2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)
Amity University, Noida, India. Sep 3-4, 2021

2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) | 978-1-6654-1703-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRITO51393.2021.9596500

# A Review on Abstractive Text Summarization Using Deep Learning

Abdul Ghafoor Etemad
*Sharda University,*
Greater Noida, India
ghafooretemad3@gmail.com

Dr. Ali Imam Abidi
*Sharda University,*
Greater Noida, India
ali.abidi@sharda.ac.in

Megha Chhabra
*Sharda University,*
Greater Noida, India
megha.chhabra@sharda.ac.in

***Abstract*— The task of creating a summary from a somewhat longer text is known as text summarization. This summary is comprised of a few sentences and would still express the main idea of the text. It has applications in sentiment analysis, document summarization, search engine queries, and business analysis etc. Over time, a lot of research has happened on the topic of abstract text summarization, especially with the emergence of deep learning. Many new Recurrent Neural Network and Convolutional Neural Network models have been proposed by the researchers. The performance of the models based on the Recurrent Neural Network and Convolutional Neural Network has been satisfactory. Models based on the pre-trained Transformer have recently demonstrated state-of-the-art performance in the field of Natural Language Processing in recent research. The amount of work done in the field is quite extensive and challenging to be brought together, this review attempts to do the same.**

***Keywords*— Abstract Text Summarization, Extract Text Summarization, Recurrent Neural Network (RNN), Sequence-to-Sequence, Convolutional Neural Network(CNN), Deep Learning, Attention Mechanism, Longest Short Term Memory(LSTM), Transformer.***

## I. Introduction

With the advent of millions of websites, blogs and social media platforms data volume is growing at an inexplicable rate. Dealing with this ever-surmounting heap of data presents many challenges and one among them is text summarization. Text summarization has many applications in different internet based domains, for example search engines are used for making queries, e-commerce websites using sentiment analysis to gauge customer feedback regarding specific products and so on. [1], [3], [11] Text summarization is the task of constructing a summary with a few sentences from a large corpus, such that the summary should consist of the main idea of the original corpus. Summarizing a large text is still a problem, generally, Extractive and Abstractive are two categories of text summarization task, where in Extractive summarization; the summarizer finds the important words and sentences in the original text then copies those without modification to generate the summary.

In Abstractive summarization; the summary is generated by a summarizer itself, it means the words, phrases, and sentences are not extracted from the source, the summarizer generates new words, phrases, and sentences for generating the summary. In extractive text summarization, the summary is not much readable because the sentences are just copied from the source document, the summary is incoherent or may have grammatical mistakes, but in abstractive text summarization the phrases are generated by the summarizer, and summary is concise, readable, semantically, and syntactically correct. Abstract text summarization needs more data than extractive to train the model, for example headline of the news can be the summary and body of the news can be the text. [1]. According to source, summarization can be divided into two categories: multi document and single document summarization, in single document the input is only one document both the extractive and abstractive summarization techniques can be applied. Multi document summarization is more complicated in this approach as more than one document is fed to the model to generate a novel, readable, coherent, and concise summary. [9] Summarization is a mapping task which maps an input sequence to the output sequence. For the task of mapping a deep learning model called sequence-to-sequence (seq-to-seq) models have been successful in many applications like: Text Summarization, Video Captioning, Machine Translation and Speech Recognition [6], [7]. Extractive Text Summarization is the oldest approach of summarization, It is used to extract and summarize the key semantic information in the source text. Before the emergence of pre-trained models and transformer, two special types of neural network (Recurrent Neural Network and Convolutional Neural Network) were used with sequence to sequence model for the task of summarization. [8]. Sequence-to-sequence framework with RNN and CNN produced the state-of-the-art performance in machine translation which is a part of Natural Language Processing.

However, text summarization is different from machine translation in machine translation - the input sequence and output sequence are almost the same length, but in abstract text summarization the length in the output is lesser than input, we compress the main idea of the input sequence in Lossy Manner, where in machine translation its lossless. [10] [12] [17],[18] [25]. Pre-trained models which train on the large unlabeled datasets brought a revolution in the all-NLP tasks by fine-tuning it on the downstream tasks [24].

The remainder of this document is structured as follows: In section two, related works in the field of abstract text summarization is mentioned, in section three comparison table and conclusion is discussed.

## II. Literature Survey

### *A. RNN-based Models*

Rush et al. [13] proposed a fully data-driven method for abstract text summarizing, using a local attention-based model to create each word of the summary conditioned on the input sequence. In this study, an attentional feed-forward neural network was utilized to construct the summary and a convolutional model was employed to encode the original text. Chopra et al. [15]. In the extension of this work used RNN for the decoder which showed a better performance for the both dataset Gigaword and DUC. He also equipped the model with a convolutional attention-based encoder. Nallapati et al. [10] employed a state-of-the-art attentional encoder-decoder recurrent neural network to perform on two different corpora. In this research many novel models are proposed that addresses many critical problems of text summarization that are not adequately modeled by basic models such as capturing the hierarchy of the sentence-to-word structure, modeling keywords by feature rich encoder and switch generator-pointer for words that are rarely unseen in the training set. The base of this model is a neural machine translation which is used in Bahdanau et al. [3]. The model consists of a bidirectional encoder with GRU-RNN [5] and unidirectional decoder with GRU-RNN that has the same hidden state size of the decoder. An attention mechanism is used on the source-hidden-state and softmax layer. For generating the word summary this model is equipped with a 'switch', the switch will decide whether to copy the word from the source or generate the word by the generator for summary. [10] For more details refer to the original paper [10]. See et al. [32] proposed a new pointer-generator network that implicitly combines abstraction and extraction. This architecture can copy the words from source by copying mechanism and generates the words using generators.

### *B. Beyond RNN-based Models*

Sequence to sequence models have reached state-of-the-art performance in several Natural Language tasks, such as machine translation, natural language generation, and so on. by the emergence of Longest Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). The standard RNN has many drawbacks like gradient vanishing, and exploding, long term dependencies, parallelization and computational constrained, however, the problem of gradient vanishing has been solved by LSTM but still the other drawbacks like gradient exploding, parallelization and handling long-term dependence has not resolved. Recently researchers found that the Convolutional Neural Network (CNN) can resolve most of the drawbacks of the RNN based models, CNN can overcome the problem of parallelization and computational complexity, computational complexity of CNN is linear to the length of data. Song et al. [14] proposed a LSTM-CNN model for text summarization. This model overcame several problems in the field of text summarization, extract text summarization models concerned syntactical structure and abstract text summarization concerned with semantics, this model improved both summarization models. The essential phrases from the original text are extracted using the MOSP approach, and then the collocation of the phrases is learned. The model is put to the test on two separate datasets, and the results reveal that it outperforms the current state-of-the-art technique in both syntactic and semantic aspects. Convolutional sequence to sequence model is proposed by Zhang et al. [17], in this model multi-layer CNN is stacked over each other to address the traditional CNN problem which encodes fixed size context [14] - a copying mechanism for dealing with rare words and decreasing the softmax layer. Also, in this novel model position embedding for the input and output, a hierarchical word and sentence level attention is used. The GAN (Generative Adversarial Network) is made up of two elements (Generator and Discriminator), the generator part generates the summary from the given input X = {x1, x2, ……….. xi} to the Y = {y1, y2, y3, …yn} based on the policy G. The Discriminator part validates the generated summary with the human generated summary and computes the differences. The overall network tries to generate a summary that cannot be differentiated from the human generated summary. In another research, [1] developed a new model using deep learning and semantic data transformation for abstract text summarization. The novelty of this model is the combination of deep learning with semantic based data transformation. In the most recent research [8] proposed a new model, this model works based on the strategy of human-like reading.

### *C. Pre-trained and Transformer based Models*

Vaswani et al. [30] introduced Transformer as a basis model for most of today's state-of-the-art NLP models. Transformer has a self-attention mechanism to focus on specific parts of the input sequence, as well as an encoder-decoder architecture with six encoder layers and the same number of decoder layers. Refer to the original study for more information. Devlin et al. [20] proposed BERT which is a deep bidirectional model that is trained on large amounts of unlabeled text and can be fine-tuned on the task-specific without architecture changes. Zihang Dai. et al [21] introduced a model that solves the drawbacks that exist in the [Ashish Vaswani et al. Attention is All you need] transformer. Transformers encode the fixed-length content which causes fragmentation error and long-term dependence handling issues. Transformer XL uses segment recurrence to overcome the problem of long-term dependence. It uses the computed hidden state values of the previous hidden state instead of calculating from scratch, it helps to have a memory from the past hidden state that prevents the fragmentation error. Also this model uses relative positional encoding. BART is a denoising autoencoder and autoregressive decoder model proposed by Mike Lewis et al. [22]. It is a transformer-based model which randomly corrupts a part of the sentence by a denoising function, similar to the BERT [20] and a learning model to predict [reconstruct the original text] the corrupted tokens like GPT with modification of ReLU to GeLUs. Hoang et al. [23] proposed a model based on GPT and provided two solutions for efficiently adapting pre-trained transformer for text summarization; source embedding: which adds a source embedding to input representation to encode the token type so that the model can identify whether this token belongs to input sequence or output summary, and domain adaptive training: This helps the model to grasp the overall structure and distribution of languages before being fine-tuned on the text summarizing task. Zhang et al. [19] proposed a text summarization model that encodes the input sequence using BERT [20] and in the decoder the models initially generate a draft summary using transformer encoder and then the model randomly masks the drafted summary and feeds it to the BERT for predicting the masked tokens and the final summary is generated by combination of both drafted summaries. XLNET is a combination of both auto-regressive and auto-encoding methods while avoiding their limitations. Instead of employing a predetermined forward or backward factorization order, XLNET maximizes the anticipated log probability of a sequence for all possible permutations. [27]. Kaiqiang Song et al. [28] proposed a neural text summarization model that uses a controlling over copying mechanism that controls the amount of copy rate in the summary. This model generates a range of hypothesis summaries, the summaries with high copy rates are more extractive and summaries with lower-level summaries are more abstractive. RoBERTa is a replication study of BERT proposed by Yinhan Liu [22], the author has changed the key hyper-parameters of the BERT model like: training with longer sequence, increasing the batch size, dynamic masking, longer training process, removing next sentence prediction, and increasing the dataset size. Pre-trained Sequence-to-Sequence and Saliency Models is proposed by Saito et al. [2] which is based on RoBERTa [22] with an additional attached saliency model that identifies important parts of the text and feeds it as additional input to the encoder. In this research, Gunel et al. [29] extended the transform encoder-decoder. In order to capture the entity level knowledge, the model trained on WikiData for including the fact that exists in the source document on the summary. This model also uses transform XL for handling long sequences. Raffel et al. [26] proposed a general framework which is based on the transform that produces state of the art performance on many NLP tasks like: text summarization, question-answering, machine translation etc. The basic idea behind this work is to produce a text-to-text framework that takes a text as input and produces a new text as output - basically it treats all the text processing problems as text-to-text problems. In this work Raffel et al. [26] has not proposed a new method - instead the research shows where the field stands, also in this work "Colossal Clean Crawled Corpus" [C4] which is hundreds of gigabytes clean English dataset is produced.

### 1) *Sequence-to-sequence framework*

Machine translation, text summarization, image captioning, and sentiment analysis are among natural language processing applications that use the sequence to sequence architecture. In the recent research sequence to sequence framework uses deep neural networks and have two components: an encoder that takes a sequence of data as input and the output of an encoder fed to the decoder part to generate the final output. [17] For long-term dependencies, Recurrent Neural Network (RNN) and its variation Long Short Term Memory (LSTM) are commonly used to build this architecture. [16] One word is provided to the encoder at each timestamp, and encoders then send the input through an embedding layer to convert it to a distributed representation. Then it's paired with a multilayer neural network that generates hidden states based on the previous input. Decoder takes the output of encoder as input and then transforms it to a distributed representation by passing through an embedding layer then using a softmax layer it generates the summary. [10], [14] Figure 3 gives an overall semantic of sequence-to-sequence framework in with the encoder firstly handles the input sequence x=x1, x2, x, x3, ……, xm where m represents the number of input elements with corresponding hidden state z = z1, z2, z3, …., zm. The decoder uses z as input to generate the output y=y1, y2, y3, …., yn one by one from left to right. While producing the output yi+1 the decoder produces new hidden state hi+1 via the previous hidden state hi with representation vector gi from the previous target word language ti and conditional input ci based on the input z.

Based on this formulation, many sequence to sequence models have novelties on RNN types. [17], [16]

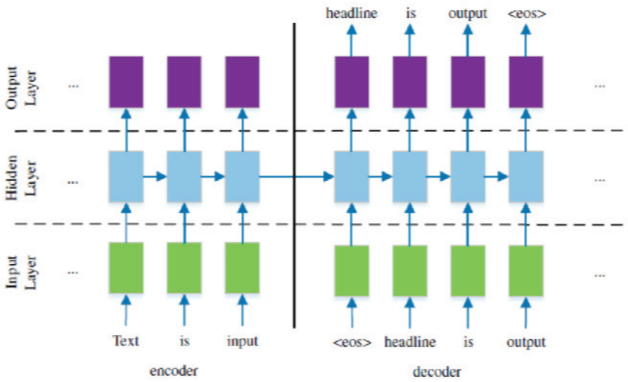

Fig. 1. Sequence to Sequence Architecture [17]

### 2) *Transformer*

Vaswani et al. [30] presented Transformer, which is essentially an encoder decoder architecture with a multi-head attention mechanism. For many natural language processing jobs, this design provided state-of-the-art performance. Transformer basically has Two component from high level of view:

*a) Encoder:* Self-attention layer and feed-forward layer are two sub-layers in each encoder. First of all, the transformer encoder encodes the input text into the vector using the embedding algorithms and then applies positional encoding in order to maintain the sequence of tokens. Then the input passes through the self-attention layer, in this layer, the attention for each individual word is calculated using the key, query, and value vector by dot products of randomly initialized vectors. After calculating the attention of each word, all the attention vectors are concatenated in order to calculate the final attention vector. Output of the attention layer passes through the feed-forward layer. This process continues in the six layers of encoder, then the output of the last encoder passes to the decoder part.

*b) Decoder:* Each decoder contains three sub-layers: encoder-decoder attention, feed-forward, and multi-head attention layer. Decoder takes the input of the last encoder and applies encoder-decoder attention (self-attention) and produces an output. Decoder also uses multi-head attention for focusing on appropriate parts of the sentences. This process repeats until the special token (End of Sentence) is reached, then the decoder feeds each step on the bottom most part of the decoder for generating the final output. Transformer contains 6 layers of encoder and 6 layers of decoder in its original structure.

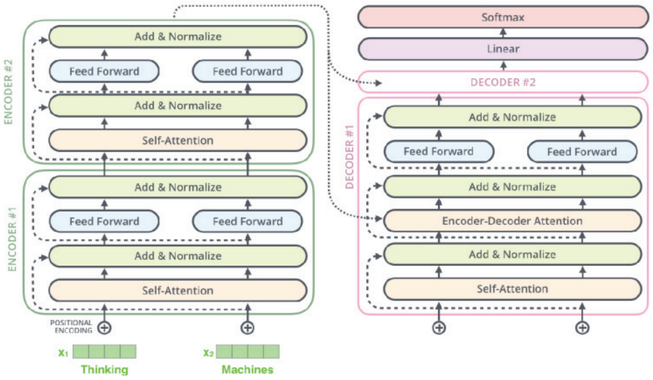

Fig. 2. Transformer architecture [30]

## Evaluation Metrics

When we defined abstractive and extractive summarization, we used a score function. However, we did not define what it is. In fact, it is quite hard to define it. One way to define it is, given a model S is by using the log probability of a summary given a document and a model:

*Equation 1: Generating summary*

$$S(x,y)=\log(p(y|x;\theta))\approx\sum\log(p(y_{(i+1)}|x,y[0:i],\theta))$$

where y [0: i] are all previous outputs before output i. Here, we make the assumption that a word in the summary is only determined by all previous words and the input text. What was done in traditional summarization was to separate the probability into two parts by using Bayes' rule:

*Equation 2: Generating summary*

$$argmax(log\ log\ (p(x))\ ) = argmax(log\ (p(y)p(x|y)))$$

where p(y) is the language model and p(x|y) is the summarization model. In this work, we will approximate p(y|x) directly with our neural network. We have just defined a score function given a model. However, the way summaries are evaluated in the general case is not clear. For summarization, we can use the measurements which are used in machine translation: BLEU and ROUGE scores. BLEU (Bilingual Evaluation Understudy) is a precision oriented score. It is defined as follows:

*Equation 3: Calculating BLEU evaluation metric*

$$BLEU = \frac{Number\ of\ words\ in\ the\ summary\ which\ are\ in\ gold\ standard}{Total\ number\ of\ words\ in\ the\ summary}$$

The recall side of BLEU is called ROUGE (for Recall-Oriented Understudy for Gisting Evaluation). There are different versions of ROUGE. The direct BLEU equivalent is ROUGE-1 and counts unigram overlaps:

*Equation 4: ROUGE evaluation metric*

$$ROUGE-1 = \frac{(\sum_{Reference\ Summary}\quad \sum_{unigram}\quad Count_{match}(unigram))}{(\sum_{Reference\ Summary}\quad \sum_{unigram}\quad Count(unigram))}$$

This definition can be generalized to N-grams:

$$ROUGE-N = \frac{(\sum_{Reference\ Summary}\quad \sum_{Ngram}\quad Count_{match}(Ngram))}{(\sum_{Reference\ Summary}\quad \sum_{Ngram}\quad Count_{Ngram}Count(unigram))}$$

There are other extensions of ROUGE. For instance, ROUGE-L takes the longest common sequence into account and Rouge-S and Rouge-SU consider skip sequences. Evaluating a summary is still a topic of research and there is no perfect consensus to do it yet. [31]

TABLE I. COMPARISON TABLE OF ABSTRACT TEXT SUMMARIZATION MODELS USING DEEP LEARNING

| Researcher | Year | Types of model | Techniques | Rouge 1 | Rouge 2 | Rouge L | Dataset |
|---|---|---|---|---|---|---|---|
| Rush et al. | 2015 | Convolutional Neural Network | Attention | 28.48 | 8.91 | 23.97 | DUC 2004 |
| Nallapati et al. | 2016 | Sequence to Sequence RNN | Attention, Switching Generator Pointer, Hierarchical attention, large vocabulary trick | 46.6<br>39.6 | 23.1<br>16.2 | 43.03<br>35.3 | DUC 2002,<br>CNN Daily Mail |
| Jonas Gehring et al. | 2017 | Convolutional Seq2Seq | GLU, Attention | 30.44<br>38.5 | 10.84<br>17.48 | 26.90<br>33.29 | DUC2004,<br>Gigaword |
| Shengli Song et al. | 2018 | LSTM-CNN | LSTM, Phrase extraction, Phrase location information | 34.9 | 17.8 |  | CNN Daily Mail |
| Yong Zhang et al. | 2019 | Convolutional Sequence to Sequence | Hierarchical attention mechanism, copying mechanism | 37.95<br>29.74<br>42.04 | 18.64<br>9.85<br>19.77 | 35.11<br>25.81<br>39.42 | Gigaword,<br>DUC,<br>CNN Daily Mail |
| Panagiotis Kouris et al. | 2019 | Deep learning with semantic data transformation | - | 46.30<br>28.73 | 23.88<br>10.07 | 43.94<br>26.12 | Gigaword,<br>DUC 2004 |
| Haoyu Zhang et al. | 2019 | Transformer with BERT | BERT, Copying Mechanism, Attention | 41.71 | 19.49 | 38.79 | CNN-Daily Mail |
| Andrew Hoang et al. | 2019 | Transformer | Domain Adaptive, Source Embedding | 39.01<br>36.73<br>40.87 | 17.87<br>14.93<br>28.59 | 36.17<br>29.66<br>37.62 | CNN Daily Mail<br>Xsum<br>Newsroom |
| Haoyu Zhang et al. | 2019 | BERT | Two Stage in Decoder layer: First Transformer Decoder for generating | 41.71 | 19.49 | 38.79 | CNN Daily Mail |

|  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
|  |  |  | draft output and second BERT for final output |  |  |  |  |
| Kaiqiang Song et al. | 2019 | Transformer | Controlling Copy Rate, Transformer | 40.89<br>45.93 | 19.11<br>24.14 | 37.60<br>42.51 | Gigaword,<br>Newsroom |
| Mike Lewis et al. | 2019 | BART | Transformer,<br>a bidirectional encoder over corrupted text and a<br>left-to-right autoregressive decoder | 44.16<br>45.14 | 21.28<br>22.27 | 40.90<br>37.25 | CNN Daily Mail<br>Xsum |
| Ming Yang,<br>Qiang Qu et al. | 2020 | Knowledge Based attention, Deep Learning, generative adversarial Network | - | 42.16<br>36.78 | 19.17<br>18.65 | 38.35<br>33.96 | CNN Daily Mail,<br>Gigaword |
| Itsumi Saito et al. | 2020 | RoBERTa Base | RoBERTa Base, saliency model | 45.80<br>45.42 | 22.53<br>22.13 | 42.48<br>36.92 | CNN Daily Mail<br>Xsum |
| Beliz Gunel et al. | 2020 | Transformer XL | Entity Level Knowledge, Transformer XL, | 34.273 | 13.018 | 32.048 | CNN Daily Mail |
| Colin Raffel et al. |  | T5 | Transformer, BERT Style | 43.52 | 21.55 | 40.69 | CNN Daily Mail |

## III. CONCLUSION

On the basis of the published papers on the field of abstract text summarization using deep learning models, all the researchers focused on overcoming the two main problems of text summarization which are syntactic and semantic. According to the performance of the different models which is mentioned in the table 1, there are two types of models that have demonstrated cutting-edge performance: From Sequence to Sequence based models Zhang et al. [17] introduced a model that is based on the Convolutional Neural Network with Hierarchical attention mechanism - this model tested three different carpus Gigaword, DUC, CNN Daily Mail; From Transformer based models Raffel et al. [26] proposed T5 model which outperformed in the most NLP tasks.

## REFERENCES

[1] "Abstractive and Extractive Text Summarization Using Document Context Vector and Recurrent Neural Networks," Chandra Khatri, G. S., 2018.

[2] "Abstractive Summarization using Combination of Pre-trained Sequence-to-Sequence and Saliency Models," by Itsumi Saito, Kyosuke Nishida, Kosuke Nishida, and Junji Tomita. 2020, arXiv preprint arXiv:2003.13028.

[3] "Neural machine translation by learning to align and translate at the same time." Yoshua Bengio, Bahdanau, Dzmitry, Kyunghyun Cho, and Dzmitry Bahdanau. 2014 arXiv preprint arXiv:1409.0473

[4] "Learning with convolutional sequences."Jonas Gehring . et al 1243-1252 in Proceedings of the International Conference on Machine Learning. 2017 – PMLR

[5] "On sequence modeling, empirical evaluation of gated recurrent neural networks."Yoshua Bengio, Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Junyoung Chung 2014 arXiv preprint arXiv:1412.3555

[6] "Abstractive text summarization based on deep learning and semantic content generalization," by Panagiotis Kouris, Georgios Alexandridis, and Andreas Stafylopatis. In Proceedings of the Association for Computational Linguistics' 57th Annual Meeting, pp. 5082-5092, 2019.

[7] "For paragraphs and documents, a hierarchical neural autoencoder."Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. Li, Jiwei, Minh-Thang Luong, and Dan Jurafsky.

[8] "Investigating a human-like reading approach for abstractive text summarization." Yang, Min .et al. 2019.

[9] "A reviewe automatic text summarization." Tas, Oguzhan .et al, 2007.

[10] "Abstractive text summarization using sequence-to-sequence rnns and beyond." Nallapati, Ramesh .et al, 2016.

[11] "Long story short–global unsupervised models for keyphrase based meeting summarization." Riedhammer, Korbinian, .et al 2010.

[12] "A deep reinforced model for abstractive summarization." Paulus, Romain .et al, 2017.

[13] "A neural attention model for abstractive sentence summarization." Rush, Alexander .et al arXiv preprint 2015.

[14] "Abstractive text summarization using LSTM-CNN based deep learning." Song, Shengli .et al, 2019.

[15] "Abstractive sentence summarization with attentive recurrent neural networks." Chopra, Sumit, Michael Auli, and Alexander M. Rush. 2016.

[16] "On the factual correctness and robustness of deep abstractive text summarization." Lux, Klaus-Michael, 2020.

[17] "Abstract text summarization with a convolutional Seq2seq model." Zhang, Yong, .et al, 2019.

[18] "Deconvolutional paragraph representation learning." Zhang, Yizhe .et al , 2017

[19] "Pretraining-based natural language generation for text summarization." Zhang, Haoyu, Jianjun Xu, and Ji Wang. , 2019.

[20] "Bert: Pre-training of deep bidirectional transformers for language understanding." Devlin, Jacob, .et al, 2018.

[21] "Transformer-xl: Attentive language models beyond a fixed-length context." Dai, Zihang .et al, 2019.

[22] "Multilingual denoising pre-training for neural machine translation." Liu, Yinhan .et al, 2020.

[23] "Efficient adaptation of pretrained transformers for abstractive summarization." Hoang, Andrew .et al, 2019.

[24] "Pretraining-based natural language generation for text summarization."Zhang, Haoyu, .et al, 2019.

[25] "Impact of Patch-Size on Classification Accuracy of Latent Fingerprint Image in Stacked Convolutional Auto-encoder based Segmentation and Detection." Chhabra, Megha, .et al, 2020.

[26] "Exploring Transfer Learning with T5: The Text-To-Text Transfer Transformer." Roberts, A., and C. Raffel, 2020.

[27] "Xlnet: Generalized autoregressive pretraining for language understanding." Yang, Zhilin, .et al, 2019.

[28] "Controlling the amount of verbatim copying in abstractive summarization." Song, Kaiqiang, et al. 2020.

[29] "Mind The Facts: Knowledge-Boosted Coherent Abstractive Text Summarization." aGunel, Beliz, , 2020.

[30] "Attention is all you need."Vaswani, Ashish, .et al , 2017.

[31] "Looking for a few good metrics: ROUGE and its evaluation." Lin, Chin-Yew, and F. J. Och. . 2004.

[32] "Get to the point: Summarization with pointer-generator networks." See, Abigail, Peter J. Liu, and Christopher D. Manning. 2017.