IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, May 2022
Impact Factor: 6.252

Text Summarization Using Deep Neural Networks

Ranjeet Pawar and Mithun Mhatre
Department of Information Technology
Bharati Vidyapeeth Institute of Technology, Navi Mumbai, Maharashtra, India

Abstract: *A Deep Learning approach for text summarization has been explored in this paper. Automatic text summarization is a technique that compresses large amounts of text to a shorter summarized format including the important information. Here we present an approach to designing an automatic text summarizer that generates a summary by extracting sentences. The Deep learning approach deals with the abstractive text summarization for a single document based on the Sequence-to-Sequence model with LSTM. The system has the backend model for the processing of the document, a frontend for the user to input the document which is forwarded to the deep learning model through which the summarized document is given to the user. The implementation of the backend is done in python for the creation and the training of the deep learning model. Two evaluation techniques ROUGE and BLUE have been used for the evaluation of the model accuracy.*

Keywords: Agricultural supply chain, Blockchain, Information database, Resource wastages

I. INTRODUCTION

The amount of data produced in the world is increasing day by day. Data amounting to more than 2.5 quintillion bytes have been produced in the last two years which is almost equal to 90 percent of the total data the world has ever produced. This rise in the amount of data is due to the widespread reach of the internet in the world and many social media apps, blogs, posts related apps etc. There is a large amount of textual data produced due to the extensive use of the internet by various means such as social media, digital marketing, newspapers, reviews, blogs etc.

Data is getting incremented every second and the rate of addition of data to the internet also keeps increasing. With the production and availability of such huge amounts of data there arises a need for processing the data and deriving insights from the textual data. One way to do it is using summarization. A precise summary of data helps readers to understand the gist of the data and derive usable and necessary information from it. With this in mind, a need for making a system which produces precise and accurate summaries arises. The system can save a lot of time for the users and also in cases, where a user needs to store data, it can also save a lot of space.

Text summarization is divided into two main types: abstractive and extractive text summarization. In these two branches, there are a lot of data summarization techniques available in the market. Many algorithmic techniques pertaining to machine learning and deep learning have also been employed to procure good results for text summarization and a lot of text summarization methods in many different languages has also been developed through various researchers, these research techniques are published in various the research papers of various professional organizations such as IEEE, Springer, Elsevier to name a few.

Since as said earlier there has been a lot of research and development done for summarization in the English language, we use this as a base for making a system capable of summarizing and producing summarized results. In the upcoming documentation we present a deep learning technique for abstractive text summarization using sequence - to - sequence model with LSTM.

1.1 Background

Significant amount of work has been done on automatic summarization. Since earlier models greatly focused on extractive methods, abstractive summarization is a relatively new research area with less experimentations. However, a recent advance in research using abstract methods has made it quite a popular field. In 2015, Cho et al described state-of-art performance of encoder decoder networks, for which output possesses the same structure of input. In the same year, Rush et al developed a neural attention feed-forward model for sentence-level summarization tasks which performed well on the DUC-2004 competition. In 2016, Chopra et al designed attentive recurrent neural networks to improve the results

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, May 2022
Impact Factor: 6.252

on the same task. There are many developments in the deep learning methodologies and NLP has progressed quite much, many researches in the field of using encoder decoder model with different RNNs like LSTMs and BILSTMs along with attention mechanism etc, have led to the development of quite good summarization models.

1.2 Project Idea

In the implementation of the project we used encoder decoders for the sequence to sequence model and explored using LSTM cells. Basically the model will be based on a frontend from where the user will input the document which will be in English language, this document will be sent to the model which will summarize the document and produce the result i.e. the summary. This result is reverted back to the user to the frontend from where he can view his summarized document.

1.3 Motivation

Now a days summarization of document is very crucial and important. For summarization of legalise document and for usability for government work, document summarization is very helpful. Also summarization involves so many aspects of deep learning and machine learning which involves the NLP and deep neural network. Here main motivation is that the deep neural networks which can be highly use in so many research papers and give high accuracy. The need for summarization in real world scenarios such as movie reviews, restaurant reviews etc. also in the fields of medical science grants it a huge scope making it important and interesting to understand the current working and limitations of deep learning.

1.4 Project Challenges

The challenges faced in the development of related to the hardware and software configurations of the system used for developing of the project are:

1. Model Training: The sequence to sequence model has to be trained for a huge number of epochs hence, time required for the project model is huge, since the accuracy of the model is dependent on the number of epoch the deep learning model is trained , hence this cannot be compromised.
2. Hardware Limitation: Since our systems lack powerful GPUs for the processing of the model, it takes more time for training the model.
3. There was a need to translate the document to Hindi if the document given by the user is in another language , we had to limit it to English language for translation since the scope would expand a bit too much if users started to input other languages for summarization to be summarized in Hindi.

1.5 Proposed Solution

The project employs abstractive text summarization for text summarization using deep learning. The already available data present on Kaggle has been downloaded from the website and after following the preprocessing steps, it is given to the encoder decoder model which converts the single document to English summary.

The solution for the system consists of a front end in the form of a website from which the user provides the document by interacting with the UI and passed to the above models and the output of the model in the form of text summary in English is provided as output to the user.

II. Literature Review

2.1 Related work and state of art

[1] Summcoder: An Unsupervised Framework For Extractive Textsummarization Based On Deep Auto-Encoders.

In this paper the author used a novel methodology for generic extractive text summarization of single documents. Their approach generates a summary according to three sentence selection metrics: sentence content relevance, sentence novelty, and sentence position relevance. A sentence ranking and a selection technique are developed to generate the document summary by ranking the sentences according to the final score obtained through the fusion of the three sentence selection metrics. The comparative study is conducted using the well-known Recall- Oriented Understudy for Gisting Evaluation (ROUGE) metric for text summarization. Their approach obtained highly competitive performance on the

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, May 2022
Impact Factor: 6.252

DUC 2002 dataset, and found that it outperforms most of the recent state-of-the-art methods, even those based on supervised learning.

[2] Automatic Text Summarization Using Supervised Machinelearning Technique For Hindi Language

This paper discusses single document automatic text summarization for Hindi text using Supervised Machine Learning Technique (SVM). It was performed on news articles of different categories such as Bollywood, politics and sports. The performance of the technique was compared with the human generated summaries. In this experiment, each sentence in the document was represented by a set of various features namely- sentence paragraph position, sentence overall position, numeric data, presence of inverted commas, sentence length and keywords in sentences. Then they were classified into one of four classes namely- most important, important, less important and not important. From the experimental results, it shows that the summarization is more difficult if they need more compression.

In future, more features like named entity recognition, cue words, context information, world knowledge etc., can be added in their work to improvise the technique and can also be extended to work on multiple documents.

[3] Dual Encoding For Abstractive Text Summarization

In this paper the author proposed method employs a dual encoder including the primary and the secondary encoders and 1 decoder. The primary encoder calculates the semantic vectors for each word in the input sequence. The secondary encoder first calculates the importance weight for each word in the input sequence and then recalculates the corresponding semantic vectors. The decoder with attention mechanism decodes by stages and generates a partial fixed-length output sequence at each stage. This paper also discusses dual encoding abstractive text summarization using deep learning.

[4] News Article Summarization With Attention-Based Deep Recurrent Neural Networks

In this paper the author focuses on an approach for building a text automatic summarization model for news articles, generating a one-sentence summarization that mimics the style of a news title given some paragraphs using Encoder-decoder using LSTM and GRU cells, and with/without attention. In this paper the results were measured by calculating their respective ROUGE scores the authors managed to build and train two relatively complex deep learning models and outperformed our baseline model, which is a simple feed forward neural network.

[5] An Improvised Extractive Approach to Hindi Text Summarization

This paper emphasizes an extractive approach for text summarization and its implementation on Java. In this paper a survey of the extractive and abstractive text summarization for English language has been carried out to form the basis and understanding of the techniques that can be used for Hindi text summarization. The system proposed in this paper followed 4 major steps: pre-processing, thematic word generation, sentence scoring and summary generation.

Preprocessing involved removal of stopwords in Hindi (a collection of 170 stopwords were present in the Hindi WordNet used). In the thematic word generation steps the words pertaining to the theme of the document to be summarized are extracted and ranked. From the sentences the Hindi words are reduced to their radix term and then thematic terms are determined. The sentences are scored on their relevance and ranking based on the previous steps and the summary is generated.

The average accuracy of the system with the expert's manual summary is found to be 85 % and also the average retention ratio is 81.1 %. This is found to be a considerably good percentage when compared with the system without the post processing stage.

[6] A Novel Technique for Multi Document Hindi Text Summarization

Extractive text summarization using machine learning techniques. The extraction technique of text summarization consists of selecting important sentences from source documents and arranging them in the destination documents. Fuzzy logic is used for sentence ranking. The summary generated by the system is found very close to the summary generated by humans. The Precision, Recall & F-score values show very good accuracy of summary generated by the system. The system achieves an average precision of 73% over multiple Hindi documents.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, May 2022
Impact Factor: 6.252

[7] Neural Abstractive Text Summarization With Sequence- To-Sequence Models

In this paper the author provided a comprehensive survey on the recent advances of seq2seq models for the task of abstractive text summarization. They discussed two categories of training methodologies, i.e., word-level and sequence-level training. The first model is an Encoder- decoder using RNN Seq2Seq Model which uses attention and pointing/copying mechanisms. The second model is Encoder-decoder using CNN Seq2Seq Model which uses position embedding mechanism. They measured results by calculating models respective ROUGE and BERT Scores.

2.2 Limitation of State of the Art techniques

S No.	Title	Challenges /Limitations
1.	SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders	Seeking improvised model for betteraccuracy Restricted to single document textsummarization
2	Automatic text summarization using supervised machine learning technique for hindi language	Summarization is more difficult if they needmore compression Restricted to Single Document TextSummarization
3	Dual Encoding for Abstractive Text Summarization	Seeking improvised model for betteraccuracy Restricted to single document textsummarization
4	News Article Summarization with Attention-based Deep Recurrent Neural Networks	The paper only serves the purpose of text summarization for only single document textsummarization The obtained model has relatively low
5	A novel technique for multi document Hindi text summarization	The model gives a good accuracy but lacks in terms of correct summarization.
6	An Improvised Extractive Approach to Hindi Text Summarization	Although the retention rate acquired is good, the accuracy may dip below 80% in certain cases, hence few enhancements in the model might be made. Web mining for extracting summaries for multi documents can be done, but the challenge is maintaining the accuracy when more documents are involved.
7	Neural Abstractive TextSummarization with Sequence-to-Sequence Models	Finer tuning for discriminator modelrequired. Models are trained longer they sometimes collapse a subset of filters to a single oscillating mode

2.3 Discussion and Future Direction

The future direction of the development of the works presented in the above papers vary with the papers, these are as follows for each of the above papers. The project documented in the report performs abstractive text summarization in English for a single document. Further the process can be implemented for multiple documents. Document input by the user is summarized by the system also the user, this can be further improved for languages other than English.

2.4 Concluding Remarks

As the availability of data in the form of text increases day by day, it becomes so difficult to read the whole textual data in order to find the required information which is both difficult as well as a time-consuming task for a human being. So, at that time Automatic Text Summarization performs an important role by providing a summary of a whole text document by extracting only the useful information and sentences. There are different approaches to text summarization. The real-world applications of text summarization can be: documents summarization, news and articles summarization, review systems, recommendation systems, social media monitoring, survey responses systems.

The paper provides a literature review of various research works in the field of automatic text summarization. This research area can be explored more by looking in existing systems and working on different and new techniques of NLP and Machine Learning.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, May 2022
Impact Factor: 6.252

III. Problem Definition and Scope

3.1 Problem Statement

To design a text summarizer using Sequence-to-Sequence model with LSTM Mechanism to generate summary for articles in English language.

3.2 Goals and Objectives

- To generate a summary for a document in English language using sequence to sequence encoder-decoder model.
- To obtain a summarized document of the parent document without losing the semantic meaning of the document.
- To provide the reader with filtered description of source text and a non-redundant presentation of facts found in the text.

3.3 Scope and Major Constraints

The main purpose is to provide reliable summaries of web pages or uploaded files depending on the user's choice. The unnecessary sentences will be discarded to obtain the most important sentences. The product includes the following components:

- Text Parser: It will divide the texts into paragraphs, sentences and words.
- Feature Vector Creator: This component will calculate and get the feature representations of sentences.
- Encoder and Decoder: The root part of Deep Learning. Mainly working with sequence of input data, in our case , the sentences
- Classifier: The classifier determines if a sentence is a summary sentence or not.

3.4 Hardware and Software Requirements for Project Development

A. Software Requirements

- OS requirements: Any OS will can be used to develop the model
- Linux
- Windows
- MacOS

B. Environment

- Any one from the below will do
- Anaconda – Jupyter Notebook / PyCharm / Spyder
- Google colaboratory

C. Library Imports

- numpy: for handling arrays
- pandas: for DataFrame
- re (regex): for cleaning text
- tensorflow (keras): for machine learning or deep learning
- nltk (NLP tool kit) : used for building Python programs that work with human language data for applying in statistical natural language processing (NLP)
- attention: for attention mechanism

D. Hardware Requirements

- CPU- Intel, AMD, etc.
- RAM - Minimum 4gb (16 gb recommended)
- Storage - 256gb space recommended
- GPU- Min 2gb recommended

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, May 2022
Impact Factor: 6.252

3.5 Expected Outcomes

- A summary generated for a document in English language
- A summarized document which doesn't lose the semantic meaning of the original document.
- Providing the user with filtered description of source text and a non-redundant presentation of facts present in the text.
- The Obtained Model should have high accuracy.

IV. System Analysis And Design

4.1 Specific Requirements

4.1.1 User Requirements

The user interfaces will be an icon in the browser. While using the browser, a user's click will trigger a panel containing 3 buttons. With these buttons the user will decide if the text to be summarized from the current website or a text file from his/her hard drive. The settings button will be used for determining the length of the sentence. After the selection of the text to be summarized will be sent to the server and the resulting package will be the summary of the text.

4.1.2 External Interface Requirements

In the external interface requirements two parts have been added: the software interfaces and the communication interfaces.

- Software Interfaces: In this system, open source API's will be used for tokenization and parsing texts to paragraphs, sentences and words. APIs will be used for getting words' roots, positions in sentence, stems and suffixes.
- Communication Interfaces: The only communication is between the extension and the server. JQuery- AJAX will be used to send queries and receive ones. HTTP will be used as the protocol.

4.1.3 Functional Requirements Text Summarizer Requirements

- The system should provide text parser functions which can take the whole text and separate into sentences, paragraphs and words.
- The system should provide a text-to-feature function which can take the necessary part and obtain a feature vector.
- The system should provide a well-trained encoder decoder model to generate better summaries.
- The system needs a classifier which is well-trained to select summary sentences.

Summarize Web Page Requirements

- A function which detects body parts and selects text. This function needs to extract unnecessary text from html.
- The system should provide communication between server and client with necessary network functions such as send and receive.

Summarize File Requirements:

- The system should provide a summarize file button with complete functionality.
- After the user selects the target file, the user presses the summarize file button and the web page application sends the file to the server.
- A set of functions that provide the reading from file depends on file extension.
- The system should provide communication between server and client with necessary network functions such as send file and receive file.

Train System Requirements

- The system should provide a login screen for admin.
- The system should provide taking new data from admin to train Autoencoders or classifiers to improve reliability.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, May 2022
Impact Factor: 6.252

4.1.4 Performance Requirements

Calculation time and response time should be as little as possible, because one of the software's features are timesaving. Whole cycle of summarizing a page/file should not be more than 30 seconds in order to write a 3 page long document. The capacity of servers should be as high as possible. Calculation and response times are very low, and this comes with the fact that there can be so many sessions at the same times. The software is only used in India, so you do not need to consider global sessions. 1 minute degradation of response time should be acceptable. The certain session limit is also acceptable at early stages of development. It can be confirmed to the user with "servers are not ready at this time" message.

4.2 System Analysis

Evaluation metrics: ROUGE (Recall-Oriented Understanding for GistingEvaluation)

- ROUGE is a **recall-oriented measure** that works by comparing the number of machine- generated words that are a part of the reference sentence with respect to the total numberof words in the reference sentence.
- ROUGE-N measures the number of matching 'n-grams' between our model-generatedtext and a 'reference'.
- The N represents the n-gram that we are using.

ROUGE-N Recall

$$\frac{\sum s \in \{ReferenceSummaries\} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum s \in \{ReferenceSummaries\} \sum_{gram_n \in S} Count(gram_n)}$$

ROUGE-N Precision

$$\frac{number\ of\ n-grams\ found\ in\ the\ model\ and\ reference}{number\ of\ n-grams\ in\ model}$$

ROUGE-N F1 Score

$$2*\frac{precision * recall}{precision+recall}$$

Evaluation metrics: BLEU(Bilingual Evaluation Understudy)

- This metric is basically calculated by comparing the number of machine-generated words that are a part of the reference sentence with respect to the total number of words in the machine-generated output.
- This metric has precision values between [0,1], where 1 represents a perfect match and 0 represents a complete mismatch.

$$BLEU = BP.exp\left(\sum_{n=1}^{N} w_n log p_n\right)$$

4.3 System Designing/Modeling

4.3.1 System Architecture

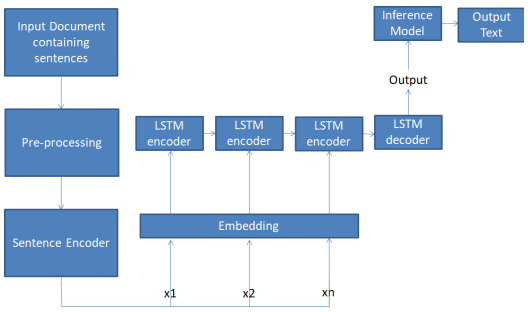


Fig. No. 4.3.1

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, May 2022
Impact Factor: 6.252

4.3.2 Modeling Designing /UML Diagram

Use Case Diagram

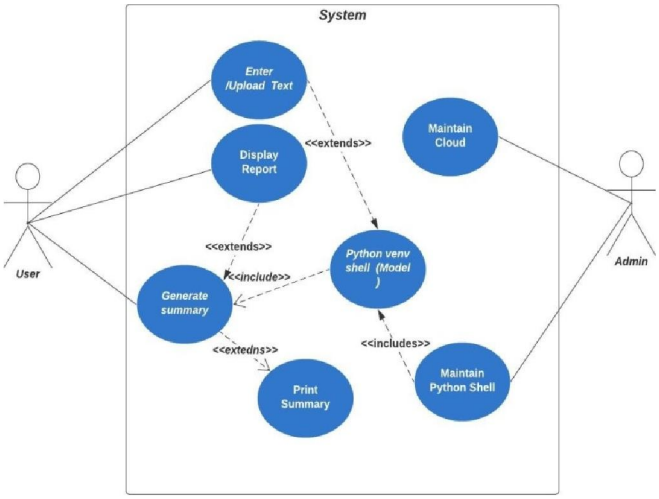


Fig. No. 4.3.2.1

Class Diagram

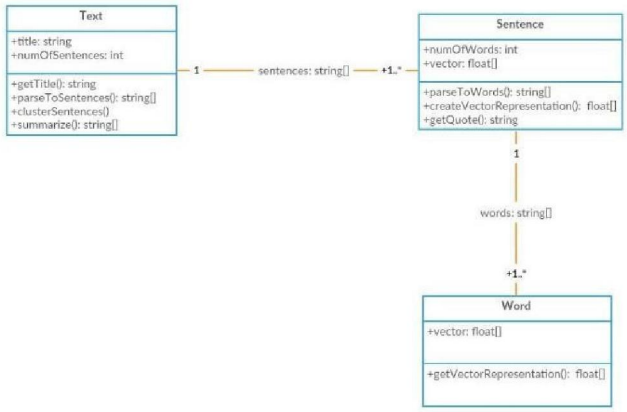


Fig. No. 4.3.2.2

Activity Diagram

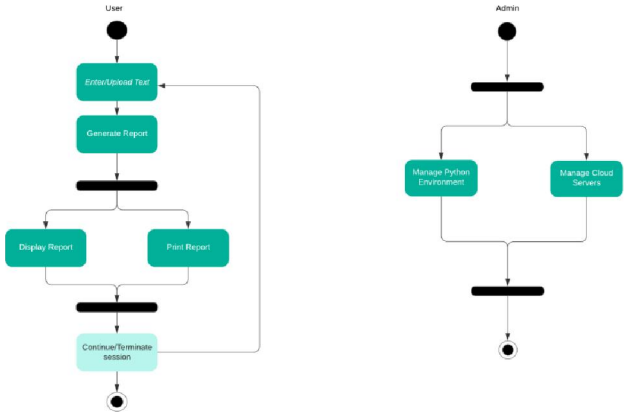


Fig. No. 4.3.2.3

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, May 2022
Impact Factor: 6.252

Data Flow Diagram

DFD 0

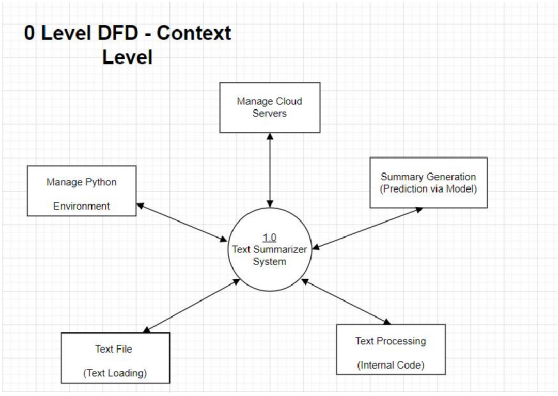


Fig. No. 4.3.2.4

DFD 1

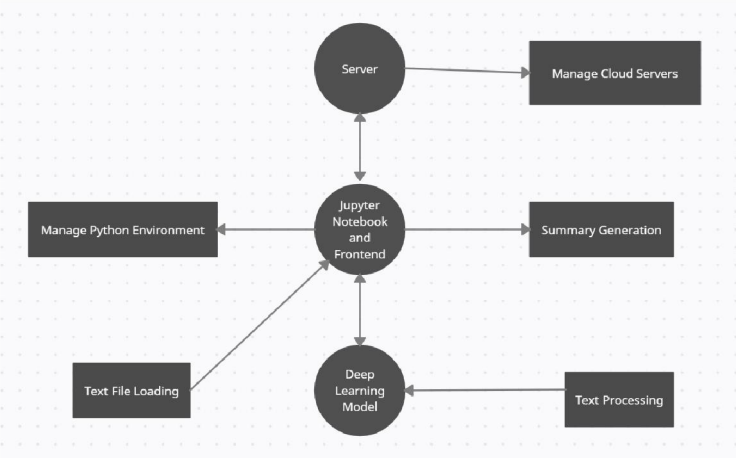


Fig. No. 4.3.2.5

V. METHODOLOGY

Text Summarization Using an Encoder-Decoder Sequence-to-Sequence Model

- Step 1 - Importing the Dataset
- Step 2 - Cleaning the Data
- Step 3 - Determining the Maximum Permissible Sequence Lengths
- Step 4 - Selecting Plausible Texts and Summaries
- Step 5 - Tokenizing the Text
- Step 6 - Removing Empty Text and Summaries
- Step 7 - Creating the Model
- Step 8 - Training the Model
- Step 9 - Generating Predictions

We feed in our input (text from news articles) to the Encoder unit. Encoder reads the input sequence and summarizes the information in the form of internal state vectors (in case of LSTM these are called the hidden state and cell state vectors). The encoder generates the context vector, which gets passed to the decoder unit as input. The outputs generated by the encoder are discarded and only the context vector is passed over to the decoder.

The decoder unit generates an output sequence based on the context vector.

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

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, May 2022
Impact Factor: 6.252

- Training phase
- Inference phase

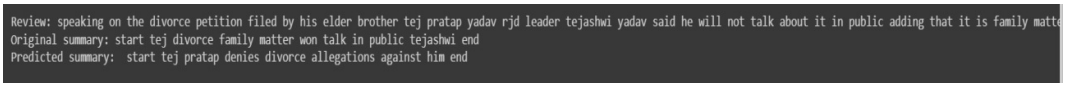
Training phase: In the training phase, we will first set up the encoder and decoder. We will then train the model to predict the target sequence offset by one time step.

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. In the end, the inference architecture drives us to decode a test sequence.

Encoder: The input length that the encoder accepts is equal to the maximum text length which is estimated. This is then given to an Embedding Layer of dimension {(total number of words captured in the text vocabulary) * (number of nodes in an embedding layer)}. This is followed by 3 LSTM networks wherein each layer returns the LSTM output, as well as the hidden and cell states recorded at the previous time steps.

Decoder: In the decoder, an embedding layer is defined followed by an LSTM network. The initial state of the LSTM network is the last hidden cell state taken from the encoder. The output of the LSTM is given to a dense layer wrapped in a Time Distributed layer with an attached softmax activation function.

Altogether, the model accepts encoder (text) and decoder (summary) as input and it outputs the summary. The prediction happens through predicting the upcoming word of the summary from the previous word of the summary.



V. CONCLUSION

In this paper, we present an approach to design an automatic text summarizer for English text that generates a summary by extracting sentences. The Deep learning approach deals with the abstractive text summarization for a single document based on Sequence-to-Sequence model with LSTM. We obtained results that were measured by calculating ROUGE and BLUE scores with respect to the actual references. We think the deficiencies currently embedded in our language model can be improved by better fine-tuning the model, deeper learning method exploration, as well as larger training dataset. The approach gives reasonably good performance. Furthermore the possibilities of the further expansion of the model to text summarization for multiple documents and for other languages can be explored in the future.

REFERENCES

[1]. Joshi, Akanksha & Fidalgo, Eduardo & Alegre, Enrique & Fernández-Robles, Laura. (2019). SummCoder: An unsupervised framework for extractive text summarization based on deep auto- encoders. Expert Systems with Applications. 129. 200-215. 10.1016/j.eswa.2019.03.045.

[2]. S. Vijay, V. Rai, S. Gupta, A. Vijayvargia and D. M. Sharma, "Extractive text summarisation in Hindi," 2017 International Conference on Asian Language Processing (IALP), 2017, pp. 318- 321, doi: 10.1109/IALP.2017.8300607.

[3]. Mr. Sarda A.T., Mrs. Kulkarni A.R. "Text Summarization using Neural Networks and Rhetorical Structure Theory" International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 6, June 2015.

[4]. Arti S. Bhoir, Archana Gulati "Multi-document Hindi Text Summarization using Fuzzy Logic Method" International Journal of Advance Foundation And Research In Science & Engineering (IJAFRSE) Volume 2, Special Issue , Vivruti 2016.

[5]. Nitika Jhatta, Ashok Kumar Bathla "A Review paper on Text Summarization of Hindi Documents" IJRCAR VOL.3 ISSUE.5 may 2015.

[6]. K. Yao, L. Zhang, D. Du, T. Luo, L. Tao and Y. Wu, "Dual Encoding for Abstractive Text Summarization," in IEEE Transactions on Cybernetics, vol. 50, no. 3, pp. 985-996, March 2020, doi: 10.1109/TCYB.2018.2876317.

[7]. Gaikwad, Deepali K and Mahender, C Namrata. A Review Paper on Text Summarization International Journal of Advanced Research in Computer and Communication Engineering, 5(3). 2016. ACM.

[8]. Chopra, S., Auli M. & Rush, A.M. (2016) Abstractive Sentence Summarization with Attentive Recurrent Neural

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 5, May 2022
Impact Factor: 6.252

Networks. 2016 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology

[9]. Tian Shi, Yaser Keneshloo, Naren Ramakrishnan, and Chandan K. Reddy. 2020. Neural Abstractive Text Summarization with Sequence-to-Sequence Models. ACM Trans. Data Sci. 1, 1, Article 1 (January 2020),35 pages. https://doi.org/10.1145/3419106