Comparative Study of Abstractive Text Summarization Techniques

Aryan Ringshia[1], Neil Desai[2], Umang Jhunjhunwala[3], Ved Mistry[4], Prachi Tawde[5]

[1,2,3,4] Student, Dept. of Information Technology, Dwarkadas J. Sanghvi College of Engineering, Maharashtra India
[5]Professor, Dept. of Information Technology, Dwarkadas J. Sanghvi College of Engineering, Maharashtra, India

---***---

Abstract - *The volume of textual data has expanded exponentially in recent years, making it a nuisance to store valuable information. Mining this enormous dataset containing dissimilar structured or unstructured data to identify hidden patterns for actionable and data driven insights can be troublesome. This information must be summarized to obtain meaningful knowledge in an acceptable time.*

This paper examines techniques for abstractive text summarization and validates these techniques on CNN/Daily Mail dataset using ROUGE scores. In addition to the use cases, their features and limitations in the real world are presented. The difficulties that arise during the summarization process and the solutions put forward in each approach are scrutinized, investigated, and explored. After evaluating the various approaches, it was found that the most common strategies for abstractive text summarization are recurrent neural networks with an attention mechanism and the transformer architecture. On experimenting, the results displayed that text summarization with Pegasus large model achieved the highest values for ROUGE-1 and ROUGE-L (44.17, 41.11 respectively). A detailed study was done to see how the best results were attained by the models applying a Transformer.

Key Words: **Abstractive Text Summarization, Attention mechanism, BERT, Neural Networks, Pegasus, Transformer.**

1. INTRODUCTION

There is a need to provide a satisfactory system for extracting information more quickly and efficiently because of the rate at which the internet has grown and consequently, the enormous volume of online information and documents. Manual extraction of summary from a large written document is quite challenging for humans. Automatic text summarization is key for addressing the issues of locating relevant papers among the vast number of documents available and extracting important information from them. A summary, according to Radev et al. [1]., is "a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually, significantly less than that". Text summarizing involves extracting and looking for the most meaningful and noteworthiest information in a document or collection of linked texts and condensing it into a shorter version while maintaining the overall meaning. The task of producing a succinct and fluent summary while keeping vital information content and overall meaning is known as automatic text summarization. In recent years, several methods for automatic text summarizing have been found and developed, and they are now widely utilized in a variety of fields. Automatic text summarizing is difficult and time-consuming for computers because they lack human knowledge and language competence. Humans, as opposed to computers, can summarize a document by reading it entirely to have a better comprehension of it and then writing a summary that highlights the essential ideas. Text summarizing raises a number of difficulties, including, but not limited to; text identification, text interpretation, summary generation, and examination of the created summary. The number of input documents (single or several), the goal (generic, domain-specific, or query-based), and the output all influence how text summarization is done (extractive or abstractive). The emergence and progress of automatic text summarization systems, which have produced considerable results in many languages, necessitates a review of these methods.

There are two different ways of text summarization:

Extractive Summarization: Extractive summarization attempts to summarize articles by finding key sentences or phrases from the original text and piecing together sections of the content to create a reduced version. The summary is then created using the retrieved sentences.

Unlike extraction, Abstractive Summarization relies on the ability to paraphrase and condense parts of a document utilizing advanced natural language approaches. Abstractive machine learning algorithms can construct new phrases and sentences to capture the meaning of the source content. When done effectively in deep learning issues, such abstraction can help overcome grammatical mistakes.

2. LITERATURE REVIEW

The earliest approaches in abstractive summarization relied on statistical methods - heavily reliant on heuristics and dictionaries for substitution of words. The methods considered for review in this paper are

2.1 RNN – Recurrent Neural Network Encoder-Decoder

In 2015, deep learning techniques were used for the first time in abstractive text summarization by Nallapati et al., and the proposed approach [2] was based on the encoder-decoder architecture.

The encoder-decoder models have been constructed to resolve Sequence to Sequence problems (Seq2Seq). The Seq2Seq models convert the neural network's input sequence right into a similar sequence of letters, words, or sentences. This model is utilized in numerous NLP applications, inclusive of system translation and textual content summarization. The input sequence in textual content summarization is the report to be summarized, and the output sequence is the summary.

Two RNNs, one functioning as an encoder and the other as a decoder, make up the basic encoder-decoder architecture for language problems. The encoder receives the input sequence and summarizes the data in internal state vectors, also known as context vectors. The encoder's outputs are discarded, leaving just the internal states. To help the decoder make accurate predictions, this context vector seeks to include information for all input elements. The RNN is a decoder in which the starting states are set to the final states, allowing the final state's context vector to be input into the decoder network's first cell. The START token is always the initial input to the decoder. Each decoder's recurrent unit uses this to create an output as well as its own hidden state by taking a hidden state from the preceding unit. This procedure is continued until the END token is encountered. Finally, at each time step, the loss on projected outputs is computed, and the mistakes are backpropagated over time to adjust the network parameters.

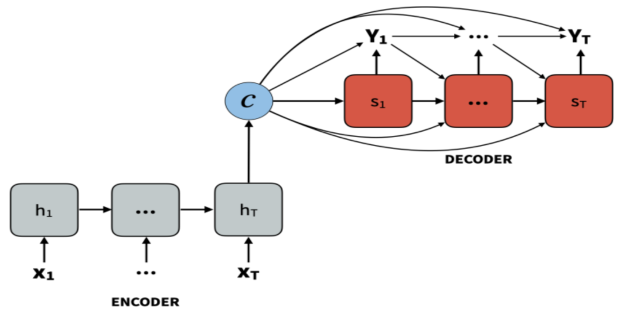

Fig -1: RNN - RNN encoder-decoder

For this architecture the chosen baseline model is LEAD-3. The main drawback with this approach is that it requires an extensive dataset which takes a long time to train.

2.2 Attention-Based Summarization ABS

The attention mechanism was introduced by Bahdanau, Cho, and Bengio [3] in the context of machine translation before being utilized for NLP tasks such as text summarization. [4].

When given long phrases, a basic encoder-decoder architecture may have problems because the size of the encoding is fixed for the input string which implies that it cannot examine all the parts of the long input.

The attention mechanism was created to help recall the information that has a substantial influence on the summary. At each output word, the attention mechanism is used to calculate the weight between the output word and each input word; the weights sum up to one. The usage of weights has the advantage of showing which input word in relation to the output word merits extra attention. After passing each input word, the weighted average of the decoder's last hidden layers is calculated and given to the SoftMax layer along with the last hidden levels in the current phase. Based on the context vectors linked with the source location and previously created target words, the model predicts a target word. Bidirectional RNNs are used in the attention mechanism. The level of attention given to words is represented by the context vector.

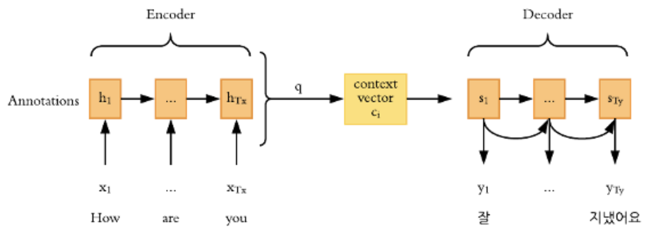

Fig -2: Attention mechanism[5]

2.3 Bidirectional encoder-decoder - BiSum

Proposed by X. Wan et.al [6], the model follows these steps: 1. Summary generation by the backward decoder from right to left, similar to the Seq2Seq-Attn model; 2. Both the encoder and the backward decoder should use an attention mechanism so that the forward decoder may construct the summary from left to right. The forward and reverse decoders both support the pointer approach.

The forward decoder is initialized with the backward encoder's last hidden state, whereas the backward decoder is initialized with the forward encoder's last hidden state. As a result, the model is capable of thinking about the past and future, resulting in balanced outputs. The input, as well as the input in reverse, are encoded into hidden states. The hidden states from both the forward and backward directions are then combined and sent to the decoder. When encoding longer sequences, bidirectional encoders have demonstrated better performance.

2.4 Pointer-Generator Networks + Coverage Mechanism

Neural sequence-to-sequence models tend to repeat themselves and reproduce factual details incorrectly. To resolve this problem, See, Liu, and Manning proposed Pointer-Generator Networks [7], in which they use a hybrid pointer-generator network that allows for both word copying and word generation from a predefined vocabulary. While maintaining the ability to generate fresh words through the generator, pointing facilitates the accurate reproduction of information. Coverage is a system for keeping track of what has been summarized and preventing duplication.

Drawback – Instead of introducing new terminology, the basic premise of this technique is to present a summary based on the source text.

2.5 Double attention pointer network

Xhixin Li et al. [8] developed a double attention pointer network-based encoder-decoder model (DAPT). In DAPT, important information from the encoder is collected by the self-attention mechanism. The soft attention and the pointer network provide more consistent core content and combine the two results in precise and reasonable summaries. Furthermore, the improved coverage mechanism is employed to avoid duplication and enhance the overall level of the summaries produced.

2.6 Transformer Architecture

Vaswani et al. [9] proposed a new basic network design, the Transformer, based entirely on attention mechanisms, with no recurrence and convolutions. The model uses positional encoding to keep track of the order of sequence. Only attention mechanism and feedforward networks are utilized for mapping dependencies between input and output.

The attention is calculated using Scaled Dot-Product Attention - dot product between a query, a key, and a value matrix Q, K, and V.

$$\mathrm{Attention}(Q,K,V)=\mathrm{softmax}((QK^T)/\sqrt{(d_k)})V$$

Equation (1) where K refers to key matrix, V refers to value matrix, Q refers to matrix of queries and d⍰ is the dimensionality of query/key matrix.

The scaled Dot-Product mechanism runs in parallel on numerous linear projections of input, the output is concatenated, multiplied by an extra weight matrix, and the multi-head attention layer output is then utilized as input to the feedforward layer.

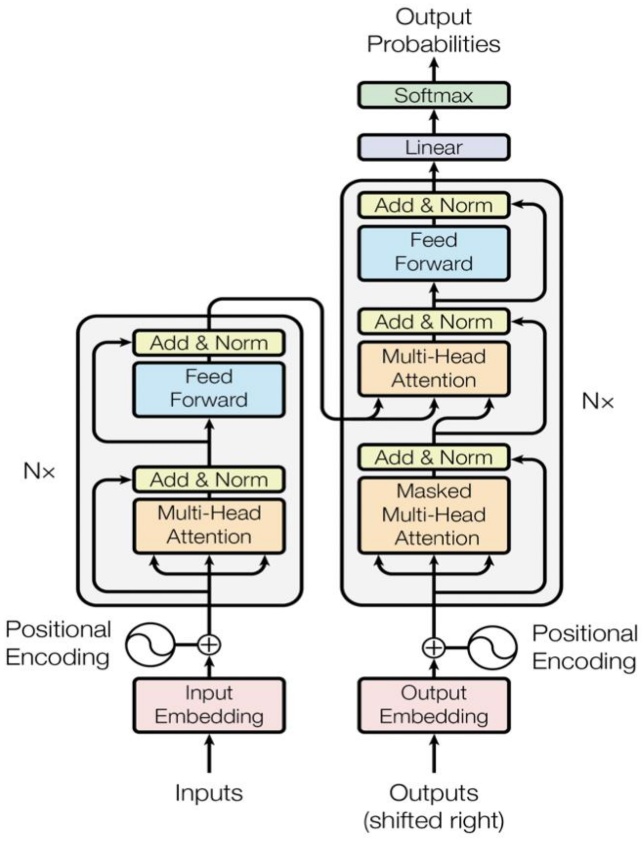

Fig-3: Transformer architecture[5]

The major advantage of The Transformer is that it lends itself well to parallelization. In comparison to encoder-decoder models with attention, Transformer achieves state-of-the-art performance whilst taking less training time. The sequential architecture limits efficiency during training, as it does not fully utilize the capability of Graphics Processing Units (GPU).

The following transformer variations were evaluated for the literature review:

2.6.1 BERT - Bidirectional Encoder Representations from Transformers (Encoders or autoencoding Transformer)

Proposed by Devlin et al. [10] and applied to text summarization [11], BERT is designed to condition both left and right context at the same time to pre-train deep bidirectional representations from the unlabelled text. BERT is a bidirectionally trained language model, which means it can read text from both sides sequentially. As a result, now a greater comprehension of language context and flow is attained than in the case of single-direction language models. BERT has two objectives:

A. Masked Language Modelling involves hiding a word in a phrase and then having the model identify which word was concealed (masked) based on the hidden word's context.

B.Next Sentence Prediction - assists the model in determining if two provided phrases have a logical, sequential relationship or if their relationship is just random.

Masked Language Modelling and Next Sentence Prediction are both used to train the model. This is done to lower the loss function of the two approaches when combined.

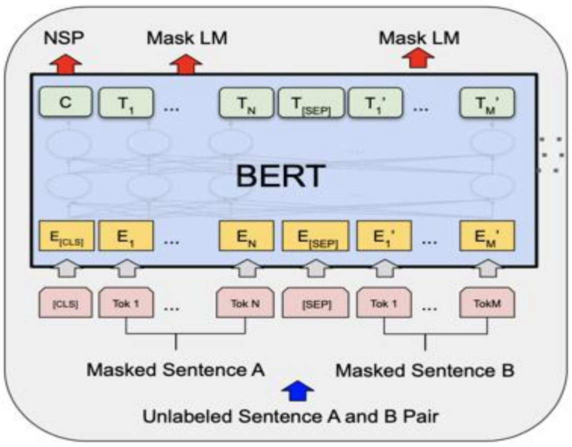

Fig-4: BERT [10]

2.6.2 PEGASUS - Abstractive Summarization Pre-training with Extracted Gap-sentences (sequence-to-sequence Transformer) [12]

Uses 2 objectives:

a. Gap sentence generation (GSG): complete sentences from the document are masked off, and the model is trained to predict these sentences using the remaining sentences. It has been proven that hiding the most important sentences from a paper is more effective.

b. Encoder pre-trained as masked language model (MLM): Words from sequences are randomly masked, and the sequence's remaining words are used to predict the masked words.

Both GSG and MLM are used at the same time.

Pegasus may require post-processing to correct errors and improve summary text output.

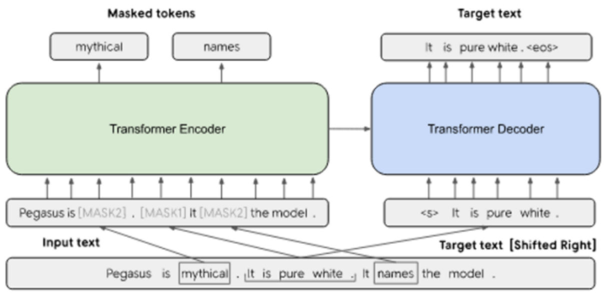
Masked tokens
mythical
names
Target text
It is pure white . <eos>
Transformer Encoder
Transformer Decoder
Pegasus is [MASK2] . [MASK1] it [MASK2] the model .
<s> It is pure white .
Input text
Target text [Shifted Right]
Pegasus is mythical . It is pure white . It names the model .

Fig-5: PEGASUS, [MASK1] – GSG, [MASK2] – MLM [12]

2.6.3 BART- (sequence-to-sequence Transformer)

Bart[13] employs a typical machine translation architecture that includes a left-to-right decoder and a bidirectional encoder (similar to BERT). The pretraining task entails changing the order of the original phrases at random and implementing a novel in-filling strategy in which text spans are replaced with a single mask token.

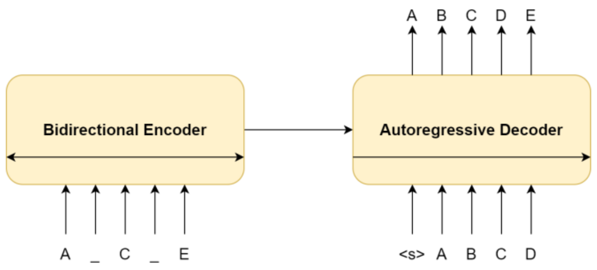

Fig-6: BART

2.6.4 T5 - Text-To-Text Transfer Transformer (Sequence-to-sequence Transformer) [14]

In contrast to BERT, which fine-tunes the model for each task separately, the text-to-text framework employs the same hyperparameters, loss function, and model for all tasks. In this technique, the inputs are represented in such a way that the model recognizes a task, and the output is just the "text" form of the projected outcome. Relative scalar embeddings are used in T5.

The three objectives that are concerned with T5 are as follows:

a. Language model: It is an autoregressive modelling approach for predicting the next word.

b. BERT-style objective: Masking/replacing words chosen randomly with a random different word and the model predicts the original text.

c. Deshuffling: The inputs are shuffled randomly and the model tries to predict the original text.

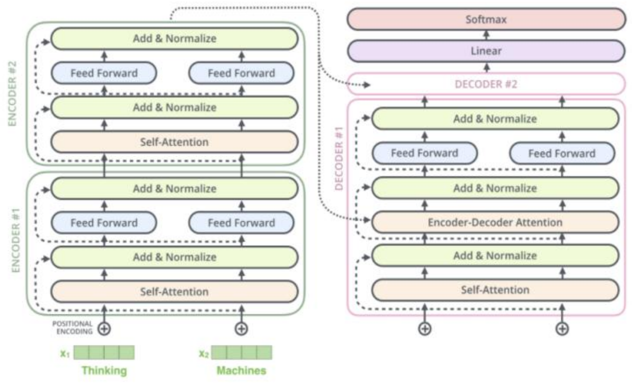

Fig-7: T5 [15]

3. DATASETS

CNN/Daily Mail is a text summary dataset [2]. Human-produced abstractive summary bullets were constructed from CNN and Daily Mail news articles as questions (with one of the entities obscured) and stories as the appropriate sections from which the system is supposed to answer the fill-in-the-blank inquiry. The programs that crawl, extract, and produce pairs of excerpts and questions from these websites were released by the authors.

According to their scripts, the corpus has 286,817 training pairings, 13,368 validation pairs, and 11,487 test pairs. On average, the source texts in the training set comprise 766 words and 29.74 sentences, whereas the summaries have 53 words and 3.72 sentences.

In 2015 and 2016, the Gigaword dataset from Stanford University's Linguistics Department was the most popular dataset for training models. The New York Times, Associated Press, and Washington Post are among the seven news organizations represented in Gigaword, which has around 10 million papers. Gigaword is one of the largest and most diverse summarization datasets, despite the fact that it contains headlines rather than summaries; as a result, it is deemed to contain single-sentence summaries.

CNN/Daily Mail dataset has been extensively used in the latest studies for training and evaluation.

4. EVALUATION

For evaluation, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores are used. They are a set of metrics, each having precision, recall, and F-1 score

ROUGE-N measures the number of matching n-grams between a reference and model-generated text. An n-gram is a collection of tokens or words. A unigram (1 gram) is made up of a single word, whereas a bigram (2 gram) is made up of two consecutive words. The N in ROUGE-N stands for the n-gram being used.

ROUGE-L uses Longest Common Subsequence (LCS) to determine the longest matching sequence of words. The main benefit of using LCS is that it bases itself on in-sequence matches that indicate sentence level word order instead of successive matches. A predetermined n-gram length is not needed since it naturally includes the lengthiest in-sequence common n-grams. Longer shared sequences indicate more similarity.

All ROUGE variations can be used to measure recall, precision, or the F1 score. The F1 score was chosen for this paper because it is less impacted by summary length and provides a good balance between recall and precision. ROUGE does not cater to different words that have the same meaning.

5. RESULTS

Table-1: Summary of approaches

Approach	Summary of Approach
RNN RNN Encoder Decoder	Uses 2 RNN cells to comprehend input and generate output sequence.
BiSum	Generates summary in both directions using an additional backward decoder.
ABS	Attention mechanism added to Sequential architecture.
Pointer-Generator Networks+Coverage	Allows both copying and generating words.
DAPT	Uses self-attention mechanism, soft attention, and pointer network.
Base Transformer	Follows an encoder-decoder structure but without recurrence and convolutions.
BERT	Uses MLM (Masked LM) and NSP (Next Sentence Prediction).
PEGASUS	Uses GSG and pre-trains the encoder as a MLM.
T5	BERT model with decoder added - uses same loss function and hyperparameters for all NLP tasks.
BART	Denoising autoencoder.

Table-2: Scores on CNN/DailyMail Dataset

Approach	R1	R2	RL
RNN Encoder Decoder LEAD-3	40.42	17.62	36.67
BiSum	37.01	15.95	33.66
ABS	41.16	15.75	39.08
Pointer-Generator Networks + Coverage	39.53	17.28	36.38
DAPT + imp-coverage (RL+MLE(ss))	40.72	18.28	37.35
Base Transformer	39.5	16.06	36.63
BERT - BERTSUMABS	41.72	19.39	38.76
PEGASUS Large	44.17	21.47	41.11
T5	43.52	21.55	40.69
BART	44.16	21.28	40.9

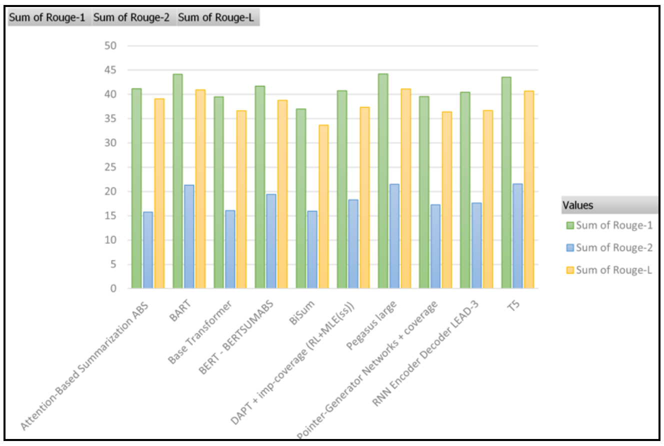

Chart-1: Scores on CNN/Dailymail dataset

Table-3: Scores on Gigaword Dataset

Approach	R1	R2	RL
ABS	30.88	-	-
Pointer-Generator Networks + Coverage	39.53	17.28	36.38
DAPT + imp-coverage (RL+MLE(ss))	40.72	18.28	37.35
Base Transformer	37.57	18.9	34.69
BERT - BERTSUMABS	36.97	17.96	33.87
PEGASUS Large	39.12	19.86	36.24
T5	-	-	-
BART	37.28	18.58	34.53

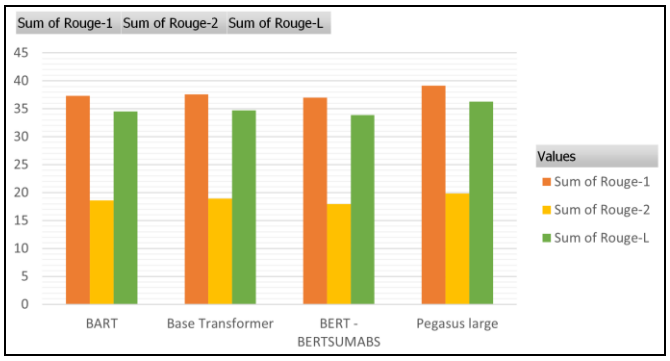

Chart-2: Scores on Gigaword dataset

ROUGE F1 results on CNN/DailyMail and Gigaword test set (The abbreviations R1 and R2 stand for unigram and bigram overlap, respectively; RL stands for the longest common subsequence.). The results for comparative systems are collected from the authors' publications or produced using open source software on the dataset.

3. CONCLUSIONS

The importance of text summarization has grown in recent years as a result of a large amount of data available on the internet. Natural language processing has a wide range of applications, with automatic text summarization being one of the most common. There are two types of text summarization methods: extractive and abstractive. The area of automatic text summarizing has a long history of research, and the focus is shifting from extractive to abstractive summarization. Abstractive summary methodology generates a relevant, precise, content-rich, and less repetitive summary. The extractive text summarization approach, on the other hand, provides a summary based on linguistics and statistical factors that include words and phrases from the original text. Abstractive summarization is a difficult field since it focuses on developing summaries that are closer to human intellect. As a result, this study puts the methodologies for abstractive summarization to the test, as well as the benefits and drawbacks of various approaches. The ROUGE scores obtained from the literature are reported, based on which, it can be concluded that the best results were realized by the models that apply Transformer.

ACKNOWLEDGEMENT

We would like to express our appreciation for Professor Abhijit Joshi, our research mentor, for his patient supervision, ardent support, and constructive criticisms of this research study. We thank Dr. Vinaya Sawant, for her advice that greatly improved the manuscript and assistance in keeping our progress on schedule.

REFERENCES

[1] D. R. Radev, E. Hovy, and K. McKeown, "Introduction to the special issue on summarization. Computational linguistics", vol. 28, No. 4, pp. 399–408,Dec. 2002.

[2] R. Nallapati, B. Zhou, C. D. Santos, Ç. Gulçehre, and B. Xiang, "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond," unpublished.

[3] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," unpublished.

[4] A. M. Rush, S. Chopra, and J. Weston, "A Neural Attention Model for Abstractive Sentence Summarization," unpublished.

[5] K. Loginova, "Attention in NLP", unpublished.

[6] X. Wan, C. Li, R. Wang, D. Xiao, and C. Shi, "Abstractive document summarization via bidirectional decoder," in *Advanced Data Mining and Applications*, G. Gan, B.

Li, X. Li, and S. Wang, Eds., Springer International Publishing, Cham, Switzerland, 2018, pp. 364–377.

[7] A. See, P. J. Liu, and C. D. Manning, "Get To The Point: Summarization with Pointer-Generator Networks," unpublished.

[8] Z. Li, Z. Peng, S. Tang, C. Zhang, and H. Ma, "Text Summarization Method Based on Double Attention Pointer Network," *IEEE Access*, vol. 8, pp. 11279-11288,Jan. 2020.

[9] A. Vaswani *et al.*, "Attention Is All You Need," unpublished.

[10] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," unpublished.

[11] Y. Liu and M. Lapata, "Neural Machine Translation by Jointly Learning to Align and Translate," unpublished.

[12] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization," unpublished.

[13] M. Lewis *et al.*, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," unpublished.

[14] C. Raffel *et al.*, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," unpublished.

[15] Q. Chen, "T5: a detailed explanation", unpublished.

[16] D. Suleiman and A. A. Awajan, "Deep Learning Based Abstractive Text Summarization: Approaches, Datasets, Evaluation Measures, and Challenges," *Mathematical Problems in Engineering, Hindawi,* vol. 2020, pp. 1-29,Aug. 2020.

[17] F. Jonsson, "Evaluation of the Transformer Model for Abstractive Text Summarization," unpublished.

BIOGRAPHIES

Aryan Ringshia is currently pursuing the B.Tech. degree with the Department of Information Technology at Dwarkadas J. Sanghvi College of Engineering, Mumbai, India. He was born in Mumbai, India. From 2021 to 2022, he was a junior mentor in DJ init.ai student chapter of the Information Technology Department at Dwarkakdas J Sanghvi College of Engineering where he worked on research and industry oriented projects to advance in domain of AIML with guidance from seniors and taught AIML concepts to freshers. His research interests include machine learning, deep learning, and natural language processing.

Neil Desai is a B.Tech. student in the Department of Information Technology at Dwarkadas J. Sanghvi College of Engineering in Mumbai, India. Currently he is working as a Teaching Assistant in the Data Science Department at Dwarkadas J Sanghvi College of Engineering where he is helping the faculty design laboratory experiments for the department. Natural Language Processing, Deep Learning, and Image Processing Applications are among his research interests.

Umang Jhunjhunwala was born in Mumbai, India and is currently pursuing his B.Tech. degree in Information Technology at D.J. Sanghvi College of Engineering in Mumbai, India. He worked as a summer data science intern at Xelpmoc Design and Tech Ltd for four months in Kolkata, India where worked as part of team to develop a system capable of performing complex natural processing tasks. His research interests include analytics, natural language processing and machine learning.

Ved Mistry was born in Mumbai, India and is currently pursuing the B.Tech. degree with the Department of Information Technology at Dwarkadas J. Sanghvi College of Engineering, Mumbai, India. He served as a junior mentor in the DJ init.ai student chapter of the Information Technology Department at Dwarkakdas J Sanghvi College of Engineering from 2021 to 2022 where he worked on research projects with the help of his seniors and taught machine learning concepts to his juniors. His research interests include Computer Vision and Image Processing, Deep Learning and Natural Language Processing.

Prachi Tawde received the M.E. degree in computer engineering from University of Mumbai, Mumbai, India in 2017 and the B.E. degree in information technology from University of Mumbai, 1st Class, Mumbai, India in 2012. She is currently an Assistant Professor at Department of Information Technology, Dwarkadas J. Sanghvi College of Engineering, Mumbai University. She has previously published in the International Conference on Big Data Analytics and Computational Intelligence (ICBDAC), IEEE, 2017. Her research interests include application of natural language processing in educational domain, machine learning and analytics.