Abstractive Text Summarisation

Submitted By

Group No - 39

Bhukya Vasanth Kumar - B180441CS - vasanthkumar_b180441cs@nitc.ac.in

Sevakula Jyothi - B180359CS - jyothi_b180359cs@nitc.ac.in

Billa Amulya - B180404CS - billa_b180404cs@nitc.ac.in

Under the Guidance of

Dr. Raju Hazari
Assistant Professor, Dept. of CSE

Department of Computer Science & Engineering
National Institute of Technology Calicut
Calicut, Kerala, India - 673 601

***Abstract*—Natural language processing (NLP) describes the interaction between human language and computers by automatic manipulation of natural language like text and speech by a software. NLP is the thing that makes it workable for PCs to peruse the text and decipher it. In recent years, availability of the data generated has increased to exponential scale. Summarisers make it simpler for users to summarize the data without pursuing it totally. There are two different approaches, namely Extractive Summarisation and Abstractive Summarisation. This work is based on the Abstractive Summarisation which is quite advanced than the former. Abstractive text Summariser generates new sentences, possibly rephrasing or using the words that were not in the original text in a human readable format. This ensures that the core information is conveyed through the shortest text possible. In this work, the main goal was to increase the efficiency and diminish train loss of sequence to sequence model for making a superior abstractive text summariser.**

***Index Terms*—Summary, NLP, Abstraction, Extraction, Encoder, Decoder, Seq2Seq**

I. INTRODUCTION

The advancement of natural language processing (NLP) over the years has created more possibilities in the way we can manipulate data. With the phenomenal growth in the internet and technology, everything now in front of us is data that is being used for various purposes. There are massive amounts of information and documents online that serve various tasks. With the increase in availability of documents, demand for exhaustive research in the area of automatic text summarization is expanding.

According to [1], summary can be defined as ”a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually, significantly less than that.”. [2] says, The Global Natural Language Processing (NLP) Market was valued at USD 10.72 billion in 2020, and it is expected to be worth USD 48.46 billion by 2026, registering a CAGR of 26.84% during the forecast period (2021-2026).

The automated text summarisation should produce an accurate summary from the given text without losing the key data and comprehensive meaning. The summary generated should help to understand required information in less time, thereby reducing the reading time. The automated text summary enables people to concentrate on key terms of the given text that are worth noting. Since the last few years, various approaches and models have been designed to automate the summary of particular texts. Text summarisation is widely used in areas like news article summary, tracking patient's health history, search engines, etc. Apps like In-shorts use this method to summarise news and provide headlines. Search engines like Google Chrome use them to generate snippets of products, and to facilitate headlines for news across the globe.

A human can produce a summary of a text in various ways based on how they understand it and the keywords they use. Doing the same with machines is challenging and hence, text summarisation is considered to be limited. Since, the machines have a deficiency of human knowledge, text summarisation turns out to be a non-trivial task[3] and the results vary.

II. PROBLEM DEFINITION

To generate a short, precise summary of a longer text by retaining the key information of the text using Natural language processing (NLP) techniques.

A. Input and Output

Input:
1. Data(Sentence/Paragraph)
2. Original summary of data (Used for calculating accuracy)

Output:
A short summary is generated based on abstractive text summarisation.

B. Diagramatic Representation

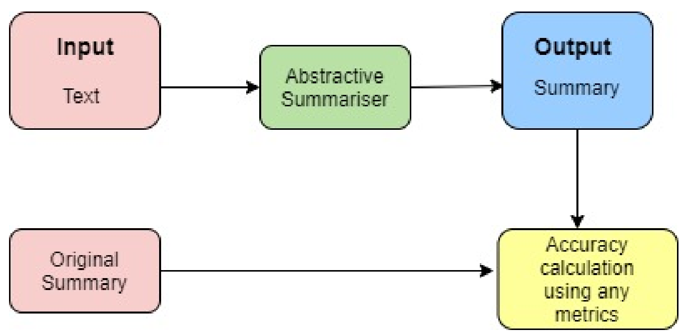

Fig. 1. Input and Output Diagram

III. LITERATURE SURVEY

A summary can be defined as a text that represents the main ideas or key information in an original text in less space. Suppose, if all the sentences in the original text are important, then the process of summarization would be less effective as the size of the summary would effect its informativeness. The main challenge in the process of summarization is identifying the informative segments and finally generating it as a consice text[1].

The evaluation of quality of a summary is one of the most difficult problem this is because there is nothing called ”ideal” summary. In many cases such as news articles, human summarizes they only meet upto an 60% of the time measuring sentence content overlap[13]. In general, summarization can be defined as combination of topic identification, interpretation and generation. For this identification part the main goal is to retain the most important, central topics. For interpretation part, the main goal is to perform compression through re-interpreting and fusing the pulled out topics into more succinct ones. This is quite important because abstracts are usually much shorter than their equivalent extracts. Where as for the generation part reformulation of the extracted and fused material into a text with new phrases is quite challenging.

Abstractive techniques need a more profound examination of the text i.e it needs deeper analysis. These techniques can produce new sentences,and improve the focus of the summary to maintain a decent compression rate[4]. An RNN Encoder decoder based architecture, which is based on sequence to sequence model is applied to process the data in sequential manner such that the input of any state may depend on the output of the previous states [5,6], this scenario is most likely to occur in a sentence, where the meaning of a word is closely related to the previous words meaning.

Teacher forcing is a fast and effective way to train RNNs. However, this approach may result in more fragile/unstable models.In [14], Seq2seq Model with Attention using GRU and Teacher Forcing is implemented,which would work fine on shorter summaries (Approx. 50 words).

Raphal et al. surveyed several abstractive text summarisation processes in general [15]. Their study differentiated between different model architectures, such as reinforcement learning (RL), supervised learning, and attention mechanism. In addition, comparisons in terms of word embedding, data processing, training, and validation had been performed. However, there are no comparisons of the quality of several models that generated summaries.

Sutskever et al. [7] describes an end-to-end approach to *sequence to sequence* learning using a Multilayer LSTM. The neural network contains encoder and decoder. Encoder uses a fixed length of text as input and Decoder represents the output. In [8], a bi-directional RNN with LSTM's in encoding layer and attention mechanism in decoding layer and the sequence to sequence model is used to generate a abstractive summary of text, thereby increasing efficiency and reducing the training loss. According to [9], Text-To-Text Transfer Transformer (T5) based abstractive text summarization method shows better performance than baseline attention based seq2seq approach.

Shi et al. presented a comprehensive survey of several abstractive text summarisation models, which are based on sequence-to-sequence encoder-decoder architecture for convolutional and RNN seq2seq models. The focus was the structure of the network, training strategy, and the algorithms employed to generate the summary [16].

Study [17], explains the problem of vanishing gradients, which happens while training a long sequence with an RNN, is solved with gated RNNs. Allowing the gradients to backpropagate along a linear path using gates, each with a weight and bias, can address this problem.The weights and biases of the gates are updated during training. LSTM and GRU,two RNN types, are the most often used gated RNNs.

Rnn encoder-decoder summarisation is utilised with LSTM units and attention to generate headlines from the text of news articles. This model is used in several NLP applications, such as machine translation and text summarisation.The vector representation of the current input word and the output of the hidden states of all previous words are merged and supplied to the next hidden state in the RNN encoder-decoder model at specified hidden states on the encoder side.The model is quite effective in predicting headlines from the same newspapers as it was trained on. [18]

From [19], we can notice that LSTM and GRU are generally used for abstractive summarisation.since LSTM has a memory unit that provides extra control but the computation time of the GRU is less.Also,while it is easier to tune the parameters with LSTM, the GRU takes less time to train. [20] says that, Before being used for NLP applications like text summarisation, the attention mechanism was used for neural machine translation.

In the paper [21], dual attention was used.Abstractive summarization necessitates the fusion of several elements of the original text, which might lead to the creation of fake facts. According to a survey, over 30% of the outputs from neural summarization systems have this issue.To avoid generating fake facts in a summary, The dual-attention sequence-to-sequence framework is then suggested. Two bidirectional GRU encoders and one dual attention decoder make up the proposed dual attention technique.

Recent approaches that used deep learning for abstractive text summarisation and metrics for evaluating these approaches and the challenges faced while implementing these approaches and their solutions were mentioned in [4]. The RNN and attention mechanism were the most commonly employed deep learning techniques.we can also notice that few methods applied LSTM to resolve the gradient vanishing problem that occured when using an RNN, while other approaches applied a GRU.

IV. BASICS OF NERURAL NETWORK

A. History

Initially in 1943, Warren McCulloch and Walter Pitts wrote a paper on how neurons work they proposed a simple neural network using the electrical circuits. Nathanial Rochester from the IBM research laboratories led the first effort to simulate a neural network. That first attempt failed. But later attempts were successful. [22] Neural networks have a unique kind of advantages such as parallel distribution, self-organizing, adaptive, self-learning and also fault tolerance, excellent performance and has the ability to adapt to different factors and also works pretty well with nonlinear problems.

B. Neural Network

The nodes, also known as processing elements (PE), and the connections form the foundation of the ANN. Each node has its own input, which it uses to receive communications from other nodes and/or the environment, as well as an output, which it uses to interact with other nodes or the environment. The 'Law of Learning' describes the process through which nodes adapt themselves. The NN's environment is represented by data [23].

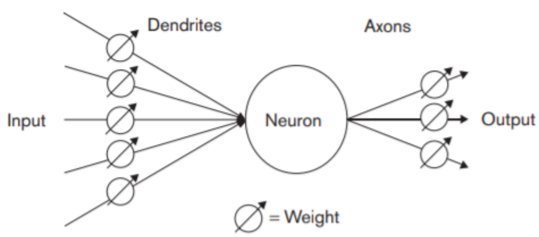

Fig. 2. Diagram of a single processing element (PE) containing a neuron, weighted dendrites, and axons to process the input data and calculate an output

Neural networks are adaptive systems that are inspired by the human brain's functioning processes [24]. They are systems that can change their internal structure in response to a function goal. They are especially well-suited to handling nonlinear issues, as they can recreate the fuzzy rules that control the best solution for these situations [25].

Neural networks used in deep learning consist of different layers connected to each other and work on the structure and functions of a human brain. It Learns from huge volumes of data and uses Complex algorithms to train a neural network. [26]

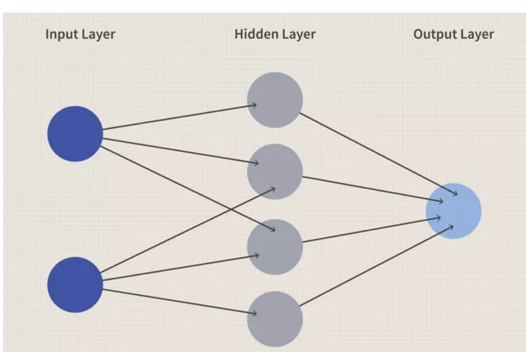

Fig. 3. A simple Neural Network

Layers of interconnected nodes make up a neural network. Basically, a perceptron is a linear machine learning algorithm for binary classification tasks. In the neural network, each node is a perceptron, which works in the same way as a multiple linear regression. The perceptron converts the signal from a multiple linear regression into a nonlinear activation function.

C. FEED FORWARD NEURAL NETWORK

In a feed forward network information flows in forward direction from the input nodes through the hidden layers (if they are present) to the output nodes. There are no cycles or loops in the network. It passes via the input layer, then the hidden layer, and finally the output layer, where we expect to achieve the intended result. A multilayer perceptron is another name for a feedforward neural network. It's a network with no closed paths or loops in the directed graph that establishes the linkages. [27]

ISSUES

- Feed forward neural network are not designed for sequence or time series data. Hence, results with time series or sequential data are bad.
- Feed forward neural network does not model memory. Thus, it cannot have future scope in prediction tasks.
- It'll have the loss of neighborhood information.

Considering the above issues, it is important to go with Recurrent Neural Network.

D. RECURRENT NEURAL NETWORK

Traditional feed-forward networks cannot be used for learning and prediction when dealing with sequential or time series data. It is necessary to develop a method that can preserve past or historical data in order to estimate future values. Recurrent neural networks, or RNNs for short, are a type of artificial neural network that can deal with sequential data and can be trained to remember information from the past.

RNNs are a type of neural network that are designed for capturing information from sequence or time series data. RNNs are good for processing sequence of data for predictions. Also, RNNs train faster and uses less computational feature as there are less Tensor operations to compute. RNNs are used in various application such as Google Image Search, Google Translation, Google lenses. Subjects like Speech recognition, language translation, stock prediction and Spam mail detection uses RNN and their features. Thus, it is valid to say that RNNs are good at Modelling Sequence Data.

Consider the following review as an example.

Superb! The box of cookies are so delicious that I want to eat them daily as snacks. I haven't completed the box yet, but will surely buy a lot in coming days.

Here, the brain subconsciously remembers only the key words like Superb and delicious. Other words like I, as, the, a, etc will not have much importance. So the final short review can be termed as 'great', or something which is positive. The same has to be done using RNNs. Here, each word in the review sentence is feeded to RNN, each at a time. The words get transformed to machine readable vectors. Then each vector is processed to RNN in a sequence. While processing, it passes previous head state to next sequence in hidden state. The hidden state holds the information on previous data.

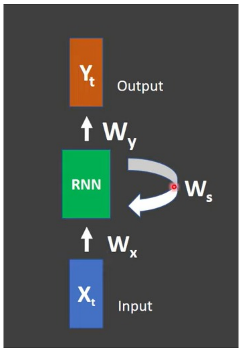

Fig. 4. A simple RNN

A simple RNN undergoes following mechanism. Below formulas represent RNN and the way data is feeded into them inorder to get the output.

$$S_t = F_w(S_{t-1}, X_t) \tag{1}$$

$$S_t = tanh(S_{t-1}, X_t) \tag{2}$$

$$Y_t = W_y S_t \tag{3}$$

E. VANISHING GRADIENT PROBLEM

Backpropagation is used to find neural network gradients. Simply put, backpropagation finds the network's derivatives by travelling from the final layer to the first one layer by layer. To compute the derivatives of the initial layers, the derivatives of each layer are multiplied down the network (from the final layer to the start) using the chain rule.

When n hidden layers utilise a Sigmoid function as an activation, n tiny derivatives are multiplied together. As we propagate down to the initial layers, the gradient reduces exponentially.

The weights and biases of the first layers will not be updated adequately with each training session if the gradient is modest. Because these first layers are typically critical for recognising the main elements of the input data, they can lead to overall network inaccuracy.

Therefore, the gradient might exponentially shrink. The gradient for next node will be even smaller if the previous node has smaller gradient. Thus, the adjustments tends to be negligible as it goes further. Hence, due to the issues of long-term dependency, we need to go through LSTM.

F. LSTM: LONG SHORT TERM MEMORY

LSTM basically learns to keep only relevant information to make the predictions. LSTMs are specifically developed to prevent the problem of long-term dependency. They don't have to work hard to remember knowledge for lengthy periods of time; it's nearly second nature to them. The are capable of learning long-term dependencies among the information for long periods of time is their default behaviour.

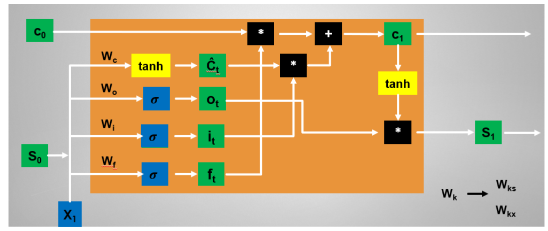

Fig. 5. A simple LSTM

- FORGET GATE: This gate decides what information should be thrown away or kept. Th information from the previous hidden state (s(t-1)) and information from the current input (x(t)) is passed through the sigmoid function. Values come out between 0 and 1. The closer to 0 means to forget, and the closer to 1 means to keep.

$$f_t = sigmoid(W_f S_{t-1} + W_f X_t) \tag{4}$$

- INPUT GATE: The input gate is used to update the cell state. First, we use a sigmoid function to combine the previous hidden state (s(t-1)) and the current input (x(t)). To help regulate the network, you additionally provide use tanh function to squish values between -1 and 1. The sigmoid result is then multiplied by the tanh output. The sigmoid output will determine which information from the tanh output should be kept.

$$i_t = sigmoid(W_i S_{t-1} + W_i X_t) \tag{5}$$

- CELL STATE: First, the cell state gets point-wise multiplied by the forget vector. This has the possibility of dropping values in the cell state if it gets multiplied by values near 0. Then we multiply the input vector to the candidate vector. The point-wise addition of both the results in new cell state.

$$C_t^1 = tanh(W_C S_{t-1} + W_c X_t) \tag{6}$$

$$c_t = (i_t * C_t^1) + (f_t * c_{t-1}) \tag{7}$$

- OUTPUT GATE: The output gate decides what the next hidden state should be. The hidden state is also used for predictions.

$$o_t = sigmoid(W_o S_{t-1} + W_o X_t) \tag{8}$$

$$h_t = o_t * tanh(c_t) \tag{9}$$

V. MOTIVATION

NLP has various applications. Study [28], [29] tells that Automatic Summarisation can be done more quickly with the help of NLP. Automatic summarisation is critical not just for summarising the importance of documents and data, but also for comprehending the emotional meanings of the data, as in obtaining information from social media. When utilised to present an overview of a news item or blog post while maintaining a strategic distance from multiple sources and maximising the diversity of content acquired, automatic summarising is especially useful. The difficulty of locating a crucial piece of data from a massive database can be minimised using this method.

A. *Handling Large Volume of Data*

Normal language handling assists PCs with talking with individuals in their own language and scales other language-related assignments. For example, NLP makes it possible for PCs to comprehend the message, hear discourse, decipher it, measure feeling and sort out what parts are huge. The current machines can break down more language-based information than people, without exhaustion and in an anticipated, reasonable way. Considering the stunning measure of unstructured information that is made every day, from clinical records to online media, technology will be vital to totally dissect text and discourse information proficiently. From [30], It is evident that, given the massive volume of unstructured data generated every day, from medical records to social media, automation will be essential for efficiently analysing text and audio data.

B. *Structuring a highly unstructured data source*

Human language is incredibly complex and varied. We communicate in a seemingly endless number of ways, both verbally and in writing. Not only are there several dialects and languages, but each language has its own set of grammar and sentence structure rules, as well as words and slang. When we write, we frequently misspell words, abbreviate them, or forget to use punctuation. We have regional accents, mutter, stammer, and use terminology from several languages when we speak.

NLP is important because it helps to resolve ambiguity in language and gives crucial quantitative structure to the data for downstream applications such as speech recognition and text analytics.

C. *Motivation Behind LSTM*

Study [31], tells that the duration of the network is unknown. The network is reduced to a certain value of the error on the sample means that the training has been completed. This value does not give us optimum results. From [31] [32], it is evident that there is difficulty of showing the problem to the network. Problems have to be translated into numerical values before being introduced to ANN. The display mechanism to be determined will directly influence the performance of the network and thus it is dependent on the user's ability. Therefore, Feed forward Neural Networks is studied.

Feed forward neural network has issues with sequence of data or time series data as they do not have loop. They'll have the loss of neighborhood information. Thus, the advanced mechanism RNN i.e Recurrent Neural Network is studied.

RNN suffers with the Vanishing gradient problem. The gradient for next node will be even smaller if the previous node has smaller gradient. Thus, the adjustments tends to be negligible as it goes further. Hence, due to the issues of long-term dependency, LSTM is favourable.

VI. DESIGN

A. *Seq2Seq Model*

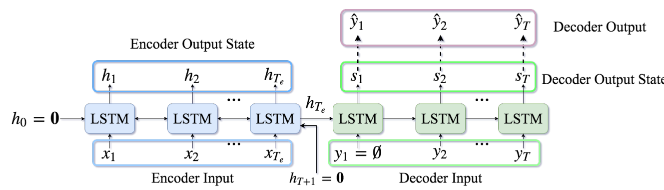

Fig. 6. Seq2Seq Model

The above shown Encoder-Decoder Architecture is a Simple seq2seq model. The encoder has a input range of Te units and after evaluation the decoder has delivered the output of range T units following the condition T¡Te. Every encoder in the state ht, can receive the previous encoder's hidden state ht-1, this is valid in both unidirectional and bidirectional LSTM but the bidirectional LSTM has an additional feature of accessing the next encoder's hidden state which is ht+1.

B. *Flowgraph*

The following Flow Graph displays the flow of control information of the model.

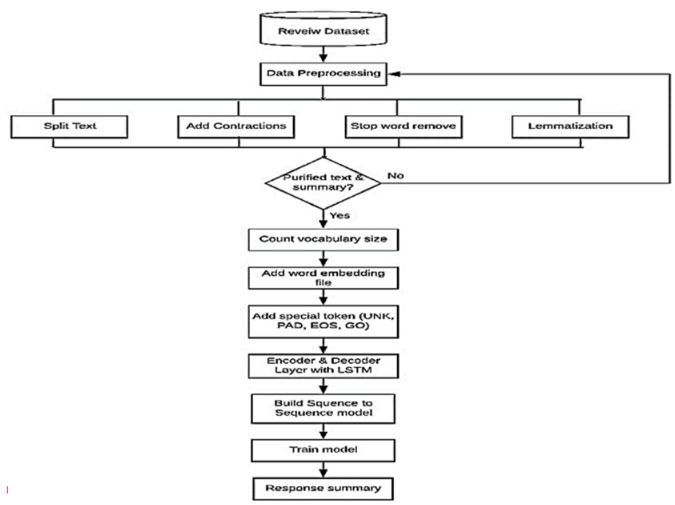

Fig. 7. FlowChart

VII. PREPROCESSING

A. DATA SET

The data used is a raw information that is unprocessed and reflects both human and machine observations of the world. The data-set used in this Project is Amazon-Fine-Food-Reviews. The data-set will contain 10 coloumns which includes: ID, ProductID, UserID, ProfileName, Time, Summary, Review, etc. The dataset has been taken from Kaggle website which has almost 294 MB size. There are almost 560,000 reviews in the dataset.

```
reviewsData=pd.read_csv("/content/drive/My Drive/
    Reviews.csv",nrows=30,000) #Taking 30,000 out of
     500,000 reviews
print(reviewsData.shape) # SHAPE of DATSET
reviewsData.head(n=10)
```

Listing 1. Dataset

B. IMPORTING NECESSARY LIBRARIES

A library is basically a collection of modules that can be called and used. This project needs various inbuilt libraries such numpy, pandas, tensorflow, keras, etc.

C. CLEANING THE DATA

The practise of correcting or deleting incorrect, corrupted, improperly formatted, duplicate, or incomplete data from a dataset is known as data cleaning [33].

```
#Reducing the length of dataset for better training
    and performance
reviewsData.drop_duplicates(subset=['Text'],inplace=
    True) #Dropping the rows with Duplicates values
    of 'Text'
reviewsData.dropna(axis=0,inplace=True) #Dropping
    the rows with Missing values
```

Listing 2. Cleaning the Datset

D. DATA ENCODING

For better model training, the dataset is preprocessed by converting everything to lowercase. While moving further we tried to remove HTML tags, stop words, punctuations. Special characters and any text inside parenthesis, and contraction mapping are also removed.

```
import nltk
nltk.download('stopwords')

stop_words = set(stopwords.words('english'))
def text_cleaner(text,num):
    newString = text.lower()  #converts all
    uppercase characters in the string into
    lowercase characters and returns it
    newString = BeautifulSoup(newString, "lxml").
    text #parses the string into an lxml.html
    newString = re.sub(r'\([^)]*\)', '', newString)
    #used to replace a string that matches a regular
     expression instead of perfect match
    newString = re.sub('"','', newString)
    newString = ' '.join([contraction_mapping[t] if
    t in contraction_mapping else t for t in
    newString.split(" ")]) #for expanding
    contractions using the contraction_mapping
    dictionary
    newString = re.sub(r"'s\b","",newString)
    newString = re.sub("[^a-zA-Z]", " ", newString)
    if(num==0):
      tokens = [w for w in newString.split() if not
    w in stop_words]  #converting the strings into
    tokens
    else :
      tokens = newString.split()
    long_words=[]
    for i in tokens:
        if len(i)>1:    #removing short words
            long_words.append(i)
    return (" ".join(long_words)).strip()

#Calling the function
cleaned_text = []
for t in reviewsData['Text']:
    cleaned_text.append(text_cleaner(t,0))
```

Listing 3. Preprocessing Data

E. SPLITTING DATA INTO TRAINING SET AND TEST SET

The splitting of the Data into Training Set and Test Set is important because there will not be enough data in training or testing set for the model to learn effective mapping from inputs to outputs.

```
#Splitting the Dataset
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(np.
    array(df['text']),np.array(df['summary']),
    test_size=0.2,random_state=0,shuffle=True)
```

Listing 4. Train and Test Set

VIII. DATA VISUALIZATION

Data visualisation is the process of converting information into a visual representation, such as a map or graph, in order to make data easier to comprehend and extract insights from. Data visualization's major purpose is to make it easier to spot patterns, trends, and outliers in massive data sets. Information graphics, information visualisation, and statistical graphics are all terms that are frequently used interchangeably. [34]

```
import matplotlib.pyplot as plt
text_word_count = []
summary_word_count = []

#Populating the lists with sentence lengths
for i in reviewsData['Cleaned_Text']:
      text_word_count.append(len(i.split()))

for i in reviewsData['Cleaned_Summary']:
      summary_word_count.append(len(i.split()))

length_df = pd.DataFrame({'text':text_word_count, '
    summary':summary_word_count})
length_df.hist(bins = 30)
plt.show()
```

Listing 5. Populating the lists with sentence lengths

The following plot shows the histogram representation with lists with both text lengths and summary lengths. text_word_count is the list of lengths of each review whereas summary_word_count is the list of lengths of each summary.

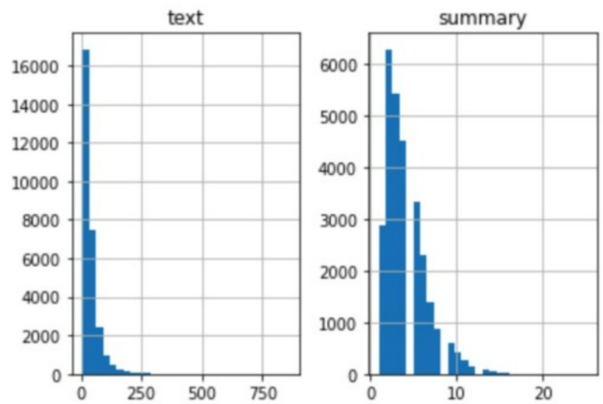

Fig. 8. Histogram Plot

IX. RESULTS

A. Sample Output

REVIEW: buy bob red pack boxes bags boxes size makes easier store good flour bag good size baking small family also good shape flour

VECTOR : 1, 6, 11, 59, 43, 2, 0, 0

ORIGINAL SUMMARY: good for gluten free

PREDICTED SUMMARY: great chips

X. WORK DONE

1. Studied and examined various papers and completed the literature work for this topic
2. Inspected some popular text summarising web applications to understand the features they provide [10]
3. Briefly discussed the proper parameters, inputs and outputs of the model from our understanding.
4. Explored various methods to generate abstract summary for a given text.
5. Selected amazon-fine-food-reviews from kaggle website which is almost 294 megabyte with 560,000 reviews approx.
6. Performed Data Preprocessing which includes cleaning the data, encoding the data by expanding all the contractions, and removing all the unwanted characters and obtained purified text and summary. Later the data is splitted into train and test data.
7. Added special tokens at the start of summary and end of summary to differentiate the given summary and text for better decoding.
8. Calculated the percentage of rare words and total coverage of rare words in the text.
9. Also calculated the maximum length of reviews and summary for making all the reviews length equal.
10. Padding zeros to the maximum length for both training and testing data has been done.
11. Implementation of the Model
12. Results from the implementation is discussed.

XI. WORK PLANS FOR S8 SEMESTER

1. Making minor changes in the model to increase the accuracy.
2. Designing the new model inorder to increase the accuracy
3. Implementing the algorithm of the new model.
4. Finding the end results using bleu score and various other metrices.
5. Comparing the previous and improved design results to observe the difference.

XII. SUMMARY

Our work deals with the implemenation of a model for abstractive text summarisation. The work uses deep learning in order to increase the efficiency. Decreased train loss is also a major contribution in this the sequence to sequence model.

In this Report, we discussed how exactly abstractive text summarisation is a challenging task, yet much needed in present times. The structure of our paper is as follows. In the section II, we clearly specified the problem definition with a brief input-output statement along with a pictorial representation. In section III, we mentioned most closely related works that has been done by researches before, that were based on our problem. The section IV gives a brief idea on the basics of Neural Networks. It further explains the advancement in Neural Networks. Feed forward neural networks and RNN are explained later on. LSTM is finally explained with the internal mechanism. Later on, in the section V, motivation behind the NLP and LSTM is given. We went on further explaining the design and the flow graph in section VI. The entire preprocessing we did in implementation is explained in the section VII which includes cleaning and encoding of data. Section VIII gives the data visualisation. The last section describes the results from the implementation.

XIII. ACKNOWLEDGEMENT

It is with great respect that we remember the names of all who have been a great help and guidance throughout our project. With a profound sense of gratitude, we would like to express our heartfelt thanks to our guide and project coordinator, *Dr. Raju Hazari, Assistant Professor, Department of Computer Science and Engineering* for his expert guidance, co-operation, and immense encouragement in pursuing this project.

We also express our sincere regards to *Dr. P Arun Raj Kumar, Assistant Professor, Department of Computer Science and Engineering* for their valuable coordination. Our sincere thanks are extended to all the teachers of the Department of Computer Science and Engineering and to all our friends for their help and support.

References

[1] Dragomir R Radev, Eduard Hovy, and Kathleen McKeown. 2002." Introduction to the special issue on summarization,"Computational linguistics 28, 4 (2002).

[2] https://www.mordorintelligence.com/industry-reports/natural-language-processing-market.

[3] Mehdi Allahyari, Seyedamin Pouriyeh and Mehdi Assef "Text Summarization Techniques: A Brief Survey".

[4] Saranyamol C S, Sindhu L, "A Survey on Automatic Text Summarization", International Journal of Computer Science and Information Technologies, 2014,Vol. 5 Issue 6.

[5] C. L. Giles, G. M. Kuhn, and R. J. Williams,"Dynamic recurrent neural networks: theory and applications," IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 153–156, 1994.

[6] A. J. Robinson,"An application of recurrent nets to phone probability estimation," IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 298–305, 1994.

[7] Sutskever et al "Sequence to Sequence Learning with Neural Networks," Conference on Neural Information Processing Systems (NIPS,2014).

[8] Abu Kaisar Mohammad Masum, Sheikh Abujar,"Abstractive method of text summarization with sequence to sequence RNNs,"IEEE (2019).

[9] Ekaterina Zolotareva, Tsegaye Misikir Tashu, "Abstractive Text Summarization using Transfer Learning,"2020.

[10] https://quillbot.com/summarize

[11] https://github.com/google-research/google-research/tree/master/rouge

[12] https://github.com/neural-dialogue-metrics/BLEU

[13] Hovy, E. and C.-Y. Lin. 1999. "Automated text summarization in SUMMARIST". In I. Mani and M. T. Maybury, editors, Advances in Automatic Text Summarization. MIT Press, Cambridge, pages 81–94.

[14] https://medium.com/@dusejashivam

[15] N. Raphal, H. Duwarah, and P. Daniel, "Survey on abstractive text summarization,"in Proceedings of the 2018 International Conference on Communication and Signal Processing (ICCSP), pp. 513–517, Chennai, 2018.

[16] T. Shi, Y. Keneshloo, N. Ramakrishnan, and C. K. Reddy,Neural Abstractive Text Summarization with Sequence-ToSequence Models: A Survey, http://arxiv.org/abs/1812.02303,2020.

[17] Deep Learning Based Abstractive Text Summarization: Approaches, Datasets, Evaluation Measures, and Challenges by Dima Suleiman and Arafat Awajan

[18] Generating News Headlines with Recurrent Neural Networks by Konstantin Lopyrev

[19] Abstractive text summarization using LSTM-CNN based deep learning by S Song, H Huang

[20] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in Proceedings of the International Conference on Learning Representations, Canada, 2014

[21] Z. Cao, F. Wei, W. Li, and S. Li, "Faithful to the original: fact aware neural abstractive summarization," in Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), New Orleans, LA, USA, February 2018.

[22] http://www2.psych.utoronto.ca/users/reingold/courses/ai/cache/neural4.html

[23] Widrow G, Hoff ME. Adaptive switching circuits. Institute of radio engineers, Western Electronic show and Convention, Convention record. 1960, part 4:96–104

[24] McCulloch WS, Pitts WH. A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 1943; 5:115–133. KK The first mathematical model of logical functioning of brain cortex (formal neuron) is exposed in the famous work of McCulloch and Pitts

[25] Introduction to Artificial neural networks by European Journal of Gastroenterology Hepatology (EUR J GASTROEN HEPAT), Jan 2008

[26] https://www.ibm.com/cloud/learn/neural-networks

[27] https://www.educba.com/feedforward-neural-networks

[28] https://www.analyticsinsight.net/what-is-nlp-and-why-is-it-important/

[29] https://towardsdatascience.com/why-nlp-is-important-and-itll-be-the-future-our-future-59d7b1600dda

[30] https://www.sas.com/en_sg/insights/analytics/what-is-natural-language-processing-nlp.html

[31] https://intellipaat.com/community/21886/advantages-and-disadvantages-of-neural-networks

[32] https://www.asquero.com/article/advantages-and-disadvantages-of-artificial-neural-networks/

[33] https://medium.com/analytics-vidhya/data-cleaning-and-preprocessing-a4b751f4066f

[34] https://searchbusinessanalytics.techtarget.com/definition/data-visualization