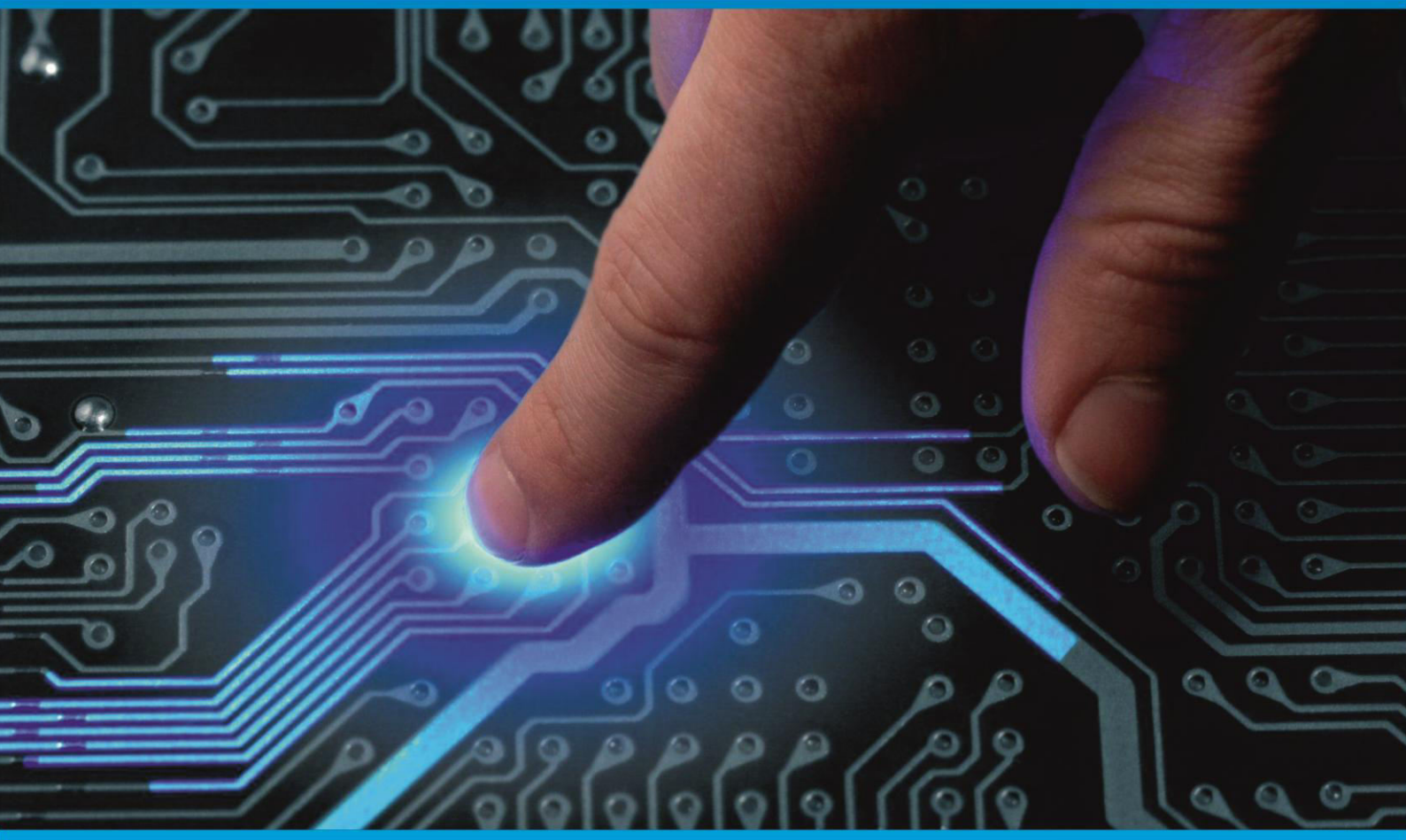
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 4, April 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.488

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904091 |

A Hybrid Approach of Text Summarization Using a Combination of Graph-based LexRank Algorithm and Text-to-Text Transfer Transformer

Rafat Ansari[1], Snehal Ghadge[2], Sweta Rajashekar[3], Prof. Vijaya Sagvekar[4]

B.E Student, Department of Information Technology, PVPPCOE, University of Mumbai, Mumbai, India[1, 2, 3]

Asst. Professor, Department of Information Technology, PVPPCOE, University of Mumbai, Mumbai, India[4]

ABSTRACT: In this paper, we present a hybrid model for single document summarization to overcome the shortcomings of both purely extractive and purely abstractive techniques of text summarization. Our model first extracts salient sentences from original text using theLexRank algorithm where each sentence is treated as a node in the graph. Next, we put these salient sentences together and get a partial condensed extractive summary of the document. Then, we use our fine-tuned T5 model to perform the abstractive task rewrite and paraphrase the sentences in the partial summary and generate the final summary. The novelty of our approach lies in combining extractive technique with abstractive and taking the advantage of both approaches. The experimental results on the News summary dataset show that the model performs well and has achieved ROUGE scores which are in competition with the existing text summarization techniques.

KEYWORDS: Automatic text summarization, Abstractive summarization, Extractive summarization, Natural language processing, LexRank, T5, Transformer, Transfer Learning.

I. INTRODUCTION

We can observe an exponential growth of textual data on the internet which can be in the form of news articles, blogs, scientific papers, medical records, legal documents, etc. People tend to read and reread these bulky amounts of information to ensure a proper understanding of what is written. This process is time-consuming and may require expertise in the context of the data to comprehend its idea within a short span of time.Nevertheless, these data are an invaluable source of information and knowledge which needs to be effectively summarized to be useful. Manually summarizing them is not only tedious but it also becomes impractical with the upsurge of information every day. There is a need for technologies that can do all the sorting and quickly identify the significant information on their own. Automatic text summarization techniques have gained momentum in recent years due to their practical applications. Automatic text summarization is a process of generating a condensed version of the given text that centralizes the idea of the overall text. It serves as a tool that help users efficiently find valuable information from large text inputs. When we as humans summarize a piece of text, we tend to read the entire text to develop our understanding, and then write a summary highlighting the main idea of the text. Since computers lack human knowledge and language capability, it makes automatic text summarization a very challenging yet fundamental task.

1.1 Classification of text summarization approaches

Depending upon the method used to generate summary, text summarization can be classified into two types:
1. Extractive summarization: In this approach, the system selects the key sentences from the input document(s) then concatenates them to form a summary.
2. Abstractive summarization: In this approach, the system forms a summary using words and phrases not present in the input document(s).

Depending upon number of documents given as input, text summarization can be classified into two types:
1. Single document summarization: In this approach, a single document is given as an input and the most useful and relevant sentences from the entire document are selected to produce a summary.
2. Multi document summarization: In this approach, more than one document is given as an input and the most useful and relevant sentences that best represent the given documents cluster are selected to produce a summary.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904091 |

Depending upon the length of the generated summary, text summarization can be classified into two types:
1. Indicative summarization: In this approach, the length of the summary is only 5% of the input document.
2. Informative summarization: In this approach, the length of the summary is 20% of the input document.

Although the summaries written by humans are generally not extractive, most researches nowadays have focused on the extractive approach. It is observed that purely extractive summarization methods often times gave better results compared to abstractive summarization [4]. This is due to the fact that abstractive summarization methods deal with problems such as limited semantic knowledge, Out-of-vocabulary (OOV) words processing and natural language generation which are relatively harder to solve than data-driven approaches like sentence extraction. On the other hand, extractive methods tend to produce redundant sentences in the summary. In this paper, we present a hybrid model for single document summarization to overcome the shortcomings of both purely extractive and purely abstractive techniques of text summarization. Our model first extracts salient sentences from the input text using the LexRank [4] algorithm. Next, we put these salient sentences together and get a condensed version of the original text. Then we use our fine-tuned transfer learning model T5 for the abstractive task that rewrites the extracted sentences and generates the final summary. We have attempted to obtain informative summary to capture the main idea of the entire text input.

II. RELATED WORK

Luhn [1] introduced the first extractive summarization method to extract the key sentences from the input text using word frequency. He proposed that any sentence with maximum occurrences of the highest frequency words are more important to the meaning of the document than the rest.

Nenkova and Vanderwende [2] proposed a multi-document summarization method called SumBasic. The algorithm first computes the probability of each word by simply counting its frequency in the document set. Each sentence is given a score as the average of the probabilities of the words in it. Sentences with higher scores are selected for the final summary.

Mihalcea and Tarau [3] presented the TextRank algorithm heavily inspired by PageRank used to rank the importance of web pages [8]. The only difference is instead of web pages here the sentences are ranked according to the similarity measures such as string kernels, cosine similarity, longest common subsequence, etc.

Günes and Dragomir [4] put forth the idea of LexRank algorithm which is based on the concept of eigen vector centrality in a graphical representation of sentences. Here, a centroidsentence is selected which works as the mean for all other sentences in the document. The sentences are then ranked according to their similarities. A sentence which is similar to many other sentences of the text has a high probability of being important.

Nallapati, Zhou, et al. [5] performed abstractive text summarization using attentional sequence-to-sequence encoder-decoder RNN that was originally developed for machine translation [9]. An encoder first encodes the source sentences as a list of fixed-length vector representations, each of which captures a word and its surrounding context. A decoder then outputs a summary based on the encoded vectors. The decoder is fed with a context vector to identify and give attention to important words/phrases in the input text.

See, Liu, et al. [6] proposed an extended encoder-decoder RNN [5] and deployed it with an additional pointer network introduced by Vinyals, et al. [10]. This model allowed both copying words from the source text via pointing and generating words from a fixed vocabulary. Pointers can be seen as an extension of attention that focuses on rare or OOV words that the attention mechanism had struggled with.

Devlin, Chang, et al [7] introduced Bidirectional Encoder Representations from Transformers (BERT) which is a Transformer-based transfer learning technique for natural language processing. It has two models: (1) $BERT_{BASE}$: It consists of 12 transformer layers along with 12 attention layers and 110 million parameters, and (2) $BERT_{LARGE}$: It consists of 24 transformer layers along with 16 attention layers and 340 million parameters. BERT is pre-trained on a large corpus of unlabelled text which include the entire Wikipedia (2,500M words) and Book Corpus (800M words). BERT's pre-trained model can be used for a variety of NLP tasks including text summarization. It has achieved state-of-the-art results in the year 2018.
Radford, Wu et al. [11] presented theGenerative Pretrained Transformer *(*GPT-2) which largely resembles their first model the GPT, with a few modifications. GPT-2 is a language model with 1.5 billion parameters, trained to predict the

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904091 |

next word on 40 GB of text.GPT-2 is based on thetransformer, which is an attention model.It learns to focus its attention on the previous token that is most relevant to predict the next word in a sentence generating a summary.
Lewis, Liu, et al. [12] proposed Bidirectional Autoencoder Representations from Transformers (BART) that make use of standard seq2seq/NMT architecture with a bidirectional encoder (like BERT) [7] and a left-to-right decoder (like GPT) [12]. It leverages denoising autoencoder to pre-train sequence-to-sequence models. BART corrupts text with an arbitrary noising function and learns to reconstruct text similar to reference summary.

Raffel, Shazeer, et al. [13] proposed a unified framework Text-to-Text Transfer Transformer (T5) that converts all NLP tasks into a text-to-text problem. Tasks such as translation, classification, summarization, and question answering. All of them are treated as a text-to-text conversion problem, rather than seen as separate distinct problems. The input and output are always text strings, in contrast to BERT models that can only output either a class label or a span of the input. T5 is an extremely large neural network model, pre-trained on huge unlabeled text dataset Colossal Clean Crawled Corpus (C4), a cleaned version of Common Crawl that is two orders of magnitude larger than Wikipedia. It has achieved state-of-the-art results on many benchmarks covering summarization, question answering, and text classification.

III. DATASET

We evaluate our model on the News summary dataset which consists of 4515 news articles and contains Author name, Headlines, Date, Article URL, Short summary and Complete text of the article. News articles have been scraped from Hindu, Indian times and The Guardian. The time period of the news articles ranges from February to August 2017.

IV. EXPERIMENTS

In this section, we briefly describe our proposed hybrid architecture for text summarization. The two fundamental elements that form the basis of our proposed method are extractor and abstractor. Figure 1 shows the outline of the proposed methodology.

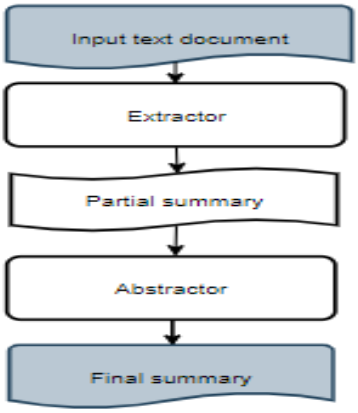


Figure 1: Outline of the proposed methodology

A. Extractor: For the purpose of extractive summarization, the extractor employs the graph-based LexRank algorithm [4].Initially, the input news article requires some text preprocessing such as tokenization (splitting the sentences), lowercasing all the words, stop words removal (removal of commonly used words in English like 'a', 'the', etc), word embedding using bag of words model to represent N-dimensional vectors where N is the number of all possible words in the English language. For each word that occurs in a sentence, the value of the corresponding dimension in the vector representation of the sentence is the number of occurrences of the word in the sentence times the IDF (inversedocument frequency) of the word. The similarity between any two sentences in the text is then defined by the IDF modified cosine similarity between the corresponding vectors of the two sentences:

$$\text{idf-modified-cosine}(x,y) = \frac{\sum_{w \in x,y} \text{tf}_{w,x}\text{tf}_{w,y}(\text{idf}_w)^2}{\sqrt{\sum_{x_i \in x}(\text{tf}_{x_i,x}\text{idf}_{x_i})^2} \times \sqrt{\sum_{y_i \in y}(\text{tf}_{y_i,y}\text{idf}_{y_i})^2}}$$

[4]

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

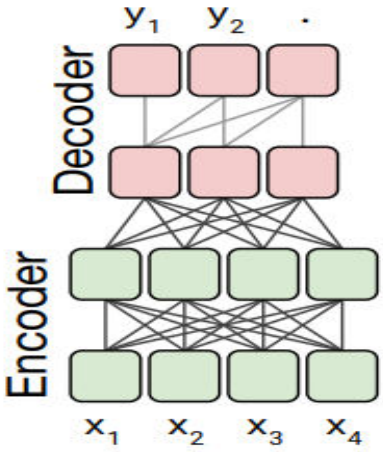
| DOI: 10.15680/IJIRCCE.2021.0904091 |

where $tf_{w,s}$is the number of occurrences of the word w in the sentence s. Based on the obtained cosine similarities, we create an adjacency matrix for a graph representation of sentences where vertices represent the sentences and edges are defined in terms of the similarity relation between pairs of sentences. Next, we compute the overall centrality value of each node using the Power method which is analogous to PageRank [8] except that it operates in an undirected graph of sentences. The Power method computes the eigen vector centrality or lexrank score of each sentence using the below formula.

$$\mathbf{p} = [d\mathbf{U} + (1-d)\mathbf{B}]^{\mathrm{T}}\mathbf{p} \quad [4]$$

where matrix B is obtained from the adjacency matrix of the similarity graph by dividing each element by the corresponding row sum, N is the total number of nodes in the graph, d is the damping factor, U is a square matrix with all elements being equal to 1/N. Extractor selects the nodes with high lexrank scores and sorts them in the same order as they appear in the text. The obtained output is a partial summary that serves as an input to the abstractor.

B. Abstractor: For abstractive summarization, we have used pre-trained Text-to-Text Transfer Transformer (T5) and fine-tuned it for summarization on news summary dataset. This type of learning is commonly referred to as transfer learning where we reuse a pre-trained model on a different problem. T5 model is essentially an encoder-decoder seq2seq transformer architecture wherein the encoder is trained in a BERT-style involving fully visible masking (i.e. every token contributes to the attention calculation of every other token in the sequence), and the decoder is trained in a GPT-style involving causal masking (i.e. every token is attended by all the tokens occurring before that token in the sequence). Figure 2 shows the T5 model architecture that we used for our fine-tuning task.



[13]

Figure 2: T5 model uses fully visible masking in the encoder and the encoder-decoder attention, with causal masking in the decoder. Dark grey lines correspond to fully-visible masking and light grey lines correspond to causal masking. Dot notation "." indicates a special end-of-sequence token that represents the end of a prediction. x and y represents the input and output sequences.

We had divided the dataset in the ratio 4:1 for training and validation respectively. The model is then trained onour modified values for hyperparameters different from the original T5 model. After training, we have saved this model to perform the abstractive task. When the abstractor receives the partial summary as the input, we first need to append a string 'summarize:' prior to the partial summary. This is done because T5 has similar formatting for the summarization process. Fine-tuned model is then applied to the partial summary to predict the final summary of our proposed architecture.

V. RESULTS

Recall Oriented Understudy for Gisting Evaluation (ROUGE) [14] is a set of metrics widely used to automatically determine the quality of a summary by comparing it to human (reference) summaries. There are several variations of ROUGE. We have used ROUGE-1, ROUGE-2 and ROUGE-L metric to evaluate the summaries of our proposed model and other existing well-known summarization methods. ROUGE-1 computes the overlap of unigrams between the system summary and reference summary. Similarly, ROUGE-2 computes the overlap of bigrams between the system

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904091 |

and reference summaries. ROUGE-L measures longest matching sequence of words using Longest Common Sequence (LCS). For more details on other variants of ROUGE refer [14].

The ROUGE-1, ROUGE-2 and ROUGE-L scores of several text summarization methods when compared with our proposed model are shown in TABLE 1. Figure 3 shows a graphical representation for the same. It can be seen that the indicators have slightly improved. TABLE 2 is an example showing the output of the model. This example is taken from the testing set. It can be seen that our summary is readable, processes OOV words aptly, generates novel words if needed and is very close to the reference summary.

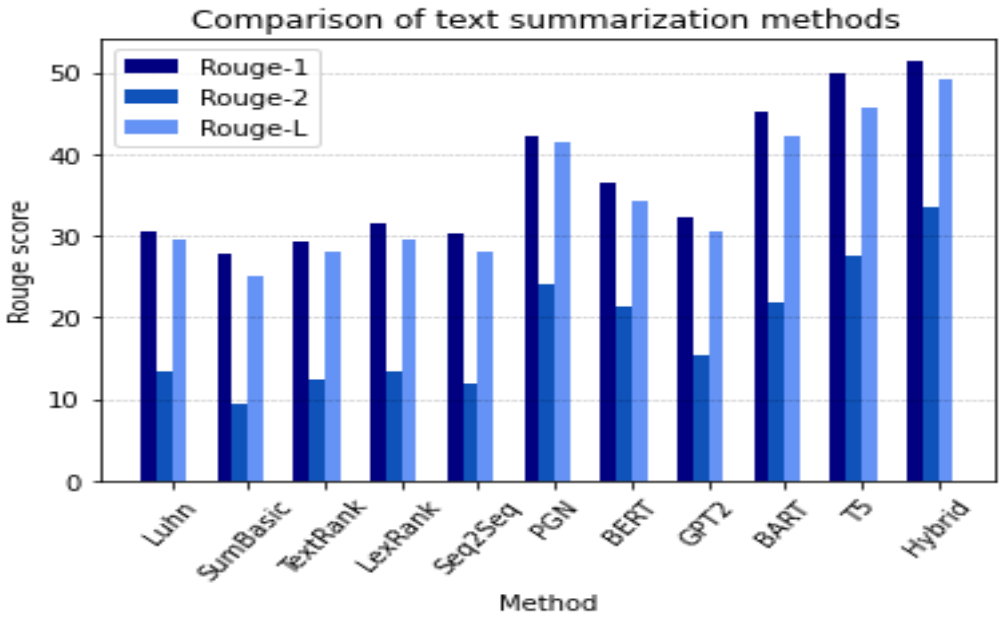


Figure 4: Bar graph displaying comparative analysis of various text summarization methods.

Method	ROUGE-1	ROUGE-2	ROUGE-L
Luhn	30.43	13.36	29.42
SumBasic	27.85	9.40	24.98
TextRank	29.34	12.34	28.02
LexRank	31.53	13.40	29.53
Sequnce-to-Sequence	30.27	11.82	28.16
Pointer Generator Network	42.11	24.15	41.35
BERT	36.37	21.28	34.17
GPT-2	32.34	15.39	30.58
BART	45.06	21.88	42.31
T5	49.82	27.43	45.68
Our hybrid model	**51.43**	**33.36**	**49.12**

*Table 1: *Result of various text summarization methods on News summary dataset. Bold text indicates significant improvement against other methods.*

*It may be noted that these results were evaluated using low computational resources. The specified methods may give better results if tested with high performance computational resources.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904091 |

Article: Mumbai, Mar 26 (PTI) Banker Uday Kotak has said the BJPs win in the recent state polls is positive for economic reforms and the government should now focus on re-activating "animal spirits" of the private sector to invest. "I think we just need to make sure that we are focused on growing the economy, getting the private sectors animal spirits back and that will create jobs," Kotak told PTI over the weekend. Terming BJPs landslide victory in the crucial state of Uttar Pradesh as a positive development, Kotak said it is a "big signal" for stability and continuity in economic policies. A two-thirds majority in the countrys most populous state, that sends the maximum number of Parliamentarians (80 to the Lower House and 30 to the Upper House, where Opposition Congress is in majority), will help BJP gain upper hand after the 2018 elections to the Rajya Sabha. The Narendra Modi government has been facing stiff opposition in getting crucial bills passed in the Rajya Sabha due to lack of adequate numbers. The BJP and allies have won 324 seats in the 403- member UP assembly. It can be noted that the private sector, struggling with over capacity amid tepid consumer demand on one hand and record high debt on the other, has been keeping away from capex (capital expenditure) since 2012.
Reference summary: Kotak Mahindra's Uday Kotak has said BJP's win in the state polls is positive for economic reforms, and the government should focus on re-activating "animal spirits" of the private sector to invest. "We just need to make sure that we are focused on growing the economy, getting the private sector's animal spirits back, and that will create jobs," Kotak said.
Our model summary: BJP's win in the recent state polls is positive for economic reforms, Banker Uday Kotak has said. "I think we just need to make sure that we are focused on growing the economy, getting the private sectors animal spirits back and that will create jobs," he added. Notably, BJP and its allies have won 324 seats in the 403-member UP assembly.

Table 2: An example from the proposed model

VI. CONCLUSION

In this work, we have described our hybrid model for text summarization. The model first extracts the salient sentences from the input text using the LexRank algorithm and later, the extracted sentences are condensed using our fine-tuned T5 Transformer. We applied our model to a news summary dataset and achieved results in competition with other state-of-the-art text summarization techniques. Combined use of extractive and abstractive methods resulted in the production of good quality summary.There are few limitations of the model which can be improved in future works. The model is trained and tested on a single dataset. In future works, we plan to evaluate our hybrid model on other text datasets. It exhibits low abstractive abilities, such as producing some novel words when required but attaining higher levels of abstraction still remains an open research question.

ACKNOWLEDGEMENT

Authors expresses their sincere gratitude to their Prof. Vijaya Sagvekar for her continuous guidance, feedback and support throughout the course of the project.

REFERENCES

[1] Luhn, H.P., 1958. The Automatic Creation of Literature Abstracts. In IBM Journal of Research and Development, Vol. 2, No. 2, pp. 159-165, April 1958.
[2] Nenkova, A. and L. Vanderwende. 2005. The impact of frequency on summarization. MSR-TR-2005-101.
[3] Mihalcea, R., &Tarau, P. 2004. Textrank: Bringing order into texts. In Lin, D., & Wu, D. (Eds.), Proceedings of EMNLP 2004, pp. 404–411 Barcelona, Spain. Association for Computational Linguistics.
[4]Günes Erkan and Dragomir R Radev. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. (JAIR) 22, 1 (2004), 457–479.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 4, April 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0904091 |

[5] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, C¸ aglarGulc¸ehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In Computational Natural Language Learning.
[6] Abigail See, Peter J. Liu, and Christopher D. Manning 2017. Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368.
[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova 2019. BERT: Pretraining of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
[8] Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd, 1998. The PageRank Citation Ranking: Bringing Order to the Web.
[9] Bahdanau, Kyunghyun Cho, YoshuaBengio. 2014. Neural machine translation by jointly learning to align and translate. ICLR.
[10] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In Neural Information Processing Systems.
[11] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.
[12] Mike Lewis, Yinhan Liu, Naman Goyal, MarjanGhazvininejad, Abdelrahman Mohamed, Omer Levy, VesStoyanov, Luke Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461, 2019.
[13] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu, 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Google, Mountain View, CA 94043, USA.
[14] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Text Summarization Branches Out: Proceedings of the ACL-04 Workshop. 74– 81.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor:
7.488

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर
NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details