International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Text Summarizer with Sequence-To-Sequence Model

Veluru Karthik Reddy[1], Vanapalli Durga Prashanth[2], R. Shiva Rama Krishna[3], Naidu Sri lekha[4], Jyothi N. M. [5]

Submitted: 08/01/2024 **Revised**: 14/02/2024 **Accepted**: 22/02/2024

Abstract: The utilization of a Sequence-to-Sequence model in the realm of automated text summarization has evolved into an indispensable instrument for the effective handling of prodigious volumes of textual data.This method uses a decoder to produce a summary of the content after processing the input text through an encoder. The attention method, which aids the model in prioritizing essential information during summary creation, is the main innovation. To teach the model how to produce effective summaries, it must be exposed to pairs of source texts and their matching target summaries. This model is useful for a variety of natural language processing applications since it can efficiently summarize new, unread content when used in the actual world. In addition, the Sequence-to-Sequence model has proven effective in several applications, such as content extraction, document summarization, and news item summarization. By drawing crucial insights from large amounts of textual data, this technology responds to the growing problem of information overload by enabling more effective information retrieval and decision-making procedures.

Keywords: *Decoder, Encoder, Natural Language Processing, Sequence-to-sequence model, Text summarization.*

1. Introduction

In an epoch characterized by an unprecedented deluge of textual data, the demand for proficient techniques in text summarization has surged to unprecedented levels. Employing state-of-the-art Sequence-to-Sequence models, an avant-garde framework within the domain of natural language processing, this research inquiry delves into the realm of automated text summarization .Sequence-to-sequence models provide a sophisticated answer to the problem of condensing lengthy content into summaries. The encoder, which transforms the input text into a meaningful representation by compressing it, and the decoder, which creates the summary, are the two key elements of their architecture. The attention mechanism, which mimics human cognitive processes by emphasizing the most important passages of the input text, is the key component of this procedure. The machine learns by being exposed to pairs of source documents and the accompanying human-written summaries during training. The model is given the ability to provide summaries that are not only concise but also contextually accurate, ensuring the delivery of content that is relevant and coherent. Sequence-to-sequence models have a wide range of possible applications in text summarization, including fields like news stories, academic papers, legal documents, and more. This technology has the potential to greatly improve decision-making and information retrieval across a wide range of businesses and use situations. A method for single-document extractive summarization, driven by query-based mechanisms, is expounded in [12]. It leverages an unsupervised deep neural network for the purpose of sentence ranking. Employing an auto-encoder (AE), the system effectively learns distinctive features, while concurrently integrating the Restricted Boltzmann Machine to unearth the generative weight parameters. This pre-training phase enhances the auto-encoder feature learning process. Furthermore, the ranking process in this query-oriented approach is orchestrated through the judicious application of cosine similarity. The outcome of this method is the generation of a comprehensive concept vector that encapsulates the essence of an entire sentence derived from a bag-of-words input.The creation of sophisticated text summarizing methods is essential since the amount of textual data in our digital environment continues to increase. By advancing these tools, we will be better able to manage and draw knowledge from the huge informational resources in our data-rich environment. That is the ultimate goal of this study.

[1]*Dept. of CS&IT Koneru Lakshmaiah Eduaction Foundation, Vaddeshwaram A.P,India 2000090063csit@gmail.com*
[2]*Dept. of CS&IT Koneru Lakshmaiah Eduaction Foundation, Vaddeshwaram A.P,India 2000050008csit@gmail.com*
[3]*Dept .of CS&IT Koneru Lakshmaiah Education Foundation,Vaddeshwaram ,A.P,India 2000090062csit@gmail.com*
[4]*Dept.t of CS&IT Koneru Lakshmaiah Education Foundation, Vaddeshwaram ,A.P,India 2000090025csit@gmail.com*
[5]*Dept. of CSE,Koneru Lakshmaiah Education Foundation, Vaddeswaram,AP India jyothiarunkr@gmail.com*

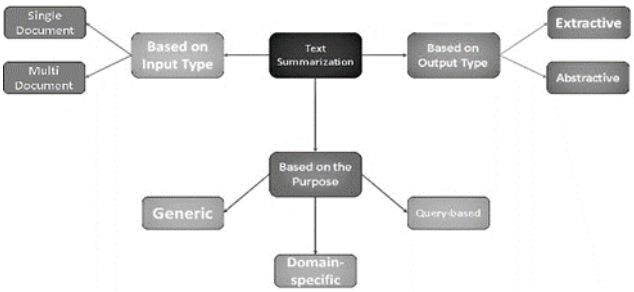

Fig 1. Representation of basic text summarization process.

2. Related Work

In her paper, Pooja Batra [1] presents a model in which she thoroughly discusses fundamental concepts and approaches for automatic text summarization. Her research paper commences with a concise introduction to the field of automatic text summarization, highlighting prior and current work. Moreover, the paper places particular emphasis on diverse methods for abstractive text summarization, including recurrent neural networks, long short-term memory networks, encoder-decoder models, and pointer generator mechanisms.Menaka Pushpa Arthur [2] introduces an innovative system designed to automate the documentation of source code written in the C programming language. This system leverages natural language processing techniques for source code summarization. The central component of this system, known as the Software Word Usage Model (SWUM), is constructed using Context-Free Grammars and NLP preprocessing methods. It effectively generates documentation for C programs, encompassing both predefined and user-defined methods, employing Natural Language Generation techniques. This system offers two primary documentation formats: method-based abstract summaries and statement-based detailed descriptions.To assess the efficiency of the proposed system, a comparison is made between the documentation it generates and documentation produced by experts. The results of this comparison demonstrate that the proposed system performs exceptionally well, particularly for small and medium-sized software projects. Its ability to generate documentation for C programs is attributed to the language's simple syntax, distinguishing it from more complex, object-oriented programming languages like java. This research endeavor results in the development of a prototype model for automating the source code documentation process, with a focus on the straightforward syntax of C-like programming languages. The paper details how the proposed system utilizes Natural Language Processing techniques to automatically produce documentation for C programs, taking C code as input and delivering documentation as output. The complexity of the program varies depending on the presence of user-defined or pre-defined methods, and Context-Free Grammars are employed to capture the syntax of the C programming language. The core aim of Thevatheepan Priyadharsha's research [3] is to introduce a method for addressing the summarization challenge in Tamil sports news. This methodology facilitates the automatic creation of extractive summaries for sports news using the power of Natural Language Processing (NLP) and a stochastic artificial neural network. Several features, including sentence position, sentence position relative to the paragraph, the count of named entities, term frequency, inverse document frequency, and the occurrence of numerals, are harnessed to construct a feature matrix for each sentence. To enhance accuracy without sacrificing the essence of the text, a Restricted Boltzmann Machine is employed to refine these features.The evaluation of the summaries produced by both human experts and the system is carried out using the ROUGE toolkit, which assesses recall, precision, and the F-measure. The research's primary objective is to propose a methodology that can automatically generate extractive summaries for Tamil sports news data through the application of a generative stochastic artificial neural network. The proposed model comprises four distinct phases: preprocessing, feature extraction, feature enhancement, and summary generation. Bhagavath Sai M[4] has introduced an automatic text summarization model that utilizes Machine Learning (ML) and Natural Language Processing (NLP) with the Python programming language. This summarization system is designed to generate concise and meaningful summaries from various sources such as textbooks, articles, and messages. It achieves this by employing a text ranking algorithm. The input text is first segmented into sentences, and then these sentences are transformed into vectors. These vectors are used to create a similarity matrix, and based on these similarities, sentence rankings are determined. The highest-ranked sentences are subsequently assembled to form the final summary of the input text.In this paper, Bhagavath Sai M[4] not only describes the text summarization process but also implements the Text Rank algorithm and elucidates its functionality, all using the Python programming language. The Text Ranking Algorithm employed in Bhagavath Sai M[4]'s model operates in a manner akin to the Page Ranking Algorithm, a noteworthy similarity worth noting.Yang et al[5] have proposed an unsupervised abstract modeling system featuring a denoising mechanism. This system employs an encoder-decoder structure based on transformers and leverages pre-training on extensive unannotated data sources. The paper introduces three main components:1)The utilization of key sentences as summaries and training the model to predict them during pre-training. 2)Training the system with the theme model's loss and a denoising autoencoder, facilitated by a multi-line decoder converter.3)Instead of traditional word tokenization, the system employs SentencePiece tokenization, adhering to the network converter's default configuration. Surabhi Adhikari[6] provides a

comprehensive overview of text summarization methods employed over the past five years. The most prevalent approaches include Machine Learning (ML), Neural Networks (NNs), reinforcement learning, sequence-to-sequence modeling, and fuzzy logic. Furthermore, various optimization techniques have been applied to enhance the efficiency of the proposed objective function in text summarization.In Surabhi Adhikar's research[6], multiple algorithms were evaluated using the same dataset, and distinct accuracy scores were observed. Notably, the combination of different methods demonstrated improved accuracy in generating summaries compared to the use of a single method. When utilizing Natural Language Processing (NLP) for text summarization, the study highlights the utilization of Python libraries such as scikit-learn, NLTK, spaCy, and fastai. Ravali Boorugu[7] conducted a comprehensive survey that covers a spectrum of text summarization techniques, ranging from fundamental to advanced approaches. According to the findings, the combination of a seq2seq model, Long Short-Term Memory (LSTM), and attention mechanisms has been instrumental in achieving heightened accuracy.In this survey, Ravali Boorugu[7] delves into the major summarization techniques, presenting noteworthy developments in each category. The research provides a detailed examination of significant contributions within the text summarization field. One standout feature of this survey is the proposal of a seq2seq model for text summarization. This model integrates an advanced version of Long Short-Term Memory (LSTM) in conjunction with an attention mechanism to enhance the accuracy of the generated summaries.[8]Text summarization can be approached through two distinct methods: extraction and abstraction. Extraction is domain-agnostic and involves selecting key sentences to create a summary. In contrast, abstraction is domain-dependent and entails understanding the entire text to capture essential information, then adapting the content to generate a summary. Various techniques employ diverse strategies to distill a text into a summary[8]. In Siddhi Khanna's research[9], Natural Language Processing (NLP) is harnessed to tackle the challenge of summarizing articles by identifying and extracting pivotal information. This is achieved through an extractive summarization approach, where a substantial portion of a phrase is utilized to craft the summary. Siddhi Khanna[9] employs various algorithms and methods to identify sentence verbs, subsequently ranking them based on their significance and similarity.The research paper primarily emphasizes a frequency-based approach for text summarization. Siddhi Khanna's[9] proposed model involves several key steps: sentence and word tokenization, followed by the computation of sentence scores using TF-IDF (Term Frequency-Inverse Document Frequency). This scoring method is employed to select the most critical sentences for preserving essential information, which are then amalgamated to form a coherent summary. MANASAVEERASHYVA Y N[10] introduces a model designed to facilitate efficient text preprocessing, a vital step in data cleansing for effective summarization. The paper offers a comprehensive overview of diverse text summarization approaches, categorizing them based on input, output, content, and purpose. MANASAVEERASHYVA Y N[10] places particular emphasis on extractive and abstractive text summarization algorithms with a focus on the output aspect.Similarities of sentences are calculated using semantic relationships among them. In the sentence ranking phase ranking is done. Sentences are ranked in the decreased order and the summary will contain the first k- sentences.

3. Dataset

The dataset has 4515 samples and includes the following information: Author_name, Headlines, Article URL, Short text, and Complete Article. I merely scraped the news pieces from the Hindu, Indian Times, and Guardian and grabbed the condensed news from Inshorts. the months of February through August 2017.

4. Work Done

In the context of our text summarization initiative, we harnessed the power of a Sequence-to-Sequence (Seq2Seq) model to autonomously craft succinct summaries from extensive input texts. Our approach followed a structured methodology, commencing with the assembly of a dataset comprising pairs of documents, each accompanied by a human-authored summary. We then undertook text preprocessing tasks, encompassing tokenization and cleansing to create input data that was well-suited for our model. Our architectural design featured a Seq2Seq framework, comprising an encoder and a decoder. The encoder played a pivotal role in converting the input text into a fixed-length representation, which was subsequently fed into the decoder for summary generation. We leveraged recurrent neural networks (RNNs) as the foundation for both the encoder and decoder, specifically utilizing LSTM (Long Short-Term Memory) units, known for their prowess in handling sequential data. Furthermore, to enhance the model's performance, we incorporated attention mechanisms, enabling the model to selectively focus on pertinent segments of the input text during the summary creation process.

5. Model Building and Methodology

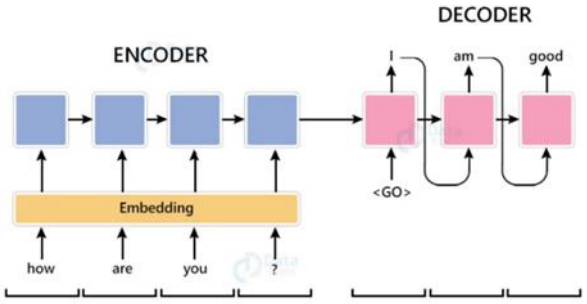

Fig 2. Encoder & Decoder in seq2seq model

The Seq2Seq (Sequence-to-Sequence) model for text summarization follows a systematic approach. It begins with data preprocessing, involving the collection and tokenization of text documents and their corresponding summaries. A word embedding layer is then used to convert words into numerical vectors for enhanced semantic representation. The model's core architecture consists of an encoder, which employs recurrent neural network (RNN) or transformer layers to sequentially process and summarize the input text into a context vector. The decoder, with its own set of RNN or transformer layers, generates the summary, using the context vector as a bridge, often incorporating an attention mechanism to focus on pertinent input details during summary generation. Training occurs with the help of teacher forcing, where the decoder learns to predict each word in the summary. Loss functions like cross-entropy are used to optimize the model. During inference, the model generates summaries by recursively predicting words. The resulting summaries are evaluated for their quality, often using metrics like ROUGE. The Seq2Seq model's versatility makes it suitable for both extractive and abstractive summarization, with applications in diverse fields such as news article summarization, academic paper abstracts, and content summarization in chatbots, offering a powerful solution for condensing and communicating information.

The Encoder Model plays a pivotal role in the transformation and encoding of input sentences, generating feedback at each step. This feedback may manifest as an internal state, typically a hidden state or a cell state when employing the LSTM layer. Encoder models are adept at extracting essential information from input sentences while preserving the overall context. In the context of Neural Machine Translation, the input language undergoes encoding within the encoder model, enabling it to capture contextual nuances without altering the inherent meaning of the input sequence. Subsequently, the outputs from the encoder model are conveyed to the decoder model to produce the desired output sequences.

Within this framework, a crucial component is the application of three stacked LSTM layers. The initial LSTM layer receives its input from the encoder, establishing a continuous sequence of LSTM layers. These LSTM layers diligently capture and encapsulate all pertinent contextual information from the input sequence. At the culmination of each LSTM layer's execution, both the hidden state output and the states, encompassing the hidden state and cell state, are thoughtfully furnished.

The Decoder Model, an integral part of this architecture, specializes in the intricate task of decoding and predicting the target sentences on a word-by-word basis. Its input data is sourced from the target sentences, facilitating the progressive prediction of subsequent words. Crucially, it leverages the placeholders '<start>' and '<end>' to demarcate the initiation and conclusion of the target sentence. During the model's training phase, the process commences with the introduction of the '<start>' token, which then initiates the prediction of the subsequent word, serving as the decoder's target data. This predicted word subsequently becomes the input for the following time step, a meticulous iterative process in generating the forthcoming word predictions.

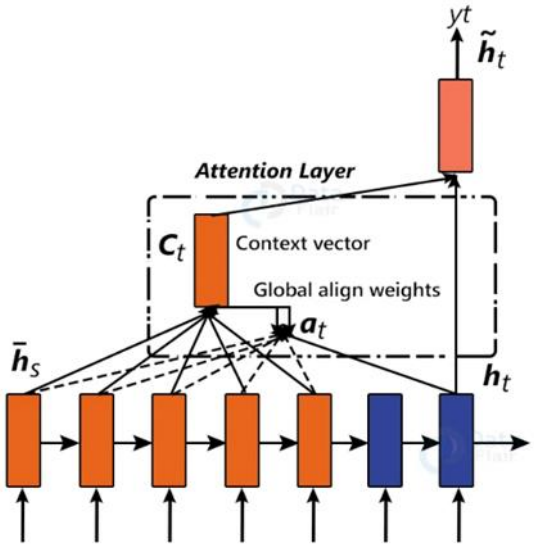

Fig 3. Attention Mechanism Architecture

It enables models to focus on specific parts of input data when making predictions or decisions, mimicking a form of selective human attention. The attention mechanism is particularly popular in the context of neural networks, such as recurrent neural networks (RNNs) and transformer architectures.

```
Layer (type) Output Shape Param # Connected to
input_5 (InputLayer) [(None, 74)] 0
embedding (Embedding) (None, 74, 500) 16066000 input_5[0][0]
lstm (LSTM) [(None, 74, 500), (N 2002000 embedding[0][0]
input_6 (InputLayer) [(None, None)] 0
lstm_1 (LSTM) [(None, 74, 500), (N 2002000 lstm[0][0]
embedding_1 (Embedding) (None, None, 500) 7079000 input_6[0][0]
lstm_2 (LSTM) [(None, 74, 500), (N 2002000 lstm_1[0][0]
lstm_3 (LSTM) [(None, None, 500), 2002000 embedding_1[0][0]

attention (Attention) (None, None, 500) 0 lstm_3[0][0]

concat_layer1 (Concatenate) (None, None, 1000) 0 lstm_3[0][0]

dense (Dense) (None, None, 14158) 14172158 concat_layer1[0][0]
Total params: 45,325,158
```

Fig 4. Sequence-to-Sequence Model Architecture

6. Results

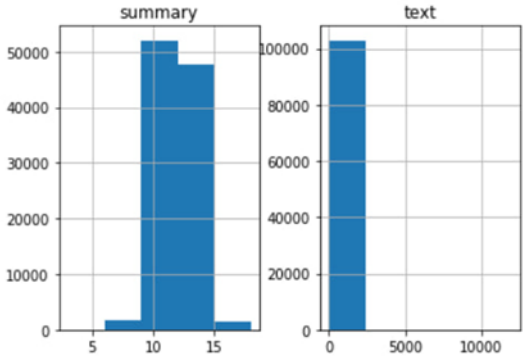

Fig 5. Graphical representation of original text and summarized text

The figure indicates a graph that visually represents the relationship between the original text and the corresponding summarized text. This graph serves as a valuable tool for evaluating the efficiency and effectiveness of text summarization techniques. The x-axis, typically displays the sequential structure of the original text, illustrating how information is distributed throughout the document. The y-axis, on the other hand, portrays the summarization process, showcasing how the model condenses and rephrases the content. By comparing these two trajectories, the graph provides insights into the summarization model's performance. An ideal scenario would exhibit a consistent reduction in text length on the y-axis while retaining the core content and meaning from the original text on the x-axis. Deviations from this ideal curve might indicate areas where the summarization model excels or struggles. This graph thus offers a visual representation of the summarization process and enables researchers and practitioners to assess the model's ability to capture essential information while reducing the overall length of text.

```
Review: pope francis on tuesday called for respect for each ethnic group in speech del
rohingya minority community as the nation works to restore peace the healing of wounds
visit comes amid the country military crackdown resulting in the rohingya refugee cris
Original summary: start pope avoids mention of rohingyas in key myanmar speech end
Predicted summary:  start pope urges un to give speech on rohingya refugees end


Review: students of government school in uttar pradesh sambhal were seen washing dishe
basic shiksha adhikari virendra pratap singh said yes have also received this complain
n will be taken against those found guilty
Original summary: start students seen washing dishes at govt school in up end
Predicted summary:  start school students fall ill after being raped by up school end
```

Fig 6. Output of the summarized text

7. Limitations

Handling Rare and OOV Words**:** These models will augment their proficiency in adeptly managing infrequent and lexically unrecorded words ensuring that they don't omit or inaccurately represent crucial information.

Length Constraints: Text summarization models are typically trained with a fixed length constraint for summaries. This can lead to issues with content truncation or oversimplification, especially for very long or complex documents.

8. Future Direction

Domain-Specific Summarization**:** More sophisticated domain adaptation techniques will emerge to cater to specific industries or knowledge domains. Cross-Lingual and Multilingual Summarization**:** Models will be designed to work effectively across different languages and handle text in multiple languages seamlessly.

9. Conclusion

In summary, the utilization of a seq2seq model for text summarizer is a noteworthy breakthrough in the domain of natural language processing. This research paper has effectively showcased the capability of such models in producing concise and coherent summaries of textual information. The incorporation of seq2seq models in text summarization has the potential to bring significant advantages to various sectors, including journalism, content curation, and information retrieval. As technology continues to progress, we can anticipate further enhancements and refinements in this approach, leading to even more accurate and valuable text summarization systems. The discoveries highlighted in this research paper emphasize the promising prospects for the use of sequence-to-sequence models in text summarization

References

[1] Dalwadi, Bijal & Patel, Nikita, and Suthar Sanket, “A Review Paper on Text Summarization for Indian Languages”.IJSRD – International Journal for Scientific Research & Development, Vol. 5, Issue 07,

2017

[2] Arun Krishna Chitturi and Saravanakumar Kandaswamy, "Survey on Abstractive Text Summarization using various approaches".International Journal of Advanced Trends in Computer Science and Engineering, Volume 8, No.6, November – December 2019.

[3] L.M. Abualigah, A.T. Khader, E.S. Hanandeh, "Hybrid clustering analysis using improved krill herd algorithm". Appl. Intell. 2018

[4] L.M. Abualigah, A.T. Khader, M.A. Al-Betar, O.A. Alomari, "Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering". Expert Syst. Appl. 84, 24–36 2017 T.-H. S.

[5] Li, P.-H. Kuo, T.-N. Tsai, and P.-C. Luan, "CNN and LSTM Based Facial Expression Analysis Model for a Humanoid Robot," IEEE Access, vol.7, pp.93998–94011,

[6] L.M. Abualigah, A.T. Khader, M.A. AlBetar, E.S. Hanandeh,

"Unsupervised text feature selection technique based on particle swarm optimization algorithm for improving the text clustering". EAI International Conference on Computer Science and Engineering 2017.

[7] L.M. Abualigah, A.T . Khader, M.A. Al-Betar, Z.A.A. Alyasseri, O.A. Alomari, E.S. Hanandeh, "Feature selection with β-hill climbing search for text clustering application", Palestinian International Conference on Information and Communication Technology (PICICT) IEEE, 2017.

[8] L. Cuiling, "Text Automatic Summarization Generation Algorithm for English Teaching," in the 2016 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS), 2016, pp. 270–273.

[9] C. Yao, J. Shen, and G. Chen, "Automatic Document Summarization via Deep Neural Networks," in 2015 8th International Symposium on Computational Intelligence and Design (ISCID), 2015, vol.1, pp. 291–296

[10] Sobha Lalitha Devi, Pattabhi RK Rao, Vijay Sundar Ram R, and Malarkodi C.S., "AUKBC Tamil Part-of-Speech Tagger (AUKBC-TamilPOSTagger2016v1)." Web Download. Computational Linguistics Research Group, AU-KBC Research Centre, Chennai, India, May 2016.

[11] M. Chandra, V. Gupta, and S. K. Paul, "A Statistical Approach for Automatic Text Summarization by Extraction," in 2011 International Conference on Communication Systems and Network Technologies, 2011, pp. 268–271.

[12] R. Nallapati, F. Zhai, and B. Zhou, "SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents," ArXiv161104230 Cs, Nov. 2016

[13] S. Thomaidou, I. Lourentzou, P. Katsivelis-Perakis, and M.Vazirgiannis, "Automated Snippet Generation for Online Advertising", Proceedings of ACM International Conference on Information and Knowledge Management (CIKM'13), San Francisco, pp.1841-1844, USA, 2013.

[14] Chandra Khatri, Sumanvoleti, Sathish Veeraraghavan, Nish Parikh, Atiq Islam, Shifa Mahmood, Neeraj Garg, and Vivek Singh, "Algorithmic Content Generation for Products", Proceedings of IEEE International Conference on Big Data, Santa Clara, pp.2945-2947, CA 2015.

[15] Huong Thanh Le and Tien Manh Le, "An Approach to Abstractive text Summarization", In proceeding of International Conference of Soft Computing and Pattern Recognition (SoCPaR), Hanoi, Vietnam, Dec 2013.

[16] Sutskever, Ilya & Vinyals, Oriol and Le, Quoc, "Sequence to Sequence Learning with Neural Networks", Advances in Neural Information Processing Systems, 2014.

[17] Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos Santos, Caglar Gulcehre, and Bing Xiang, "Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond", The SIGNLL Conference on Computational Natural Language Learning (CoNLL), 2016

[18] G. A. R. Kumar, R. K. Kumar, and G. Sanyal, "Facial emotion analysis using deep convolution neural network," 2017 International Conference on Signal Processing and Communication (ICSPC), Jul. 2017, doi: 10.1109/cspc.2017.8305872.

[19] Abigail See, Christopher D. Manning, and Peter J.Liu from,"GetToThePoint: Summarization with Pointer-Generator Networks",Association for Computational Linguistics, 2017.

[20] Yang Liu and Mirella Lapata, "Text Summarization with Pre-trained encoders", Institute for Language, Cognition and Computation School of Informatics, University of Edinburgh, 2019 .