ISSN 2278-3091
Volume 13, No.2, March - April 2024
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse011322024.pdf
https://doi.org/10.30534/ijatcse/2024/011322024

A Comparative Study of Transformer-based Models for Text Summarization of News Articles

Charles Munyao Muia[1], Aaron Mogeni Oirere[2], Rachael Njeri Ndung'u[3]
[1] Department of Computer Science, Murang'a University of Technology, Kenya, munyaomuia@mut.ac.ke
[2] Department of Computer Science, Murang'a University of Technology, Kenya, amogeni@mut.ac.ke
[3] Department of Information Technology, Murang'a University of Technology, Kenya, rndungu@mut.ac.ke

Received Date : February 12, 2024 Accepted Date: March 14, 2024 Published Date: April 06, 2024

ABSTRACT

Transformer-based models such as GPT, T5, BART, and PEGASUS have made substantial progress in text summarization, a sub-domain of natural language processing that entails extracting important information from lengthy texts. The main objective of this research was to conduct a comparative analysis of these four transformer-based models based on their performance in text summarization of news articles. In achieving this objective, the transformer models pre-trained on extensive datasets were fine-tuned on the CNN/DailyMail dataset using a low learning rate to preserve the learned representations. The T5 transformer records the highest scores of 35.12, 22.75, 32.82, and 28.59 in ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-Lsum respectively, surpassing GPT, BART, and PEGASUS across all ROUGE metrics. The findings deduced from this study establish the proficiency of encoder-decoder models such as T5 in summary generation. Furthermore, the findings also demonstrated that the fine-tuning process's effectiveness in pre-trained models is improved when the pre-training objective closely aligns with the downstream task.

Key words: Natural Language Processing, ROUGE Metrics, Text Summarization, Transformers.

1. INTRODUCTION

In an era of information abundance, news articles have become a primary source of information and knowledge dissemination [1]. Staying informed is essential yet time-consuming, with the constant influx of diverse news stories. Text summarization, a transformative branch of natural language processing, offers an elegant solution to this problem [1], [2]. The ability to automatically extract the essential information from lengthy news articles and present it in a concise, coherent summary not only enhances information accessibility but also frees readers from the demanding task of reading through voluminous text [3]. This study, therefore, explores text summarization within news article contexts, emphasizing the revolutionary importance of transformers in the domain of natural language processing.

Transformers, a class of neural networks, have redefined the landscape of natural language processing [4]. Since their inception, these models have consistently outclassed performance benchmarks across various language tasks. The self-attention mechanism, which was introduced in the paper "Attention is All You Need" by Vaswani *et al.*, is the central component of the architecture of transformer-based models [5], [6],[7]. With the introduction of this attention mechanism, the models may weigh various sections of the input sequences and give varying attention weights based on previous sequences. In the self-attention mechanism, each token has three vectors: Key (K), Query (Q), and Value (V), which are the linear projections of the input embeddings. The attention score between the vectors is calculated using a dot product to compute the relevance between corresponding tokens. The attention scores are then converted into probabilistic distributions using a softmax activation function [6]. Transformers' ability to process sequential data representations has improved due to their increased capacity to pay attention to different parts of the input sequence dynamically. Such advancements have led to remarkable gains in several natural language processing tasks and paved the way for text summarization techniques capable of generating more contextually aware and human-like summaries.

Therefore, this paper's primary objective is to compare and analyze how well four state-of-the-art transformer-based news article summarization models perform. The following is a brief description of how the rest of this paper is organized: in

Section 2, we have the related works, and in Section 3.0, the methodology is described. Section 4 covers the results of this study and a discussion of their implications. In Section 5, the study's conclusions are discussed, along with recommendations for further research.

2. RELATED WORKS

Text summarization is among the application areas of the natural language processing domain, and it deals with the extraction of the most significant content from the original text while maintaining its main idea [8]. Two main types of text summarization exist, namely extractive and abstractive. While the abstractive type of summarization goes further and creates succinct summaries by interpretation and paraphrasing, extractive summarization identifies and chooses the most significant sentences from the provided text [9], [10].

The trajectory of text summarization milestones has undergone remarkable transformations, progressing from frequency-based approaches to the incorporation of sophisticated machine-learning techniques [11]. The latest stride in this evolution involves transformer-based models, which marks a key advancement that has significantly elevated natural language processing. Text summarization researchers commonly leverage diverse datasets, including the CNN/DailyMail, Newsroom, New York Times (NYT), Document Understanding Conference (DUC), and the Gigaword datasets to explore and enhance techniques in this dynamic and rapidly evolving domain [12].

In this study [13], three state-of-the-art transformers were compared across few-shot and zero-shot learning for both abstractive text summarization for multi-documents, whereby the summary was based on a user-defined query. The implementation details of this study were as follows: a batch size of 8 and 512 tokens as the maximum sequence length, 20 training epochs, and three warm-up steps on a V100X GPU-powered machine. Four datasets, which were sourced from the TensorFlow datasets catalogue, were used in this study. This study's findings pointed out statistically significant differences between transformer-based models trained in zero-shot settings; however, the difference becomes negligible after a few examples in few-shot learning.

In this study [14], a comparative analysis of the T5 transformer was conducted on abstractive text summarization across three benchmark datasets: CNN/DailyMail, MSMO, and the XSUM. The evaluation criteria employed to gauge the efficiency of the T5 model were the BLEU and ROUGE metrics. The experiment was run on PyTorch with the Adam optimizer, with 1e-3 as the learning rate for model optimization across the training data's epochs and eight as the batch size. The MSMO dataset emerged as exceptionally high-performing, showcasing the T5 model's exceptional proficiency with the highest recorded scores. The findings of this study demonstrated that pre-trained transformers can produce concise summaries of the provided input text.

A comparative analysis of auto-encoder transformers, auto-regressive, and sequence-to-sequence-based models was conducted in this study [15] for both extractive and abstractive summarization. The dataset used in this study was the BBC news dataset, and the evaluation metrics were ROUGE-1, 2, and L, which provided a comprehensive analysis of the model's performance across different aspects of summarization quality. The experiments in this study were conducted using a varied set of hyperparameters to come up with the optimal set to achieve promising results. The findings of this study indicated that abstractive summarization takes more time than extractive but delivers a better summary in terms of coherence and fluency.

However, there is a need to conduct a comparative study under the same environment set-up to guarantee the comparability of the evaluation results, given that the existing studies in this domain have been conducted under different experimental variations. This has, therefore, inspired the need to conduct the comparative study using the same dataset and hyperparameters across the four selected models while ensuring fairness and uniformity without the risk of introducing bias and inconsistencies in the evaluation results.

3. METHODOLOGY

This section covers the transformer-based models selected for this comparative study, the detailed experimental set-up, the dataset, the training process, and the evaluation metrics used.

3.1 The Transformers

3.1.1 GPT

Generative Pre-trained Transformers (GPT) are a series of transformer-based models that OpenAI introduced. Predicting the subsequent token in a series using the previous context is a well-known autoregressive feature of GPT models. The multi-head attention and a position-wise feedforward network (FFN) are included in a stack of transformer layers that make up its architectural design. [16]. Massive text data corpora have been leveraged to pretrain these GPT models, and they have delivered cutting-edge performance in a number of natural language processing applications. With different transformer layers, attention heads, and model parameters,

GPT has different variants, such as gpt-2, gpt-2 medium, gpt-2 large, and gpt-2 xl.

3.1.2 T5

T5 (Text-To-Text Transfer Transformer) stands out as a versatile transformer model that was developed by Google Research. Its unique characteristic lies in a unified text-to-text framework that maps any NLP task as a text-to-text problem [17]. The architecture consists of a composition of encoders and decoders, where the encoder processes the input text and the decoder provides the corresponding output text. To handle input text effectively and make predictions, the encoder and decoder stacks integrate self-attention and feedforward neural network layers [17]. The final decoder layer's output undergoes processing through a dense layer with softmax as the activation function. T5 has various variants, such as T5-Small, T5-Base, T5-Large, T5-3B, and T5-11B, each distinguished by varying parameters.

3.1.3 BART

The transformer-based BART model, which Facebook AI released, refers to a Bidirectional and Auto-Regressive Transformer. BART operates on a typical architecture that follows a sequence-to-sequence approach to process input. The architecture is made up of a bi-directional encoder and a characteristic autoregressive bi-directional decoder [18]. One notable aspect of BART is its auto-regressive nature, where the model is trained to generate target sequences from corrupted input sequences. The encoder receives a corrupted version of the tokens, while the decoder receives the original tokens to mask particular works for prediction.

3.1.4 PEGASUS

PEGASUS is a sequence-to-sequence type of transformer explicitly designed for abstractive summarization. Developed by Google Research, PEGASUS is pre-trained using a large corpus through the gap-sentence generation approach, whereby some sentences are removed from the provided input sequence, and the transformer is required to restructure the output sequence from the provided input tokens in the form of a sequence that is corrupted [19]. The architecture of PEGASUS includes an encoder-decoder framework, similar to BART, and its pre-training objectives include Masked Language Modelling (MLM) and Gap Sentence Generation (GSG). Table 1 below shows the number of parameters across variants of the described transformer-based models and their specific types of architecture.

Table 1: Variations of the Transformer Models.

Transformer	Parameters		Type of Architecture
	base	large	
GPT	124M	762M	Decoder
T5	220M	770M	Encoder-Decoder
PEGASUS	175M	568M	Encoder-Decoder
BART	140M	400M	Encoder-Decoder

3.2 Experiment Set-up

3.2.1 Experimental Materials

The experiment was conducted using a Windows 10 (64-bit) ProBook laptop with 16GB RAM, AMD Radeon GPU, and core i7 Processor. Google Collaboratory environment was used to run the models on a Python Notebook. The Colab environment offered supplementary hardware accelerating GPUs and provided access to CUDA version 12.2, the Transformers library version 4.35.2, and PyTorch version 2.0.

3.2.2 Dataset Description

The Hugging Face hub provided the CNN/DailyMail dataset used in this experiment. This dataset is a widely adopted benchmark in natural language processing, particularly for applications such as text summarization. This dataset includes over 300,000 pairs of articles and the human-generated summaries that correspond with them. For this study, we used the 3.0.0 version of this dataset, which is a non-anonymized version specifically curated for abstractive summarization.

The CNN/DailyMail Dataset's train, validation, and test set distribution is displayed in Table 2 below. The train split contains 287,226 instances, whereas the validation and the test split have 13,368 and 11,490 instances, respectively.

Table 2: Distribution of the CNN/DailyMail Dataset

CNN/DailyMail Dataset Distribution	
Dataset Split	**No. of Instances**
Train Set	287,226
Validation Set	13,368
Test Set	11,490

3.2.3 Data Pre-Processing

Data preprocessing was conducted to prepare the dataset into a requisite format for the transformers. Tokenization was carried out to transform the input sequences into a numerical representation that could be fed into the models using the default auto tokenizers for each model. The tokenizer receives the input text and converts it into input tensors, each with a

corresponding attention mask and input ids. The input sequences were also truncated and padded to fit the set input length of 512 tokens to allow the model to process all the sequences simultaneously in a batch. Longer input texts were pruned off, while shorter inputs were padded with special tokens to fit the model's token limit. Data splitting was done guided by the conventions of deep learning into a ratio split whereby 70% went into the train set, 20% was used for validation, and 10% went to the test set.

3.2.4 Hyperparameters

All of the selected transformers in this study were subjected to the same set of hyperparameters to guarantee the comparability of the results. Therefore, we used a fixed number of attention heads in the model's architecture, set at 8, and configured the dimension of the feedforward neural network in each transformer block at 2048. Given that this was a fine-tuning task, a low learning rate of 1e-5 and a weight decay of 10^{-2} were adopted. The AdamW optimizer was set for these pre-trained models to optimize the model and avoid catastrophic forgetting of the learned representations. With a batch size of 6, each transformer model was trained for five epochs to minimize the memory consumed during each training and validation iteration. To expedite the training process, the number of steps required to accumulate gradients before executing an update pass was fixed at 16. The set number of epochs guided the mode in which evaluation was done and the save strategy for the models to regulate the number of saved checkpoints.

3.2.5 Model Training

The pre-trained models were loaded directly from the Hugging Face hub together with their learned weights. Training the models was initiated by calling the Trainer containing the defined training arguments to configure the training loop. All the parameters of the pre-trained models were updated over a lower learning rate to avoid undoing the learned representations from the pre-training steps. The training entailed calculating the loss function, backpropagating to calculate the gradient, and then updating the parameters guided by the defined gradient accumulation steps. This process was informed by the target task being similar to the pre-training task of the models, and also as observed in this study [20]. During the training phase, we adopted a learning rate of 1e-5, the AdamW optimizer for model optimization, and set cross-entropy loss as the requisite loss function. The per-device train and validation batch size were set to 6 to track the model's performance during the training loop.

3.2.6 Evaluation

This study used the ROUGE (Recall Oriented Understudy for Gisting Evaluation) metrics to offer a quantitative measure of how well the generated text aligns with the reference summary. Four different ROUGE metrics, namely, ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-Lsum, were used to evaluate the experimental results quantitively. These metrics provide a direct way of comparing the performance of different models, and they are a widely adopted evaluation benchmark in text summarization research [21]. The metrics were defined through the compute metric function and passed to the Trainer to return the rouge scores.

The mathematical formulation of the ROUGE metric is described below in the following equation.

$$ROUGE\text{-}N = \frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)}$$

Whereby;

{reference} denotes the reference summaries.

Count (N_n-$gram$) represents the count of n-grams in the reference summary.

$Count_{match}$ (N_n-$gram$) is the number of n-grams in the reference and generated summary.

The following is a step-wise description of the procedural algorithm describing the flow of the experiment, starting with the loading of the dataset up to model evaluation.

Algorithm 1: The flow of the experiment

Input: A sample input text

Output: Rouge Scores of the generated output

Begin Algorithm

i. Loading the Dataset
ii. Preprocessing the Dataset
iii. Load the Pre-trained Models from Hugging Face
iv. Define the training arguments
v. Train the models using the Trainer
vi. Evaluating the models using ROUGE metrics

End Algorithm

4. RESULTS

This section provides a comprehensive performance evaluation report of the four transformers—GPT, BART, T5, and PEGASUS. We present four ROUGE metric scores for each considered model, providing a detailed insight into their summarization capabilities. Table 3 below displays the recorded Rouge metrics, showing that T5 performs better, with

a rouge score of 35.12, 22.75, 32.82, and 28.59 for Rouge-1, Rouge-2, Rouge-L, and Rouge-Lsum, respectively.

Table 3: ROUGE scores of the Transformers.

Transformer	Rouge-1	Rouge-2	Rouge-L	Rouge-Lsum
GPT	24.83	16.92	22.14	21.07
BART	27.61	18.37	28.52	25.84
T5	35.12	22.75	32.82	28.59
PEGASUS	33.69	21.58	28.43	23.76

Based on the Rouge metrics, T5 is superior in comparison with the other models since this model returns the highest Rouge scores for the four Rouge metrics considered in this study. The ability of the T5 model to effectively capture unigrams and bigrams and maintain coherence in the generated summaries places T5 as the most effective model for a text summarization task. BART and PEGASUS closely compete with T5, with BART excelling in Rouge-L and Rouge-Lsum, while PEGASUS consistently performs well across all metrics but still falls short of T5. The GPT model exhibits lower Rouge scores overall, with 24.83, 16.92, 22.14, and 21.07 for the four Rouge metrics used in this study. Complementing Table 3, the bar graph in Figure 1 below also demonstrates the performance of each model on Rouge metrics.

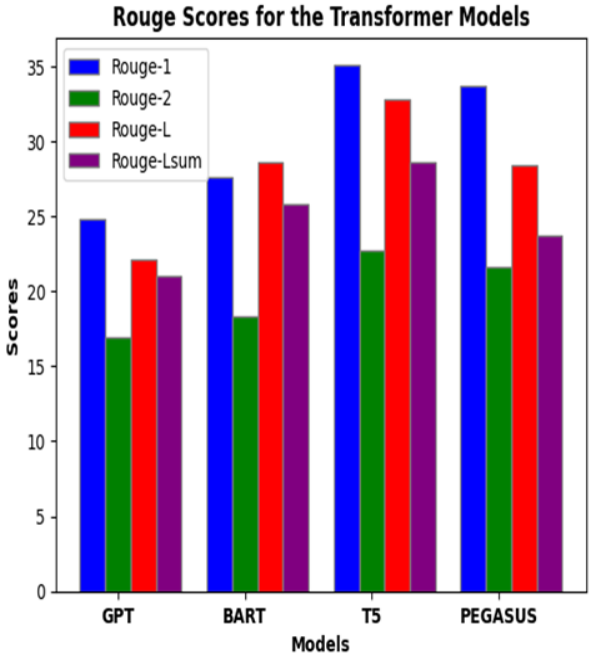

Figure 1: Rouge Scores for GPT, BART, T5, and PEGASUS on the CNN/DailyMail dataset

The time each model takes across the epochs of the training data offers insights into the convergence patterns of the models as they go through the training loop. The GPT model displays a comparatively constant reduction in time as the number of epochs rises, which is suggestive of the model's parameters and rapid convergence. T5 shows an initial increase in the first epochs followed by a stable gradual decrease, showing that this model requires a few epochs to adapt and stabilize. PEGASUS demonstrates an almost similar curve to GPT; however, it starts with a higher initial time, implying a potentially slower convergence rate. BART shows a steady reduction in the final epochs, emphasizing its ability to adapt to the training data despite a high initial time in the first epochs. The line graph in Figure 2 illustrates the convergence patterns of the models used in this study.

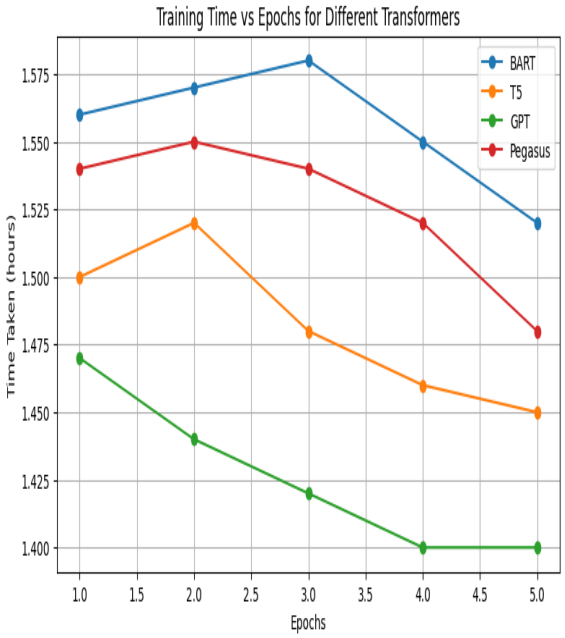

Figure 2: Training Time (Hours) taken by the Transformers across 5 epochs

4.1 Discussion

The performance metrics emphasize T5's notable superiority over GPT, BART, and PEGASUS. T5 consistently records high scores across all Rouge metrics, showcasing its exceptional ability to capture unigram and bigram overlaps and maintain coherence in generated summaries. This performance is attributed to its text-to-text framework that transforms any given natural language processing task into a text generation task using prefixes without changing its objective function or architecture. BART records competitive Rouge scores but falls slightly behind T5, particularly in Rouge-1 and Rouge-2, showing that it struggles in accurately representing unigrams and bigrams of a particular text since its decoder has to restructure the provided sequence from the

partially masked input sequences. PEGASUS excels in Rouge-L and Rouge-Lsum since the gap sentence generation objective masks the essential sentences from the input sequence. It feeds them to the decoder, which generates them together as a single output sequence. GPT trailing in all Rouge metrics shows that this architecture was ideally pre-trained for text generation compared to the other three models, which were pre-trained for summarization, among other closely related tasks.

The line graph in Figure 2 represents the time taken by four transformers (GPT, T5, BART, and PEGASUS) over five epochs, revealing distinctive patterns in their convergence behavior. GPT exhibits a consistent decrease in time, highlighting its steady convergence due to fewer parameters, making this model more diminutive than the other models with many parameters. T5 displays initial fluctuations in the first epochs but stabilizes and converges smoothly, demonstrating an adaptive learning curve across the training data, which is a desirable trait in neural networks. BART shows slow convergence, particularly in the early epochs, indicative of its complex model architecture. The curve displayed by this model is attributed to its autoregressive nature that makes its architecture operate bi-directionally. Although PEGASUS shows a consistent decrease in time, it starts with a higher initial duration, implying a potentially slower initial convergence due to its vast number of parameters and a large model dimension. The noted observations emphasize the need to consider the convergence pattern when selecting a transformer-based model tailored to specific task requirements.

5. CONCLUSION

The comparative study conducted in this paper for the transformer-based models for text summarization has underscored T5's summarization capability and desirable convergence behaviour. T5 consistently outperformed GPT, BART, and PEGASUS across all Rouge metrics, showcasing its proficiency compared to the other models. T5's convergence pattern also demonstrated an adaptive learning curve, showing the model's ability to converge desirably after a few epochs. The findings also highlight the crucial role of a closely aligned pre-training objective and model generalizability, indicating that optimal fine-tuning occurs when the pre-training objective is almost similar to the task at hand. These findings contribute to the broader understanding of natural language processing and offer valuable implications for future research in text summarization. Experimenting using datasets other than the CNN/DailyMail dataset could be part of this study's future research direction. Another potential direction for prospective study includes enhancing the T5, the best model, to provide better summaries.

REFERENCES

1. I. Awasthi, K. Gupta, P. S. Bhogal, S. S. Anand, and P. K. Soni, **"Natural language processing (NLP) based text summarization - A survey,"** in 2021 6th *International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, India: IEEE, doi: 10.1109/ICICT50816.2021.9358703.
2. P. Raundale and H. Shekhar**, "Analytical study of text summarization techniques,"** in 2021 *Asian Conference on Innovation in Technology (ASIANCON)*, PUNE, India: IEEE, Aug. 2021, pp. 1–4. doi: 10.1109/ASIANCON51346.2021.9544804.
3. A. P. Widyassari, A. Affandy, E. Noersasongko, A. Z. Fanani, A. Syukur, and R. S. Basuki, **"Literature review of automatic text summarization: Research trend, dataset, and method,"** in 2019 *International Conference on Information and Communications Technology (ICOIACT)*, Yogyakarta, Indonesia: IEEE, Jul. 2019, pp. 491–496. doi: 10.1109/ICOIACT46704.2019.8938454.
4. L. Tunstall, L. von Werra, and T. Wolf, **Natural language processing with transformers: building language applications with Hugging Face,** [First edition]. Sebastopol, CA: *O'Reilly Media*, Inc., 2022.
5. F. Zhang, G. An, and Q. Ruan, **"Transformer-based natural language understanding and generation,"** in 2022 16th *IEEE International Conference on Signal Processing (ICSP)*, Beijing, China: IEEE, Oct. 2022, pp. 281–284. doi: 10.1109/ICSP56322.2022.9965301.
6. A. Gillioz, J. Casas, E. Mugellini, and O. A. Khaled, **"Overview of the transformer-based models for NLP tasks,"** presented at the 2020 *Federated Conference on Computer Science and Information Systems*, Sep. 2020, pp. 179–183. doi: 10.15439/2020F20.
7. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). **Attention is all you need.** *Advances in neural information processing systems*, 30
8. H. S., A. S., A. V., and R. K. Grace, **"Summarization of news articles using transformers,"** in 2022 *5th International Conference on Advances in Science and Technology (ICAST)*, Mumbai, India: IEEE, Dec. 2022, pp. 159–163. doi: 10.1109/ICAST55766.2022.10039608.
9. H. Siddiqui, S. Siddiqui, M. Rawat, A. Maan, S. Dhiman, and M. Asad, **"Text summarization using extractive techniques,"** in 2021 *3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Greater Noida, India: IEEE, Dec. 2021, pp. 28–31. doi: 10.1109/ICAC3N53548.2021.9725501.
10. M. A. I. Talukder, S. Abujar, A. K. M. Masum, S. Akter, and S. A. Hossain, **"Comparative study on abstractive text summarization,"** in 2020 11th *International Conference on Computing, Communication and*

Networking Technologies (ICCCNT), Kharagpur, India: IEEE, Jul. 2020, pp. 1–4. doi 10.1109/ICCCNT49239.2020.9225657.

11. C. Orăsan, **"Automatic summarization: 25 years on,"** Nat. Lang. Eng., vol. 25, no. 06, pp. 735–751, Nov. 2019, doi: 10.1017/S1351324919000524.
12. J. Li, C. Zhang, X. Chen, Y. Cao, P. Liao, and P. Zhang, **"Abstractive text summarization with multi-head attention,"** in 2019 *International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary: IEEE, Jul. 2019, pp. 1–8. doi: 10.1109/IJCNN.2019.8851885.
13. T. Goodwin, M. Savery, and D. Demner-Fushman, **"Flight of the pegasus? Comparing transformers on few-shot and zero-shot multi-document abstractive summarization,"** in *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain pp. 5640–5646. doi: 10.18653/v1/2020.coling-main.494.
14. T. T, M. Borah, P. Dadure, and P. Pakray, "**Comparative analysis of T5 model for abstractive text summarization on different datasets,**" *SSRN Journal*, 2022, doi: 10.2139/ssrn.4096413.
15. A. Choudhary, M. Alugubelly, and R. Bhargava**, "A comparative study on transformer-based news summarization,"** in 2023 15th *International Conference on Developments in eSystems Engineering (DeSE)*, Baghdad & Anbar, Iraq: IEEE, Jan. 2023, pp. 256–261. doi: 10.1109/DeSE58274.2023.10099798.
16. Q. Zhu, L. Li, L. Bai, and F. Hu, **"Chinese text summarization based on fine-tuned gpt2,"** in *Third International Conference on Electronics and Communication; Network and Computer Technology (ECNCT 2021)*, Harbin, China doi: 10.1117/12.2629132.
17. C. Raffel et al., **"Exploring the limits of transfer learning with a unified text-to-text transformer,"** *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, Jan. 2020.
18. Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. "**Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.**" *arXiv preprint arXiv:1910.13461* (2019).
19. J. Zhang, Y. Zhao, M. A. Saleh, and P. Liu, "PEGASUS: **Pre-training with extracted gap-sentences for abstractive summarization,"** *International Conference on Machine Learning*, vol. 1, pp. 11328–11339, Jul. 2020.
20. J. Kamiri, G. M. Wambugu, and A. M. Oirere, **"A comparative study of deep learning and transfer learning in detection of diabetic retinopathy,"** *IJCATR*, vol. 11, no. 07, pp. 247–254, Jul. 2022, doi: 10.7753/IJCATR1107.1001.
21. M. Bhandari, P. Gour, A. Ashfaq, P. Liu, and G. Neubig, "**Re-evaluating evaluation in text summarization,"** 2020, doi: 10.48550/.2010.07100.