Research Paper Summarization using NLP

SHIVENDU SHUKRE[1], SAMARSINH SALUNKHE[2], PRANAV RATHI[3], VEDANT SHINDE[4], PROF. M. V. MANE[5]

[1, 2, 3, 4, 5] *Dept. of Computer Engineering Pune Institute of Computer Technology, Pune, India*

Abstract-* Keeping up with the most recent developments in their respective domains is extremely difficult for researchers and practitioners in the age of information overload, where many research articles are released every day. It is becoming more and more challenging to sift through several articles to locate pertinent and useful information due to the sheer volume of scientific publications. As a result, there is an increasing need for automated methods for summarizing research publications so that users may quickly understand the important ideas without using excessive time and effort. This research article's goal is to present a thorough analysis of the current approaches and tools used for NLP-based research paper summarizing. We want to investigate the various approaches, from conventional extractive procedures to more sophisticated abstractive ones, and talk about their advantages, disadvantages, and prospective uses. We want to shed light on the current trends and problems in this sector by evaluating and contrasting the state-of-the-art approaches, offering scholars and practitioners an invaluable resource to direct future research and development initiatives. In summary, the goal of this study is to present a thorough review of the most cutting-edge approaches of employing NLP to summarize research papers. We aim to create a deeper awareness of the difficulties and opportunities that lie ahead by investigating the numerous methods, algorithms, and evaluation measures used in this area. Through this study, we hope to spur new research and innovation in the field of research paper summary, allowing academics and industry professionals to stay up to date on advancements and utilize the body of scientific literature more effectively.

***Indexed Terms-* Text Summarization, Natural Language Processing, Deep Learning, Abstractive Text Summarization, ROUGE**

I. INTRODUCTION

The field of information retrieval is experiencing rapid advancement in the digitalized world. To stay updated, people rely on various resources, but they prefer information that is concise and to the point, considering time constraints. However, a significant challenge arises when reading news articles or online reviews, as they often require thorough reading to draw conclusions. To address this issue, text summarization has emerged as a means to improve information retrieval. Text summarization involves extracting the essential information from a text in a concise, well-organized, and easily understandable manner. It relies on Natural Language Processing (NLP) techniques to generate summaries that can be interpreted by humans.

Text summarization can be categorized into two types: extractive summarization and abstractive summarization. Extractive summarization involves selecting the most relevant sentences from the original text, essentially extracting a subset of sentences. It can be compared to highlighting key sentences from the original document. On the other hand, abstractive summarization focuses on capturing the main essence of the text and generating a summary using its own words. Abstractive summarization can be seen as recreating the original text with new phrases to form a summary.

Both extractive and abstractive summarization methods have their own advantages and disadvantages. Extractive summarization ensures accuracy by selecting important sentences, but it may result in incoherent summaries. Abstractive summarization produces summaries that are concise and readable but may lose some key facts, especially when dealing with large documents.

Recently, pre-training approaches have been proposed for text summarization, in which a large language model is pre-trained on a large corpus of text before being fine-tuned for summarization. For example, the BART model proposed by Lewis et al. (2020) uses a denoising autoencoder pre-training task to improve the model's ability to generate high-quality summaries. Similarly, the Pegasus model proposed by Zhang et al. (2020) uses extracted gap-sentences as a form of pre-training data.

II. RELATED WORK

BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension by Lewis et al. (2020) [1] - This paper introduces BART (Bidirectional and Auto-Regressive Transformer), a pre-training approach for sequence-to-sequence models that achieves state-of-the-art results on a range of text generation tasks, including summarization.

Fine-tuning Language Models from Human Preferences by Kim et al. (2020 [2]) - This paper proposes a method for fine-tuning language models for summarization by using human preferences as feedback. The authors show that this approach can significantly improve the quality of the generated summaries.

Pegasus: Pre-training with Extracted Gap-sentences for Abstractive Summarization by Zhang et al. (2020) [3] - This paper introduces Pegasus, a pre-training approach for abstractive summarization that uses gap-sentences, or sentences that have been removed from the original text, as a form of pre-training data. The authors show that Pegasus outperforms existing state-of-the-art approaches on several benchmark datasets.

Controllable Abstractive Summarization by Gehrman et al. (2018) [5] - This paper proposes a method for controllable abstractive summarization, in which the user can specify desired attributes of the summary (such as length or content) and the model generates a summary that satisfies these constraints.

Summarizing Long Articles Using Context-aware Neural Networks by Wang et al. (2018) [6] - This paper introduces a context-aware neural network approach for summarizing long articles, in which the model takes into account the context of each sentence in the article when generating the summary.

A Neural Attention Model for Abstractive Sentence Summarization by Rush et al. (2015) [7] - This paper proposes a sequence-to-sequence neural network model with attention mechanism for abstractive sentence summarization. The model generates summaries by attending to different parts of the input sentence.

Get To The Point: Summarization with Pointer-Generator Networks by See et al. (2017) [8] - This paper introduces a pointer-generator network for abstractive summarization that can generate words from the source text or copy them directly, addressing the problem of out-of-vocabulary words.

Deep Reinforcement Learning for Sequence-to-Sequence Models by Paulus et al. (2017) [9] - This paper explores the application of reinforcement learning techniques to sequence-to-sequence models for abstractive summarization. The model is trained using a combination of supervised learning and reinforcement learning with reward models based on ROUGE scores.

Ranking Sentences for Extractive Summarization with Reinforcement Learning by Lin and Bilmes (2018) [10] - This paper proposes a reinforcement learning framework for extractive summarization, where a model learns to rank sentences based on their importance to the summary. The model is trained using a combination of supervised learning and reinforcement learning with rewards based on ROUGE scores.

BERTSUM: Text Summarization as Sequence-to-Sequence Learning with Transformers by Liu et al. (2019) [11] - This paper presents BERTSUM, a transformer-based model for abstractive text summarization. The model leverages the pre-trained BERT model and introduces a novel document-level encoder to capture global information for summary generation.

III. THEORY AND IMPLEMENTATION

A. Transformer

Recurrent models, including Recurring Neural Networks and Long Short-Term Memory, have long been used to solve issues with language modelling and machine translation. But because these models are fundamentally sequential, they become problematic as sequence lengths increase because batching among cases is constrained by memory issues. RNNs handle data sequentially (or one data element at a time), thus just adding more processing power won't dramatically speed them up. As a result, it is challenging for us to train RNNs on massive amounts of data. Because there is less room for parallelization, they train more slowly. The disadvantages of sequential processing are still present and harmful, notwithstanding recent successes in significantly improving the computational efficiency of such models.

Transformers enter the scene at this point. A team of academics from Google (and UoT) created transformers, a type of neural network design, in 2017. They completely rely on an attention mechanism to identify global dependencies between the input and the output, avoiding the use of the notion of repetition. Transformers are far more parallelizable than sequential models and can produce extremely high translation quality even with little training time. They are also not as challenging to train on very huge volumes of data. For instance, the GPT-3 (Generative Pre-Trained Transformer-3) model was trained using data that was around 45 terabytes in size.

B. T5 Transformer

Google Research created the T5 (Text-to-Text Transfer Transformer) model, a flexible and potent transformer-based language model. It is intended to use a uniform framework to address a variety of natural language processing tasks. T5 is taught in a "text-to-text" approach, in contrast to earlier models that were trained for specific tasks like translation or summarization, meaning it learns to transform input text into output text regardless of the specific job.

T5 adopts the transformer architecture, which consists of numerous layers of feed-forward neural networks and self-attention processes. It is made up of a decoder and an encoder that use the same transformer architecture. T5 is exposed to a range of text-to-text tasks while being trained. For instance, it is taught to do the task of translating a sentence from one language to another rather than training it especially for translation. T5 learns to generalize across numerous activities and can be adjusted for purposes by exposing the model to a variety of tasks in this generalized style.

T5 is fine-tuned by retraining the pre-trained model on downstream tasks or datasets. To perform the goal task, such as text classification, question-answering, summarizing, or any other text-based task, the model must be able to adapt to its particular requirements. To fine-tune a system, task-specific input-output pairs are often provided, and model parameters are optimized using methods like backpropagation and gradient descent.

On numerous benchmarks and tasks for natural language processing, T5 has demonstrated cutting-edge performance. Because of its adaptability and generalization skills, it is a useful tool for a variety of text-based applications, enabling academics and practitioners to handle multiple tasks with a single model.

The transformer architecture, which has evolved to be a standard for many cutting-edge language models, is followed by the T5 model. It is made up of an encoder and a decoder, both of which have numerous layers of feed-forward neural networks and self-attention mechanisms. When encoding or decoding, the self-attention mechanism enables the model to consider the relative weights of the various words in a sentence or sequence. This approach enables the model to comprehend the context and relationships within the input text and aids in capturing word dependencies. The information is subsequently processed by the feed-forward neural networks in each layer, and non-linear modifications are used to improve the representation of the input text. The model can learn hierarchical representations of the text thanks to the multi-layered structure, with higher layers capturing more abstract and sophisticated elements.

T5 is taught how to convert input text into output text regardless of the task at hand using a process known as "text-to-text" training. It has been honed to do a wide variety of tasks, including text classification, translation, summarization, and question-answering. Massive amounts of data and computational resources are used to train the model, which frequently takes days or weeks to complete on powerful hardware. T5 receives task-specific input-output pairs in a standardized text format during training. It is taught to map an input sentence like "Translate this English sentence to French" to the appropriate translated sentence, as opposed to being trained especially for translation, for instance; various mappings store information about blocked voters allowed recipients, and proposal deposits. Various events occur when a proposal is added, when someone votes, and when a proposal is tallied.

T5 learns to generalize across several activities and may apply its expertise to new, unforeseen tasks by employing the text-to-text format and extensive training on a variety of tasks. One of T5's advantages is its capacity to generalize, which enables effective fine-tuning and adaptation to certain downstream applications.

C. Working

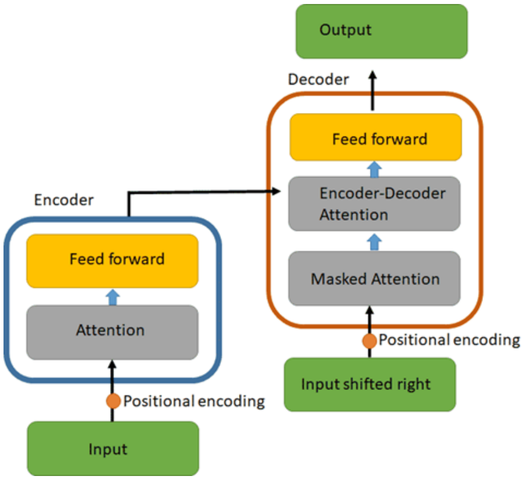

Figure 1. Working of T5

1. Input Encoding

The input text is initially tokenized by utilizing methods like Byte Pair Encoding (BPE) or Sentence-Piece to divide it into smaller units like words or sub-word units. The mapping of each token to an individual numerical representation known as an embedding follows. The meaning and context of the input tokens are captured by these embeddings.

2. Encoder

A stack of identical encoder layers is passed through with the input tokens that have been encoded. A feed-forward neural network and a self-attention mechanism make up the two sub-layers of each encoder layer. The model can examine the connections between various tokens in the input sequence thanks to the self-attention mechanism. Each token's relevance in relation to other tokens is highlighted by the computation of attention weights for each token. The attention weights direct the model's processing emphasis to pertinent areas of the input.

The representations derived from the self-attention mechanism are subjected to non-linear changes by the feed-forward neural network within each encoder layer. It improves the characteristics and extracts intricate patterns from the sequence of input.

The concurrent processing of the input tokens by the encoder layers enables the model to collect dependencies and contextual data across the whole sequence.

3. Decoder

T5 uses both an encoder and a decoder when the work at hand calls for the creation of output text, such as when translating or summarizing. Additionally, a stack of identical layers makes up the decoder. A masked self-attention mechanism, an encoder-decoder attention mechanism, and a feed-forward neural network are the three sub-layers that make up each decoder layer.

The decoder's masked self-attention method enables the model to pay attention to earlier points in the target sequence while it is being generated. This prohibits the model from accessing future tokens, which would be unavailable during inference, and guarantees that it pays attention to relevant information while predicting the next token.

The decoder can concentrate on important areas of the encoder's output thanks to the encoder-decoder attention mechanism. It enables the model to produce precise and contextually relevant output tokens by utilizing the learned representations from the encoder.

4. Training and Fine Tuning

T5 is first pre-trained utilizing a text-to-text transfer learning framework on a vast corpus of various tasks. The model is trained to anticipate masked tokens in the input sequence, carry out sentence categorization, produce text, and manage additional text-to-text transformations during pre-training. T5 gains general language comprehension and transferable skills thanks to this pre-training.

T5 can be fine-tuned on downstream activities after pre-training. Input-output pairs specific to the task are provided, and the model is further trained on the target task, as part of the fine-tuning process. Utilizing methods like backpropagation and gradient descent, the model's parameters are optimized to minimize the discrepancy between the model's predictions and the desired outputs for the supplied task-specific data. T5 can be fine-tuned to conform its previously learned information to the needs of the target task.

5. Inference

A fresh, unread input text is sent via the encoder during inference. The decoder produces the desired output text based on the task. For instance, in translation, the decoder creates the translated text from the input that has been encoded.

Beam search or other decoding techniques can be used to investigate a variety of alternative output sequences throughout the decoding process. Beam search iteratively chooses the most likely tokens based on a scoring system, keeping track of the top-k most probable sequences at each decoding stage to progressively build the final output.

The T5 model effectively encodes, decodes, and generates output text for diverse natural language processing tasks by utilizing the transformer architecture, self-attention mechanisms, feed-forward networks, and masked decoding.

IV. CONCLUSION & FUTURE WORK

In this project, we are going to implement robust models for Abstractive Text Summarization of scientific documents. With the help of T5 Transformer we will be creating the training and testing dataset.

We would be training these models mainly on Google Colab and will preprocess the data according to the model and finally we will compare the accuracies of the models. We aim to achieve better accuracies that the models which are already presented in the reference papers. Our efforts will help in improving the automatic summarization of documents.

Most existing summarization systems focus on single document summarization. Future work could explore techniques for summarizing multiple documents on a given topic, which is a more challenging task due to the need to identify and reconcile different perspectives and information across the documents.

Many applications require summarization of texts in a specific domain (e.g., scientific papers, legal documents, news articles). Future work could explore domain-specific approaches that leverage domain-specific knowledge and language models to improve the quality of the summarization.

Current summarization systems are typically fully automated, but interactive summarization systems that allow users to provide feedback and interact with the system could lead to more personalized and accurate summaries.

REFERENCES

[1] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

[2] Ziegler, Daniel & Stiennon, Nisan & Wu, Jeffrey & Brown, Tom & Radford, Alec & Amodei, Dario & Christiano, Paul & Irving, Geoffrey. (2019). Fine-Tuning Language Models from Human Preferences.

[3] Zhang, Jingqing & Zhao, Yao & Saleh, Mohammad & Liu, Peter. (2019). PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization.

[4] S M, Meena M P, Ramkumar R.E, Asmitha Selvan, Emil. (2020). Text Summarization Using Text Frequency Ranking Sentence Prediction. 1-5. 10.1109/ICCCSP49186.2020.9315203.

[5] Angela Fan, David Grangier, and Michael Auli. 2018. Controllable Abstractive Summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

[6] P. Janjanam and C. P. Reddy, "Text Summarization: An Essential Study," 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), 2019, pp. 1-6, doi: 10.1109/ICCIDS.2019.8862030

[7] Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

[8] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

[9] Alexios Gidiotis and Grigorios Tsoumakas. 2020. A Divide-and-Conquer Approach to the Summarization of Long Documents. IEEE/ACM Trans. Audio, Speech and Lang. Proc. 28 (2020), 3029–3040. https://doi.org/10.1109/TASLP.2020.3037401.

[10] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking Sentences for Extractive Summarization with Reinforcement Learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.

[11] A. Nikiforovskaya, N. Kapralov, A. Vlasova, O. Shpynov and A. Shpilman, "Automatic generation of reviews of scientific papers," 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), 2020, pp. 314-319, doi: 10.1109/ICMLA51294.2020.00058.

[12] P. R. Dedhia, H. P. Pachgade, A. P. Malani, N. Raul and M. Naik, "Study on Abstractive Text Summarization Techniques," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020, pp. 1-8, doi: 10.1109/ic-ETITE47903.2020.087.

[13] S M, Meena M P, Ramkumar R.E, Asmitha Selvan, Emil. (2020). Text Summarization Using Text Frequency Ranking Sentence Prediction. 1-5. 10.1109/ICCCSP49186.2020.9315203.

[14] S. Alhojely and J. Kalita, "Recent Progress on Text Summarization," 2020 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2020, pp. 1503-1509, doi: 10.1109/CSCI51800.2020.00278.

[15] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880, Online. Association for Computational Linguistics.

[16] Angela Fan, David Grangier, and Michael Auli. 2018. Controllable Abstractive Summarization. In Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

[17] D. Bartakke, S. Kumar and A. Junnarkar, "Text Summarization and Dimensionality Reduction using Learning Approach," 2020 IEEE International Conference for Innovation in Technology (INOCON), Bangluru, India, 2020,

pp. 1-5, doi: 10.1109/INOCON50539.2020.9298250.

[18] S. K. Dam, M. Shirajum Munir, A. D. Raha, A. Adhikary, S. -B. Park and C. S. Hong, "RNN-based Text Summarization for Communication Cost Reduction: Toward a Semantic Communication," 2023 International Conference on Information Networking (ICOIN), Bangkok, Thailand, 2023, pp. 423-426, doi: 10.1109/ICOIN56518.2023.10048944.

[19] T. Islam, M. Hossain and M. F. Arefin, "Comparative Analysis of Different Text Summarization Techniques Using Enhanced Tokenization," 2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 2021, pp. 1-6, doi: 10.1109/STI53101.2021.9732589.

[20] Y. Chen and Q. Song, "News Text Summarization Method based on BART-TextRank Model," 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 2021, pp. 2005-2010, doi: 10.1109/IAEAC50856.2021.9390683.