix is generated to assess the performance accuracy of the LSTM classifier. Confusion matrix provides a summary of correct and incorrect predictions as shown in Table V [10]. Another performance metric calculated is the accuracy score, which is the ratio of correct predicted reviews to the total number of reviews presented in Equation (2) [10]. The results of confusion matrix are summarized and presented in Table 5, in addition the accuracy score is found to be around 89.9%.

TABLE V. CONFUSION MATRIX RESULTS

| | Positive | Negative |
|---|---|---|
| Positive | 4553 | 447 |
| Negative | 4436 | 564 |

Considering the accuracy score of the classifier, these results show that the classifier achieves an appropriate precision for the intended dataset. A better precision can be achieved by further cleaning the data and employing the ensemble classification approaches.

## IV. CONCLUSION

Sentiment Analysis also referred to as opinion mining is the process of extracting opinions from text data and classifies it into positive, negative or neutral ones. In this paper, a Long Short-Term Memory classifier is used with Adam optimizer to automatically categorize the preprocessed IMDb movie reviews. In total 10k reviews are considered, 5k for positive and 5k for negative sentiments. Results have concluded that the highest accuracy attained by the devised approach is of 89.9%.

A superior accuracy can be attained by using further data preconditioning techniques. Furthermore, higher classification accuracy can be achieved by employing the ensemble classifiers or deep learning approaches. Exploring these opportunities is another prospect.

## V. ACKNOWLEDGEMENT

The author would like to extend his sincere thanks to the anonymous reviewers for their useful feedback. Author is also thankful to engineer A. Alamodi for her help during basic simulation model and manuscript draft preparation.

## REFERENCES

[1] Sailunaz, K., Dhaliwal, M., Rokne, J., & Alhajj, R. (2018). Emotion detection from text and speech: a survey. Social Network Analysis and Mining, 8(1), 28.

[2] Bandhakavi, A., Wiratunga, N., Padmanabhan, D., & Massie, S. (2017). Lexicon based feature extraction for emotion text classification. Pattern recognition letters, 93, 133-142.

[3] Joshi, P. (2017). Artificial intelligence with python. Packt Publishing Ltd.

[4] A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, and C. Potts,"Learning word vectors for sentiment analysis." In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol.1, pp. 142-150, 2011.

[5] G. E. . R. D. Meyer, “An analysis for unreplicated fractional factorials,” Technometrics, vol. 28, pp. 11–18, 1986.

[6] Srinivasa-Desikan, B. (2018). Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras. Packt Publishing Ltd.

[7] Hoque, M. T., Islam, A., Ahmed, E., Mamun, K. A., & Huda, M. N. (2019, February). Analyzing Performance of Different Machine Learning Approaches With Doc2vec for Classifying Sentiment of Bengali Natural Language. In 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE) (pp. 1-5). IEEE.

[8] Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., & Xu, B. (2016, August). Attention-based bidirectional long short-term memory networks for relation classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 207-212).

[9] Jiang, S., & Chen, Y. (2017, September). Hand Gesture Recognition by Using 3DCNN and LSTM with Adam Optimizer. In Pacific Rim Conference on Multimedia (pp. 743-753). Springer, Cham.

[10] B. M. and V. B., “Sentiment Analysis using Support Vector Machine based on Feature Selection and Semantic Analysis,” International Journal of Computer Applications, vol. 146, no. 13, pp. 26–30, 2016.

[11] Onan, A., Korukoğlu, S., & Bulut, H. (2016). A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. Expert Systems with Applications, 62, 1-16.