Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 17, no. 10, October 2021, pp. 900-906
DOI: 10.23940/ijpe.21.10.p8.900906

Fine-Tuned T5 for Abstractive Summarization

Abdul Ghafoor Etemad*, Ali Imam Abidi, and Megha Chhabra

Department of Computer Science & Engineering, School of Engineering & Technology, Sharda University, Greater Noida, 201310, India

Abstract

Abstract Text Summarization can be understood as the task of constructing a summary from a relatively larger text. This summary would comprise of only a comparatively much smaller number of sentences than the actual text and would still express the main idea. Its applications lie in sentiment analysis, document summarization, search engine queries, business analysis, etc. Over time, a lot of research has happened on the topic of abstract text summarization, especially with the emergence of pre-trained models proposed by researchers. In this research a pre-trained model was fine-tuned on Xsum and Gigaword datasets and produced state-of-the-art performance in the abstractive summarization.

Keywords: abstract text summarization; extract text summarization; convolutional neural network; recurrent neural network; sequence-to-sequence model; transformer; encoder-decoder; attention mechanism

(Submitted on May 20, 2021; Revised on August 27, 2021; Accepted on September 25, 2021)

1. Introduction

With the advent of millions of websites, blogs, and social media platforms, data volume has been only growing. Dealing with this ever-surmounting heap of data presents many challenges and one amongst them is text summarization. Text summarization has many applications in different internet-based domains. For example, search engines are used for making queries and e-commerce websites use sentiment analysis to gauge customer feedback regarding specific products and so on [1]. Text summarization is the task of generating 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, the task of summarization is divided into two (Abstractive, Extractive) categories. 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 easily 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. The summary is concise, readable, semantically, and syntactically correct. Abstract text summarization needs more data than extractive summarization to train the model. For example, a headline of the news can be the summary and the body of the news can be the text [1]. According to the source, summarization can be divided into two categories: single document and multi document summarization. In single document, the input is only one document and 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, and concise summary. 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, models have been successful in many applications such as: Machine Translation, Speech Recognition, Video Captioning, and text summarization [2]. Extractive Text Summarization is the oldest approach of summarization. It is used to mine the core semantic information in the original text and summarize it. Before the emergence of pre-trained models and transformers, two special types of neural networks (Recurrent Neural Network and Convolutional Neural Network) were used with the sequence-to-sequence model for the task of summarization [3]. The sequence-to-sequence framework with RNN and CNN produced state-of-the-art performance in machine translation which is a part of Natural Language Processing.

* Corresponding author.
E-mail address: ghafooretemad3@gmail.com

However, text summarization is different from machine translation - the input sequence and output sequence are almost the same length. However, in abstract text summarization, the length in the output is less than the input. We compress the main idea of the input sequence in Lossy Manner, where in machine translates its lossless [4-7]. Pre-trained models which train on large unlabeled datasets brought a revolution in the all-NLP tasks by fine-tuning it on the downstream tasks. In this research a pre-trained model proposed by Colin Raffel [7] is fine-tuned in the Xsum and Gigaword datasets and produces state-of-the-art performance and abstractive text summarization. Xsum is a collection of BBC articles with its summary written by the author of the article.

The rest of this paper is organized as follows: In Section 2, related works in the field of abstract text summarization is mentioned and in Section 3 the comparison table is depicted. In Section 4 the results and discussion are mentioned. Finally, the conclusion is discussed in Section 5.

2. Literature Survey

2.1. RNN-based Models

Rush et al [8] introduced a fully data-driven approach for abstract text summarization, which was a local attention-based model used to generate each word of the summary conditioned to the input sequence. In this research a convolutional model was used to encode the original text, and an attentional feed-forward neural network was used to generate the summary. Chopra et al [9] used RNN for the decoder which showed better performance for both the Gigaword dataset and DUC. They also equipped the model with a convolutional attention-based encoder. Nallapati et al. [4] used an attentional encoder-decoder recurrent neural network that showed state-of-the-art performance on two different corpora. In this research many novel models were proposed that addressed 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. [10]. The model consists of a bidirectional encoder with GRU-RNN [11] 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 [4]. Abigail See introduced a new pointer-generator network that merges the abstraction with the extraction implicitly. This architecture can copy the words from the source by copying mechanism and generating the words using generators.

2.2. Beyond RNN-based Models

With the emergence of Longest Short-Term Memory (LSTM and Gated Recurrent Unit (GRU)), sequence to sequence models have achieved state of the art performance in many Natural Language tasks like machine translation, natural language generation, and so on. The standard RNN has many drawbacks such as gradient vanishing, exploding, long term dependencies, parallelization, and being computational constrained. The problem of gradient vanishing has been solved by LSTM, but there still exists other drawbacks like gradient exploding, parallelization, and handling long-term dependence. Recently researchers found that Convolutional Neural Network [CNN] can resolve most of the drawbacks of the RNN based models. CNN can overcome the problem parallelization and computational complexity. Computational complexity of CNN is linear to the length of data. Song, Huang [12] 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. This new model uses a MOSP method for extracting the key phrases from original text and then learning the collocation of the phrases. The model is tested with two different datasets and the result shows that the model outperforms the state-of-the-art approach both in terms of syntactic and semantic aspects.

The convolutional sequence to sequence model is proposed by Yong Zhang [6]. In this model, a multi-layer CNN is stacked over each other to address the traditional CNN problem which encodes fixed size context [12] - 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. Generative Adversarial Network (GAN) consists of two parts [Generator and Discriminator], the generator part generates the summary from the given input $X = \{x_1, x_2, \ldots\ldots x_i\}$ to the $Y = \{y_1, y_2, y_3, \ldots y_n\}$ 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 abstractive text summarization. The novelty of this model is the combination of deep learning with semantic based data transformation. In the most recent research, a new model was proposed based on the strategy of human-like reading [3].

2.3. Pre-Trained and Transformer based Models

A transformer is proposed by Ashish Vaswani et al. [13], which is a base model for most of today's state of the art NLP models. The transformer uses a self-attention mechanism in order to focus on different parts of the input sequence and it uses an encoder-decoder structure. It has 6 layers of encoder and the same number of layers in the decoder. For more details, refer to the original paper. BERT is a deep bidirectional model which is trained on unlabeled text and can be fine-tuned on the specific task without architecture changes [14]. Zihang Dai et al. [15] proposed a new model that solves the drawbacks that exist in transformer by Ashish Vaswani et al. 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 auto-encoder and auto-regressive decoder model proposed by Mike Lewis et al [16]. It is a transformer-based model which randomly corrupts a part of the sentence by a denoising function, similar to BERT [14] and uses a learning model to predict [reconstruct the original text] the corrupted tokens like GPT with modification of ReLU to GeLUs. [17] proposed a model based on GPT and provided two solutions for efficiently adapting pre-trained transformers for text summarization; source embedding, which adds a source embedding to the input representation to encode the token type so that the model can identify whether this token belongs to the input sequence or output summary, and domain adaptive training; this allows the model to understand general structure and language distribution before being fine-tuned on the text summarization task directly.

Haoyu Zhang et al. [18] proposed a text summarization model that encodes the input sequence using BERT [14]. In the decoder the models initially generate a draft summary using a transformer encoder and then the model randomly masks the drafted summary and feeds it to the BERT for predicting the masked tokens. The final summary is generated by combining both drafted summaries. XLNET is a combination of both auto-regressive and auto-encoding methods while avoiding their limitation. XLNET maximizes the expected log likelihood of a sequence with respect to all possible permutations instead of using fixed forward or backward factorization order. [19]. Kaiqiang Song [20] 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 hypotheses 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 [16]. The author has changed the key hyper-parameters of the BERT model as such: training with longer sequence, increasing the batch size, dynamic masking, longer training process, removing next sentence prediction, and increasing the dataset size.

The Pre-trained Sequence-to-Sequence and Saliency Models is proposed by Saito which is based on RoBERTa [16] 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, Beliz Gunel [21] extended the transform encoder-decoder. In order to capture the entity level knowledge, the model is trained on Wiki-data for including the facts that exist in the source document on the summary. This model also uses transform XL for handling long sequences. Colin Raffel et al. [7] 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 [7] has not proposed a new method - instead the research shows where the field stands. Also in this work is the “Colossal Clean Crawled Corpus” [C4], which produced hundreds of gigabytes of a clean English dataset.

3. Methodology

3.1. Model Architecture

The transformer was proposed as an attention mechanisms-based network architecture by Ashish Vaswani et al. [13] at Google Brains/Research which is only an encoder-decoder architecture attached with a multi-head attention mechanism. This architecture produced state-of-the-art performance on the most natural language processing tasks.
The transformer basically has two components from a high level of view:

Encoder: Each encoder contains two sub-layers: self-attention layer and feed-forward layer.
First of all, 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 is passed through the self-attention layer. In this layer the attention for each individual word gets calculated using the key, query, and value vector by dot products of

randomly initialized vectors. After calculating attention of each word, all the attention vectors concatenate in order to calculate the final attention vector. The output of the attention layer is passed through the feed-forward layer. This process continues in the six layers of the encoder, then the output of the last encoder passes to the decoder part.

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. The decoder also uses a multi-head attention for focusing on appropriate part 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. The transformer contains 6 layers of encoder and 6 layers of decoder on its original structure as shown in Figure 1.

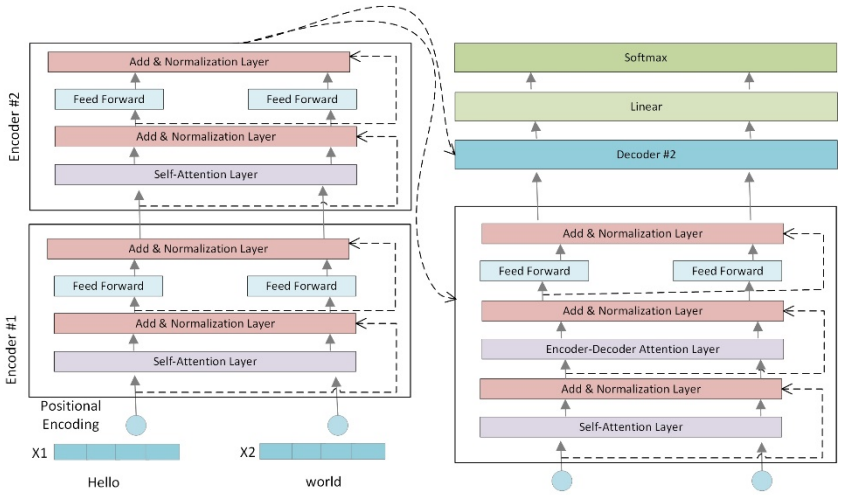

Figure 1. The original layered structure of Transformer

3.2. Base Model

Text to Text Transfer Transformer (T5) is the base model for this experiment which is proposed by Colin Raffel [7]. T5 is a general transformer-based text to text framework that produced state-of-the-art performance on many NLP tasks such as text summarization, question-answering, machine translation, etc.

The basic idea behind this framework is to convert all the NLP tasks to as text-to-text problem. This framework accepts text as input and produces a new text as output as shown in the Figure 2 below.

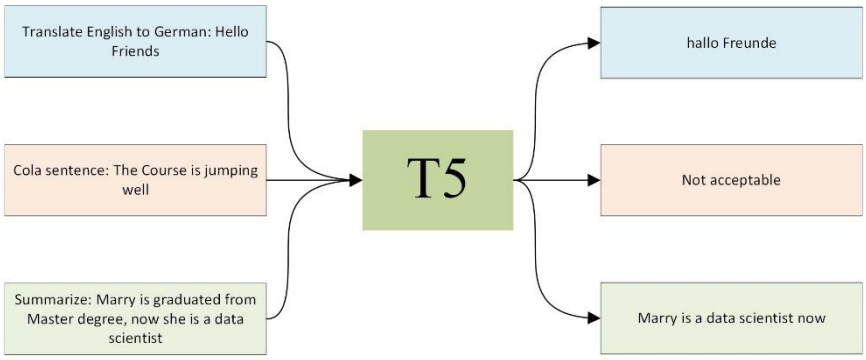

Figure 2. The T5 Transformer Framework

3.3. The Proposed Model

Models are pre-trained on large datasets in an unsupervised manner. They are then fine-tuned for downstream tasks which produced state-of-the-art performance in NLP and other tasks. In this experiment the advantage of pre-trained models has been utilized. Since this model is a fine-tuned pre-trained model (T5), all the parameters of the base model are copied. Then, the model is trained on the Xsum dataset for abstractive text summarization.

Model Hyper-parameters: Since the model is a fine-tuned model, the hyper-parameters must be the same to the original model. The following hyper-parameters were set to the model: vocabulary size to 32128, size of encoder layer to 512, size of Key, Query, and Value vectors per attention head to 64, number of encoder layers to 6, number of decoder layers to 6, number of heads to 8, dropout rate to 0.1, and activation function to ReLU.

3.4. Evaluation Metrics

When abstractive and extractive summarization was defined, a score function was used but not defined. One way of doing it would have been given a model S and using the log probability of a summary given a document and a model as shown in equation (1). where $y[0:i]$ are all previous outputs before output i.

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

Here, the assumption was made 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 as in equation (2) where $p(y)$ is the language model and $p(x|y)$ is the corresponding summarization model.

$$argmax\left(log\big(p(x)\big)\right) = argmax\big(log\ log\ \big(p(y)p(x \vee y)\big)\big) \tag{2}$$

In this work, $p(y|x)$ will be approximated directly with our neural network. A score function has just been defined given a model. The way summaries are evaluated, however 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 shown in equation (3)

$$BLEU = \frac{Number\ of\ words \in the\ summary\ which\ are \in gold\ standard}{Total\ number\ of\ words \in the\ summary} \tag{3}$$

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. It is shown in equation (4).

$$ROUGE - 1 = \frac{(\Sigma_{Reference\ Summary}\ \ \Sigma_{unigram}\ \ Count_{match}(unigram))}{(\Sigma_{Reference\ Summary}\ \ \Sigma_{unigram}\ \ Count(unigram))} \tag{4}$$

This definition can be generalized to N-grams as in equation (5)

$$ROUGE - N = \frac{(\Sigma_{Reference\ Summary}\ \ \Sigma_{Ngram}\ \ count_{match}(Ngram))}{(\Sigma_{Reference\ Summary}\ \ \Sigma_{Ngram}\ \ count_{Ngram}count(Ngram))} \tag{5}$$

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 [22].

3.5. Dataset

Improvement of Text Summarization models are dependent on multiple elements and one of those effective elements is the training, validation, and testing dataset. In this research the Xsum dataset is used to fine-tune the pre-trained model T5, which is proposed by Colin Raffel [7]. The result of the training is shown on the Table 1. Xsum is a collection of the BBC article with its summary generated by the author of the articles. Xsum contains 204045 articles for the training set, 11333 articles for testing and the same number of articles for the validation set. In this experiment the model is trained and tested on the whole dataset. Gigaword is a collection of English articles with its summaries that contain 3 million articles. In this experiment T5 is fine-tuned on 100000 articles and tested in 1900 articles due to limitation of the resources.

4. Result & Discussion

The following table (Table 1) is the result of the fine-tuned T5 model on the Gigaword and Xsum dataset.

Table 1. Experiment result on Gigaword and Xsum dataset

Model	Rouge1	Rouge 2	Rouge L	Rouge LSUM	Dataset
BART-RXF	40.45	20.69	36.56		Gigaword
ProphetnMet	39.51	20.42	36.69		
ERNIE-GENLARGE	39.46	20.34	36.74		
FT5	**43.02**	14.50	**37.43**	**37.49**	
FT5	30.91	5.26	20.85	20.85	Xsum

As shown in the above Table 1, the FT5 model produced state-of-the-art performance in ROUGE 1 and ROUGE L in the Gigaword dataset. Since the summaries of the Xsum dataset are longer than the model output size, it produced a low ROUGE score. However, the summaries generated by the model are very coherent and are more similar to the human generated summary.

4.1. Generated Summary

The Generated Summary from the Xsum dataset by the model is shown in the table below (Table 2). The generated summary is more readable, concise, and more similar to the human generated summary.

Table 2. Proposed model generated summary from an Xsum dataset sample

Article	Summary	Dataset
Samsung said: "Shipments of the Galaxy Note 7 are being temporarily delayed for additional quality assurance inspections." There are reports in South Korea and the US of the Galaxy Note 7 "exploding" either during or just after charging. However, it is unclear whether the delay is because of these reports. Pictures and videos shared online depict charred and burnt handsets. Shares fell as much as 3.5% during trade in Seoul before making a partial recovery to close 2% down on the day. Sister company Samsung SDI told Reuters that while it was a supplier of Galaxy Note 7 batteries, it had received no information to suggest the batteries were faulty. A YouTube user who says they live in the US uploaded a video of a Galaxy Note 7 with burnt rubber casing and damaged screen under the name Ariel Gonzalez on 29 August. He said the handset "caught fire" shortly after he unplugged the official Samsung charger, less than a fortnight after purchasing it. "I came home after work, put it to charge for a little bit before I had class, went to put it on my waist and it caught fire," he said.He added that while he was unharmed, his carpet was burnt in the incident. At least five other claims of phones "exploding" had been made by 24 August, according to the Korean news agency Yonhap News. Further images of a burnt Galaxy Note 7 were uploaded to Kakao Story, a popular social media site in Korea, on 30 August.A user wrote: "There was another explosion of the Galaxy Note 7. It was my friend's phone. A Samsung employee checked the site and he is currently in talks over the compensation with Samsung. You should use its original charger just in case and leave the phone far away from where you are while charging." The post has since been deleted, according to Business Korea Rival Apple is due to hold an event on 7 September, where it is expected to announce its latest iPhone. "The timing could not be worse for Samsung," said Roberta Cozza, research director at Gartner. "Samsung was back on track with its premium phones after the Galaxy S7 earlier this year. If it plans on issuing a recall, it will have to be done quickly, as such issues can be very damaging. "The Galaxy Note 7 was very well received when it was launched earlier this month, so this is a delicate moment."	"The timing could not be worse for Samsung," said Gartner analyst Roberta Cozza. "Samsung was back on track with its premium phones after the Galaxy S7 earlier this year." There are reports in South Korea and the US of the Galaxy Note 7 "exploding" either during or just after charging. Samsung said: "Shipments of the Galaxy Note 7 are being temporarily delayed for additional quality assurance inspections."	Xsum

5. Conclusions

Abstract Text Summarization is one of the most challenging and important tasks in Natural Language Processing (NLP). In this research, a T5 model which was proposed by Google researchers has been fine-tuned on the Xsum and Gigaword dataset. The fine-tuned model produced state-of-the-art performance in the Gigaword dataset and produced human level summaries in the Xsum dataset.

References

1. Khatri, C., Singh, G., and Parikh, N. Abstractive and extractive text summarization using document context vector and recurrent neural networks. *arXiv preprint arXiv:1807.08000*, 2018.

2. Kouris, P., Alexandridis, G., and Stafylopatis, A. Abstractive text summarization based on deep learning and semantic content generalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp.5082-5092, 2019.
3. Yang, M., Qu, Q., Tu, W., Shen, Y., Zhao, Z., and Chen, X. Exploring human-like reading strategy for abstractive text summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1), pp.7362-7369, 2019.
4. Nallapati, R., Zhou, B., Gulcehre, C., and Xiang, B. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.
5. Paulus, R., Xiong, C., and Socher, R. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.
6. Zhang, Y., Li, D., Wang, Y., Fang, Y., and Xiao, W. Abstract text summarization with a convolutional Seq2seq model. *Applied Sciences*, 9(8), pp.1665, 2019.
7. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.
8. Rush, A.M., Chopra, S., and Weston, J. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.
9. Chopra, S., Auli, M., and Rush, A.M. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp.93-98, 2016.
10. Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
11. Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
12. Song, S., Huang, H., and Ruan, T. Abstractive text summarization using LSTM-CNN based deep learning. *Multimedia Tools and Applications*, 78(1), pp.857-875, 2019.
13. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp.5998-6008, 2017.
14. Devlin, J., Chang, M.W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
15. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., and Salakhutdinov, R. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.
16. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.
17. Hoang, A., Bosselut, A., Celikyilmaz, A., and Choi, Y. Efficient adaptation of pretrained transformers for abstractive summarization. *arXiv preprint arXiv:1906.00138*, 2019.
18. Zhang, H., Xu, J., and Wang, J. Pretraining-based natural language generation for text summarization. *arXiv preprint arXiv:1902.09243*, 2019.
19. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., and Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.
20. Song, K., Wang, B., Feng, Z., Liu, R., and Liu, F. Controlling the amount of verbatim copying in abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(5), pp.8902-8909, 2020.
21. Gunel, B., Zhu, C., Zeng, M., and Huang, X. Mind the facts: Knowledge-boosted coherent abstractive text summarization. *arXiv preprint arXiv:2006.15435*, 2020.
22. Lin, C.Y. and Och, F.J. Looking for a few good metrics: ROUGE and its evaluation. In *Ntcir Workshop*, 2004.

Abdul Ghafoor Etemad is an aspiring scholar who is pursuing a master's degree from Department of Computer Science & Engineering, School of Engineering & Technology, Sharda University, Greater Noida, India. His research interests include Data Summarization etc.

Ali Imam Abidi is an Assistant Professor at the Department of Computer Science & Engineering, School of Engineering & Technology, Sharda University, Greater Noida, India. His research interests include Computer Vision, Feature Data Analysis etc.

Megha Chhabra is Assistant Professor at the Department of Computer Science & Engineering, School of Engineering & Technology, Sharda University, Greater Noida, India. Her research interests include Machine Learning, Image Forensics etc.