Islamic University Journal of Applied Sciences (IUJAS)
https://journals.iu.edu.sa/jesc

Volume VIII, Issue II, December 2026, Pages 39-56

Assessing Transformer Models for Abstractive Summarization of Scientific Articles

Emad Nabil *

Computer Science Department, Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah 42351, KSA, emadnabil@iu.edu.sa

AI Center, Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah 42351, KSA

*Corresponding Author: (Emad Nabil), *Email Address*: emadnabil@iu.edu.sa

Abstract

The rapid growth of academic literature has intensified the need for effective automatic text summarization techniques capable of producing concise and informative representations of scientific documents. While extractive methods are widely used, they are limited in their ability to generate coherent and semantically rich summaries. Recent advances in Transformer-based architectures have enabled significant progress in abstractive summarization; however, their effectiveness on domain-specific datasets, such as scientific articles, remains an open challenge. In this study, we investigate the performance of three pre-trained Transformer-based models—T5, BART, and GPT-2—on the task of abstractive summarization using the CL-SciSumm 2019 dataset. A total of 19 experimental configurations are conducted to analyze the impact of generation parameters, including beam size, length penalties, and n-gram constraints, on summarization quality. The models are evaluated using ROUGE metrics, with a focus on ROUGE-2.To complement content-based evaluation, this work incorporates linguistic acceptability assessment using the Corpus of Linguistic Acceptability (CoLA), a benchmark dataset for evaluating grammatical correctness. The results show that BART achieves the best performance with an ROUGE-2 F1-score of 0.40664, while T5 demonstrates superior grammatical acceptability, achieving 93.36%, but BART achieves a very near performance to T5. Ultimately, these findings demonstrate the potential of pre-trained neural networks, particularly the BART architecture, to drive the future of complex, generative NLP applications, transforming how academic research is processed and understood.

Keywords: Abstractive summarization; Text summarization; Pre-trained Models; CL-SciSumm 2019

https://doi.org/10.63070/jesc.2026.033
Received 06 April 2026; Revised 20 May 2026; Accepted 10 June 2026; Available online 15 July 2026.
Published by Islamic University of Madinah on behalf of ***Islamic University Journal of Applied Sciences.***
This is an open access article published under the Creative Commons Attribution 4.0 International license (CC BY 4.0):https://creativecommons.org/licenses/by/4.0/

1. Introduction

Over the past two decades, the volume of digital textual content—ranging from news articles and social media posts to books and scholarly publications—has grown exponentially [1]. This rapid expansion poses significant challenges for efficient information retrieval and knowledge synthesis, thereby necessitating advanced solutions in the field of Natural Language Processing (NLP). NLP enables machines to process and generate human language through two primary components: Natural Language Understanding (NLU), which focuses on extracting semantic meaning and contextual relationships, and Natural Language Generation (NLG), which produces coherent human-readable text [2].

Within this domain, Automatic Text Summarization (ATS) has emerged as a critical task aimed at condensing large volumes of text into concise and informative representations [3]. In academic research, summarization is particularly important, as researchers are required to review extensive and continuously growing bodies of literature. The scale and complexity of scholarly publications make manual analysis increasingly time-consuming, thereby highlighting the need for effective automated summarization systems [3].

Although most academic papers include author-written abstracts, relying solely on them is often insufficient. Abstracts may not fully capture the detailed contributions of a study and can reflect the authors' subjective emphasis. Moreover, abstracts frequently serve as concise previews rather than comprehensive and objective representations of the underlying work [4]. Consequently, automated summarization systems can provide more balanced and task-specific summaries of scientific documents.

Summarization approaches are typically categorized into extractive, abstractive, and hybrid methods. Extractive approaches select and concatenate salient sentences directly from the source text, often preserving grammatical correctness but lacking flexibility and coherence. In contrast, abstractive summarization aims to generate novel sentences that capture the underlying meaning of the source material, requiring deeper semantic understanding and advanced language generation capabilities. Hybrid approaches attempt to combine the strengths of both paradigms [5].

Recent advances in deep learning, particularly Transformer-based architectures, have significantly improved the performance of abstractive summarization systems. Pre-trained models such as T5, BART, and GPT-2 leverage large-scale training and self-attention mechanisms to model complex linguistic

patterns and generate coherent summaries. However, the effectiveness of these models on domain-specific tasks, such as scientific article summarization, remains an active area of research due to challenges including long document structures, specialized terminology, and the limited availability of high-quality annotated datasets.

In this study, we investigate the performance of several pre-trained Transformer-based models—namely T5, BART, and GPT-2—on the task of scientific article summarization using the CL-SciSumm 2019 dataset. A series of controlled experiments is conducted to evaluate how different model configurations influence summarization quality. In addition to standard evaluation using ROUGE metrics [6], this work also explores the use of linguistic acceptability measures to assess the grammatical quality of generated summaries.

The main contributions of this work can be summarized as follows:

- A comparative evaluation of multiple Transformer-based models for abstractive summarization of scientific texts.
- An analysis of the impact of generation parameters on summarization performance.
- An exploration of grammatical acceptability as a complementary evaluation criterion alongside traditional metrics.

The remainder of this paper is organized as follows. Section 2 reviews related work in scientific text summarization. Section 3 describes the proposed methodology and model configurations. Section 4 presents the experimental setup and results. Section 5 concludes the paper and outlines directions for future work.

2. Literature Review

2.1.Evolution of Text Summarization

Automatic Text Summarization (ATS) has been an active area of research for several decades. Early approaches were predominantly extractive, relying on statistical and heuristic methods to identify and select the most salient sentences from a document. These methods were constrained by limited Natural Language Understanding (NLU) capabilities and often failed to capture deeper semantic relationships within the text.

With advances in computational linguistics and machine learning, summarization techniques have evolved toward more sophisticated approaches. In particular, abstractive summarization has gained increasing attention due to its ability to generate coherent and concise summaries that are not restricted to the original wording of the source text. This shift has been largely driven by improvements in deep learning models, especially Transformer-based architectures, which enable more effective modeling of long-range dependencies and contextual information. Figure 1 illustrates the general framework of automatic summarization.

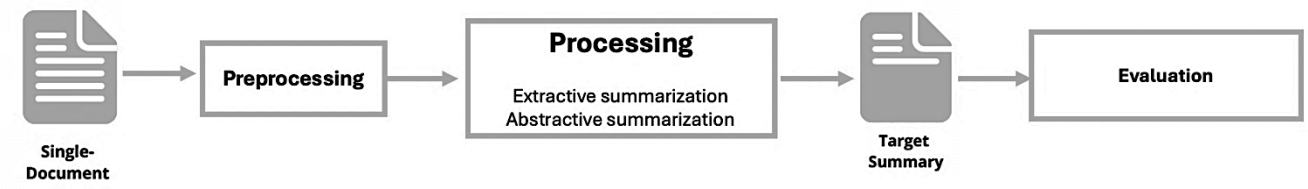


Figure 1: Illustration of summarization approach*es*.

2.2.Summarization of Scientific Documents

Summarizing scientific literature presents unique challenges compared to general-domain text summarization. Scientific documents are typically lengthy, structurally complex, and contain domain-specific terminology. To address these challenges, a variety of approaches have been proposed in the literature.

Early work by [7] employed statistical methods combined with GloVe word embeddings for extractive summarization of scientific articles. Their findings highlighted the importance of adapting summary length dynamically based on document characteristics rather than enforcing fixed-length summaries.

Several studies have focused on leveraging citation information to improve summarization quality. For instance, the work in [8] introduced an automated scientific document summarization system evaluated on the CL-SciSumm 2016 and 2017 datasets. Their approach consists of a two-stage pipeline: first identifying relevant text spans based on similarity between citation contexts and reference documents, followed by optimization using a Differential Evolution framework (MOOSciSumm). Similarly, the NaCTem-UoM system [9], developed for the CL-SciSumm 2019 shared task, formulates summarization

as a binary classification problem and incorporates trigram blocking to reduce redundancy in the generated summaries.

Other research has explored citation contextualization and discourse-aware summarization. The framework proposed in [10] utilizes query reformulation, word embeddings, and supervised learning to identify discourse facets associated with citations, enabling more context-aware summaries. In a related approach, the Poli2Summ system [11] introduces a multi-stage pipeline involving text span identification, citation classification, and supervised ranking of extracted content.

In addition to citation-based methods, alternative techniques have been proposed to improve coherence and diversity in summaries. The rule-based approach in [12] groups text segments based on similarity and ranks sentences using heuristic features such as sentence length and position. Meanwhile, the work in [13] applies a Determinantal Point Process (DPP) to model summarization as the selection of a diverse and low-redundancy subset of sentences.

Neural network-based methods have also been explored. The study in [14] employs a Convolutional Neural Network (CNN) to score sentences by combining word embeddings with citation-related features. More recently, pre-trained Transformer models have been utilized to enhance summarization performance. For example, the work in [14] leverages BERT-based encoders to identify cited text spans and examines their impact on extractive summarization tasks, demonstrating the benefits of adapting general-domain models to scientific text.

2.3. Knowledge Gap

Despite the extensive research in scientific document summarization, most approaches predominantly focus on extractive methods that rely heavily on sentence selection and ranking. While these approaches are effective in preserving factual accuracy, they often fail to produce coherent, concise summaries that fully capture the underlying meaning of the source text.

Summarization techniques can broadly be categorized into extractive and abstractive methods, as shown in Table 1. Extractive summarization can be viewed as selecting and assembling the most important parts of the original text, whereas abstractive summarization involves understanding the content and generating

new text that conveys the same meaning in a more natural and compact form. Consequently, true semantic understanding is more closely associated with abstractive approaches.

Table 1: A Summary of Differences between Extractive versus Abstractive Summarization Methods

Aspect	Extractive Summarization	Abstractive Summarization
Definition	Selects and combines sentences directly from the original text	Generates new sentences that convey the meaning of the original text
Approach	Sentence selection, ranking, and scoring	Natural language generation and semantic rewriting
Output Style	Uses exact words and sentences from the source	Produces paraphrased and newly generated text
Understanding Level	Limited semantic understanding	Deeper semantic and contextual understanding
Fluency & Coherence	May be disjointed or less smooth	More fluent and human-like
Common Methods	• Frequency-based (TF-IDF) • Graph-based (TextRank) • Machine learning ranking models • Clustering-based methods • Citation-based summarization (important in scientific texts)	• Sequence-to-sequence models • Encoder–decoder architectures • Transformer-based models such as T5, BART, GPT-2

Hybrid methods, which combine extractive and abstractive techniques, have shown promise in leveraging the strengths of both. However, identifying the most effective abstractive models to integrate within such frameworks remains an open challenge, so in our study, we show which abstractive method is more efficient among three famous pre-trained Transformer-based models.

Recent advances in pre-trained Transformer-based models, such as T5, BART, and GPT-2, have significantly improved the capabilities of abstractive summarization. Nevertheless, their application to scientific datasets—particularly CL-SciSumm 2019—remains relatively underexplored, especially in terms of systematic comparative evaluation and the impact of model configurations on performance.

Therefore, this study addresses this gap by conducting a comprehensive comparative analysis of these Transformer-based models for abstractive summarization of scientific documents. In addition, it examines the influence of generation parameters and employs complementary evaluation metrics to assess both content quality and linguistic acceptability.

3. Methodology

This section elucidates the various components and steps employed by our system. The methodology is delineated into three phases, as depicted in Figure 2: Pre-processing Data, Processing, and Evaluation.

3.1.Pre-Processing Phase

This step aims to clean data and convert it into a coherent format so that it can be easily handled. First, convert the format from XML to CSV; this conversion in the format will help facilitate working and iterating over the data easily. And null and duplicate rows are removed. Then, we have applied the following steps:

- Removing stop words.
- Removing special characters.
- Lowercase characters.

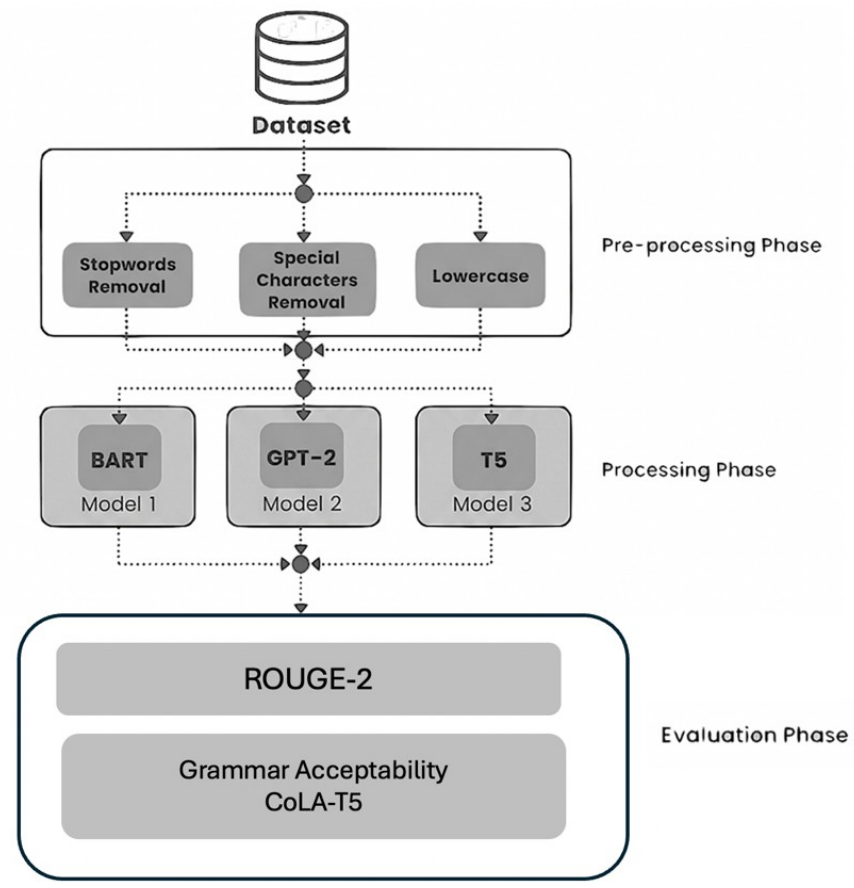


Figure 2. The Architecture of the methodology

3.2.Processing Phase

The summarization was carried out using three Transformer models, explained in Table 2, which are deep learning models that use the self-attention mechanism, differentially weighing the significance of each part of the input data. Transformers offer a collection of deep-learning NLP models that have been pre-trained for summarization tasks. And these pre-trained models are T5, BART, and GPT-2 pre-trained models.

Table 2: A brief summary of the three models used in our system

Feature	GPT-2	T5	BART
Model Type	Causal Language Model (Autoregressive)	Sequence-to-Sequence (Text-to-Text)	Denoising Autoencoder / Sequence-to-Sequence
Transformer Type	Decoder-only	Encoder-Decoder	Encoder-Decoder
Directionality	Unidirectional (Left-to-Right)	Bidirectional (Encoder) + Unidirectional (Decoder)	Bidirectional (Encoder) + Unidirectional (Decoder)
Training Approach	Unsupervised pre-training focused on text generation.	Multi-task pre-training formulating all NLP tasks as text-to-text generation.	Pre-trained by corrupting text and learning to reconstruct the original sequence.
Use of Attention	Masked Multi-Head Self-Attention.	Multi-Head Self-Attention (Encoder), Masked Self-Attention + Cross-Attention (Decoder).	Multi-Head Self-Attention (Encoder), Masked Self-Attention + Cross-Attention (Decoder).
Pre-training Objectives	Next-token prediction (Standard Language Modeling).	Span corruption (Masked Language Modeling variant replacing spans with sentinel tokens).	Text denoising (Token masking, token deletion, text infilling, sentence permutation, document rotation).
Number of Parameters	117M (Small) to 1.5B (Extra Large).	60M (Small) to 11B (XXL).	139M (Base) to 406M (Large).
Input Length Limitations	1024 tokens.	512 tokens typically during pre-training (uses relative position embeddings, enabling longer sequences).	1024 tokens.
Typical Hyperparameters	Layers: 12-48Heads: 12-25Hidden Size (dmodel): 768-1600Optimizer: Adam	Layers: 12-24 per stackHidden Size (dmodel): 768-1024Optimizer: AdaFactor	Layers: 6-12 per stackHidden Size (dmodel): 768-1024Activation: GeLU
Contextual Understanding	Strong forward-looking context, but limited by lack of bidirectional (past and future) visibility.	Excellent bidirectional understanding explicitly tied to generation capabilities.	Deep bidirectional context combined with robust autoregressive generation.
Main Advantages	Excellent at zero-shot generation and fluent, coherent text synthesis.	Unified framework that simplifies fine-tuning; highly adaptable to diverse tasks without architectural changes.	Exceptionally strong at generation tasks that require comprehension, such as abstractive summarization and translation.

3.2.1. T5 Model

T5 Model [15] is developed by Google researchers as part of their efforts to investigate the boundaries of transfer learning in natural language processing. The model takes a novel text-to-text approach, with each

NLP task framed as translating one type of text into another. This method makes it easier to apply the model to a wide range of activities, from translation to question-answering.

Architecture: T5's design is based on the Transformer model, which has proven to be extremely effective in a variety of NLP tasks. It is made up of an encoder and a decoder, both of which use the Transformer mechanism. The encoder is responsible for processing the input text. It is bidirectional, which means it examines the full input sequence (words or sub-words) at once. This method enables the model to comprehend the context surrounding each word in the input. The output text is then generated one piece at a time by the decoder. It's autoregressive, which means it predicts the future word based on the preceding ones. This sequential generation ensures that the result is consistent and contextually appropriate.

Adaptation and Training: All tasks in T5 training are converted into text-to-text format. A translation task, for example, is defined as converting a sentence from one language to another, whereas a classification task includes converting a sentence to a label. T5 is trained on a vast corpus of text during pre-training, gaining a wide knowledge of language and context. This stage consists of learning from a combination of unsupervised and supervised tasks. T5 is fine-tuned to certain activities during this process. This entails training the model on task-specific datasets, which allows it to specialize and increase its performance on certain tasks.

Capabilities and Use Cases: T5's text-to-text foundation makes it extremely adaptable. It can be used for a variety of purposes, including but not limited to.

- Text Summarization is the process of creating succinct summaries of lengthy documents.
- Question Answering: Answering questions based on the situation.
- Text Translation is the process of translating text between languages.
- Text classification is the process of categorizing or labeling text.

3.2.2. BART Model

BART [16], created by Facebook AI researchers, is a transformational paradigm in the field of natural language processing (NLP). It combines the advantages of two powerful architectures: BERT's

bidirectional nature and GPT's autoregressive capabilities. This hybrid model is intended for both understanding and creating language, making it suitable for a wide range of NLP tasks.

Architecture: BART's architecture combines an encoder and a decoder, both of which are based on the Transformer model.

Encoder: BART's encoder, like BERT's, is bidirectional. It processes the entire input text, capturing context from both sides. This enables BART to have a thorough comprehension of the input text, taking into account the context around each word. The decoder, on the other hand, is autoregressive, similar to GPT. It produces output in a sequential manner, predicting each upcoming word based on the previous ones. This method is critical for activities requiring text generation since it ensures that the output is coherent and contextually linked with the input.

Adaptation and Training: BART's training process is unusual among its peers because it includes a distinct pre-training objective. BART has been pre-trained with a denoising autoencoder objective. The input text is purposely distorted (e.g., by removing some words or mixing phrases) and then fed into the model in this procedure. The model's job is to recreate the original, undamaged text. BART can learn both context understanding and text creation with this training method. BART conducts fine-tuning on task-relevant datasets for specific tasks. This step customizes the model's capabilities for certain applications, increasing its efficacy in those domains.

Capabilities and Use Cases: BART's hybrid nature makes it extremely adaptable in a variety of NLP applications. Among its applications are:

- Text Summarization is the process of creating succinct summaries from lengthy texts while maintaining important information.
- Machine Translation: The process of translating text between languages using a thorough understanding of linguistic nuances.
- Text Generation: The process of producing coherent and contextually relevant text, which is useful in chatbots and creative writing.
- Answering Questions: Providing correct responses to questions depending on given or inferred context.
- Text Classification is the process of categorizing or labeling text based on its content.

3.2.3. GPT-2 Model

GPT-2 [17], created by OpenAI, is a significant achievement in the field of natural language processing (NLP). It's an autoregressive language model that employs deep learning to generate writing that looks like it came from a human. GPT-2 is well-known for its capacity to generate coherent and contextually relevant text passages, making it a game-changing tool in the field of AI-driven text synthesis.

Architecture: GPT-2's design is based on the Transformer model, which is well-known for its ability to handle sequential data such as text. Transformer Model: The Transformer architecture, which was specifically built for managing sequential data, lies at the heart of GPT-2. Unlike typical models that handle input sequentially, the Transformer supports parallel processing, which greatly reduces training and prediction durations. GPT-2 is autoregressive in nature. This means that it guesses each word based on the words that preceded it, synthesizing text one word at a time. This method allows the model to maintain coherence across longer stretches.

Training and Data: The training process and data used to train GPT-2 are critical to its capabilities. GPT-2 was trained using an unsupervised learning technique. Without any special task-oriented training, it was fed a vast dataset of different internet material and trained to anticipate the next word in a sentence.

Large-Scale Dataset: The model was trained using a dataset known as "WebText," which contains a massive amount of text from the internet. GPT-2 was able to grasp and develop a wide range of writing styles and themes as a result of this comprehensive training.

Capabilities and Use Cases: GPT-2's ability to generate text allows it to thrive in a variety of applications:

- GPT-2 may generate innovative and diverse content ranging from composing articles to producing fictitious novels.
- Language Translation: Though it is not its major job, GPT-2 can perform some translation tasks.
- Text Summarization: It can reduce large texts into shorter summaries while maintaining vital information.
- Question Answering: Using the knowledge in its training data, GPT-2 may generate responses to questions.

3.3.Evaluation Phase

The chosen evaluation metric is ROUGE 2-F1 score [18] (Recall-Oriented Understudy for Gisting Evaluation), which measures the overlapping of textual units (n-grams, word sequences) between the generated summary and a set of gold reference summaries. [1] Since ROUGE is based on overlapping n-grams, there are variations of the ROUGE metrics. Our interest is in ROUGE-2, which means bigram overlapping, because this is the widely used metric. To calculate the F1 Score, we first need to calculate Recall (R) as shown in Eq. 1, and calculate Precision (P) as shown in Eq. 2, then we can calculate the F1 Score in Eq. 3

$$R = \text{(Number of overlapping bigrams)} / \text{(Total bigrams in REFERENCE summary)} \quad \text{Eq. 1}$$

$$P = \text{(Number of overlapping bigrams)} / \text{(Total bigrams in GENERATED summary)} \quad \text{Eq. 2}$$

$$F1 = 2 \times (P \times R) / (P + R) \quad \text{Eq. 3}$$

Another metric employed is the grammatical acceptability of the generated summary. To achieve this, we utilized a trained T5 model on the Corpus of Linguistic Acceptability (CoLA) dataset. The CoLA [19] serves as a widely recognized dataset for training classification models that assess the grammatical and syntactic correctness of given sentences. The Corpus of Linguistic Acceptability (CoLA) comprises 10,657 sentences sourced from 23 linguistics publications. These sentences have been meticulously annotated by their original authors to determine their grammatical acceptability.

4. Experiments and Results

Experiment setup: The tests were carried out on a RYZEN 7 9 Gen 16GB RAM system equipped with an Nvidia RTX 3060 6GB GPU.

Dataset: All the experiments were conducted on the same dataset (CL-SciSumm 2019) mentioned in the previous section.

Experiments: This study conducted 19 trials to analyze the effectiveness of three advanced NLP models in text summarization, GPT-2, BART, and T5, by analyzing how well they preserve correctness and coherence when summarizing content. The experiments, which were separated into five for GPT-2, four for BART, and ten for T5, demonstrated that the BART model performed better. BART's performance

was underlined by attaining the greatest ROUGE-2 F1 score of 0.40664, which was credited to its hybrid structure of bidirectional encoding and autoregressive decoding.

Furthermore, critical configuration settings such as Num-Beams, n-gram penalty, and length parameters were fine-tuned in the text generation segment to optimize coherence, relevance, and efficiency, ensuring that the generated outputs were not only contextually accurate but also met the desired length and uniqueness standards. Several fundamental configuration settings are critical for text-generating tasks, notably with models such as GPT-2, BART, and T5.

- Num-Beams (Beam Search Size): Specifies the number of beams to search for. A greater number enhances the likelihood of obtaining a more accurate result, but at the expense of computing efficiency.
- n-gram Penalty: The creation of repeating n-grams is prevented.
- Penalty for Length (Len-penalty): Modifies the length of the generated output. A higher penalty produces shorter sequences, whereas a smaller penalty produces longer sequences.
- Minimum Length (Min-len): This property specifies the minimum length of the generated text.
- Maximum Length (Max-len): This parameter specifies the maximum length of the generated text. Ensures that the model does not provide excessively long outputs.
- Early Stopping: If enabled, creation will cease as soon as any of the beams creates an end-of-sentence token.

Each of the following tables, Table 3, Table 4, and Table 5, shows the different configuration settings that were used in each model and the effect of each one on the output results. Table 6 shows a comparison of the best-performing setting for each model using ROUGE-2 F1 as a metric. As shown in Table 6, the BART model surpassed T5 and GPT2.

To evaluate the grammatical quality of the generated summaries, a version of the T5 model was fine-tuned for linguistic acceptability classification. The model was trained on the CoLA dataset, which is specifically annotated for grammatical acceptability. Table 7 presents a comparison of the grammatical quality of summaries generated by BART, GPT-2, and T5. The results indicate that both T5 and BART achieve high levels of grammatical acceptability, outperforming GPT-2 in this evaluation.

Table 3: T5-Model Results

Exp. #	Num_Beans	n-gram	Len_penalty	Min_len	Max_len	Early_stop	ROUGE-2 F1
1	10	3	2	50	200	True	0.29936
2	5	3	2	150	250	True	0.31535
3	5	2	2	150	250	True	0.30081
4	5	3	2	150	250	True	0.29553
5	10	3	2	100	250	True	0.34365
6	10	2	2	100	250	True	0.30520
7	10	3	2	100	250	False	**0.34417**
8	10	3	2	100	250	False	0.13111

Table 4: GPT-2 Model Results

Exp #	skip_special_tokens	Min_len	Max_len	Early_stop	ROUGE-2 F1
1	True	100	200	True	0.38009
2	True	100	250	False	0.38046
3	True	150	250	True	0.38046
4	False	150	250	False	0.32751
5	True	150	250	True	**0.38733**

Table 5: BART Model Results

Exp. #	skip_special_tokens	Min_len	Max_len	Early_stop	ROUGE-2 F1
1	True	100	200	True	**0.40664**
2	False	150	200	False	0.38588
3	True	150	200	True	0.38699
4	False	150	200	True	0.38048

Table 6: The best performing summarization model

Model	ROUGE-2 (F1-SCORE)
T5	0.34417
GPT2	0.38733
BART	**0.40664**

Table 7: The percentage of grammatical acceptance for each model

Best Model	Gramatical Acceptance Accuracy
T5	**93.36%**
GPT-2	92.54%
Bart	93.27%

The success of BART in summarization tasks can be attributed to its unique design, which effectively captures the context from both directions in its encoder and then generates a coherent, concise summary in its decoder. This approach allows for a more accurate representation and condensation of the original text. While GPT-2 and T5 are formidable in their respective strengths, BART's specific alignment with

the requirements of summarization, understanding the essence of the text, and generating a precise, context-aware summary makes it the most effective among the three for this application.

This outcome underscores the importance of choosing the right model based on the specific requirements of an NLP task. It also highlights the continuous evolution in the field of NLP, where different architectures offer varied advantages for diverse language processing tasks. The findings from using these models in summarization tasks provide valuable insights into their practical applications and pave the way for more targeted and effective use of NLP technologies in real-world scenarios.

5. Conclusion

Within the field of computational linguistics, generating automated summaries of academic literature remains a significant and ongoing challenge. This importance is underscored by the considerable time researchers, educators, and students spend analyzing complex scholarly texts. By reducing this time investment, automated summarization systems can enhance knowledge accessibility, encourage broader reading, and support more efficient learning.

To develop an effective approach for summarizing scientific documents, this study conducted a comparative evaluation of three widely used Transformer-based models [20]: T5, GPT-2, and BART. The results demonstrate that the BART model achieves the strongest performance, attaining a ROUGE-2 F1-score of 0.40664 and outperforming the other two models. In addition, an auxiliary experiment was conducted to assess the grammatical acceptability of the generated summaries. This evaluation utilized a T5 model fine-tuned on the CoLA dataset, which is annotated for linguistic acceptability. The findings indicate that both T5 and BART generate summaries with high grammatical quality, outperforming GPT-2 in this aspect.

For future work, we plan to extend this study by evaluating additional models and incorporating more comprehensive evaluation metrics beyond ROUGE-2 and grammatical acceptability, like ROUGE-L, BERTScore, SummaC, and human evaluation, which will enrich the credibility of the evaluation. Also, to have strong evidence of a model's superiority, other challenging datasets will be included in the study as future work. Furthermore, we aim to develop a hybrid summarization framework that integrates both extractive and abstractive approaches to improve the quality of scientific document summarization.

References

[1] R. R. Waly and W. H. Gomaa, "Extractive Summarization of Scientific Articles," in 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), May 2022, pp. 349-354. doi: 10.1109/MIUCC55081.2022.9781702.

[2] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," Expert Systems with Applications, vol. 165, p. 113679, 2021. doi: 10.1016/j.eswa.2020.113679.

[3] Y. M. Wazery, M. E. Saleh, A. Alharbi, and A. A. Ali, "Abstractive Arabic Text Summarization Based on Deep Learning," Computational Intelligence and Neuroscience, vol. 2022, 2022. doi: 10.1155/2022/9046638.

[4] N. I. Altmami and M. E. B. Menai, "Automatic summarization of scientific articles: A survey," Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 4, pp. 1011-1028, 2020. doi: 10.1016/j.jksuci.2020.04.020.

[5] P. Semaan, "Natural language generation: an overview," J Comput Sci Res, vol. 1, no. 3, pp. 50-57, 2012. doi: 10.14355/jcsr.2012.0103.02.

[6] S. Teufel and M. Moens, "Summarizing scientific articles: experiments with relevance and rhetorical status," Computational Linguistics, vol. 28, no. 4, pp. 409-445, 2002. doi: 10.1162/089120102762671936.

[7] N. Saini, S. Kumar, S. Saha, and P. Bhattacharyya, "Scientific Document Summarization using Citation Context and Multi-objective Optimization," in 2020 25th International Conference on Pattern Recognition (ICPR), Jan. 2021, pp. 4290-4295. doi: 10.1109/ICPR48806.2021.9412089.

[8] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," J. Mach. Learn. Res., vol. 21, no. 1, Art. no. 140, Jan. 2020. doi: 10.5555/3455716.3455856.

[9] C. Zerva, M. Q. Nghiem, N. T. Nguyen, and S. Ananiadou, "Nactem-uom@ cl-scisumm 2019," in Proceedings of the 4th Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL@ SIGIR), Jan. 2019.

[10] A. Cohan and N. Goharian, "Scientific document summarization via citation contextualization and scientific discourse," International Journal on Digital Libraries, vol. 19, no. 2, pp. 287-303, 2018. doi: 10.1007/s00799-017-0216-7.

[11] M. La Quatra, L. Cagliero, and E. Baralis, "Poli2sum@ cl-scisumm-19: Identify, classify, and summarize cited text spans by means of ensembles of supervised models," in Proceedings of the 4th Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL@ SIGIR), Jan. 2019.

[12] S. Ma, H. Zhang, T. Xu, J. Xu, S. Hu, and C. Zhang, "IR&TM-NJUST@ CLSciSumm-19," in Proceedings of the 4th Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL@ SIGIR), 2019, pp. 181-195.

[13] L. Li, Y. Zhu, Y. Xie, Z. Huang, W. Liu, X. Li, and Y. Liu, "CIST@ CLSciSumm-19: Automatic Scientific Paper Summarization with Citances and Facets," in Proceedings of the 4th Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL@ SIGIR), vol. 54, 2019.

[14] L. Chiruzzo, À. Bravo, and H. Saggion, "LaSTUS-TALN+ INCO@ CL-SciSumm 2019," in Proceedings of the 4th Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL@ SIGIR), Jan. 2019

[15] C. Zerva, M. Q. Nghiem, N. T. Nguyen, and S. Ananiadou, "Cited text span identification for scientific summarisation using pre-trained encoders," Scientometrics, vol. 125, no. 3, pp. 3109-3137, 2020. doi: 10.1007/s11192-020-03741-z.

[16] A. R. Fabbri, W. Kryściński, B. McCann, C. Xiong, R. Socher, and D. Radev, "Summeval: Re-evaluating summarization evaluation," Transactions of the Association for Computational Linguistics, vol. 9, pp. 391-409, 2021. doi: 10.1162/tacl_a_00373.

[17] M. Lewis et al., "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," arXiv preprint arXiv:1910.13461, 2019. doi: 10.48550/arXiv.1910.13461.

[18] A. Radford et al., "Language models are unsupervised multitask learners," OpenAI blog, 2019.

[19] A. Warstadt, A. Singh, and S. R. Bowman, "Neural network acceptability judgments," Transactions of the Association for Computational Linguistics, vol. 7, pp. 625-641, 2019. doi: 10.1162/tacl_a_00290.

[20] R. Haruna, A. Obiniyi, M. Abdulkarim and A. A. Afolorunsho, "Automatic Summarization of Scientific Documents Using Transformer Architectures: A Review," 2022 5th Information Technology for Education and Development (ITED), Abuja, Nigeria, 2022, pp. 1-6, doi: 10.1109/ITED56637.2022.10051602.