

---

## Sentiment Analysis from Movie Reviews Using LSTMs

Jyostna Devi Bodapati<sup>1</sup>, N. Veeranjanyulu<sup>2\*</sup>, Shareef Shaik<sup>1</sup>

<sup>1</sup> Department of CSE, Vignan's Foundation for Science, Technology and Research, Vadlamudi, Guntur, AP, India

<sup>2</sup> Department of IT, Vignan's Foundation for Science, Technology and Research, Vadlamudi, Guntur, AP, India

Corresponding Author Email: [veeru2006n@gmail.com](mailto:veeru2006n@gmail.com)

---

<https://doi.org/10.18280/isi.240119>

### ABSTRACT

**Received:** 13 November 2018

**Accepted:** 19 January 2019

---

**Keywords:**

*recurrent neural networks, gated recurrent neural networks, text mining, word embedding, SVM, deep neural networks*

With the advent of social networking and internet, it is very common for the people to share their reviews or feedback on the products they purchase or on the services they make use of or sharing their opinions on an event. These reviews could be useful for the others if analyzed properly. But analyzing the enormous textual information manually is impossible and automation is required. The objective of sentiment analysis is to determine whether the reviews or opinions given by the people give a positive sentiment or a negative sentiment. This has to be predicted based on the given textual information in the form of reviews or ratings. Earlier linear regression and SVM based models are used for this task but the introduction of deep neural networks has displaced all the classical methods and achieved greater success for the problem of automatically generating sentiment analysis information from textual descriptions. Most recent progress in this problem has been achieved through employing recurrent neural networks (RNNs) for this task. Though RNNs are able to give state of the art performance for the tasks like machine translation, caption generation and language modeling, they suffer from the vanishing or exploding gradients problems when used with long sentences. In this paper we use LSTMs, a variant of RNNs to predict the sentiment analysis for the task of movie review analysis. LSTMs are good in modeling very long sequence data. The problem is posed as a binary classification task where the review can be either positive or negative. Sentence vectorization methods are used to deal with the variability of the sentence length. In this paper we try to investigate the impact of hyper parameters like dropout, number of layers, activation functions. We have analyzed the performance of the model with different neural network configurations and reported their performance with respect to each configuration. IMDB bench mark dataset is used for the experimental studies.

---

## 1. INTRODUCTION

Sentiment analysis is the task of processing the given textual information to analyze the emotions in it. In simple words we need to analyze whether the textual information talks positive or negative feedback about the product or topic. It is also popularly known as opinion mining. It requires the knowledge of natural language processing, artificial intelligence and machine learning. Sentiment analysis is all about what other people are thinking about something. Sentiment analysis is very much useful as it provides useful inferences and also helpful to understand public opinion on a product or service.

Internet is a rich source of such textual opinion or review information. Analyzing such information would give us lots of information and future insights. For example in an online shopping website people usually write their reviews after buying and using the product. These reviews are very much helpful for the customers who wish to buy that product. The problem here is, when the number of reviews is large in number, it is not possible for the customer to read all the reviews before taking a decision. So it would be helpful if we can automate this process and the task is popularly known as sentiment analysis. Potential applications of this task are:

Movie review analysis, product analysis, twitter opinion mining etc.

In this work we have focused on understanding the polarity of the given movie reviews by classifying whether it is positively polarized or negatively polarized. This problem can be posed as a multi label classification task where the final opinion could be worse, bad, neutral, good and excellent. In this work the problem is posed as a binary classification task where the final opinion can be either positive or negative. The reviews given by different people are of different lengths with different number of words in each review. Sentence vectorization methods are used to deal with the variability of the sentence length.

In this paper we try to investigate the affect of different hyper parameters like dropout, number of layers, activation functions. We have analyzed the performance of the model with different neural network configurations and reported their performance with respect to each configuration.

The IMDB benchmark dataset is used for our experimental studies that contain movie reviews that are classified as being positive or negative. In the experiment, an LSTM model is compared to other models and the LSTM model yields the best performance on the IMDB datasets.

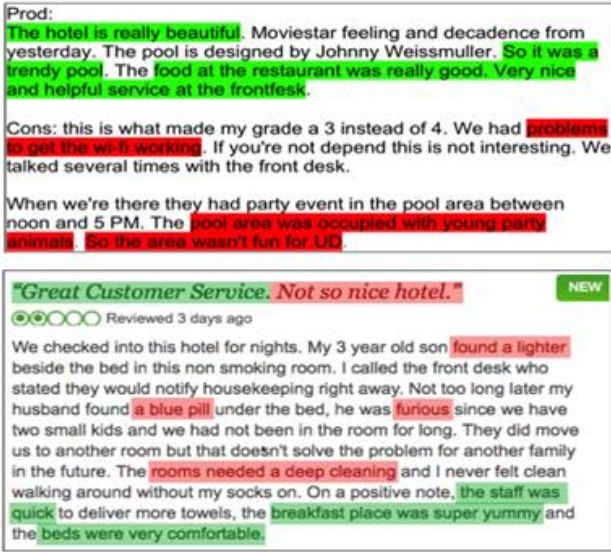


Figure 1. A sample positive and negative review

## 2. BACKGROUND WORK

Sentiment analysis is the process of analyzing the given textual information to analyze the emotions in it [1]. In recent past one or more of the following models are being used for this task. In vocabulary based methods, the important keywords would be identified and the review considered as positive, negative or neutral depending on the set of words it consists of. The task of Sentiment analysis can be achieved using two different types of techniques: Lexicon based and machine learning based techniques.

Lexicon based methods or corpus based methods leverage the set of words and semantics of the words in the given review. These are the unsupervised techniques so do not require labelled data. Machine learning based techniques are the supervised methods that rely on labeled data. These methods overcome the lexicon methods and the popular approaches are logistic regression, support vector machines (SVM), multi layer perceptron (MLP).

Among the machine learning based techniques, recently, deep neural networks [1] have displaced all the classical methods and achieve great success for the problem of automatically generating sentiment analysis information descriptions for images and videos. Most recent progress in this problem has been achieved through employing recurrent neural networks (RNNs) for this task. A Recurrent Neural Network is a type of Neural Network that is suitable to model sequence data [2]. These networks can better represent the temporal dynamics in the data.

RNNs are perfect to model sequential data as they are capable of remembering the input with its internal memory state and recurrent connections to learn and model sequential data as shown in Figure2. Formula for calculating current state:

$$h_t = f(h_{t-1}, x_t) = \tanh(w_{hh}h_{t-1} + w_{xh}x_t) \dots \dots \quad (1)$$

In Eq. (1),  $h_t$ ,  $h_{t-1}$  and  $x_t$  are current state, previous state and input state respectively and  $w_{hh}$  and  $w_{xh}$  represent weights associated with the hidden state and weights associated with the input respectively.

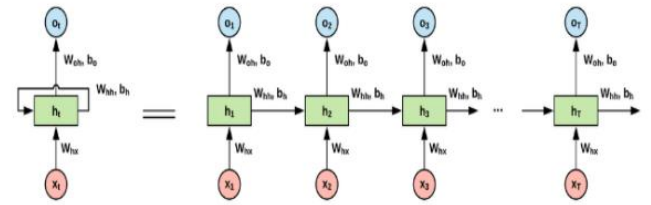


Figure 2. Working model of RNN

Though RNNs are capable of modeling long sequential data theoretically they fail to represent long sequences in real time applications [3]. This is mainly due to the vanishing or exploding gradients problem. RNNs are trained using back propagation through time (BPTT). For longer sequences when gradient flows back through time, it is possible that the gradient either can explode or can vanish and the model cannot be trained further. Therefore RNNs are not well suitable for modeling longer sequences in data.

To address the issue of exploding or vanishing gradients a variant of neural networks has been introduced called as, Long Short-Term memory networks (LSTMs), well suitable to model longer sequences in the data [4]. LSTM makes use of four gates to regulate the flow of data.

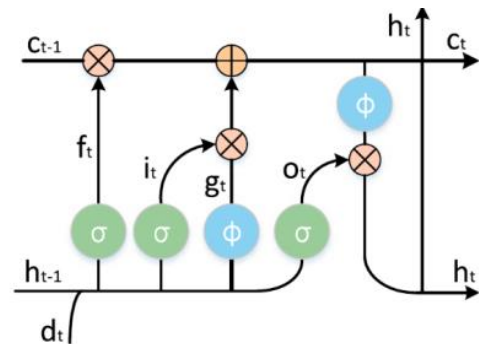


Figure 3. Illustration of the LSTM cell

Figure 3 shows the pictorial representation of LSTM cell. LSTM leverages four different gates to determine how much of the new information has to be added to the cell state (input (i)), how much of the previous cell state information has to be forgotten (forget (f)), gate (g) and output (o) gate along with the cell (c) and hidden ( $s_t$ ) state.  $\sigma$  represent logistic sigmoid. Following formulas are used to compute the state values at each gate.

$$i = \sigma(x_t U^i + s_{t-1} W^i)$$

$$f = \sigma(x_t U^f + s_{t-1} W^f)$$

$$o = \sigma(x_t U^o + s_{t-1} W^o)$$

$$g = \tanh\sigma(x_t U^g + s_{t-1} W^g)$$

$$c_t = c_{t-1}of + goi$$

$$s_t = \tanh(c_t)oo$$

GRU (gated recurrent unit) works the same way as LSTM cell but the number of gates is reduced in GRU by merging the input and forget gates functionality one gate is being designed

called update gate. Hence compared to LSTM it is simpler in operation and has very few in number of parameters which ultimately results in faster training time than LSTMs.

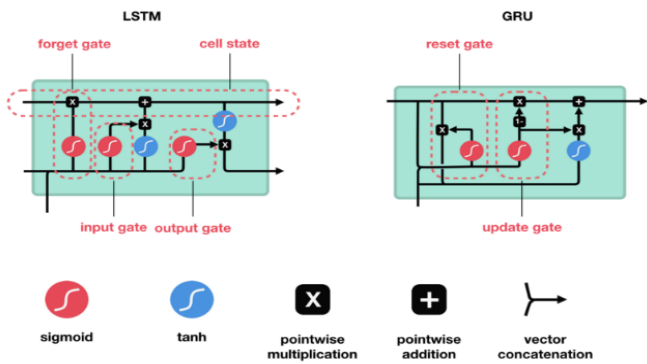


Figure 4. Illustration of LSTM vs GRU

### 2.1 Sentence vectorization

The objective of this task is to identify the polarity of the given review. Usually these reviews are in the text format and none of the neural network models can be used with the text data. The given text has to be converted to vector form to use it as input to the model. The process of converting the given reviews or any text format as a vector is known vectorization of the textual data. Two popular methods in the literature are one hot encoding and Bag-of-Words (BoW) representations [5]. In one hot encoding a binary representation is used with size of the vector equal to the number of words in the vocabulary. Each element of the vector represents the occurrence or no-occurrence of the corresponding word. In case of BoW method the  $i^{th}$  element of the vector represents the number of times  $i^{th}$  word of the vocabulary occurred in the given text. That is instead of using a binary 1 or a 0, frequency count is used to represent the number of times a particular word occurred. These methods are simple but the context of the words is not preserved in the representation. The resultant vectors are of size equal to the vocabulary size and are sparse [6].

Another method called word embedding is introduced [7] [8], where the vector size is much lesser than the size of the vocabulary. The sentences that are semantically similar are represented using similar features [9]. In the vector feature space the sentences that are semantically similar would result in higher similarity than the ones that are not.

One of the recent successful word embedding approach is the word2vec method [10]. In our work we make use of the word2vec representation as they are capable of preserving the context in the final feature representation [11] [12].

Once the representation is available any classification model can be used. As there is some sequence in the data LSTM can model the sentence vectors better and finally a SoftMax layer is used to classify the results.

## 3. PROPOSED WORK

The objective of this work is to analyze the reviews and predict the sentiment based on the reviews available. In this work we propose a sequential model to identify the sentiment analysis of the movie reviews. To process the sequential data, LSTM is proposed. LSTM is a variant of RNN that is majorly

used to process long sequential data. Following are the steps involved in the process: Identify the dataset, load the data, clean the data, then develop a vocabulary and save the processed data. In this work we use the IMDB, benchmark Movie Review Data is used for the experimental studies. IMDB is a collection of movie reviews retrieved from the imdb.com website.

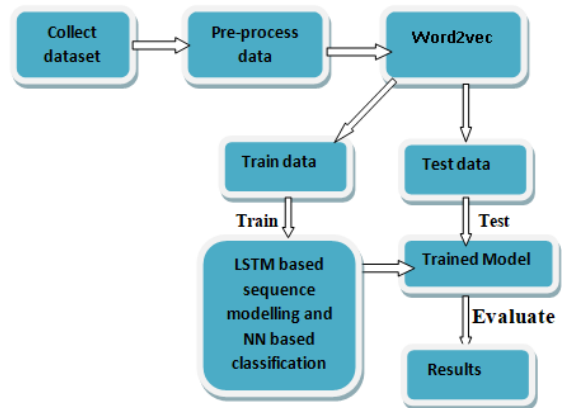


Figure 5. Block diagram of proposed sentiment classification

In the first step we have considered IMDB dataset for sentiment analysis. During the preprocessing stage, the entire text has been converted to lowercase and all the white spaces, punctuation marks and other special symbols are removed. Then tokenize the sentences and remove all the single letter words, numbers. Remove all the less frequent words also. Then the remaining words will form the vocabulary. The next step is to represent words of the vocabulary as feature vectors so that these feature representations can be used as input to the modelling. A more classical representation is the bag of words (BoW) representation. In this representation, if  $k$  is the size of the vocabulary then the feature vector size is  $k$  where every  $i^{th}$  element represents the number of occurrences of  $i^{th}$  word in the document. This method is very simple and is successfully used for language modelling and document classification. But the problem with this representation is the relationship between words is completely ignored. In the proposed work we use word2vec for word embedding. Word2Vec is one of the most popular technique to learn word embeddings using shallow neural network. Word2vec has two primary methods of contextualizing words: The Continuous Bag-of-Words model (CBOW) and the Skip-Gram model. CBOW, which is the less popular of the two models, uses source words to predict the target words. In practice, this model is very inefficient when working with a large set of words.

For our work we use the Skip-Gram model which works in the opposite fashion of the CBOW model, using target words to predict the source, or context, of the surrounding words. Here we train a simple neural network with a single hidden layer where the objective is to learn the weights of the hidden layer. The weights of the hidden layer represent the “word vectors” that we’re trying to learn.

## 4. EXPERIMENTAL RESULTS

### 4.1 Summary of the dataset

For our experimental study we use the IMDB dataset. It is

the large movie review dataset and is a bench mark for movie review dataset that contains a total of 50,000 reviews out of which 25000 are positively polarized and 25000 are negatively polarized. Among the total available reviews, 25,000 reviews are used for training and the remaining 25000 are used for evaluating the performance of the trained model. That is the same amount of data is used for training and testing. The objective of this work is to identify the polarity of the given review that is whether the review given is of positive sentiment or negative sentiment.

**Table 1.** Summary of the IMDB dataset

Dataset	# Total samples	# Train samples	#Test samples	#Classes
IMDB	50000	25000	25000	2

The task here is to classify whether the given reviews lead to a positive or negative sentiment. We use logistic regression, SVM based classification models to compare the results of our proposed model. SVM is a shallow model which is proved to be the most robust to classify the given data. But with the introduction of deep neural networks all the existing methods were fallen behind in terms of performance.

#### 4.2 Architecture of the proposed network used

We initialize word embedding layer with random values. Each word is represented with an embedding vector of size 32. Maximum length of the review is set to 500 as most of the reviews are having more than 500 words and very few are falling o the other side. The top 5000 words are used in the vocabulary and infrequent words are removed from the dictionary to avoid unnecessary computations.

During training the hyper parameters that resulted in best performance are: Dropout is applied with a rate of 0.5. Adam optimizer is used to optimize the model and binary cross entropy is used as the loss function. Initial learning rate is set to 0.001 and with a decay rate of 0.97. A batch size of 128 is adopted. The regularization parameter is set to 0.001 to avoid overfitting.

**Table 2.** Performance of the model with different configurations

Configuration of the model	Epochs	LSTM -units	Max_ review length	Accuracy
EMBEDDING LAYER + LSTM LAYER + DENSE LAYER	3	100	500	85.51 %
	3	100	1000	86.72 %
	3	100	500	87.44 %
	3	50	500	86.88 %
	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 %	

Table 2 exhibits the accuracy of the proposed model with different architectures

From Table 2 we can observe that with 100 LSTM units

when the model is trained with 50 epochs we reached the best performance when compared to the other models. When the review length is set to 1000 we can observe there is a dip in performance. When the number of LSTM units is increased to 200 then also we can observe the deterioration in performance. This could be due to overfitting problem.

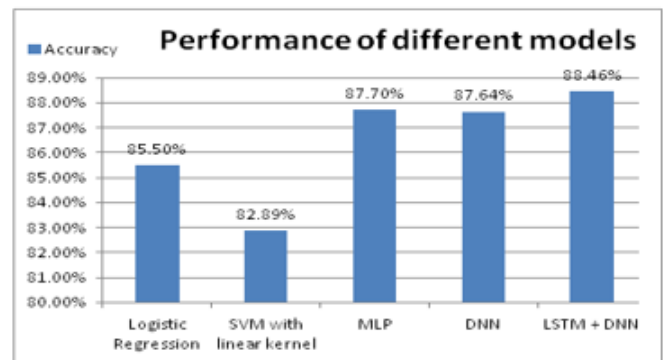
#### 4.3 Comparison with other models

This subsection gives a comparative study of performance of different classification models on the benchmark IMDB movie review dataset. The proposed model is compared with logistic regression, SVM, MLP and CNN. Except CNN all the other models are shallow models and SVM is the robust classification model compared to the other shallow models.

**Table 3.** Comparison of proposed model with existing models

SNO	Classification Model	Accuracy
1	Logistic Regression	85.5 %
2	SVM with linear kernel	82.89 %
3	MLP	87.70 %
4	DNN	87.64 %
5	LSTM + DNN	88.46 %

To compare the proposed model with the existing model a comparison study has been done using different existing models. Table 3 shows that the proposed LSTM based model with word2vec embedding gives better performance compared to other models. Linear regression is better than SVM. This could be due to the linear kernel used with the SVM model.



**Figure 6.** Comparison of proposed model with existing models

From Figure 6 it is very much evident that the proposed model is giving better performance when compared to the other models. SVM gives the least performance and MLP and DNN gives close to the LSTM based model.

#### 5. CONCLUSION AND FUTURE SCOPE

The objective of the work is to generate the polarity of the given review. For the word representations we proposed to use word2vec embeddings as it can represent the contextual information well compared to the other models. The experimental studies prove that the proposed method when used with LSTM based classification gives the best performance.

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] Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio, Y. (2014). Learning phrase representations using RNN encoder- decoder for statistical machine translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, pp. 1724–1734. <https://doi.org/10.3115/v1/D14-1>
- [2] Sutskever I, Vinyals O, Le QV. (2014). Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems.
- [3] Pascanu R, Mikolov T, Bengio Y. (2012). On the difficulty of training recurrent neural networks. In: ICML'13 Proceedings of the 30th International Conference on Machine Learning 28: 1310-1318. <https://doi.org/10.1007/s12088-011-0245-8>
- [4] Hochreiter 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>
- [5] Ramos J. (2013). Using TF-IDF to determine word relevance in document queries. In: Proceedings of the First Instructional Conference on Machine Learning, pp. 242.
- [6] Dong J, Li X, Snoek CGM. (2018). Predicting visual features from text for image and video caption retrieval. In: IEEE Transactions on Multimedia 20(12): 3377-3388. <https://doi.org/10.1109/TMM.2018.2832602>
- [7] Ain QT, Ali M, Riaz A, Noureenz A, Kamranz M, Hayat B, Rehman A. (2017). Sentiment analysis using deep learning techniques: A review. In: International Journal of Advanced Computer Science and Applications (IJACSA) 8(6): <https://doi.org/10.14569/IJACSA.2017.080657>
- [8] Sokolova M. (2018). Big text advantages and challenges: Classification perspective. In: International Journal of Data Science and Analytics 5(1): 1-10. <https://doi.org/10.1007/s41060-017-0087-5>
- [9] Bouazizi M, Ohtsuki T. (2018). Multi-class sentiment analysis in Twitter: What if classification is not the answer. In: IEEE Access 6: 64486-64502. <https://doi.org/10.1109/ACCESS.2018.2876674>
- [10] Yan XL, Subramanian P. (2018). A review on exploiting social media analytics for the growth of tourism. In: International Conference of Reliable Information and Communication Technology. Springer, Cham. [https://doi.org/10.1007/978-3-319-99007-1\\_32](https://doi.org/10.1007/978-3-319-99007-1_32)
- [11] Shafi MK, Bhat MR, Lone TA. (2018). Sentiment analysis of print media coverage using deep neural networking. In: Journal of Statistics and Management Systems 21(4): 519-527. <https://doi.org/10.1080/09720510.2018.1471263>
- [12] Petrolito R, Dell'Orletta F. (2018). Word embeddings in sentiment analysis. In: Proceedings of the Fifth Italian Conference on Computational Linguistics.