Text Summarizer

A Project Report in partial fulfillment of the degree

Bachelor of Technology

in

Computer Science and Engineering

By

19K41A0518	N. Vinusha
19K41A0520	P. Shiva Sowmya
19K41A0577	N. Anasri

Under the Guidance of
D. Ramesh
Assistant Professor School of CS&AI

Submitted to

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
S.R.ENGINEERING COLLEGE(A), ANANTHASAGAR, WARANGAL
(Affiliated to JNTUH, Accredited by NBA) Dec-2021.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Project Report entitled "Text Summarizer" is a record of bonafide work carried out by the student(s) N. Vinusha, P. shiva Sowmya, N. Anasri, bearing Roll No(s) 19K41A0518,19K41A0520,19K41A0577 during the academic year 2022-23 in partial fulfillment of the award of the degree of ***Bachelor of Technology*** in **Computer Science & Engineering** by the Jawaharlal Nehru Technological University, Hyderabad.

.

Supervisor **Head of the Department**

External Examiner

Abstract

Text Summarization is the most popular problem in this modern era. Text summarization is the process of generating short, fluent, and most importantly accurate summary of a respectively longer text document. In this approach we build model which will reduce the text size and create a summary of our text data.The main idea behind automatic text summarization is to be able to find a short subset of the most essential information from the entire set and present it in a human-readable format. It is very useful because more useful information can be read in a short time. Where human eye can miss out on crucial details that can be found in your text but our software do not miss it. It is a time Saving Process. To build this model, we have used seq2seq modeling for encoding and decoding the text and Long Short time memory(LSTM) for building the model. To train the model, we used the amazon food reviews dataset which is collected from Kaggle website. In our project we got an accuracy of 84.86% for the 1,00,000 records we have taken.

Table

1. INTRODUCTION

Text summarization history is one of the where is there is always a fight for its development. The origins of it start from 1940 at the time of world war 2 to convert russsian language into English. Though development in this area has been in code and syntax. The main hype text summarization was got in 1990’s .In those days they used to use rule based algorithms. Which used to say the importance and ranking of text .so it developed a along and in 2016 they developed a model using seq2seq.

Along with the growth of the internet and big data, making people overwhelmed by the large information and documents on the internet. This triggers the desire of many researchers to develop a technological approach that can summarize texts automatically. The area of text summarization research has been studied since the mid-20th century, which was first discussed openly by Lun (1958) with a statistical technique namely word frequency diagrams. Many different approaches have been created to date. Based on the number of the document, there is single and multi-document summarization. Meanwhile, based on the summary results there are the extractive and abstractive results.

Text summarization is the process of generating short, fluent, and most importantly accurate summary of a respectively longer text document. The main idea behind automatic text summarization is to be able to find a short subset of the most essential information from the entire set and present it in a human-readable format. As online textual data grows, automatic text summarization methods have potential to be very helpful because more useful information can be read in a short time. Automatic text summarization is used in many domains. One of the most common uses of Text Summarization is in the news field. For example, every day we read many news articles. While reading an article, generally, we come across many details that may not be needed in the article or we may across such details that are not so important for us to know. We would like to know the crux of the news article. In such a case, text summarisation techniques help us to condense the news article size into small size by giving us the crux of the news article.

There are mainly two methods for summarizing the text document that can be done by using extractive and abstractive techniques. Extractive summaries concentrate on selecting important passages, sentences, words, etc. from the primary text and connecting them into a concise form. The importance of critical sentences is concluded on the basis of analytical and semantic features of the sentences

2. LITERATURE SURVEY

1. Chin-Yew introduced Recall Oriented Understudy for Evaluation ROUGE. That is an automatic evaluation package for text summarization. The paper also introduced four different measures of ROUGE: - ROUGE-N, ROUGE-L, ROUGE-W and ROUGE-S. It measures the quality of summary by comparing the generated
summary with other ideal summaries that are created by humans. These methods are efficient for automatic evaluation of single document summary as well as multi-document summaries.

2. k shil Kumar has analyzed and compared the performance of three different algorithms. Firstly, the different text summarization techniques explained. Extraction based techniques are used to extract important keywords to be included in the summary. For comparison three comparison three keyword extraction algorithms namely Text Rank, Lex Rank, Latent Semantic Analysis (LSA) were used. Three algorithms are explained and implemented in python language. The ROUGE 1 is used to evaluate the effectiveness of the extracted keywords. The results of the algorithms compared with the handwritten summaries and evaluate the performance. In the end, the TextRank Algorithm gives abetter result than other two algorithms.

3. Pankaj Gupta has reviewed different techniques of Sentiment analysis and different techniques of text summarization. Sentiment analysis isa machine learning approach in which machine learns and analyze the sentiments, emotions present in the text. The machine learning methods like Naive Bayes Classifier and Support Machine Vectors (SVM) are used. these methods are used to determine the emotions and sentiments in the text data like reviews about movies or products. In Text summarization, uses the natural language processing (NPL) and linguistic features of sentences are used for checking the importance of
the words and sentences that can be included in the final summary. In this paper, a survey has been done of previous research work related to text summarization and Sentiment analysis, so that new research area can be explored by considering the merits and demerits of the current techniques and strategies.

4. Harsha Dave author has proposed a system to generate the abstractive summary from the extractive summary using Word Net ontology. The multiple documents had been used like text, pdf, word files etc. The author has discussed various text summarization techniques then

author discussed step by step the multiple document text summarization approaches. The experiment result is compared with the existing online extractive tools as well as with human generated summaries and shows the proposed system gives good results. At last the author proposed for the future that the *Literature Review on* Automatic Text Summarization

5. Yihong Gong, the author proposes two methods that create the generic text summaries by ranking and extracting sentences from the main text documents. The first method uses information retrieval (IR) methods that rankthe sentence relevance and provides the relevance scores to sentences and the second method uses the latent semantic analysis (LSA) technique that based on latent semantic indexing (LSI) in order to identify the semantic importance of the sentences, for summary creations. The author uses the Singular Value Decomposition (SVD) to generate the text summary. Further, this paper author explains the SVD based methods step by step. The effect of different Weighted Schemes is also checked on the performance of the summaries. The purposed methods provide generic abstractive summaries. Finally, the results are compared with the human-generated summaries. It generates better human like abstractive summaries. For future author proposed to investigate various machine learning techniques so that quality of generic text summarization can be improved.

6. Rada Mihalcea, the author introduced the Text Rank a graph-based ranking model for the processing of the text. it is an unsupervised method for keyword and sentence extraction. Text Rank uses voting based weighting mechanism and provides the score to the sentence then finally determine the importance of the sentence. The nodes in the graph represent the sentences. The significance of the sentence based on incoming and outgoing edges from nodes. The weight of each is determined based on similarity score between the sentences. TextRank derived from the Google's Page Rank algorithm. TextRank provides extractive summaries of the text. Text Rank Provides the best results.

7. Güneş Erkan , the author introduces graph-based method Lex Rank. In this, the sentence score is calculated based on Eigenvector Centrality. It is cosine transform weighting method. In this, the original text is split into sentences and a graph is built where sentences act as the nodes. The complete method is explained in the paper. The results show that Lex Rank outperforms the existing centroid based methods. This method is also performed well in case of noisy data. This method generates an extractive summary of the text.

8. Kavita , the author proposed graph-based text summarization framework Opinosis. It generates abstractive summaries. Opinosis works on redundant data like human reviews on movies or products and provides abstractive summaries. Firstly, it creates the direct Opinosis-Graph of the text. Where nodes represent the word units of the text. Three unique graph properties: Redundancy capture, Collapsible structures and Gapped subsequence capture is used to explore and explore different sub-paths that help in the creation of abstractive summaries of the text. The valid path is selected and marked with high redundancy score, collapsed path and summary generation. Then all paths ranked in descending order according to scores. The duplicate paths are removed using Jaccard measure the results are compared with human summaries. Results show Opinosis summaries has better agreement with human summaries. For future work author proposed to use a similar idea to overlay parse trees.

9. Dharmendra Hinhu , the author uses the extractive text summarization. The author gives the Wikipedia Articles as input to the system and identifies text scoring. Firstly, the sentences are Tokenized through pattern matching using regular expressions. Then we get data in form of set of words then stop words are removed from the set of words. The words are then stemmed. Then traditional methods are used for scoring of the sentences. Scoring helps in classifying the sentences if they included in summary or not. It is found that scoring sentences based on citation give better results.

10. Tacho Jo ,the author proposed a particular version of KNN (K Nearest Neighbor) where the words are assumed as features of numerical vectors represents text. The similarity between feature vectors is computed by considering the similarity among attributes as well as among values. Text summarization viewed as the task of classification. The text is partitioned into paragraphs or sentences. Each paragraph or sentence is classified into ‘summary or ‘non-summary’ by the classifier. The sentences which are classified into ‘summary’ are extracted as results from summarizing the text and other text rejected. Improved results are obtained with the proposed version of KNN in text classification and clustering. The modified version of KNN leads to a more compact representation of data item and better performance.

Author	Processing steps	Approach	Evaluation Measure	Result
A. Kukkar	→Collection of elements →feature Extraction →summary generation	Partial swarm optimisation	Rouge-1 Rouge-2 Rouge-3	79.14 75.83 75.15
Beibei Haui	Sentence intentions to generate summary	Intension based bug report summarization technique	Precision recall F-score Pyramid Precison	5% 3% 3% 5%
Angel Hernandez Castaned	Feature Extraction	Tf-idf,D2V,LDE,OHE	F-score	0.249
Shubra Goyal Jindal	→Preprocessing →keyword and sentence identification →clustering →rule designing →clustering	Tf-idf,fuzzy –c means, heirachial clustering	F-score Precision recall	80.1% 78.22% 82.12%
Ashima Kukkar	Feature Extraction and Sentence score calculation	PSO and Colony Optimisation	Precision Recall f-score	79.74% 78.79% 79.76%
Cheng Gen Yang	→Extracting important informationusing anthropogeniuc and procedural classes	TSM uses ENR and BRC textual model	f-score	0.530
Akalanka Galapathi	Selecting highest probability sentences	Logistic Regression Model	Precision Recall F-score	44% 24% 29%
Surbhi Bhatia	Ranking sentences based on most relevancy	Senti WordNet, Principle Component Analysis	Precision Recall f-score	0.49 0.27 0.35
Beomjum	→Analysing duplicate →graph generation →Sentence score	Relation based bug report summarizer	Precision Recall f-score	51.69% 51.82% 51.76%
Xiaochen Li	Vector Generation and Tokenisation	Deep sum a stepped auto encoder network	F1-score R1	0.462 0.563

3.DESIGN

3.1 REQUIREMENT SPECIFICATION(S/W & H/W)

Hardware Requirements

- ✓ **System** : Pentium 4, Intel Core i3, i5, i7 and 2GHz Minimum
- ✓ **RAM** : 4GB or above
- ✓ **Hard Disk** : 10GB or above
- ✓ **Input** : Keyboard and Mouse
- ✓ **Output** : Monitor or PC

Software Requirements

- ✓ **OS** : Windows 8 or Higher Versions
- ✓ **Platform** : Kaggle Notebook
- ✓ **Program Language** : Python

3.2 FLOWCHART(METHODOLOGY)

The whole approach is depicted by the following flowchart

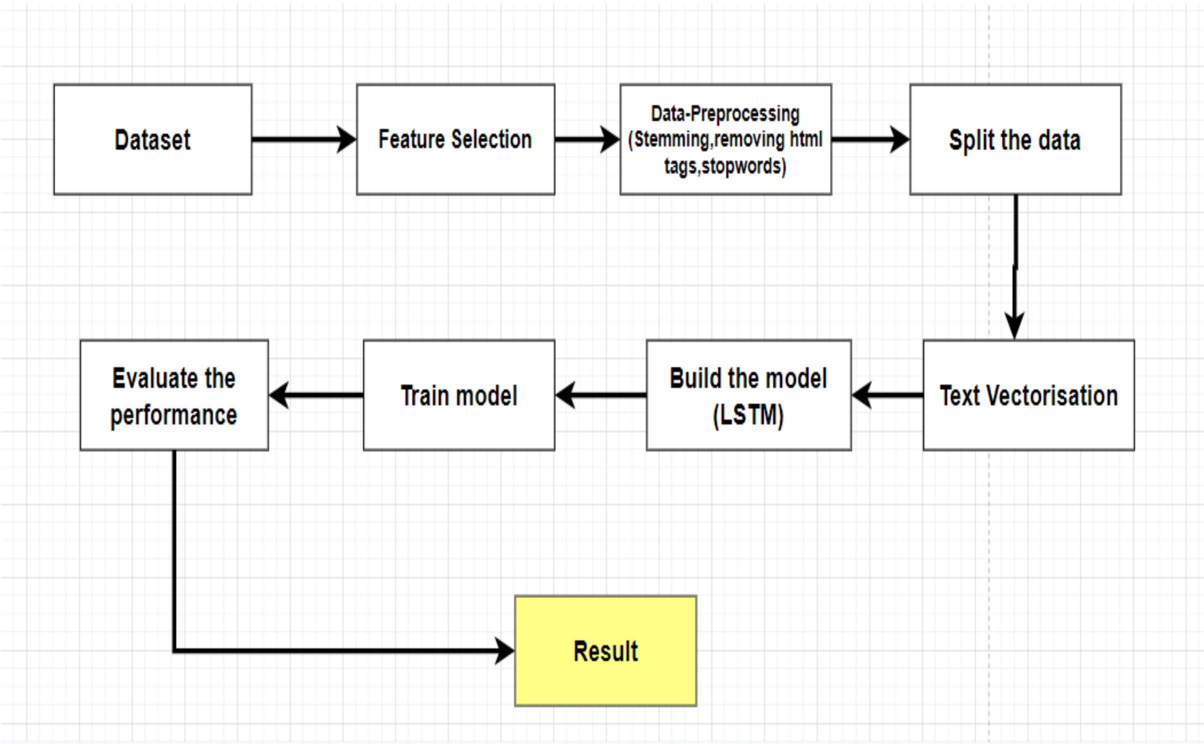

Figure 1: Flow chart of the technique

4.DATASET:

The dataset is downloaded from kaggle which was containing all the data regarding customer reviews on food .This data was extracted from the amazon. This dataset contains 5,68,454 customer reviews and 10 columns like text, summary, product id, use rid, name etc. But we are taking 1,00,000 rows from dataset for training and testing the model. The columns text and summary is used in building the model.

Id	ProductId	UserId	ProfileNar	Helpfulne:	Helpfulne:	Score	Time
1	B001E4KF(	A3SGXH7A	delmartian	1	1	5	1.3E+09
2	B00813GR	A1D87F6Z	dll pa	0	0	1	1.3E+09
3	B000LQOC	ABXLMWJI	Natalia Coı	1	1	4	1.2E+09
4	B000UA0C	A395BORC	Karl	3	3	2	1.3E+09
5	B006K2ZZ	A1UQRSCL	Michael D.	0	0	5	1.4E+09
6	B006K2ZZ	ADT0SRK1	Twoapenn	0	0	4	1.3E+09
7	B006K2ZZ	A1SP2KVK	David C. Sı	0	0	5	1.3E+09
8	B006K2ZZ	A3JRGQVE	Pamela G.	0	0	5	1.3E+09
9	B000E7L2I	A1MZYO9	R. James	1	1	5	1.3E+09

Summary	Text
Good Qual	I have bought several of the Vitality canned dog food products a
Not as Adv	Product arrived labeled as Jumbo Salted Peanuts...the peanuts
"Delight" s	This is a confection that has been around a few centuries. It is a
Cough Me	If you are looking for the secret ingredient in Robitussin I believe
Great taffy	Great taffy at a great price. There was a wide assortment of yu
Nice Taffy	I got a wild hair for taffy and ordered this five pound bag. The ta
Great! Jus	This saltwater taffy had great flavors and was very soft and chev

Figure 2: Sample data of Amazon fine food reviews dataset.

5. DATA PREPROCESSING

Data pre-processing is a technique that is used to convert raw data into a clean dataset. The data is gathered from Kaggle which is in raw format (i.e., unbalanced data and un required columnspresent in it) which is not feasible for the computer to predict the summary of the given text. Pre-processing for this text data is determined below:

5.1 Removing Unnecessary Columns:

The columns like time, profile name, product id, user id ,score is not necessary for building the model. So, we are deleting all the unnecessary columns and using only text, summary columns. Dataset after deleting all the un-necessary columns.

Summary	Text
Good Qual	I have bought several of the Vitality canned dog food products a
Not as Adv	Product arrived labeled as Jumbo Salted Peanuts...the peanuts v
"Delight" s	This is a confection that has been around a few centuries. It is a
Cough Me	If you are looking for the secret ingredient in Robitussin I believe
Great taffy	Great taffy at a great price. There was a wide assortment of yu
Nice Taffy	I got a wild hair for taffy and ordered this five pound bag. The ta
Great! Jus	This saltwater taffy had great flavors and was very soft and chev

Figure3: Dataset after feature selection

5.2 Removing Duplicate Data:

The Values in the dataset may be duplicate so, there is a need to check for data that is repeating and remove it so that it does not affect the model training and the accuracy that it is giving.

5.3 Removing Null Data:

The Values in the dataset may be Null so, there is a need to check for data that is null or not, so that it does not affect the model training and the accuracy that it is giving.

5.4 Removing Html tags:

The reviews contains some html tags in between which is not useful and it also affect the model which decreases the accuracy. BeautifulSoup is used for removing html tags.

5.5 Converting to Lower Case:

Converting the text present in the reviews column into lower case is necessary since itwill be helpful while converting the text data into vectors. These vectors will be later used fortraining the model and validating it using the validation data. All the text should be in same format for this process.

5.6 Removal of Stop Words:

Stop word removal is one of the most used preprocessing steps across different NLP applications. The idea is simply removing the words that occur commonly across all the documents in the corpus. Typically, articles and pronouns are generally classified as stop words. These wordshave no significance in some of the NLP tasks like information retrieval and classification, whichmeans these words are not very discriminative. On the contrary, in some NLP applications stop word removal will have very little impact.

5.7 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).

5.7 Padding

As we know all the neural networks needs to have the inputs that should be in similar shape and size. When we pre-process the texts and use the texts as an input for our Model. Where we know that we need to have the inputs with the same size, now here padding comes into picture. The inputs should be in same size at that time padding is necessary. The Padding method from Keras library is used to add zeroes to the formed vectors.

5.9 Text Embedding

Seq1Seq2 model :

Seq2Seq model is a model that takes a stream of sentences as an input and outputs another stream of sentences. This can be seen in neural machine Translation where input sentences is one language and output sentences are translated versions of that language. Encoder and Decoder are the two main techniques used in seq2seq modeling.

Encoder Model :

Encoder model is used to encode or transform the input sentences and generate feedback after every step. This feedback can be an internal state i.e, hidden state or cell state if we are using the LSTM layer. Encoder models capture the vital information from the input sentences while maintaining the context throughout. In Neural Machine translation, out input language will be passed into the encoder model where it will capture the contextual information without modifying the meaning of the input sequence. Outputs from the encoder model are then passed into the decoder model to get the output sequences.

Decoder Model :

The decoder model is used to decode or predict the target sentences word by word. Decoder input data takes the input of target sentences and predicts the next word which is then fed into the next layer for the prediction.<start> to start the sentence and <End> to end the target sentence are the two word that help the model to know what will be the initial variable to predict the next word and the ending variable to know the ending of the sentence. While training the model, we first provide the word<start>, the model then predicts the next word that is the decoder data target. This word is then fed as input data for the next timestep to get the next word prediction.

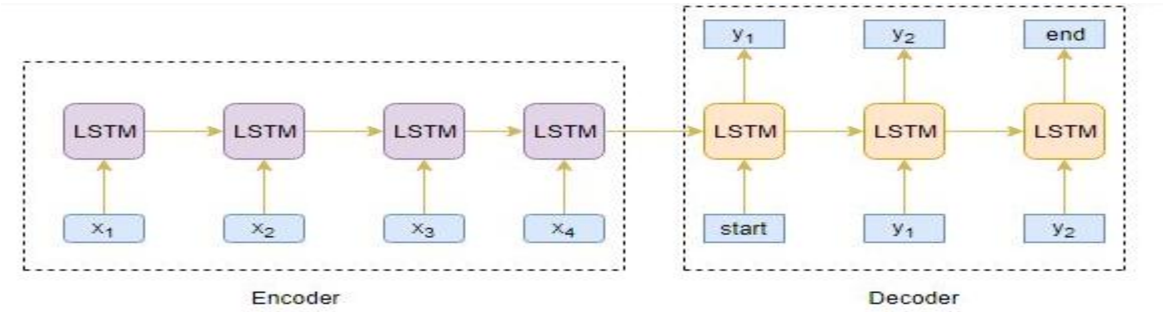

Figure 4: Architecture of Encoder-Decoder model

We can set up the Encoder-Decoder in 2 phases:

- Training phase
- Inference phase

Inference Phase

After training, the model is tested on new source sequences for which the target sequence is unknown. So, we need to set up the inference architecture to decode a test sequence.

Training phase

In the training phase, we will first set up the encoder and decoder. We will then train the model to predict the target sequence offset by one time step.

6. MODEL

We are using Stacked LSTM containing 3 layers of LSTM stacked on top of each other. This will make our prediction much better.

Encoder: We will initialize the encoder input tensor using the 'Input' object. The expected shape of the batch will be 74 (maximum input length)-dimensions. Then we will create an 'Embedding Layer' which will have the total number of input words as the first argument and a shape of 500 which is the latent (hidden) dimension.

LSTM: Now we will create 3 stacked LSTM layers where the first LSTM layer will have input of encoder and like that create a continuous sequence of LSTM layers.

The LSTM layer will capture all the contextual information present in the input sequence. We will return hidden state output and also states i.e. hidden state and cell state after execution of every LSTM layer.

Decoder: Like Encoder we will initialize the decoder input tensor and then pass it to the only LSTM. Here, the decoder will also have the initial state where we will pass the hidden state and cell state values that we have obtained from the encoder's LSTM layer.

Attention Layer: We will pass the encoder and decoder outputs into the attention layer and then we will concatenate attention layer outputs with the decoder outputs. Now we will create our Dense Layer that is the output layer for our model. It will have the shape of the total number of target words and a softmax activation function.

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 74)]	0	
embedding (Embedding)	(None, 74, 500)	16099500	input_1[0][0]
lstm (LSTM)	[(None, 74, 500), (N	2002000	embedding[0][0]
input_2 (InputLayer)	[(None, None)]	0	
lstm_1 (LSTM)	[(None, 74, 500), (N	2002000	lstm[0][0]
embedding_1 (Embedding)	(None, None, 500)	7085500	input_2[0][0]
lstm_2 (LSTM)	[(None, 74, 500), (N	2002000	lstm_1[0][0]
lstm_3 (LSTM)	[(None, None, 500),	2002000	embedding_1[0][0] lstm_2[0][1] lstm_2[0][2]
attention (Attention)	(None, None, 500)	0	lstm_3[0][0] lstm_2[0][0]
concat_layer1 (Concatenate)	(None, None, 1000)	0	lstm_3[0][0] attention[0][0]
dense (Dense)	(None, None, 14171)	14185171	concat_layer1[0][0]

Total params: 45,378,171
Trainable params: 45,378,171

Figure 5: Model Architecture

7. RESULTS

- **No. Of Epochs=10**

From split data 80% of training data, we made a machine model to predict. The machine model predicts the other 20 % of data as data testing to see how our model work. We use 3 stacked Long Short-Term Memory to make the machine model good enough to predict.

- **Accuracy:**

The results of this machine model are having 84.86% accuracy. It is explained from 20% data, that is classified for testing in which the training data is used to predict the machine correct around the trained model.

```
125/125 [==============================] - 58s 411ms/step - loss: 1.5360 - accuracy: 0.8109 - val_loss: 1.2848 - val_accuracy: 0.8338
Epoch 2/10
125/125 [==============================] - 50s 402ms/step - loss: 1.2589 - accuracy: 0.8334 - val_loss: 1.2287 - val_accuracy: 0.8359
Epoch 3/10
125/125 [==============================] - 50s 403ms/step - loss: 1.1963 - accuracy: 0.8355 - val_loss: 1.1883 - val_accuracy: 0.8378
Epoch 4/10
125/125 [==============================] - 50s 403ms/step - loss: 1.1460 - accuracy: 0.8373 - val_loss: 1.1571 - val_accuracy: 0.8387
Epoch 5/10
125/125 [==============================] - 50s 402ms/step - loss: 1.1026 - accuracy: 0.8391 - val_loss: 1.1368 - val_accuracy: 0.8397
Epoch 6/10
125/125 [==============================] - 50s 403ms/step - loss: 1.0616 - accuracy: 0.8411 - val_loss: 1.1247 - val_accuracy: 0.8411
Epoch 7/10
125/125 [==============================] - 50s 402ms/step - loss: 1.0226 - accuracy: 0.8433 - val_loss: 1.1087 - val_accuracy: 0.8414
Epoch 8/10
125/125 [==============================] - 50s 402ms/step - loss: 0.9871 - accuracy: 0.8450 - val_loss: 1.0944 - val_accuracy: 0.8422
Epoch 9/10
125/125 [==============================] - 50s 402ms/step - loss: 0.9529 - accuracy: 0.8467 - val_loss: 1.0946 - val_accuracy: 0.8432
Epoch 10/10
125/125 [==============================] - 50s 402ms/step - loss: 0.9202 - accuracy: 0.8486 - val_loss: 1.0911 - val_accuracy: 0.8421
```

Figure 6: accuracy scores

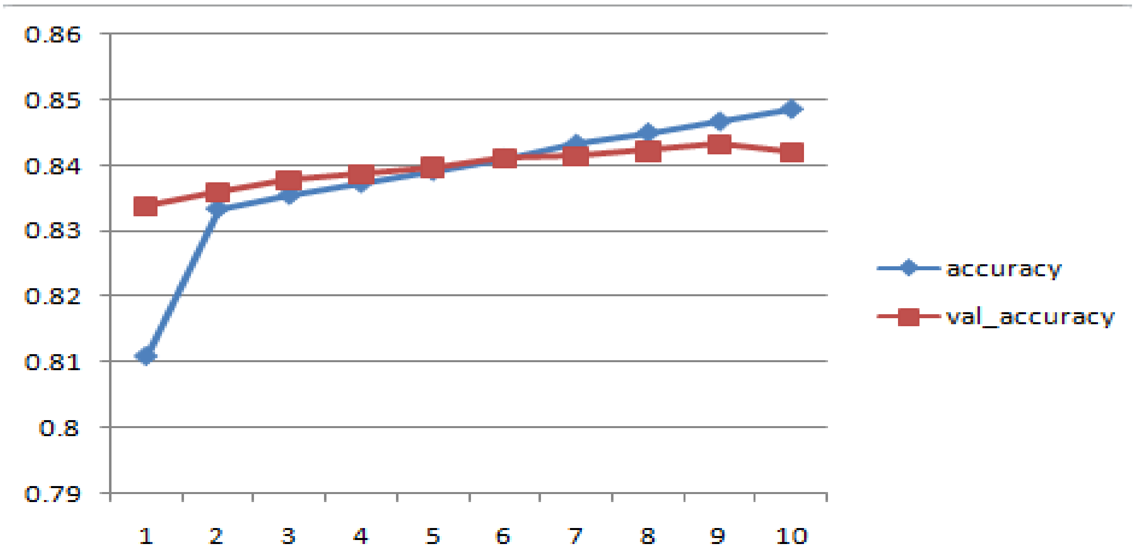

Figure7: accuracy Score Graph

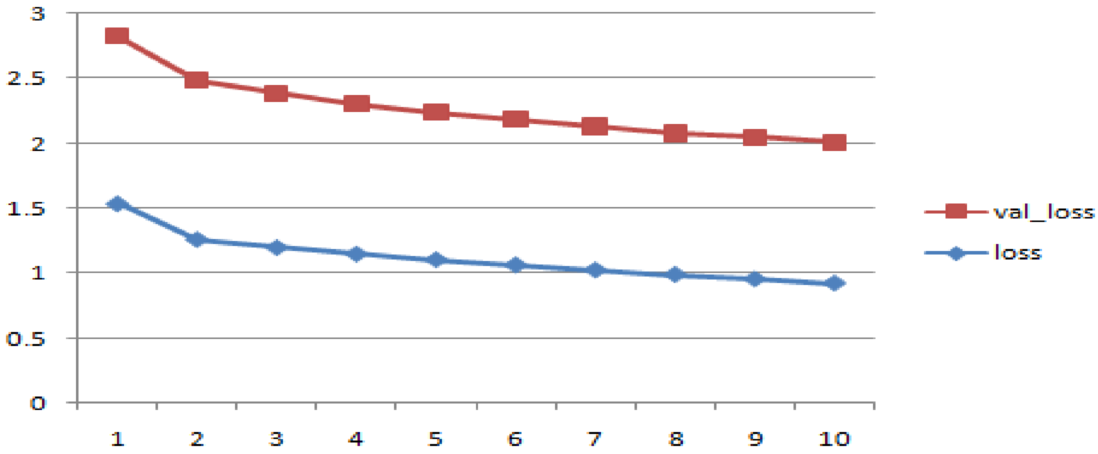

Figure8: Loss Value Graph

Testing the model

Output 1:

```
Enter :  McCann's Oatmeal is a good quality choice.  Our favorite is the Apples and Cinnamon, but we find that none of these are overly
sugary.  For a good hot breakfast in 2 minutes, this is excellent."
Review : McCann's Oatmeal is a good quality choice.  Our favorite is the Apples and Cinnamon, but we find that none of these are overly
sugary.  For a good hot breakfast in 2 minutes, this is excellent."

Predicted summary: good
```

Output 2:

```
inp_review = input("Enter : ")
print("Review :",inp_review)

inp_review = clean(inp_review,"inputs")
inp_review = ' '.join(inp_review)
inp_x= in_tokenizer.texts_to_sequences([inp_review])
inp_x= pad_sequences(inp_x,  maxlen=max_in_len, padding='post')

summary=decode_sequence(inp_x.reshape(1,max_in_len))
if 'eos' in summary :
  summary=summary.replace('eos','')
print("\nPredicted summary:",summary);print("\n")
```

```
Enter : it really crispy and the amount of chocolate filling is generous it at least times cheaper getting it in bulk of 25 packs as compared to buying it from the local
Review : it really crispy and the amount of chocolate filling is generous it at least times cheaper getting it in bulk of 25 packs as compared to buying it from the local

Predicted summary: good chocolate
```

Output 3:

```
Enter :  "These Albanese gummi bears and rings and so on are very good and tasty and high quality. The bears even have little faces. At my local candy store this
type of gummi stuff (bears, rings, snakes, balls, worms, whatever) are about $10/lb. These twin packs of 4.5 or 5 pound bags is a screaming deal as far as I'm
concerned. I'm probably 50 pounds deep in these friggin' things. Consumed!"
Review : "These Albanese gummi bears and rings and so on are very good and tasty and high quality. The bears even have little faces. At my local candy store this type of gummi stuff (bears, rings,
snakes, balls, worms, whatever) are about $10/lb. These twin packs of 4.5 or 5 pound bags is a screaming deal as far as I'm concerned. I'm probably 50 pounds deep in these friggin' things. Consumed!"
Predicted summary: best gummy bears ever
```

Output 4:

```
Enter :  The package came with the label torn off and no cooking instructions. I know how I normally cook Couscou
so tried 3/4 cup water to 1 cup couscous-brought to a boil and let sit, covered for 5min. It was mushy and tastel
ss.We have thrown out the rest of the container.
Review : The package came with the label torn off and no cooking instructions. I know how I normally cook Couscou
so tried 3/4 cup water to 1 cup couscous-brought to a boil and let sit, covered for 5min. It was mushy and tastel
ss.We have thrown out the rest of the container.

Predicted summary: poor quality
```

Figure 9: Output

8.Conclusion & FutureScope

In the new digital era where large amounts of datasets are budding on the globe, to analyze the same and extract the result is getting popular. While building the paper, we studied a lot of research in a statistical manner to understand the need and greed of visualizing the dataset and processing techniques applied on the text data. For the same, much research has been taken into count. By using seq2seqmodel with Lstm , we can predict the summarized text /review. So that the user can able to know the review without wasting time.

The proposed model able to predict the summarized text for the given input with 84.86% accuracy. Further this model can be developed by using sophisticated and advanced algorithms and we can also develop by implementing Graphical User Interface where the user can enter text and get the summarized output with good accuracy.

9.REFERENCES:

[1] Dutta, Sushama Rani, et al. "IoT in Autism Detection in Its Early Stages." Internet of Things: Enabling Technologies, Security and Social Implications. Springer, Singapore, 2021. 47-58.

[2] Dutta, Sushama Rani, Sujoy Datta, and Monideepa Roy. "Using cogency and machine learning for autism detection from a preliminary symptom." 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2019.

[3] Dutta, Sushama Rani, et al. "A Machine Learning-Based Method for Autism Diagnosis Assistance in Children." 2017 International Conference on Information Technology (ICIT). IEEE, 2017.

[4] Giri, Soumyajit, Sushama Rani, et al. "A real time epidemic alert generation system for rural areas using WBANs and kiosks." 2017 International Conference on Information Technology (ICIT). IEEE, 2017.

[5] Dutta, Sushama Rani, and Monideepa Roy. "EDOT: Context-aware tracking of similar data patterns of patients for faster diagnoses." 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT). IEEE, 2017.

[6] Dutta, Sushama Rani, and Monideepa Roy. "Providing context-aware healthcare services using circular geofencing technique." 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). IEEE, 2016.]

[7] Dutta, Sushama Rani, et al." Handling Scarcity of Data in Autism Identification using Binary Imputation Method" International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-1, May 2019

[8] Donalek, Ciro. "Supervised and unsupervised learning." Astronomy Colloquia. USA. 2011.

[9] Kukkar, Ashima, and Rajni Mohana. "An optimization technique for unsupervised automatic extractive bug report summarization." International Conference on Innovative Computing and Communications. Springer, Singapore, 2019.

[10] Huai, Beibei, et al. "Mining Intentions to Improve Bug Report Summarization." SEKE. Vol. 2018. 2018.

[11] Angel Hern´andez-Casta˜neda et al . "Extractive Automatic Text Summarization Based on Lexical- Semantic Keywords". In: IEEE Access 8 (2020), pp. 49896–49907.