Goodreads Books Reviews

A Project Report in partial fulfillment of the degree

Bachelor of Technology

in

Electronics & Communication Engineering/Computer Science & Engineering

By

19K41A04H6	Ravi Teja. S
19K41A05D9	Ujjvala Sindhu. P
19K41A05H1	Sushanth. T

Under the Guidance of

D. Ramesh

Submitted to

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
S R ENGINEERING COLLEGE(A), ANANTHASAGAR, WARANGAL
(Affiliated to JNTUH, Accredited by NBA)

November-2022

SR Engineering College
Innovation . Creativity . Entrepreneurship

DEPARTMENT OF COMPUTER SCIENCE &ENGINEERING

CERTIFICATE

This is to certify that the Project Report entitled "Goodreads Book reviews" is a record of bonafide work carried out by the student(s) S. Ravi Teja, P. Ujjvala Sindhu, T. Sushanth bearing Roll No(s) 19K41A04H6, 19K41A05D9, 19K41A05H1 during the academic year 2021-2022 in partial fulfillment of the award of the degree of ***Bachelor of Technology*** in **Electronics & Communication/Computer Science Engineering** by the Jawaharlal Nehru Technological University, Hyderabad.

.

Supervisor **Head of the Department**

External Examiner

ABSTRACT

Books are said to be the best friend a person can have. Book reading culture dates back to almost a couple of thousands of years. After ancient civilizations learned to write, they stored information in tablets or walls or stones are said to be the predecessors of books. The newest form of books is called e-books, digitalization or digital printing of paper-based books. A few years back, people had to go to the library in person to collect books but now online book stores are getting popular. With all its perks being easy, online book store comes with some penalty i.e., reader don't know about the books or the service of the book store itself. To avoid such, book readers tend to rely on reviews and ratings. Our goal is to provide rating with review provided. So that book readers can buy desired books to read and get better services from online book stores. In this report, we proposed a good reads review rating prediction using LSTM and Universal Sequence Encoder.

Table of Contents

1. INTRODUCTION

The rise in E — commerce, has brought a significant rise in the importance of customer reviews. There are hundreds of review sites online and massive amounts of reviews for every product. Customers have changed their way of shopping and according to a recent survey, 70 percent of customers say that they use rating filters to filter out low rated items in their searches. The ability to successfully decide whether a review will be helpful to other customers and thus give the product more exposure is vital to companies that support these reviews, companies like Google, Amazon and Yelp.

Online reviews play a great role in influencing the shopping decisions made by consumers. These reviews provide consumers with information and experience about product quality. Online reviews commonly comprise of a free-text format, user star-level rating Out of five and numerical scale rating that is 0-5 or 0-10. People believe that reviews will do help to the rating predication based on the idea that high star rating,5 number rating may significantly be attach with really good reviews. However, user's rating star-level information is not usually available on many online review's websites. Due to, it's not possible for a given user to rate every product. On the other hand, most online reviews are written in free-text format, and therefore difficult for computer system to understand and analyse it. Identifying ratings for online reviews lately become an important topic in Natural Language Processing.

With the rapid development of Internet, the number of netizens has risen sharply, and more and more netizens express their experience of products in online communities. As we know the Internet is growing up thus the textual information also is growing very fast. One of this textual information is the customer comments or reviews. People usually prefer to read the reviews before buying or using a service to make the right decision. This behaviour is also common before the existence of the Internet. From this amount of available data, researches attempt to handle and use these data to have a specific and useful knowledge.

In general, people are influenced by others' opinions. As a real-life example, a person will go and eat in a specific restaurant after asking the people who tried this restaurant before. This is a common behaviour, and there are several statistical studies such as:

- A study states the influence of the reviews where 64% of them spent 10 minutes to read the reviews, and 33% spent half hour or more in reading online reviews. Also, before buying: 39% read around 8 reviews or more and 12% read 16 reviews or more.
- 90% of the buyers said: their buying decision is affected by the online reviews.
- The action after reading a positive review, 48% of the survey responders are motivated to visit the business' website, and 21% are shopping around.
- A survey shows how important is the reviews before buying products: where 44.8% said it is important and 35% said it is very important while 4.7% only said the reviews are unimportant.
- 84% of the survey responders said: they trusted the online reviews as personal recommendation (vs. 80% in 2015)

2. LITERATURE REVIEW

- In Alshari, E., Azman, A., Mustapha, N., Doraisamy, S., Alksher, M. (2016). Prediction of rating from comments based on information retrieval sentiment analysis, In: 2016 Third International Conference on Information Retrieval and Knowledge Management (CAMP). The authors used the combination of sentiment analysis with information retrieval to predict the rating of Amazon comments. Vector Space Model (VSM) was applied as a supervised classifier. They compared it with the combination of VSM with sentiment analysis. The Lexical dictionary approach was used as sentiment analysis with the VSM. The obtained result shows that the usage of sentiment analysis has a positive effect on the performance of the classifier in their rating prediction.
- K. S. Srujan et al. discussed, the different pre-processing methods named as HTML tags and URLs removal, punctuation, whitespace, special character removal and stemming are used to eliminate noise. The pre-processed data is characterized using feature selection methods like term frequency-inverse document frequency (TF–IDF). The classifiers namely K-Nearest Neighbour (KNN), Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF) and Naive Bayes (NB) are used to classify sentiment of Amazon book reviews.
- C. Balasubramanian et al. a hybrid approach which consists of both models based and memory-based algorithms in order to increase the performance of the system. A similar recommender system can be incorporated for Music or Book recommendation systems. It is vital that precise recommendations are provided to the users.
- Santosh Kumar et al. performed a survey on different approaches available for recommender system and performs a comparative analysis of different algorithms. In addition, various applications have been discussed. At the end, issues and challenges in recommender systems have been discussed.

- V.K. Kiran et al. proposed approach in this study removes specification list like battery, processor, camera etc. and consumer reviews for a user mentioned product from variant websites and identifies crucial terms corresponding to the technical features of the product in the review to determine polarity of the feature and classifying it under the specification list. Each specification is assigned a score based on polarity i.e., positive/negative feedback. Overall product rate is computed by aggregating the score specific to individual features. This approach is very useful for those customers who target at specific features in a product.
- AyseCufoglu the proposed system aims to give an overview on the user profiling and its related concepts, and discuss the pros and cons of current methods for the future service personalization. Furthermore, it also gives details about the simulations which have been carried out with well-known classification and clustering algorithms with real world user profile dataset.
- Bhatt et al. proposed a system for sentiment analysis on iphone5 reviews. The methodology integrates various pre-processing techniques to reduce noisy data like HTML tags, punctuations and numbers. The features are extracted using part -of speech (POS) tagger and rule-based methods are applied to classify the reviews into different polarity. A rule-based mining of product feature sentiment is also done. And also provides a visualization and summarization.
- Tripathy et al. presented a comparison of different classifiers based on accuracy for movie review dataset. The methodology incorporated various pre-processing techniques to reduce noisy data like whitespaces, numbers, stop word removal and vague information removal. The features are extracted and represented by count vectorizer and TF–IDF. Naïve Bayes (NB) and Support Vector Machine (SVM) are used to classify the data as positive or negative. The dataset considered for training and testing of model during this work is marked dependent on polarity movie dataset and a correlation with results obtainable in existing literature has been made for basic examination. By comparing accuracy of NB with SVM, SVM achieved accuracy of 94%.
- Xing Fang et al., this paper tackles the problem of sentiment polarity categorization, which is one of the fundamental problems of sentiment analysis. A general method for sentiment polarity categorization is proposed with detailed method descriptions. Data utilized in this study are online product reviews

collected from Amazon.com. Experiments for each sentence-level categorization and review-level categorization are made with promising outcomes.

S.NO.	AUTHOR NAME	TITLE OF THE PAPER	JOURNAL NAME/YEAR	MERITS	DEMERITS
1	K. S. Srujan et al.	Classification of Amazon Book Reviews Based on Sentiment Analysis	Information Systems Design and Intelligent Applications (ISDIA), 2018, Springer.	Gave accuracy 90.15% for six books by processing of various classifiers, Random Forest ranked highest.	Varies with dimensionalities.
2	C.Balasubramanian et al.	A Personalized User-Recommendation Based on Attributes Clustering and Score Matrix	International Journal of Pure and Applied Mathematics (IJPAM), 2018.	User based Collaborative Filtering techniques made great contributions to rate prediction and Recommendation.	Hybrid approach will increase the performance of system.
3	Santosh Kumar et al.	Survey on Personalized Web Recommender System	I.J. Information Engineering and Electronic Business (IJIEEB), July 2018.	Different challenges and issues of recommendation system discussed.	Over-Specialization due to change in the interest of the user.
4	V. K. Kiran et al.	User specific product recommendation and rating system by performing	4th International Conference on Advanced Computing and Communicatio	The approach serves as a better alternative to rate a product based on its technical	It is very difficult to read through each individual Review of various items and make a good decision for

		sentiment analysis on product reviews	n Systems (ICACCS), 2017, IEEE.	specification by analyzing large number of user reviews.	an individual customer.
5	AyseCufoglu	User Profiling – A Short Review	International Journal of Computer Applications (IJCA), December 2016.	Naive Bayes Tree classifier archives better accuracy results with user profile dataset.	Lacks in representing multi -dimensionality of the user profile.
6	Aashutosh Bhatt et al.	Amazon Review Classification and Sentiment Analysis	International Journal of Computer Science and Information Technologies (IJCSIT), 2015.	Less time consumption due to Data Visualization and Summarized as bar charts and pie charts to help users to understand easily.	Data visualization takes more time and space.
7	AbinashTripa t hy et al.	Classification of Sentimental Reviews Using Machine Learning Techniques	International Conference on Recent Trends in Computing (ICRTC-2015), Elsevier.	SVM classifier gave better accuracy in predicting sentiment of a review.	Single fold is considered for testing and only two classifiers implemented.
8	X. Fang et al.	Sentiment analysis using product review data	Journal of BigData, 2015, Springer.	The POS tagging is used to extract the most relevant features to get better results in classifying the sentence as positive or negative. To analyze the quality of the online products	The fake comments about the product, which gives the bad review about the product or not identified. SA Problem can be sometimes managed by manual methods.

3. DESIGN:

3.1 Requirement Specifications (S/W & H/W)

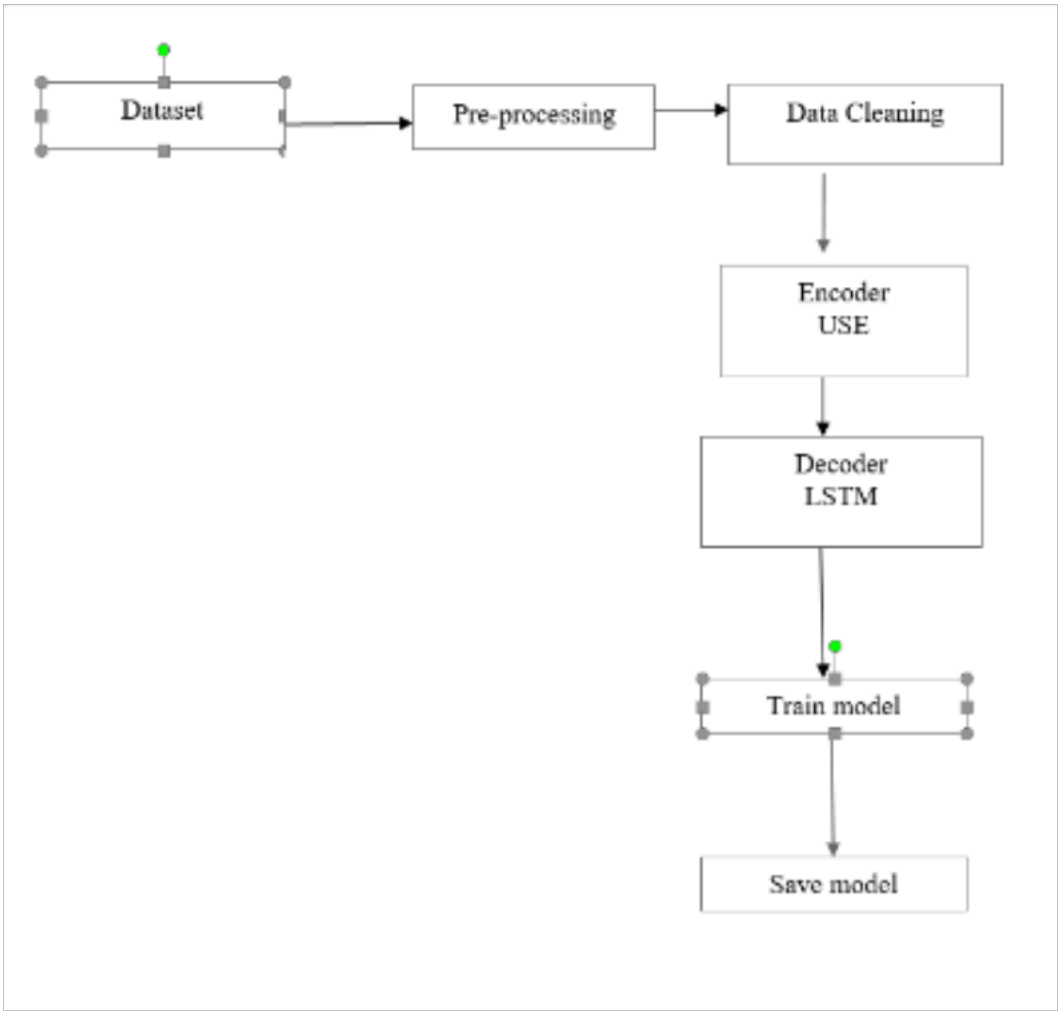
Hardware Requirements

- ✓ **System** : Processor Intel(R) Core (TM) i5-8265U CPU @ 1.60GHz, 1800 MHz, 4 Cores, 8 Logical Processors
- ✓ **RAM** : 8 GB
- ✓ **Hard Disk** : 500GB
- ✓ **Input** : Keyboard and Mouse
- ✓ **Output** : PC

Software Requirements

- ✓ **OS** : Windows 11
- ✓ **Platform** : Google Collaboratory / Jupyter Notebook
- ✓ **Program Language** : Python

3.2 Flow chart



4. DATASET

The Data Set which we used in our project is taken from the Kaggle. The information present in the data is:

Available

- user_id - Id of user
- book_id - Id of Book
- review_id - Id of review
- rating - rating from 0 to 5
- review_text - review text
- date_added - date added
- date_updated - date updated
- read_at - read at
- started_at - started at
- n_votes - no. of votes

- n_comments - no. of comments

Used in our project

- review_text - review text
- rating - rating from 0 to 5

5. DATA PREPROCESSING

1) data collection:

we have collected the dataset from the kaggle website the dataset name is Goodreads Book review This dataset contains more than 1.3M book reviews about 25,475 books and 18,892 users , which is a review subset for spoiler detection, where each book/user has at least one associated spoiler review.

2) data preparation:

our main goal is to predict the rating based on the review in text format so we need to consider only these two columns and removed columns that are user_id, book_id, 'review_id, data_added, data_updated, read_at, started_at, n_votes, n_comnents from the dataset.

3) Decreasing the number of samples:

our dataset consist of nearly 9 lakhs of samples but we have 2 lakh of samples because 9 lakhs of need model with high priority and in Google colab it need to use

the Google Collab Pro service which is of premium.

4) Data conversion:

we have converted the rating data which is numeric format into string that is '0' means very bad, '1' means bad, '2' means average, '3' means very good, '4' and '5' means excellent.

5) Encoder:

we have taken universal sentence encoder. The Universal Sentence Encoder encodes text into high dimensional vectors that can be used for text classification, semantic similarity, clustering, and other natural language tasks. The pre-trained Universal Sentence Encoder is publicly available in Tensorflow-hub. It comes with two variations i.e. one trained with Transformer encoder and other trained with Deep Averaging Network (DAN). in our model we are using transformer encoder

6) Data cleaning:

we have used data cleaning methods are stemming.Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. Stemming is important in natural language understanding (NLU) and natural language processing (NLP).

7) Decoder:

we have used LSTM model as our decoder with 3 LSTM layers and 3 Hidden layers .LSTM is takes the main keyword from sentence and stores as a constant in state machine it will treat it as constant throughout the process and it will predict the output.

8) Save model:

Finally we need to save the model with .h5 extension and we can use this model for further deployment.

6. METHODOLOGY:

The Models we used in our project are LSTM (Long short term memory) and Universal sequence encoder

Long Short-Term Memory (LSTM)

LSTM networks are an extension of recurrent neural networks (RNNs) mainly introduced to handle situations where RNNs fail. Talking about RNN, it is a network that works on the present input by taking into consideration the previous output (feedback) and storing in its memory for a short period of time (short-term memory). Out of its various applications, the most popular ones are in the fields of speech processing, non-Markovian control, and music composition. Nevertheless, there are drawbacks to RNNs. First, it fails to store information for a longer period of time. At times, a reference to certain information stored quite a long time ago is

required to predict the current output. But RNNs are absolutely incapable of handling such “long-term dependencies”. Second, there is no finer control over which part of the context needs to be carried forward and how much of the past needs to be ‘forgotten’. Other issues with RNNs are exploding and vanishing gradients (explained later) which occur during the training process of a network through backtracking. Thus, Long Short-Term Memory (LSTM) was brought into the picture. It has been so designed that the vanishing gradient problem is almost completely removed, while the training model is left unaltered. Long-time lags in certain problems are bridged using LSTMs where they also handle noise, distributed representations, and continuous values. With LSTMs, there is no need to keep a finite number of states from beforehand as required in the hidden Markov model (HMM). LSTMs provide us with a large range of parameters such as learning rates, and input and output biases. Hence, no need for fine adjustments. The complexity to update each weight is reduced to O(1) with LSTMs, similar to that of Back Propagation Through Time (BPTT), which is an advantage.

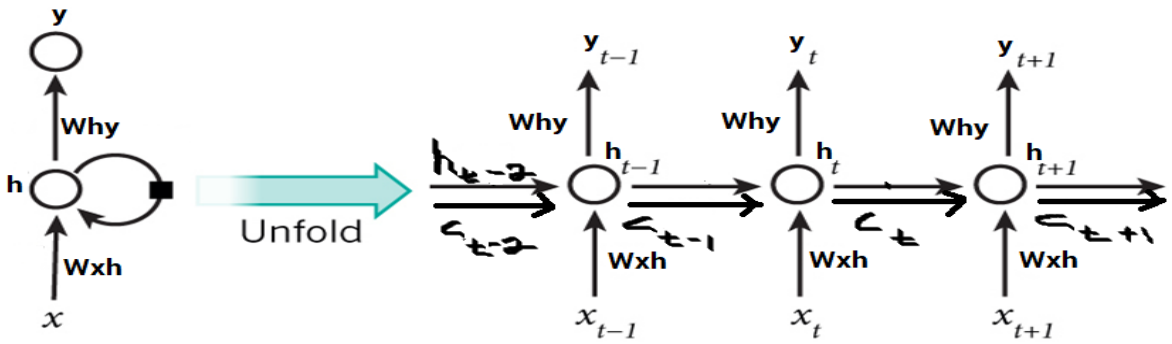
Exploding and Vanishing Gradients:

During the training process of a network, the main goal is to minimize loss (in terms of error or cost) observed in the output when training data is sent through it. We calculate the gradient, that is, loss with respect to a particular set of weights, adjust the weights accordingly and repeat this process until we get an optimal set of weights for which loss is minimum. This is the concept of backtracking. Sometimes, it so happens that the gradient is almost negligible. It must be noted that the gradient of a layer depends on certain components in the successive layers. If some of these components are small (less than 1), the result obtained, which is the gradient, will be even smaller. This is known as the scaling effect. When this gradient is multiplied with the learning rate which is in itself a small value ranging between 0.1-0.001, it results in a smaller value. As a consequence, the alteration in weights is quite small, producing almost the same output as before. Similarly, if the gradients are quite large in value due to the large values of components, the weights get updated to a value beyond the optimal value. This is known as the problem of exploding gradients. To avoid this scaling effect, the neural network unit was re-built in such a way that the scaling factor was fixed to one. The cell was then enriched by several gating units and was called LSTM.

Architecture:

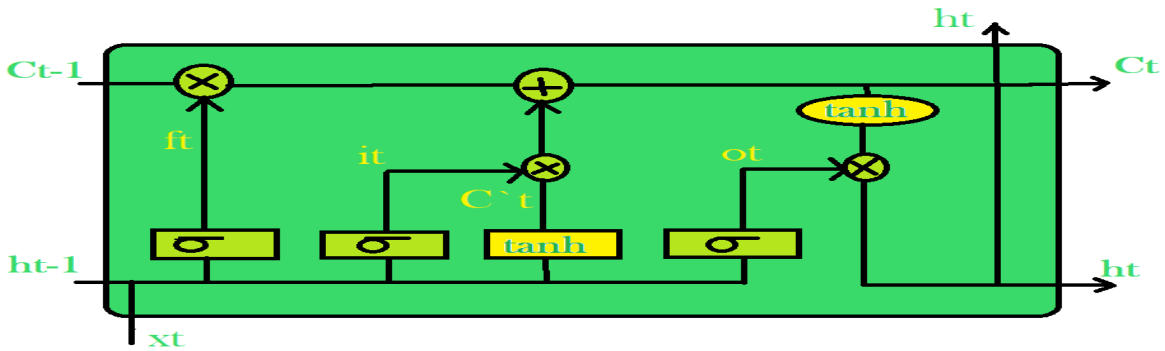
The basic difference between the architectures of RNNs and LSTMs is that the hidden layer of LSTM is a gated unit or gated cell. It consists of four layers that interact with one another in a way to produce the output of that cell along with the cell state. These two things are then passed onto the next hidden layer. Unlike RNNs which have got the only single neural net layer of tanh, LSTMs comprises of three logistic sigmoid gates and one tanh layer. Gates have been introduced in order to limit the information that is passed through the cell. They determine which part of the information will be needed by the next cell and which part is to be discarded. The output is usually in the range of 0-1 where '0' means 'reject all' and '1' means 'include all'.

Hidden layers of LSTM :



Each LSTM cell has three inputs h_{t-1}, C_{t-1} and x_t and two outputs h_t and C_t . For a given time t, h_t is the hidden state, C_t is the cell state or memory, x_t is the current data point or input. The first sigmoid layer has two inputs– h_{t-1} and x_t where h_{t-1} is the hidden state of the previous cell. It is known as the forget gate as its output selects the amount of information of the previous cell to be included. The output is a number in [0,1] which is multiplied (point-wise) with the previous cell state C_{t-1}.

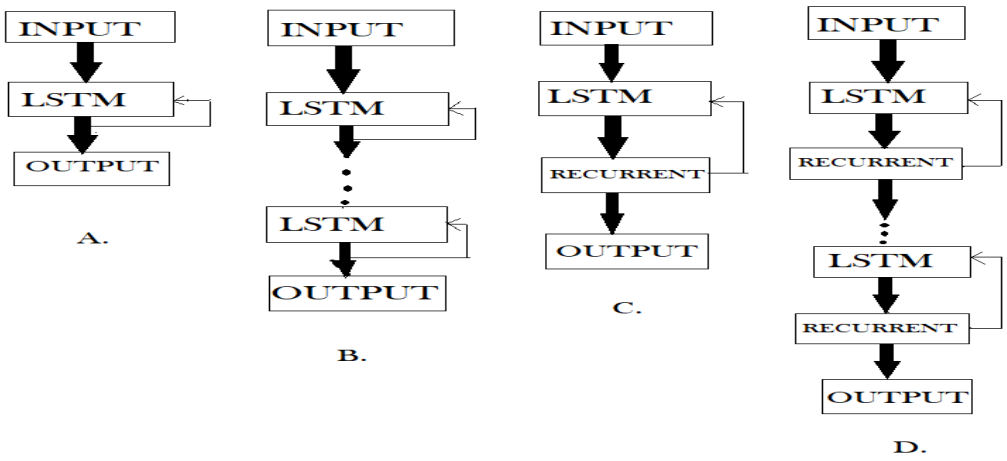
Conventional LSTM:



The second sigmoid layer is the input gate that decides what new information is to be added to the cell. It takes two inputs $h_{t-1}$and x_t. The tanh layer creates a vectorC_t of the new candidate values. Together, these two layers determine the information to be stored in the cell state. Their point-wise multiplication $(i_t?C_t)$ tells us the amount of information to be added to the cell state. The result is then added with the result of the forget gate multiplied with previous cell state $(f_t * C_{t-1})$to produce the current cell state C_t. Next, the output of the cell is calculated using a sigmoid and a tanh layer. The sigmoid layer decides which part of the cell state will be present in the output whereas tanh layer shifts the output in the range of [-1,1]. The results of the two layers undergo point-wise multiplication to produce the output ht of the cell.

Variations:

With the increasing popularity of LSTMs, various alterations have been tried on the conventional LSTM architecture to simplify the internal design of cells to make them work in a more efficient way and to reduce the computational complexity. Gers and Schmidhuber introduced peephole connections which allowed gate layers to have knowledge about the cell state at every instant. Some LSTMs also made use of a coupled input and forget gate instead of two separate gates that helped in making both the decisions simultaneously. Another variation was the use of the Gated Recurrent Unit(GRU) which improved the design complexity by reducing the number of gates. It uses a combination of the cell state and hidden state and also an update gate which has forgotten and input gates merged into it.



LSTM(Figure-A), DLSTM(Figure-B), LSTMP(Figure-C) and DLSTMP(Figure-D)

Figure-A represents what a basic LSTM network looks like. Only one layer of LSTM between an input and output layer has been shown here.

Figure-B represents Deep LSTM which includes a number of LSTM layers in between the input and output. The advantage is that the input values fed to the network not only go through several LSTM layers but also propagate through time within one LSTM cell. Hence, parameters are well distributed within multiple layers. This results in a thorough process of inputs in each time step.

Figure-C represents LSTM with the Recurrent Projection layer where the recurrent connections are taken from the projection layer to the LSTM layer input. This architecture was designed to reduce the high learning computational complexity (O(N)) for each time step) of the standard LSTM RNN.

Figure-D represents Deep LSTM with a Recurrent Projection Layer consisting of multiple LSTM layers where each layer has its own projection layer. The increased depth is quite useful in the case where the memory size is too large. Having increased depth prevents overfitting in models as the inputs to the network need to go through many nonlinear functions.

Applications:

LSTM models need to be trained with a training dataset prior to its employment in real-world applications. Some of the most demanding applications are discussed below:

- Language modelling or text generation, that involves the computation of words when a sequence of words is fed as input. Language models can be

operated at the character level, n-gram level, sentence level or even paragraph level.

- Image processing, that involves performing analysis of a picture and concluding its result into a sentence. For this, it's required to have a dataset comprising of a good amount of pictures with their corresponding descriptive captions. A model that has already been trained is used to predict features of images present in the dataset. This is photo data. The dataset is then processed in such a way that only the words that are most suggestive are present in it. This is text data. Using these two types of data, we try to fit the model. The work of the model is to generate a descriptive sentence for the picture one word at a time by taking input words that were predicted previously by the model and also the image.
- Speech and Handwriting Recognition
- Music generation which is quite similar to that of text generation where LSTMs predict musical notes instead of text by analyzing a combination of given notes fed as input.
- Language Translation involves mapping a sequence in one language to a sequence in another language. Similar to image processing, a dataset, containing phrases and their translations, is first cleaned and only a part of it is used to train the model. An encoder-decoder LSTM model is used which first converts input sequence to its vector representation (encoding) and then outputs it to its translated version.

Drawbacks:

As it is said, everything in this world comes with its own advantages and disadvantages, LSTMs too, have a few drawbacks which are discussed as below:

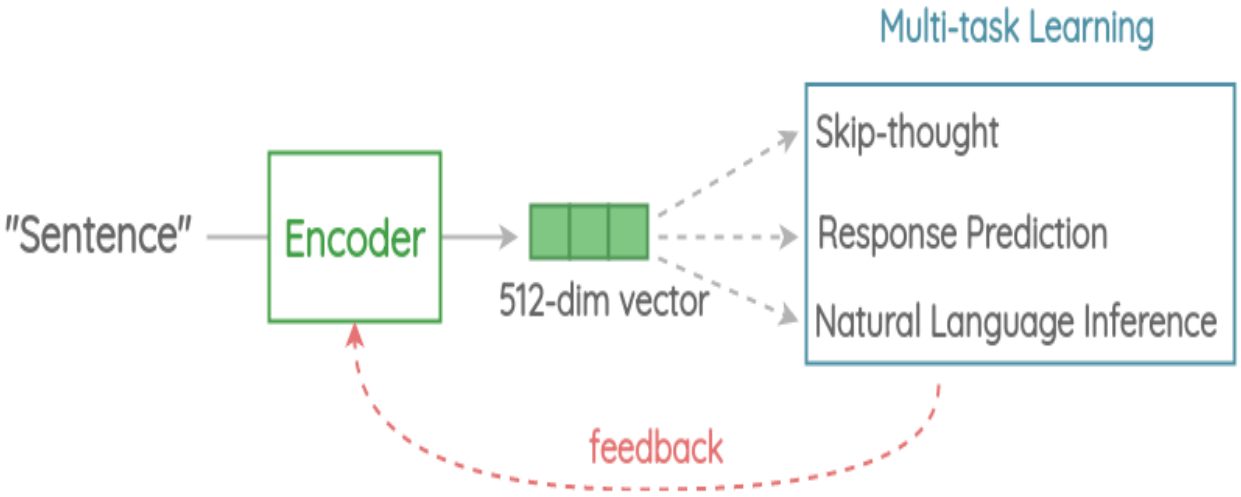
- LSTMs became popular because they could solve the problem of vanishing gradients. But it turns out, they fail to remove it completely. The problem lies in the fact that the data still has to move from cell to cell for its evaluation. Moreover, the cell has become quite complex now with the additional features (such as forget gates) being brought into the picture.
- They require a lot of resources and time to get trained and become ready for real-world applications. In technical terms, they need high memory-bandwidth because of linear layers present in each cell which the system usually fails to provide for. Thus, hardware-wise, LSTMs become quite inefficient.

- With the rise of data mining, developers are looking for a model that can remember past information for a longer time than LSTMs. The source of inspiration for such kind of model is the human habit of dividing a given piece of information into small parts for easy remembrance.
- LSTMs get affected by different random weight initialization and hence behave quite similar to that of a feed-forward neural net. They prefer small weight initialization instead.
- LSTMs are prone to overfitting and it is difficult to apply the dropout algorithm to curb this issue. Dropout is a regularization method where input and recurrent connections to LSTM units are probabilistically excluded from activation and weight updates while training a network.

Universal sequence encoder

The universal sentence encoder makes looking up embeddings at the sentence level as simple as it has previously been to look up embeddings at the word level. Then, using less supervised training data, the sentence embeddings can be easily employed to compute sentence level meaning similarity and improve performance on subsequent classification tasks. The universal sentence encoder model converts textual information into numerically represented, high-dimensional vectors called embeddings. It aims to transfer learning especially to other NLP tasks like text categorization, semantic similarity, and clustering. The freely accessible universal sentence encoder is listed in Tensor flow-hub. To learn for a wide range of jobs, it is trained on a number of data sources.

On a high level, the idea is to design an encoder that summarizes any given sentence to a 512-dimensional sentence embedding. We use this same embedding to solve multiple tasks and based on the mistakes it makes on those, we update the sentence embedding. Since the same embedding has to work on multiple generic tasks, it will capture only the most informative features and discard noise. The intuition is that this will result in an generic embedding that transfers universally to wide variety of NLP tasks such as relatedness, clustering, paraphrase detection and text classification.



Variant: **Encoder**

This is the component that encodes a sentence into fixed-length 512-dimension embedding. In the paper, there are two architectures proposed based on trade-offs in accuracy vs inference speed. Variant 1: Transformer Encoder In this variant, we use the encoder part of the original transformer architecture. The architecture consists of 6 stacked transformer layers. Each layer has a self-attention module followed by a feed-forward network. The self-attention process takes word order and surrounding context into account when generating each word representation. The output context-aware word embeddings are added element-wise and divided by the square root of the length of the sentence to account for the sentence-length difference. We get a 512-dimensional vector as output sentence embedding. This encoder has better accuracy on downstream tasks but higher memory and compute resource usage due to complex architecture. Also, the compute time scales dramatically with the length of sentence as self-attention has O(n2) time complexity with the length of the sentence. But for short sentences, it is only moderately slower.

Variant: **Deep Averaging Network (DAN)**

In this simpler variant, the encoder is based on the architecture. First, the embeddings for word and bi-grams present in a sentence are averaged together. Then, they are passed through 4-layer feed-forward deep DNN to get 512-dimensional sentence embedding as output. The embeddings for word and bi-grams are learned during training. It has slightly reduced accuracy compared to

the transformer variant, but the inference time is very efficient. Since we are only doing feedforward operations, the compute time is of linear complexity in terms of length of the input sequence.

Variant: **Multi-task Learning**

To learn the sentence embeddings, the encoder is shared and trained across a range of unsupervised tasks along with supervised training on the SNLI corpus.

7. RESULTS:

According to our requirement we have taken rating and review text columns which are of 100000 rows*2 columns

```
df_reviews=df_reviews.drop(columns=['user_id', 'book_id','review_id' ,'date_added', 'date_updated', 'read_at', 'started_at','n_votes','n_comments'], axis=0)
df_reviews
```

	rating	review_text
0	5	This is a special book. It started slow for ab...
1	3	Recommended by Don Katz. Avail for free in Dec...
2	3	A fun, fast paced science fiction thriller. I ...
3	0	Recommended reading to understand what is goin...
4	4	I really enjoyed this book, and there is a lot...
...	...	...
899995	3	3.5 stars. \n Jenna is a popular YA author and...
899996	3	This was a quick read for me. I have read a lo...
899997	4	** spoiler alert ** \n 3.5 stars. \n This book...
899998	4	** spoiler alert ** \n Another fun read from M...
899999	3	** spoiler alert ** \n 3.5 stars \n I liked it...

900000 rows × 2 columns

We have taken only 100000 rows * 2 columns in the project

```
]  1  df_reviews.drop(df_reviews.index[100000:900000], inplace=True)
   2  df_reviews
   3
```

	rating	review_text
0	5	This is a special book. It started slow for ab...
1	3	Recommended by Don Katz. Avail for free in Dec...
2	3	A fun, fast paced science fiction thriller. I ...
3	0	Recommended reading to understand what is goin...
4	4	I really enjoyed this book, and there is a lot...
...	...	...
99995	2	Underwhelming as hell. The characters were so ...
99996	4	This was so satisfying ugghhhh \n the characte...
99997	4	This was beautiful. Holy shit. I am absolute S...
99998	3	I thought this was cute and, while it was love...
99999	0	I just really can't be bothered with this anym...

100000 rows × 2 columns

Accuracy

We have gone through nearly 8 different models of lstm and we have got 65% accuracy as our final model.

```
Epoch 12/30
4218/4218 [==============================] - 68s 16ms/step - loss: 0.8269 - accuracy: 0.6361 - val_loss: 0.8673 - val_accuracy: 0.6236
Epoch 13/30
4218/4218 [==============================] - 64s 15ms/step - loss: 0.8233 - accuracy: 0.6399 - val_loss: 0.8589 - val_accuracy: 0.6251
Epoch 14/30
4218/4218 [==============================] - 65s 15ms/step - loss: 0.8216 - accuracy: 0.6382 - val_loss: 0.8632 - val_accuracy: 0.6196
Epoch 15/30
4218/4218 [==============================] - 68s 16ms/step - loss: 0.8190 - accuracy: 0.6404 - val_loss: 0.8471 - val_accuracy: 0.6307
Epoch 16/30
4218/4218 [==============================] - 64s 15ms/step - loss: 0.8170 - accuracy: 0.6435 - val_loss: 0.8460 - val_accuracy: 0.6298
Epoch 17/30
4218/4218 [==============================] - 65s 15ms/step - loss: 0.8141 - accuracy: 0.6423 - val_loss: 0.8529 - val_accuracy: 0.6283
Epoch 18/30
4218/4218 [==============================] - 69s 16ms/step - loss: 0.8127 - accuracy: 0.6435 - val_loss: 0.8460 - val_accuracy: 0.6330
Epoch 19/30
4218/4218 [==============================] - 65s 15ms/step - loss: 0.8083 - accuracy: 0.6455 - val_loss: 0.8454 - val_accuracy: 0.6245
Epoch 20/30
4218/4218 [==============================] - 64s 15ms/step - loss: 0.8069 - accuracy: 0.6460 - val_loss: 0.8456 - val_accuracy: 0.6311
Epoch 21/30
4218/4218 [==============================] - 70s 17ms/step - loss: 0.8052 - accuracy: 0.6461 - val_loss: 0.8435 - val_accuracy: 0.6300
Epoch 22/30
4218/4218 [==============================] - 65s 15ms/step - loss: 0.8017 - accuracy: 0.6471 - val_loss: 0.8443 - val_accuracy: 0.6279
Epoch 23/30
4218/4218 [==============================] - 66s 16ms/step - loss: 0.8004 - accuracy: 0.6490 - val_loss: 0.8412 - val_accuracy: 0.6307
Epoch 24/30
4218/4218 [==============================] - 69s 16ms/step - loss: 0.7977 - accuracy: 0.6504 - val_loss: 0.8387 - val_accuracy: 0.6303
Epoch 25/30
4218/4218 [==============================] - 65s 15ms/step - loss: 0.7949 - accuracy: 0.6518 - val_loss: 0.8458 - val_accuracy: 0.6311
Epoch 26/30
4218/4218 [==============================] - 69s 16ms/step - loss: 0.7920 - accuracy: 0.6518 - val_loss: 0.8407 - val_accuracy: 0.6303
Epoch 27/30
4218/4218 [==============================] - 66s 16ms/step - loss: 0.7896 - accuracy: 0.6517 - val_loss: 0.8442 - val_accuracy: 0.6266
Epoch 28/30
4218/4218 [==============================] - 66s 16ms/step - loss: 0.7864 - accuracy: 0.6535 - val_loss: 0.8468 - val_accuracy: 0.6283
Epoch 29/30
4218/4218 [==============================] - 69s 16ms/step - loss: 0.7842 - accuracy: 0.6551 - val_loss: 0.8383 - val_accuracy: 0.6345
Epoch 30/30
4218/4218 [==============================] - 65s 15ms/step - loss: 0.7831 - accuracy: 0.6557 - val_loss: 0.8408 - val_accuracy: 0.6273
CPU times: user 44min 38s, sys: 5min 39s, total: 50min 18s
Wall time: 33min 38s
```

8. CONCLUSION:

In this project, we have tried to detect the Ratings on commercial websites on a

scale of 1 to 5 based on the reviews given by the users. We made use of natural language processing to do so. We have used LSTM and universal sequence encoder to generate the rating based on the text. We got the accuracy of 50 by using the above mentioned models.As with any project, there is room for improvement here. The very nature of this project allows for multiple algorithms to be integrated together as modules and their results can be combined to increase the accuracy of the result. This model can further be improved with the addition of more algorithms into it. However, the output of these algorithms needs to be in the same format as the others. Once that condition is satisfied, the modules are easy to add as done in the code. This provides a great degree of modularity and versatility to the project

9. REFERENCES:

- The Role of Customer Product Reviews - eMarketer, Emarketer.com, 2017.

[Online]. Available:https://www.emarketer.com/Article/Role-of-Customer-Product Reviews/1008019. [Accessed: 06- Apr- 2017].

- Gesenhues, A. (2017). Survey: 90% Of Customers Say Buying Decisions Are Influenced By Online Reviews, Marketing Land, 2017. [Online]. Available: http:// marketingland.com/survey-customers-more-frustrated-by how-long-it-takes-to-resolve-a-customer-service-issue than-the-resolution-38756. [Accessed: 06- Apr-2017].
- Web Users Put More Stock in Consumer Reviews - eMarketer, Emarketer.com, 2017. [Online]. Available: https://www.emarketer.com/Article/Web-Users-Put-More Stock-Consumer-Reviews/1012929. [Accessed: 06- Apr2017].
- Internet Users Rely on Reviews When Deciding Which Products to Purchase - eMarketer,Emarketer.com,2017.[Online].Available: https://www.emarketer.com/Article/Internet-Users-Rely-on-Reviews-Deciding-Which-Products-Purchase/1014465. [Accessed: 06- Apr- 2017].
- Local Consumer Review Survey 2016 | The Impact Of Online Reviews, BrightLocal, 2017. [Online]. Available: https://www.brightlocal.com/learn/local-consumer-review survey/. [Accessed: 06- Apr- 2017].
- Liu, Bing. (2012). Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies, 5(1) 1–167.
- Liu, Bing. (2015). Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press.
- Schouten, K., Frasincar, F. (2016). Survey on Aspect-Level Sentiment Analysis, IEEE Transactions on Knowledge and Data Engineering, 28 (3) 813-830.
- Che, W., Zhao, Y., Guo, H., Su, Z., Liu, T. (2015). Sentence Compression for Aspect -Based Sentiment Analysis, IEEE/ACM Transactions on Audio, Speech, and Language Processing, 23 (12) 2111-2124.
- Sarawgi, K., Pathak, V. (2017). Opinion Mining: Aspect Level Sentiment Analysis using SentiWordNet and Amazon Web Services, International Journal of Computer Applications, 158 (6) 31-36.
- Alhojely, S. (2016). “Different Applications and Techniques for Sentiment Analysis, International Journal of Computer Applications, 154 (5) 24-28. [12] Altwairesh, N. S. (2016). Sentiment Analysis of Twitter: A Study on the Saudi Community,” Ph.D dissertation, KSU Univ., Riyadh, 2016