# IMDB Movie Review Analysis Using Bidirectional LSTM

Adarsh Gupta

Department of Computer Engineering
International Institute of Information Technology.
Bhubaneswar, India
B516003@iiit-bh.ac.in

Ayush Raizada

Department of Information Technology
International Institute of Information Technology.
Bhubaneswar, India
B416018@iiit-bh.ac.in

Sri Krishna Vijapurapu

Department of Information Technology
International Institute of Information Technology.
Bhubaneswar, India
B416018@iiit-bh.ac.in

# 1.1 Introduction:

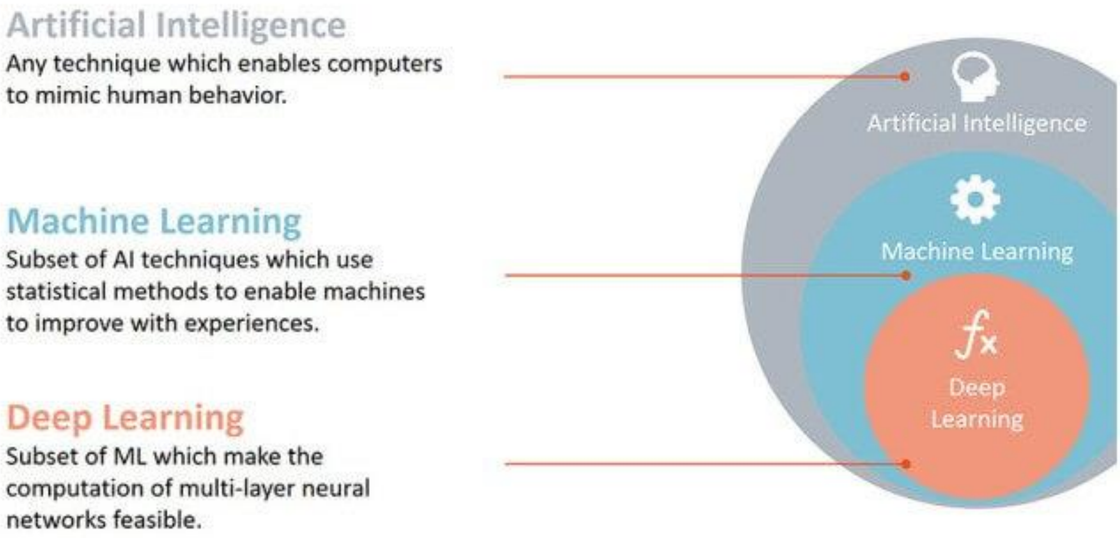
Python is an Interpreted language which in lay man’s terms means that it does not need to be compiled into machine language instruction before execution and can be used by the developer directly to run the program. This makes it comprehensive enough for the language to be interpreted by an emulator or a virtual machine on top of the native machine language which is what the hardware understands.

It is a High-Level Programming language and can be used for complicated scenarios. High-level languages deal with variables, arrays, objects, complex arithmetic or Boolean expressions, and other abstract computer science concepts to make it more comprehensive thereby exponentially increasing its usability.

Python is also a General-purpose programming language which means it can be used across domains and technologies. Python also features dynamic type system and automatic memory management supporting a wide variety of programming paradigms including object-oriented, imperative, functional and procedural to name a few.

Artificial intelligence is a conglomeration of concepts and technologies that mean different things to different people – self-driving cars, robots that impersonate humans, machine learning, and more – and its applications are everywhere you look.

An AI is a computer system that is able to perform tasks that ordinarily require human intelligence. These artificial intelligence systems are powered by machine learning. Many of them are powered by machine learning, some of them are powered by specifically deep learning, some of them are powered by very boring things like just rules.



Artificial Intelligence
Any technique which enables computers to mimic human behavior.

Machine Learning
Subset of AI techniques which use statistical methods to enable machines to improve with experiences.

Deep Learning
Subset of ML which make the computation of multi-layer neural networks feasible.

# 1.2 Objective:

It is common knowledge that, usually, moviegoers, called users or reviewers in the rest of this project, utilize movie ratings and reviews in selecting their next movie to see/watch. This is indeed the case for the authors of this project. And, unfortunately, sometimes movie reviews and ratings do not help users make the right choices, as evidenced by their emotional feelings after watching the movie. This is perhaps because users desire a certain emotional state after watching a movie, which does not match the emotions evoked by the selected movie. Clearly, user’s reviews and ratings for a movie are strongly tied to their emotions evoked by the movie. This project argues that

1) It may be useful for users in their decision-making process to choose the next movie to watch if a movie also comes with an (expected) emotion signature or an emotion map.

2) Towards goal 1, we can build automated software tools that

(i) Analyse movie reviews and ratings, and

(ii) Provide an emotional signature from the reviews and ratings of any movie.

3) Clearly, once emotion maps for all movies are at hand, if a user perhaps submits his desired emotion state and, possibly, the desired genre of the movie, it is easy to build a personalized movie recommender system for each user.

This project is a first step for goals 1 and 2 (but not for goal 3, due to space limitations), and makes an attempt to analyse the relationships between

• Users ratings for a movie and their emotions evoked by watching the movie as evidenced in their movie reviews and ratings, and

• Movies genres and users emotional responses from their movie reviews.

For our experimental evaluation, we used movie reviews from IMDB, the world’s most popular content source for movie, TV and celebrity content [1] with reviews for more than 3.5 million movies. IMDB members provide reviews and usefulness scores for other reviews. However, the overall rating of a movie is calculated by IMDB’s own rating algorithm, which, in turn, is based on reviewers scores .

# 1.3 Problem Statement:

The problem statement is, as stated in above section to predict whether a review is positive or negative for any given movie, the statistical and logical problem involved in doing so are;

1) Data is given in form of text and has to be encoded in order for a classifier to fit and predict as such
2) Cleaning or tokenizing the data alone is not sufficient because, there are too many commonly arising words such as a, an, the, me, I, you, they etc which are not of much importance and hence must be removed
3) The last problem to be tackled is normalizing the data to equivalent weights so as to get most accuracy over less data requirement.

## Libraries and Models:

### NLTK:

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries. We are using the **STOPWORDS, WordNet** to clean our text of recurring words and converting it to the most simplest form of sentence so as to improve our accuracy

### Keras:

We are using keras and underlying deep learning models to perform data cleaning, normalization and predictions. One such class we’re using is **Tokenizer,** this class allows to vectorize a text corpus, by turning each text into either a sequence of integers (each integer being the index of a token in a dictionary) or into a vector where the coefficient for each token could be binary, based on word count, based on tf-idf. In our case we converted the sentences to list first and then tokenized them with random numbers on the basis of TF-IDF.

## 2. Literature Survey:

The process of Sentiment Analysis involves the construction of the input vector space from the existing document vector space. Mainly there are two approaches to carry out vector space mapping. The machine learning based or statistical based feature extraction methods are widely used because extraction of features is done by applying statistical measures directly. Earlier works on sentiment classification using machine learning approaches were carried by Pang et al. in 2002 . Sentiment analysis was performed on IMDb movie reviews using n-gram approaches and Bag of Words (BOW) as features. The model was trained using different classifiers like Naïve Bayes (NB), Maximum Entropy (ME), and Support Vector Machines (SVM). The unigram features outperformed when compared to other features.

The use of an ensemble classifier which classifies based on the results obtained by different classifiers like NB, ME and SVM is the major highlight of the work. Many researchers have worked on extracting features based on the parts of speech tagger. Geetika et al. used unigram model to extract adjective as a feature which in turn describes the positivity or negativity of the sentence.

Identifying the semantics or the meaning of the text by a machine learning algorithm is a challenging task. Lexicon features are used in this regard to extract the opinions expressed in the text. Sarcasm detection is one of the major advantages of choosing lexicon features. Anukarsh et al. focused on the slangs and emojis which were present in the text to detect sarcasm. Use of slang and emoji dictionaries during preprocessing increased the efficiency of sarcasm detection. Capturing the sentiment orientation of the text towards a topic helps in identifying the overall polarity of the text. Taboda et al., in, used dictionaries to calculate the Semantic Orientation (SO) and termed it as Semantic Orientation CALculator (SO-CAL). Various factors such as Parts of Speech (Adjectives, Nouns, Verbs and Adverbs), Intensifiers (Somewhat, Very, Extraordinary etc.,), Negations, etc., were considered to calculate sentiment orientation. Results showed that the Lexicon based sentiment analysis gives better results and can be applied to wide domains. Similarly in , Dehkharghani developed lexicon for sentiment analysis.

Melville et al. Worked on extracting features using lexicon methods. Positive and negative word counts that are present in the text were used as the background lexicon knowledge and then the probability that a document belongs to a particular class was calculated. Use of pooling multinomial classifiers which incorporate both training examples and the background knowledge is the major contribution. Kolchyana et al., in, used both machine learning and lexicon approaches to perform sentiment analysis on Twitter data. Special lexicon features such as N-grams, Lexicon sentiment, Elongated words number, Emoticons, Punctuations, etc., were used. Use of these features increased the overall accuracy of the model. The hybrid method combines the features generated by both machine learning approach and lexicon approach. Use of a hybrid approach reduces the complexity of the overall model by retaining only the important features and thus increases time efficiency. The main advantage of using the lexicon features is that it captures the meaning or the semantics expressed in the reviews thereby contributing to the effective classification. The experimental results showed that the review classification was more accurate because of the use of semantics of the review as a feature and is comparable with the human review classification.

The polarity of a review depends on the intensity of each word present in the review and the context used by the reviewers to express their opinion. Therefore, identifying the features that extract the intensity of words based on context that inclines the polarity either towards positive or negative polarity is a challenging task. The proposed work captures the polarity of a word and determines how important the word is for the classification task. The capturing phase is done through the features generated using Hybrid Feature Extraction Method (HFEM). The HFEM combines the reduced Machine learning features with the Lexicon features to increase the performance of the model.

# 3. Data Collection:

The labelled data set consists of 50,000 IMDB movie reviews, specially selected for sentiment analysis. The sentiment of reviews is binary, meaning the IMDB rating < 5 results in a sentiment score of 0, and rating >=7 have a sentiment score of 1. No individual movie has more than 30 reviews. The 25,000 review labelled training set does not include any of the same movies as the 25,000 review test set. In addition, there are another 50,000 IMDB reviews provided without any rating labels.

### File descriptions:

- labeledTrainData - The labelled training set. The file is tab-delimited and has a header row followed by 25,000 rows containing an id, sentiment, and text for each review.
- testData - The test set. The tab-delimited file has a header row followed by 25,000 rows containing an id and text for each review. Your task is to predict the sentiment for each one.
- unlabeledTrainData - An extra training set with no labels. The tab-delimited file has a header row followed by 50,000 rows containing an id and text for each review.
- sampleSubmission - A comma-delimited sample submission file in the correct format.

### Data fields

- id - Unique ID of each review
- sentiment - Sentiment of the review; 1 for positive reviews and 0 for negative reviews
- review - Text of the review

# 4. Methodology:

## 4.1 Exploratory Data Analysis:

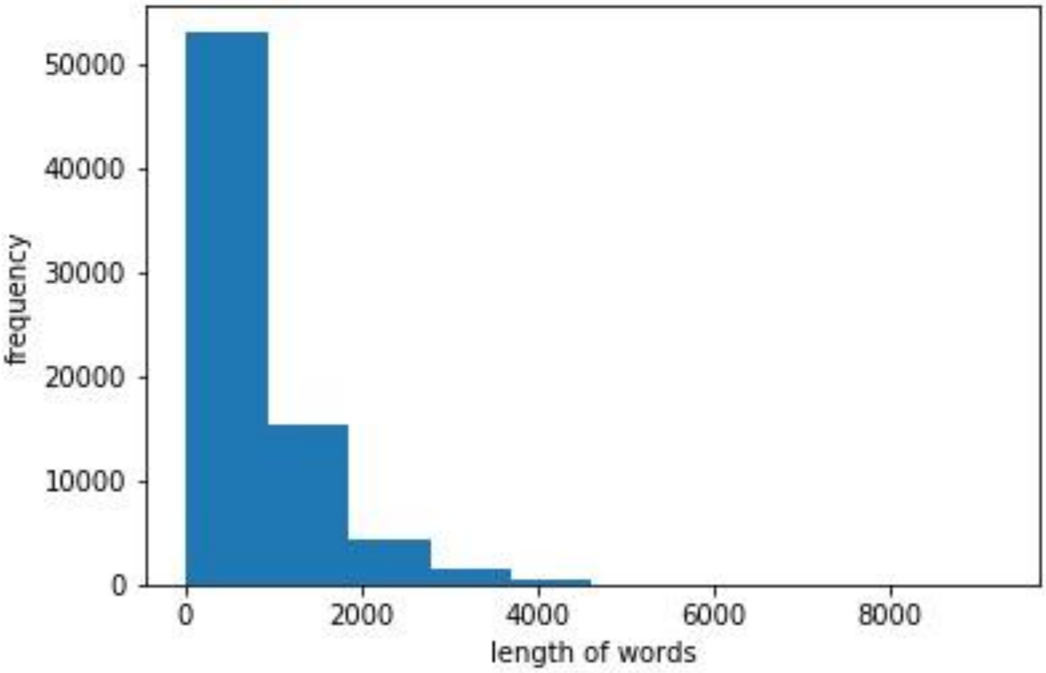
In the proposed model, sentiment analysis is employed on IMDb Movie Reviews. The input for the proposed model is the set of reviews whose polarity needs to be determined. The output corresponds to reviews with polarity assigned to each of them. The task of sentiment analysis is carried out in the following phases: preprocessing the dataset, feature Extraction, feature selection and finally classification using LSTM/RNN features. IMDb is the most commonly used website for getting information about a movie throughout the world. Because of its popularity and due to the presence of large number of reviews related to a particular movie, IMDb Movie Review Dataset is used in the proposed work. It is one of the standard benchmark datasets used for Sentiment Analysis on Movie reviews. The dataset contains 25,000 positive and negative reviews each.

We are using NLTK to remove stopwords and repetitive words that have not much impact on the sentiment which we are trying to predict also some reviews are quite long while others maybe short, so inorder to set a normalized weighted length we have to determine the mean length of reviews which can be done so using Numpy mathematical functions on said dataset.

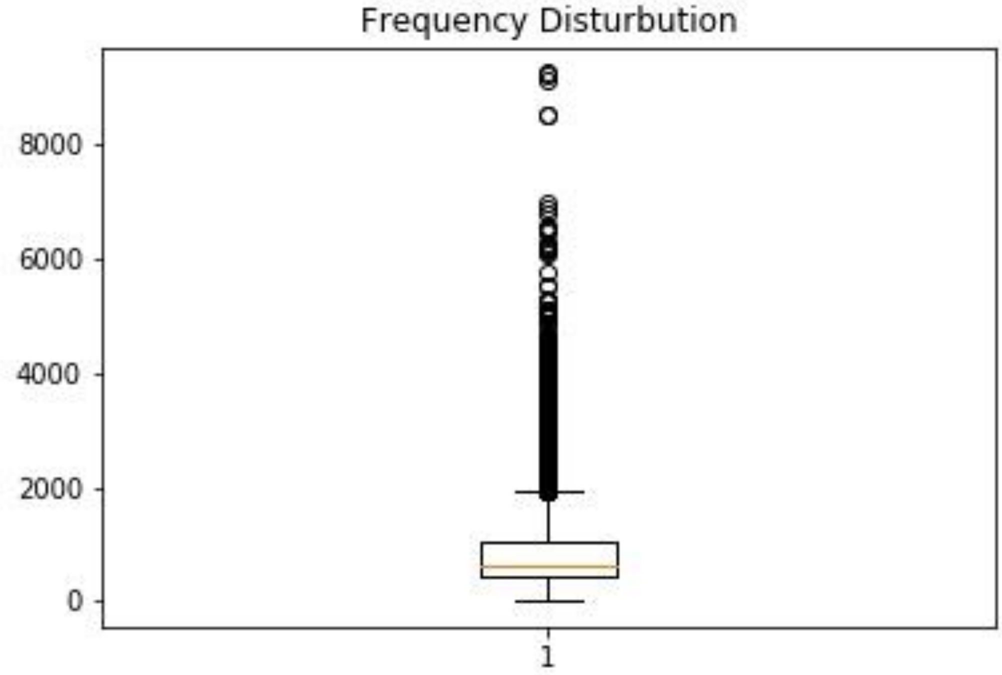
### 4.1.1 Figures :

.

**Histogram Plot**: Frequency vs Length of words



**Box Plot**: Frequency Distribution to find Maximum number of Features to be used.



# 4.2 Data Modelling:

## 4.2.1 Word Embedding:

A recent breakthrough in the field of natural language processing is called word embedding.
This is a technique where words are encoded as real-valued vectors in a high-dimensional space, where the similarity between words in terms of meaning translates to closeness in the vector space. Discrete words are mapped to vectors of continuous numbers. This is useful when working with natural language problems with neural networks and deep learning models are we require numbers as input.

Keras provides a convenient way to convert positive integer representations of words into a word embedding by an Embedding layer.

The layer takes arguments that define the mapping including the maximum number of expected words also called the vocabulary size (e.g. the largest integer value that will be seen as an integer). The layer also allows you to specify the dimensionality for each word vector, called the output dimension. Let’s say that we are only interested in the first 6,000 most used words in the dataset. Therefore our vocabulary size will be 6,000. We can choose to use a 32-dimension vector to represent each word. Finally, we may choose to cap the maximum review length at 500 words, truncating reviews longer than that and padding reviews shorter than that with 0 values. Since our mean review length has turned out to be 128.8 we have chosen to cap the maximum length to 130, also our average frequency distribution shows that at least 6000 features must be used so as to map 6000 different words.

## 4.2.2 Bidirectional LSTM:

Bidirectional LSTMs are an extension of traditional LSTMs that can improve model performance on sequence classification problems. In problems where all timesteps of the input sequence are available, Bidirectional LSTMs train two instead of one LSTMs on the input sequence. The first on the input sequence as-is and the second on a reversed copy of the input sequence. This can provide additional context to the network and result in faster and even fuller learning on the problem.

The idea of Bidirectional Recurrent Neural Networks (RNNs) is straightforward. It involves duplicating the first recurrent layer in the network so that there are now two layers side-by-side, then providing the input sequence as-is as input to the first layer and providing a reversed copy of the input sequence to the second.

Bidirectional LSTMs are supported in Keras via the Bidirectional layer wrapper. This wrapper takes a recurrent layer (e.g. the first LSTM layer) as an argument. It also allows you to specify the merge mode that is, how the forward and backward outputs should be combined before being passed on to the next layer.

### 4.2.3 Sequential Model :

The sequential model is a linear stack of layers.You create a sequential model by calling the keras_model_sequential() function then a series of layer functions:

- **Dense Layer**: A dense layer is just a regular layer of neurons in a neural network. Each neuron receives input from all the neurons in the previous layer, thus densely connected. The layer has a weight matrix **W,** a bias vector **b,** and the activations of previous layer.

  output = activation(dot(input, kernel) + bias)where activation is the element-wise activation function passed as the activation argument, kernel is a weights matrix created by the layer, and bias is a bias vector created by the layer.
- **Dropout:** Dropout is a a technique used to tackle Overfitting . The Dropout method in keras.layers module takes in a float between 0 and 1, which is the fraction of the neurons to drop Dropout works by randomly setting a fraction rate of input units to 0 at each update during training time, which helps prevent overfitting.
- **Global Max Pooling 1D:** Global max pooling is ordinary max pooling layer with pool size equals to the size of the input (minus filter size + 1, to be precise). You can see that MaxPooling1D takes a pool_length argument, whereas GlobalMaxPooling1D does not. For example, if the input of the max pooling layer is 0,1,2,2,5,1,20,1,2,2,5,1,2, global max pooling outputs 55, whereas ordinary max pooling layer with pool size equals to 3 outputs 2,2,5,5,52,2,5,5,5 (assuming stride=1).

# 5. Findings and Suggestions:

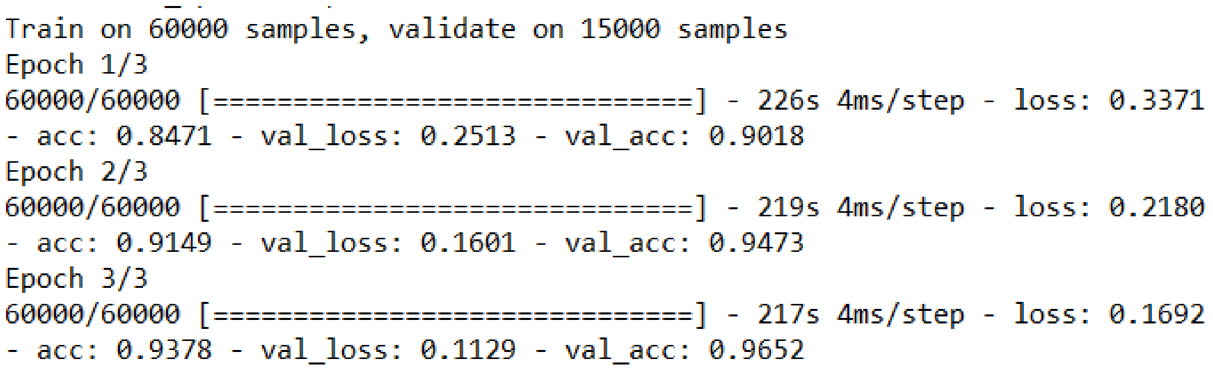
| Classifier | Number of Features | Accuracy(%) |
|---|---|---|
| Support Vector Machine SVM | 5000 | 75.467 |
| Naïve Bayes Classifier | 8000 | 54.733 |
| K-Nearest Neighbour Classification | 5000 | 72.267 |
| Bidirectional LSTM | 6000 | 94 |

From various research papers as published on the internet we have obtained information on how accurate the classisifiers were when predicting sentiment using movie reviews.

As such we observed that for any sentiment analysis the most important thing that a machine must consider is past data ie using Recurrent Neural Networks or Long Short Term Memory has substantially increased the learning rate and accuracy of prediction in sentiment analysis.

## 6. Conclusion:

In this project we successfully cleaned and normalized 60000 imdb movie review dataset and fitted it into a two layer Bidirectional LSTM neural network and made the machine learn to predict whether a review is positive or negative. We have achieved a good accuracy of 94% and also generated a function which takes input to tokenize any review and predict whether a review is positive or not.



```
Train on 60000 samples, validate on 15000 samples
Epoch 1/3
60000/60000 [==============================] - 226s 4ms/step - loss: 0.3371
- acc: 0.8471 - val_loss: 0.2513 - val_acc: 0.9018
Epoch 2/3
60000/60000 [==============================] - 219s 4ms/step - loss: 0.2180
- acc: 0.9149 - val_loss: 0.1601 - val_acc: 0.9473
Epoch 3/3
60000/60000 [==============================] - 217s 4ms/step - loss: 0.1692
- acc: 0.9378 - val_loss: 0.1129 - val_acc: 0.9652
```

This project could be essential in future generations where every product based company would definitely like to analyse which of its products are getting a positive review, this is not just for movies but can be used for any sentiment prediction provided necessary data is processed as such.

### References:

[1] Vidisha M. Pradhan, Jay Vala, A Survey on Sentiment Analysis Algorithms for Opinion Mining, https://www.researchgate.net/publication/290788060 ,

[2] V. Vanitha, V. P. Sumathi, V. Soundariya , An Exploratory Data Analysis of Movie Review Dataset, https://www.ijrte.org/wp-content/uploads/papers/v7i4s/

[3] H. M. Keerthi Kumar, B. S. Harish, H. K. Darshan, Sentiment Analysis on IMDb Movie Reviews Using Hybrid Feature Extraction Method