Goodreads Book Review Using NLP

[1]Sri Ramoju Ravi Teja,[2] Padi Ujjvala Sindhu,[3]Terala Sushanth

[1]Depatment of Electronic & Communication Engineering, SR University, Warangal, Telangana, India

[2]Depatment of Computer Science and Engineering, SR University, Warangal, Telangana, India

[3]Depatment of Computer Science and Engineering, SR University, Warangal, Telangana, India

Email: ravitejaramoju1432@gmail.com.

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.

Keywords: Goodreads book review**,** Lstm, universal sentence encoder, Pre-Processing

1.1 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.

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.

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 SURVEY:

- 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. [1] 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. [2] 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. [3] 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. [4] 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 [5] 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 [6] 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.[7] 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.,[8] 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 of 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 sentiment analysis on product reviews	4th International Conference on Advanced Computing and Communication Systems (ICACCS), 2017, IEEE.	The approach serves as a better alternative to rate a product based on its technical specification by analyzing large number of user reviews.	It is very difficult to read through each individual Review of various items and make a good decision for 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	AbinashTripathy 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. METHODOLOGY:

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

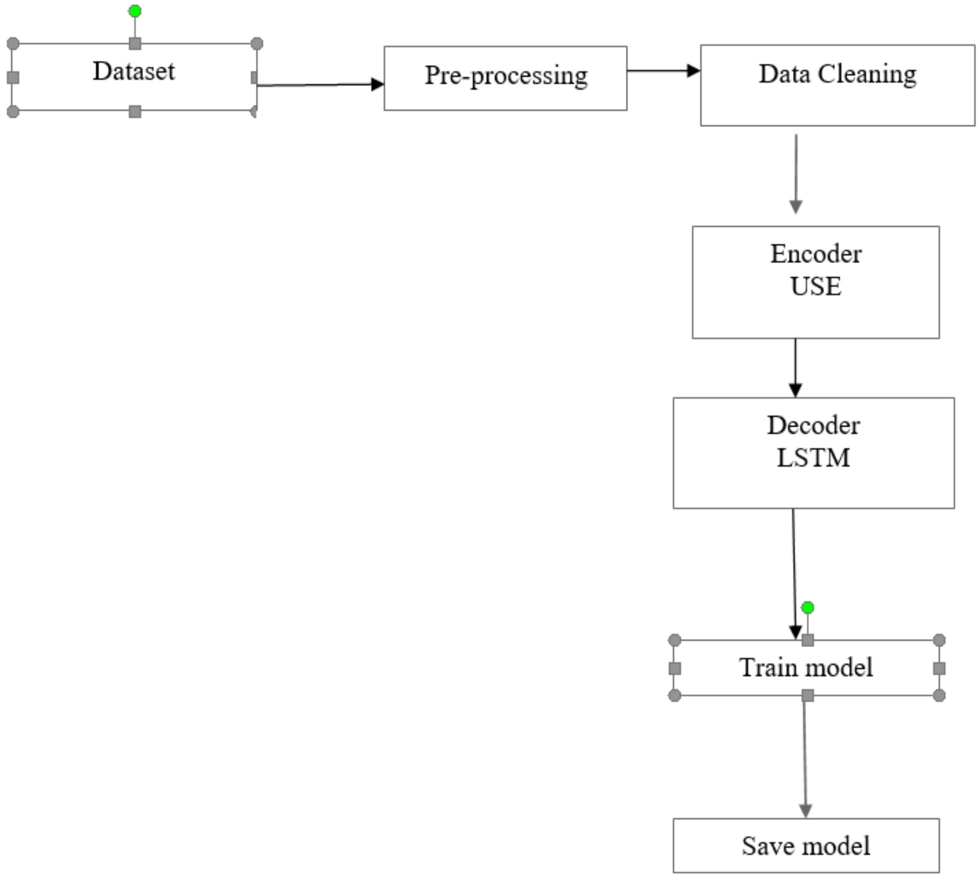

Figure 1- Flow chart of the technique.

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

	A	B	C	D	E	F	G	H	I	J	K
1	user_id	book_id	review_id	rating	review_te:	date_adde	date_upda	read_at	started_at	n_votes	n_comments
2	8842281e:	18245960	dfdbb7b0e	5	This is a	Sun Jul 30	Wed Aug 3	Sat Aug 26	Tue Aug 15	28	1
3	8842281e:	16981	a5d2c3628	3	Recomme	Mon Dec (	Wed Mar 22 11:37:04 -0700 201			1	0
4	8842281e:	28684704	2ede853b:	3	A fun,	Tue Nov 1:	Mon Mar :	Sat Mar 18	Fri Mar 17	22	0
5	8842281e:	27161156	ced5675e5	0	Recomme	Wed Nov (	Wed Nov 09 17:38:20 -0800 201			5	1
6	8842281e:	25884323	33273272!	4	I really	Mon Apr 2	Mon Apr 2	Sun Jun 26	Sat May 28	9	1
7	8842281e:	19398490	ea4a220b:	4	A	Sun Jan 03	Tue Sep 2(	Tue Sep 13	Sat Aug 20	35	5
8	8842281e:	22318578	5fe9882bf	5	5 stars for	Sun Jun 07	Wed Mar :	Sun Aug 09	Sun Jun 07	24	3
9	8842281e:	24189224	dbc01e243	0	Numerous	Fri May 29	Fri May 29 17:49:40 -0700 2015			11	5
10	8842281e:	22551730	75471007(	4	Another	Thu May 0	Wed Dec 1	Sat Jul 11 (	Wed Jun 2	20	6
11	8842281e:	22816087	d11954e6e	5	I love	Wed Apr 2	Wed Dec 1	Wed Nov (	Sun Sep 27	40	6
12	8842281e:	5577844	52c8ac494	5	A	Wed Sep 2	Wed Oct 0	Tue Sep 3(	Sun Sep 21	5	1
13	8842281e:	21792828	9bdae13d(	5	I couldn't	Mon Sep 2	Wed Mar :	Thu Nov 2(	Mon Nov :	45	4
14	8842281e:	17855756	9db60af73	4	What a	Sat Jul 26 :	Sat Jul 26 :	Sat Jul 26 13:10:06 -0		3	1
15	8842281e:	16158596	6ff8bbc48!	0	Recomme	Mon Jul 07	Mon Jul 07 10:56:39 -0700 2014			0	0
16	8842281e:	12961964	04659501I	4	A classic	Thu May 2	Sun Apr 23	Sat Apr 15	Tue Mar 2:	60	2
17	8842281e:	17315048	885c772fb	5	Mark	Sat Apr 05	Wed Mar :	Mon Aug 2	Sat Aug 16	25	5
18	8842281e:	17860739	022bb6daf	4	An	Wed Mar :	Tue Sep 23	Sun Sep 21	Sat Jul 26 (	7	0
19	8842281e:	5064	da4757c5(	5	I tore	Sun Mar 0:	Tue May 0	Sun Apr 27	Thu Apr 17	2	2
20	8842281e:	18659415	4e652faa6	4	The Circle	Wed Nov :	Fri Jan 10 :	Sun Dec 22	Wed Dec 1	30	1
21	8842281e:	13526165	51fe3e46c	5	My wife	Thu May 3	Wed Mar :	Wed Mar :	Tue Mar 1(	23	5
22	8842281e:	13453029	46a6e1a14	4	A fun fast	Tue Dec 04	Sat Jul 26 :	Tue Jul 08	Wed Jul 02	5	1
23	8842281e:	12953520	8666dfd55	4	A	Tue Jul 10	Wed Sep 1	Mon Sep 1	Tue Sep 04	1	0
24	8842281e:	13239822	a582bfa8e	3	This book	Mon Jul 02	Wed Mar :	Wed Aug 1	Sun Aug 12	7	0
25	8842281e:	9850443	9f35b2534	3	A fun,	Tue Jun 05	Sun Aug 05	Sat Aug 04	Fri Jul 27 0	2	1

Figure.2 – Datset Insights

5 DATA PRE-PROCESSING:

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 MODEL

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.

LSTM Architecture:

Long Short- Term Memory (LSTM) networks are a modified version of recurrent neural networks, which makes it easier to remember past data in memory.

ARCHITECTURE OF LSTM:

1. Forget Gate
2. Input Gate
3. Output Gate

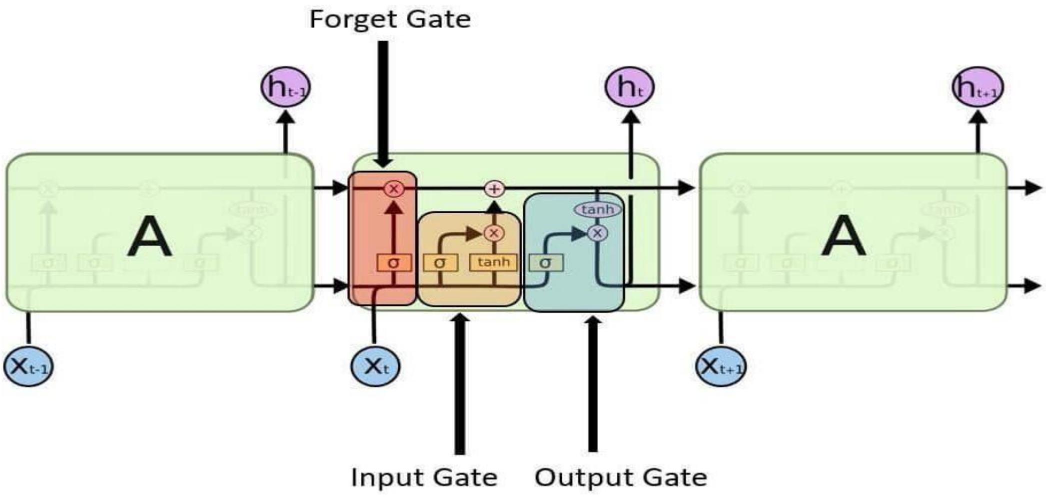

Figure 3- Architecture of LSTM

1. **Input gate-** It discover which value from input should be used to modify the memory. **Sigmoid** function decides which values to let through 0 or 1. And **tanh** function gives weightage to the values which are passed, deciding their level of importance ranging from **-1** to **1**.

$$i_t = \sigma(W_i . [h_t - 1, x_t] + bi)$$
$$C_t = \tanh\,(W_C . [h_t - 1, x_t] + b_C)$$

2. **Forget gate-** It discover the details to be discarded from the block. A sigmoid function decides it. It looks at the previous state **(ht-1)** and the content input (Xt) and outputs a number between 0(omit this) and 1(keep this) for each number in the cell state **Ct-1**.

$$f_t = \sigma(W_f . [h_t - 1, x_t] + b_f)$$

3. **Output gate-** The input and the memory of the block are used to decide the output. Sigmoid function decides which values to let through 0 or 1. And tanh function decides

which values to let through 0, 1. And tanh function gives weightage to the values which are passed, deciding their level of importance ranging from -1 to 1 and multiplied with an output of **s**igmoid.

$$O_t = \sigma(W_o[h_t - 1, x_t] + b_o$$
$$h_t = o_t * \tanh(C_t)$$

The words are inputted into an embedding lookup. In most cases, when working with a corpus of text data, the size of the vocabulary is unusually large.

This is a multidimensional, distributed representation of words in a vector space. These embeddings can be learned using other deep learning techniques like **GloVe**, we can train the model in an end-to-end fashion to determine the embedding as we teach.

These embeddings are then inputted into our **LSTM layer**, where the output is fed to a sigmoid output layer and the **LSTM cell** for the next word in our sequence.

LSTM LAYER:

We used a function to build the LSTM layers to handle the number of layers and sizes dynamically. The service will take a list of LSTM sizes, which can indicate the number of LSTM layers based on the list's length indicating a two-layered LSTM network. In our model we used a list of length 2, containing the sizes 32 and 16, indicating a two-layered LSTM network where the first layer size 32 and the second layer has hidden layer size 16.

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.

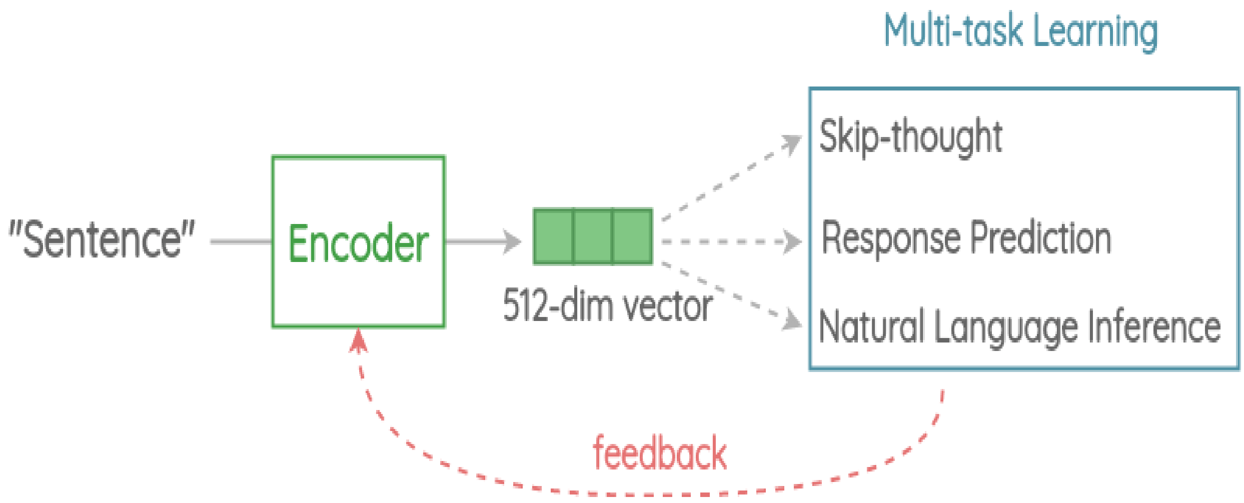

Figure 4- USE Model operation.

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

Figure 5- data set visualization.

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

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

Figure 6- data set reduction

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
```

Figure 7 – Testing and Training Accuracy

We have got 65% accuracy for our model by training with 30 epochs.

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 65 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:

[1] 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].

[2] 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].

[3] 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].

[4] 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].

[5] 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].

[6] Liu, Bing. (2012). Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies, 5(1) 1–167.

[7] Liu, Bing. (2015). Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press.

[8] Schouten, K., Frasincar, F. (2016). Survey on Aspect-Level Sentiment Analysis, IEEE Transactions on Knowledge and Data Engineering, 28 (3) 813-830.

[9] 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.

[10] 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.

[11] 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