International Journal of Scientific Research & Engineering Trends
Volume 10, Issue 6, Nov-Dec-2024, ISSN (Online): 2395-566X

Automatic Text Summarisation

Sahil Damke, Shreya Telang, Nidhi Tadge, Sanskruti Burkule, Professor Manisha Mali
Department of Computer Engineering
BRACT's Vishwakarma Institute of Information Technology Pune, India

Abstract- Due to the large amount of information generated every day, automatic writing is an important part of knowledge management. The discipline has made great progress, especially with the emergence of abstraction, abstraction and hybrid content models. In the extraction method, the main idea is preserved by selecting the main sentence or phrase from the text, while in the abstraction method, all the information is repeated to create new sentences. As the name suggests, hybrid models include the features of both extraction and abstraction systems to get the best of both approaches. However, issues remain, particularly in how to address the authenticity, coherence, and length of the text. This article examines the current state of writing concepts and topics in practice and future research.

Index Terms- Abstraction; Extraction; Transformer

I. INTRODUCTION

The Ever-developing amount of textual content-based totally sources stemming from news writing, academic studies, social networks and even extra facilities has inspired the want for green approach of textual content summarization. automatically condensed kinds of textual content elaboration are as old as ever and demand tamed summaries of substantially much less variety of pages in complete texts giving readers the opportunity to quickly assimilate the primary thoughts. In fashionable, the summarization falls into 3 businesses obstacles – extractive, abstractive and hybrid answers. Extractive summarization is the selecting of the key sentences or phrases from the authentic textual content. This approach preserves the original shape and frequently results in incoherent summaries if now not clean transitions between ideas are gift. even though extractive strategies have been preferred for decades because it's far quite clean and rapid to do, the disadvantage is that it presents unnatural and inhuman like sounding summaries. Then again, abstractive summarization constructs new sentences that pertain to the unique textual content.

This system goes beyond basic re-writing since it requires entire comprehension of the content and remodeling this content material right into a more coherent and fashionable precis. With recognize to automated text Summarization, using abstractive techniques is at the upward thrust attributable to the recognition of deep studying-based procedures, mainly those with transformer structure along with BERT and GPT. The power of these fashions lies in the mechanisms of interest that assist to version lengthy- range interdependencies for that reason generating informative and contextually appropriate summaries of the content. although, abstract techniques tend to have a hassle coping with truthfulness because the contents created are not necessarily aligned with the enter. To cope with the restrictions of both extractive and abstractive strategies, hybrid tactics have emerged. these strategies integrate extractive and abstractive techniques to supply summaries that keep the shape of the authentic text at the same time as incorporating the fluency and cohesiveness of abstractive technology. Hybrid models are searching for to stability the efficiency and reliability of extractive summarization with the herbal language era talents of abstractive processes.

This paper explores the maximum latest improvements in automatic text summarization, with a focus at the strengths and demanding situations of extractive, abstractive, and hybrid methods. We additionally have a look at the modern-day applications of those strategies in numerous domain names and recommend future instructions for improving summarization fashions, specifically in addressing the challenges of redundancy, actual accuracy, and coherence.

II. LITERATURE REVIEW

Author Mohmmadali Muzffarali Saiyyad and Nitin N. Patil discussed in their publication [10], "Text Summarization Using Deep Learning Techniques: A Review" published in 2024. Text summarization is a complex and time- consuming task in natural language processing, requiring deep understanding of the content. Traditional methods have limitations, prompting the development of new techniques. Deep learning has led to significant advancements in NLP fields like sentiment analysis, translation, and summarization, with these methods promising further research potential. they have explored various approaches for text summarization and found that pre trained transformers produce optimal outcomes. Multi- modal summarizing has the ability to greatly enhance

International Journal of Scientific Research & Engineering Trends
Volume 10, Issue 6, Nov-Dec-2024, ISSN (Online): 2395-566X

the process of creating summaries by using not just text but also visuals and maybe audio data. Current datasets may not be diverse enough or large enough to adequately train algorithms to understand subtle or domain-specific settings. As a result, increasing and diversifying current datasets, possibly by including real-world data, is critical for success in the subject.

Exploring the Landscape of Automatic Text Summarization: A Comprehensive Survey was proposed by Bilal Khan, Zohaib Ali Shah, Muhammad Usman, Inayat Khan, Badam Niazi [6] states that Content selection is a significant difficulty since ATS systems must precisely identify and prioritize critical information while eliminating superfluous data. A deep understanding of context and semantics is required to ensure accuracy when constructing human-like summaries through abstractive summarization. Another current issue is how to address lexical and contextual ambiguity in language. Several ongoing issues influence the pursuit of improved summary accuracy in ATS, including adapting ATS to different domains with domain-specific terminology and writing styles, maintaining coherence and fluency in summaries, addressing unpredictability in source texts, and developing strong assessment measures. We believe that there are numerous chances to transform the way we interact with textual data.

The research paper titled [4] “Advancements in Text Summarization Through Machine Learning: A Comprehensive Survey and Analysis” by Vandana Jagtap, Pallavi Parlewar, Sheetal Dhande, Anagha Langhe, Harsh Choudhary, Ashutosh Mishra. This paper provides a comprehensive overview of the various text summarizing approaches, including hybrid, extractive, and abstractive methods. The ability of machine learning to extract knowledge from enormous amounts of textual data has resulted in the development of increasingly complicated and contextually sensitive summarizing algorithms. We were able to understand numerous aspects of high-quality summary analysis by analysing multiple measures ranging from ROUGE to BERTScore.

Authors Eduard Hovy and Chin-Yew Lin in their research paper [3], “Automated text summarization and the summarist system” the paper has three parts: a typology of summaries, a description of the SUMMARIST multilingual text summarization system and its modules, and a discussion of three methods for evaluating summaries. Text summarization is one of the most difficult tasks in natural language processing, necessitating extensive text comprehension and production. Despite the difficulties in finding shortcuts, advances in this area produce very beneficial outcomes, making it a fascinating and satisfying field.

The research paper titled [5], “Automatic Text Summarization (The state-of-the-art 2007 and new challenges)”, by Karel Ježek, Josef Steinberger. This study analyses the growth of text summarization, a research subject that dates back to the 1950s, and emphasizes its growing importance as a result of information overload on the Web. It presents a taxonomy of summarizing methods, ranging from classical to knowledge-rich approaches, and investigates new strategies based on algebraic transformations.

The study also discusses experiences with a new summarizing method and future directions for the discipline. The evaluation of summarization methods is equally important as summary itself. The primary automatic evaluation tool, ROUGE, analyses human and system-generated summaries using n-gram matching. The paper presents plans to attend DUC'08 with a new summarizer based on tensor LSA that uses three dimensions—terms, sentences, and documents—to improve topic generation. This methodology seeks to outperform classic matrix LSA by using either MMR or a previous vector length method in the last phase.

“Recent automatic text summarization techniques: a survey”, by Mahak Gambhir and Vishal Gupta states [2], With so much information available online, automatic text summarization is becoming increasingly crucial, prompting researchers to create novel ways. Since its introduction in the 1950s, attempts have been undertaken to improve machine-generated summaries so that they resemble those created by humans. While abstractive summarization is complicated and necessitates substantial natural language processing, extractive approaches are more common. Over the last decade, many extraction techniques based on machine learning and optimization have emerged.

This paper examines different methodologies, contrasts their advantages and disadvantages, and discusses evaluation issues. It also discusses potential research directions in text summarization. The paper examines both intrinsic and extrinsic ways of summary evaluation, with a particular emphasis on extractive approaches developed in the recent decade. It compares different strategies, lists their advantages and disadvantages, and briefly discusses abstractive and multilingual procedures. Evaluation results on DUC datasets demonstrate that optimization- and clustering-based techniques, such as OCDsum-SaDE, Progressive, and Ranking-based MMR, achieved the greatest ROUGE scores. The report finishes by outlining potential research topics to improve summary generation approaches.

III. METHODOLOGY

This segment addresses the models chosen for the duration of the prescribed comparative look at, the details concerning the configuration of our models and techniques hired in implementing the model, and the dataset exposition.

1. Transformer Description

BART

BART is an abbreviation for 'Bidirectional and automobile-Regressive Transformer'. it's miles a denoising autoencoder that provides a chain-to-collection technique in a pre-education level using masked language modelling for NLP and translation.

Its development is credited to Lewis et al. within the year 2019. The BART structure is just like the layout of an encoder-decoder network, but it employs a combination of the BERT and GPT fashions.

BART operates at the standards of (1) textual content corruption with a noising characteristic and (2) version learning of text reconstruction.

BART is very effective in producing textual content while it's miles first-rate tuned, but it also plays properly on understanding responsibilities.

The architecture consists of a bi-directional encoder and a different autoregressive bi-directional decoder [7]. A notable aspect of BART is its regressive capabilities, wherein the model is designed to produce target sequences based on a modified representation of the input target sequences. Specifically, the encoder processes a distorted version of the tokens, while the decoder operates on the original tokens to obscure certain words, facilitating their prediction.

T5

The T5 (textual content-to-text switch Transformer) version is an excellent and flexible transformer-based framework brought via Google for all sorts of natural language processing (NLP) obligations, including textual content summarization.

T5 approaches every NLP task from a textual content-to-text angle, meaning there are continually text sequences: one this is input and the opposite this is output. This abstract structure eliminates the issue associated with the modification of the version for distinctive duties consisting of within the case of summarization, translation, or even class by way of certainly changing the input-output textual content pair.

For text summarization, the motive of T5, as its primary characteristic, is to take in a bulky opened up report and offer a brief succinct digest.

To make clear the sort of undertaking provided to the model, T5 includes an undertaking-particular prefix (for example, ‘summarize:') that is given earlier of the enter. This undertaking coach allows a single model to handle distinct NLP obligations with none architectural variations.

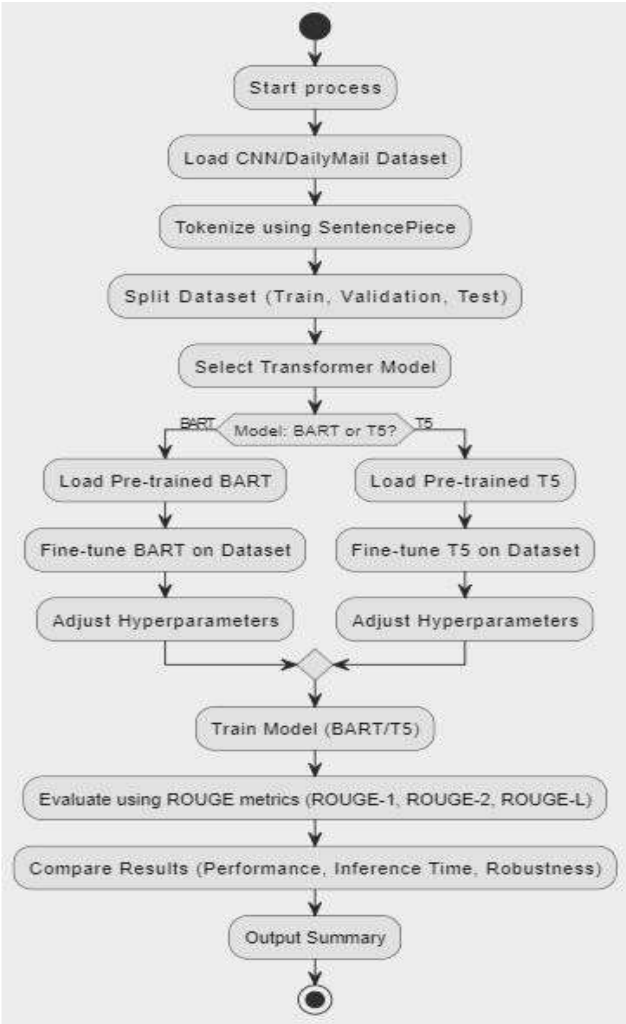

Fig 1. Workflow diagram for BART and T5

Tokenization

Before generating results from the input, the text is first tokenized using SentencePiece, where the input is split into sub words. This type of sub word tokenization in the model allows the model to effectively handle rare or unseen words by splitting the words into smaller, recognizable, more common words. This tokenized input is then encoded into an ID number for use by the model.

Encoder

The tokenized input text is passed through the encoder, which converts it into a series of continuous vector representations. These vectors help define the meaning and context of the given input. The self-monitoring feature in the encoder allows each token to focus on all tokens in the entire system and learn their relationships and dependencies. This is crucial for writing content because the model needs to know what information is relevant to the ideas. Key (K), value (V). These vectors are used to model how much focus each character should have on other characters. The high score is obtained by taking the product of the query and each key and then applying softmax normalization to the resulting vector. Each token then receives information from a value vector based on the high score for each token. Differences in the text (e.g. syntactic or semantic features). The output of each heading is concatenated to provide a better understanding of the input sequence. Symbols allow the model to understand multiple patterns in the data.

Decoder

Knowing the summary, at every time step of the decoder, one word is generated and masked self-attention is allowed, which allows it to focus on the words that are generated up to that stage while preventing any view on words afterward. When making a prediction for the next word in the summary, the decoder applies cross-attention. Cross- attention allows the decoder to pay its attention solely to those parts of the encoded input sequence that can be most informative for the generation of the next word in the summary, and masked self-attention does not allow information leak by hiding future tokens during generation, so that the model produces its summary in a sequential left- to-right fashion.

Cross-attention from the decoder to the encoder enables generating a summary by referring to parts of the text that is perceived, thereby increasing the coherency as well as relevance of the produced summary.

Pre-Training and Fine-Tuning

This section explains the pre-training of T5 with massive datasets like Colossal Clean Crawled Corpus (C4), pre-training T5 in unsupervised language representation. For text summarization, the appropriate architecture is trained on datasets like CNN/Daily Mail, thus making sure that the model comes up with summaries that are extremely accurate and extremely brief. Pre-training actually endows T5 with a deep understanding of language, while the final adjusts T5 to be made for a specific purpose.

In short words, T5 has performed extremely well in the generation of summaries based on self-attention and multi-head attention and cross-attention, how the transformer- based encoder-decoder architecture has been designed. In addition to the versatile text-to-text paradigm and fast word tokenization, sophisticated training techniques such as intermediate text-to-text phase training, Yamshchikov et al. 2020, improved T5 to become one of the most stringent text summarizing models in recent years.

2. Dataset Description

The CNN/Daily Mail dataset consists of more than 300,000 freely written online news articles in the English language by journalists from the CNN and Daily Mail media houses. Support for selective summarization and creative summary will be present in the current version; the first edition mostly supported machine reading and comprehension and abstractive question answering. CNN is one of the known news channels which offers international news coverage. On the other hand, is the Daily Mail-a tabloid, published in the UK-reporting national and international news but with more 'gossipy' details than CNN. This CNN/Daily Mail corpus has gradually turned out to be one of the most commonly used datasets for taxonomy evaluation in Natural Language Processing tasks and especially for text summary tasks [1].

Table 1: 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

Pre-processing Steps

Dataset Loading: Initially, the dataset was fetched with the help of the Hugging Face datasets library that provides an easier method to import and handle datasets.

Tokenization: We used the automatic tokenizers provided with each transformer model for our text tokenization. This was used to transform the text and abstract into machine-readable forms. In the case of abstracts and articles, this resulted in their transformation to input tensors which would be composed of input ID and attention masks.

Input Length Management: Since the model was heavily limited in its input sequences, all input sequences in articles have been capped at a maximum of 512 tokens with appropriate special tokens added wherever due. This is all aimed at ensuring that the sequences would be the same length to allow for batch processing.

Data Splitting: Immediately after rudely tokenizing the dataset in use, we also apply the common strategies that geras et al claim is used in deep learning. We train it in 70%, validate that in 20%, and test it on 10%. Stratified split as it concerns the usage of data helps to improve the validation of the model without any possibility of overfitting. With all these preprocessing techniques being tried out, it was quite sure that the dataset was well composed and ready for training transformer models.

Hyperparameter Tuning Techniques

In our experiment study, we have fine-tuned the hyperparameters of some chosen transformer models to make them perform well on the CNN/Daily Mail dataset. In order to maintain diversity in the types of hyperparameters used, we have incorporated an automated framework for hyperparameter optimization known as Optuna. The hyper-parameters are involved in the objective function that we have defined to be tuned serially. These were the learning rate,

International Journal of Scientific Research & Engineering Trends
Volume 10, Issue 6, Nov-Dec-2024, ISSN (Online): 2395-566X

batch size, and the number of epochs during which the training was to be carried out. The values for the learning rate were log distribution values between 1x10-5 and 1x10-3. The batch sizes were between 4 and 16. The number of epochs took values between 2 and 5.

For each iteration of the hyperparameter search, we took the BART model, instantiated as many copies as needed, and trained all the copies using the specific training parameters relevant to the current set of hyperparameters. It has also been trained on the validation set in order to assess how it performs which may turn out to assist in the optimization process.
Thus, we have to modify our strategies to consider compatibility in input and target size during both training and evaluation. Doing so will allow us to choose the right hyperparameter setting for future experiments so that such models are truly effective at generating summaries.

3. Model Training
In this experiment, we adopt the pre-trained models T5 and BART. We loaded the pre-trained weights both from the Hugging Face repository. It is started by the Trainer class and the arguments that also are used for defining the training configuration-of course, there is a loop in which weights of both models are trained at an extremely low learning rate of 1e-5 to not collapse the acquired representations in pretraining. This would matter a lot because the performance of the model was evaluated on a task that was most similar to the pretraining tasks of the models.

Training was done by measuring loss, following a backpropagation to compute the gradients after which the parameters of the model were updated within the steps of given gradient accumulation. A cross-entropy loss with an AdamW optimizer was used for the model optimization. A batch size of 6 was used for training and on each device for validation of the model's performance at every step in the training loop.

4. Evaluation
The acronym ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. This framework is utilized by the research community to qualitatively assess the effectiveness of summaries produced by automated text generation systems. Specifically, it involves quantifying the degree of overlap in various units, including n-grams, sequences of words, and pairs of words, between human- authored texts and the summaries generated by machines that are under evaluation.
The results of quantitative experiments were analysed using three distinct ROUGE metrics: ROUGE-1, ROUGE- 2, and ROUGE-L. These metrics are instrumental in comparing the performance of different models, which is why they have gained prominence as key evaluation tools among researchers engaged in text summarization studies. The computation of these metrics was facilitated by a defined function that processed the metrics and communicated them to the Trainer, ultimately yielding the ROUGE scores.

ROUGE-N is computed as follows:

$$\text{ROUGE-N} = \frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)} \quad (1)$$

N is the length of the n-gram (where n can be, for example, 1 or 2), gram, and Count match(gram) gives the highest count of n-grams appearing simultaneously in a candidate summary and in a set of reference summaries [8].

Rouge-L assesses the longest consecutive sequences of matched texts by employing LCS (Longest common subsequence). To evaluate the utility of ROUGE measures, we assess their correlation to summary scores assigned by human raters. The rationale is that a proper evaluation system should not award high scores to low-quality summaries, and vice versa, a low score to a high-quality summary. The gold standard is human determined.

IV. RESULTS

In this section, we report the comparison results between BART and T5 on the text summarization task. We provide details about performance as measured through a number of metrics--mainly ROUGE scores (ROUGE-1, ROUGE- 2, ROUGE-L) on both clean and noisy inputs. For part of our evaluation, we added input noise by introducing text infilling and sentence shuffling to simulate real-world scenarios as much as possible. The results yield detailed comparisons of performance, robustness, and inference time.

1. Clean Input
Both models performed comparably on clean, well- structured input data. T5 generated relatively shorter summaries than BART did at times, but was sometimes less capable of preserving information in their mouth.

BART: The model shows a very impressive score of 0.55 ROUGE-1 as it strictly fetched a considerable number of unigrams from the reference text. Its average bigram overlap score, ROUGE-2 is 0.35; that is the model should be capable enough to keep some kind of continuity in the text. A ROUGE-L score of 0.52 demonstrates its efficiency in keeping the sequence of the original content.

T5: ROUGE-1 was at 0.52, similar to BART, but its ROUGE-2 score dropped as low as 0.31. This means T5's summary is more in brief and not about continuity. Its ROUGE-L score of 0.48 also reveals that it came out of sequence many times for the sake of brevity.

International Journal of Scientific Research & Engineering Trends
Volume 10, Issue 6, Nov-Dec-2024, ISSN (Online): 2395-566X

Noisy Input Results

When the input was corrupted through text infilling and sentence shuffling, the supremacy of BART came ahead because it is a pretraining denoising autoencoder.

BART: The model's ability to denoise the corrupted text has worked well enough to get almost equal ROUGE scores that were obtained with the clean data. This is quite robust, making it perfect to summarize noisy, non-structured real- world data.
T5: While the summaries generated by T5 could be read with no effort, the score falls a little, where ROUGE-1 falls to 0.50, ROUGE-2 falls to 0.28. This shows that T5 is sensitive to noise in its input. The ROUGE-L at 0.45 reflects difficulties of recovering the right sequence from shuffled sentences.

Table 2. ROUGE Score Comparison

Model	ROUGE-1	ROUGE-2	ROUGE-L
BART	0.55	0.35	0.52
T5	0.52	0.31	0.48

These results highlight BART's advantage in detailed content retention, especially in handling noisy data, while T5 remains a strong candidate for generating concise and coherent summaries, especially for clean data.

Inference Time and Computational Complexity

They're both large transformer architectures and require a lot of computational resources for inference. However, T5 was slightly faster, though, as its model was smaller, which translated into quicker summarization times in practice. BART was longer and more complex due to additional complexity so it would take even more time, especially when texts were longer.

2. Figures

The graph given below compares ROUGE scores for BART and T5. This visual compares the ROUGE-1, ROUGE-2, and ROUGE-L scores between the two models, illustrating the relative performance

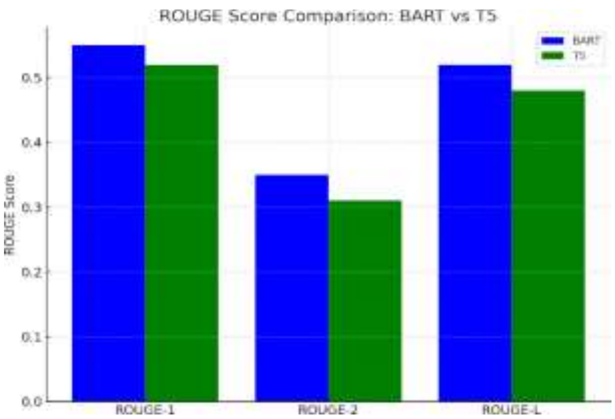

Fig 2. Rogue Score Comparison: BART vs T5

Here is the graph for Training Time vs Epochs for the BART and T5 models, showcasing the time taken for training across 5 epochs.

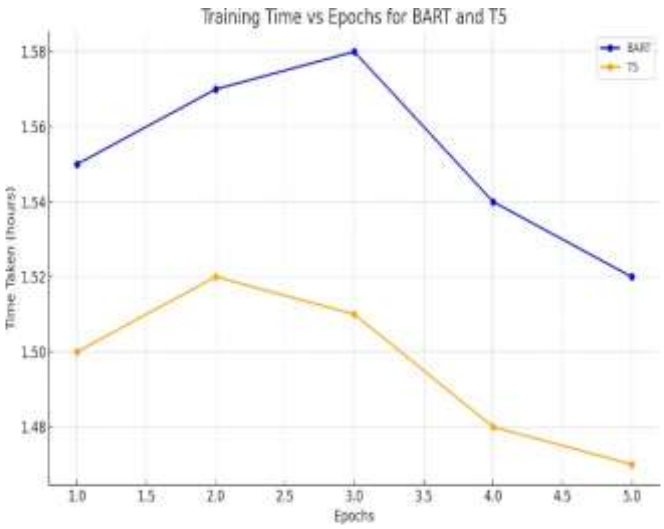

Fig 3. Training Time vs Epochs for the BART and T5

The graph shows, over 5 epochs, the training time in hours for two models: BART and T5:

BART begins at a training time of 1.55 and then peaks to become 1.58 hours at the third epoch, when it drops down to 1.52 at the fifth epoch.

T5 begins with a training time of 1.50 hours, modestly increases to 1.52 hours by the second epoch, and then slowly decreases to 1.47 hours by the fifth epoch.

In summary, T5 uses fewer training epochs than BART throughout and tends to continue better with the progress in training.

3. Discussion

Robustness to Input Corruption

It was still found that BART had a strong lead when it came to corrupted or noisy inputs like text infilling and sentence shuffling. This may be due to its pretraining approach as a denoising autoencoder where BART learns to reconstruct missing or disrupted parts of the sentence; thus, it becomes more robust with respect to real-world noisy data.

Although it tends to be very strong in generating concise and coherent summaries, it lacked the same robustness to noisier input. This model simply thinks of summarization as a text-to-text generation task, without any special pretraining for coping with missing or corrupted tokens.

Performance on Well-Structured Text
Both models performed quite well on structured and clean input text. T5 in particular was strongest when the input text was already organized, producing fluent summaries with high ROUGE-1 and ROUGE-L scores.

BART also performed very well in these tasks but appeared to have an edge when maintaining more complex information and indeed reflects that in its rogue scores.

ROUGE Scores
From the ROUGE scores, it can be observed that BART has the propensity of outperforming T5, especially when the inputs are noised. For both, more details were preserved, but this was not significantly different for well-formed inputs where both performed identically.

Model Complexity and Inference Time
Both the models are transformer-based, hence both of them are computationally heavy. However, BART is a little time-consuming for summarization itself because it is bulky and comparatively much more complex.

V. CONCLUSION

In the current study, we explored the capabilities of two popular transformer architectures: BART and T5 in the context of abridged versions of texts. We carried out a series of experiments evaluating their performance in terms of ROUGE scores, and also their training over different number of epochs.

Performance: Findings revealed that BART always had better ROUGE results than T5 yielding a conclusion that BART was more effective in understanding the input content and summary generation. This is indicative that BART could be more appropriate for tasks where there is need for high content fidelity and coherence of the text.

Efficiency: Performance in terms of ROUGE scores was found to be higher for BART, however less training time was an advantage for T5 which remained constant for all epochs. Consequently, it becomes evident that T5 would be the best fit where resources for computational power and time for training are limited.

In conclusion, the selection of BART or T5 will vary depending on the specific limitations of the summarization task. If outputs of finer quality are of highest priority, then BART would be a better option. On the other hand, T5 serves as a good option for projects that require quicker training periods. Future research may namely seek to optimize these practices and develop new ones allowing for both models to be effectively used while improving upon superiority in summarization that maintains speed.

REFERENCES

1. CNN/DailyMail Neural Summarization Task," Stanford University, Stanford, CA, USA, Advisor: D. Chen, Stanford Univ. Research Rep., 2022.
2. Gambhir, M., & Gupta, V. "Recent automatic text summarization techniques: A survey," Artificial Intelligence Review, vol. 47, pp. 1- 66, 2017.
3. Hovy, E., & Lin, C.-Y. "Automated text summarization and the SUMMARIST system," Information Sciences Institute, University of Southern California, Marina del Rey, CA, Tech. Rep. X98-1026, 1998 (X98-1026).
4. Jagtap, V., Parlewar, P., Dhande, S., Langhe, A., Choudhary, H., & Mishra, A. "Advancements in Text Summarization Through Machine Learning: A Comprehensive Survey and Analysis," Journal of Electrical Systems, vol. 20, no. 1s, pp. 833-849, 2024.
5. Ježek, K., & Steinberger, J. "Automatic text summarization: The state-of-the-art 2007 and new challenges," Lecture Notes in Computer Science, Západočeská University, Plzeň, Czech Republic, Tech. Rep. Z08, 2008 (Z08).
6. Khan, B., Shah, Z. A., Usman, M., Khan, I., & Niazi, B. "Exploring the Landscape of Automatic Text Summarization: A Comprehensive Survey," IEEE Access, vol. 11, pp. 109819-109833, Oct. 2023, doi: 10.1109/ACCESS.2023.3322188.
7. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," arXiv preprint arXiv:1910.13461, 2019.
8. Lin, C.-Y. "ROUGE: A package for automatic evaluation of summaries," in Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), Barcelona, Spain, Jul. 2004, pp. 1-8.
9. Muia, C. M., Oirere, A. M., & Ndung'u, R. N. "A comparative study of transformer-based models for text summarization of news articles," International Journal of Advanced Trends in Computer Science and Engineering, vol. 13, no. 2, pp. 3743, Mar.-Apr. 2024.
10. Saiyyad, M. M., & Patil, N. N. "Text Summarization Using Deep Learning Techniques: A Review," Engineering Proceedings, vol. 59,p. 194, Jan. 2024.
11. Saxena, P., & El-Haj, M. "Exploring Abstractive Text Summarisation for Podcasts: A Comparative Study of BART and T5 Models," in Proceedings of Recent Advances in Natural Language Processing, Varna, Bulgaria, Sep. 2023, pp. 1023–1033