Received: 20 August 2025 | Revised: 6 March 2026 | Accepted: 1 April 2026 | Published online: 16 April 2026

RESEARCH ARTICLE

Artificial Intelligence and Applications
2026, Vol. 00(00) 1–9
DOI: 10.47852/bonviewAIA62027294

BON VIEW PUBLISHING

Transformer-Based Approaches in Paraphrasing Texts

Mohamed Cherradi[1,*] and Hajar El Mahajer[1]

[1] *Computer Science Department, Abdelmalek Essaâdi University (UAE), Morocco*

Abstract: In a variety of natural language processing tasks, such as text generation, classification, sentiment analysis, and question answering, large language models like GPT have recently shown impressive capabilities. However, maintaining coherence and preserving semantic integrity make it difficult to generate high-quality paraphrases, especially for lengthy textual inputs. By utilizing three well-known and extensively used LLM architectures, including T5, Pretraining with Extracted Gap-sentences for Abstractive Summarization, and Bidirectional and Auto-Regressive Transformer, this study explores this problem by putting forth a strong multi-model framework for automatic paraphrasing. The main goal is to evaluate how well they can produce coherent and semantically correct paraphrases at the sentence and paragraph levels without requiring text segmentation. To assess output quality, the experimental setup uses common evaluation metrics, such as Recall-Oriented Understudy for Gisting Evaluation and Bilingual Evaluation Understudy scores. According to empirical findings, T5 consistently produces better results, especially in terms of semantic fidelity and linguistic fluency, even though all three models demonstrate strong paraphrasing abilities. These results highlight T5's efficacy in complex paraphrasing tasks and provide insightful information for future research in data augmentation, summarization, and automatic content rewriting.

Keywords: paraphrase generation, large language models (LLMs), T5, PEGASUS, BART

1. Introduction

As a key task in natural language processing (NLP), paraphrase generation has many uses in data augmentation, machine translation, question answering, and content creation [1]. Numerous methods, from rule-based systems to statistical and neural models, have been proposed over the past few decades to tackle this task [2]. Despite tremendous advancements, producing high-quality paraphrases, especially at the paragraph level, remains difficult because it must maintain semantic integrity while generating outputs that are varied and fluid [3]. Longer textual spans present a greater challenge because it is much harder to maintain semantic consistency and contextual coherence.

Recent advances in large language models (LLMs) have opened new opportunities to revisit this task from a generative perspective. In particular, models like T5 [4], Pretraining with Extracted Gap-sentences for Abstractive Summarization (PEGASUS) [5], and Bidirectional and Auto-Regressive Transformer (BART) [6] have demonstrated remarkable capabilities in a variety of text generation tasks, including summarization and translation. However, most existing studies focus primarily on sentence-level generation, and their application to coherent long-span paraphrasing remains relatively underexplored. Furthermore, comparative analyses of different LLM architectures under a unified experimental setting are still limited, particularly when considering parameter-efficient fine-tuning strategies. In this article, we address this gap by leveraging the autoregressive capabilities of LLMs to generate coherent and contextually faithful paraphrases for both sentences and longer spans of text.

There are three elements we can contribute. In order to systematically assess transformer-based language models for paraphrase generation across various textual granularities, we first suggest a single experimental framework. Second, by utilizing the Quantized Low-Rank Adaptation (QLoRA) technique [7], which enables effective training with fewer computational resources, we modify a pretrained BART model to examine the effects of parameter-efficient fine-tuning. Third, we empirically compare several LLM architectures (T5, PEGASUS, and BART) and evaluate how well they perform in terms of computational efficiency, semantic preservation, and fluency. The evaluation is carried out using well-known lexical metrics, like Recall-Oriented Understudy for Gisting Evaluation (ROUGE) and Bilingual Evaluation Understudy (BLEU) scores [8], which offer insights into how well various architectures work for real-world paraphrasing applications.

The remainder of this paper is organized as follows: Section 2 presents a review of related work and existing approaches to paraphrase generation. Section 3 describes our proposed methodology, including data preparation, model architectures, and fine-tuning strategies. Section 4 reports and discusses the experimental results. Finally, Section 5 concludes the paper and outlines potential directions for future research.

2. Literature Review

The most important recent developments in paraphrase generation are reviewed in this section, with an emphasis on

***Corresponding author:** Mohamed Cherradi, Computer Science Department, Abdelmalek Essaâdi University (UAE), Morocco. Email: m.cherradi@uae.ac.ma

transformer-based methods that are most pertinent to our work. Large pretrained language models have greatly enhanced the fluency and semantic fidelity of generated paraphrases, leading to significant advancements in the field. A gap-sentence generation pretraining objective was first introduced by PEGASUS [9] for abstractive summarization, but it has also shown promise for paraphrasing tasks. Similar to this, T5 [10] has gained popularity because of its adaptable text-to-text framework; [11] has shown that it can produce paraphrases that are both lexically and syntactically diverse and semantically consistent using controlled generation techniques. Additionally, BART, which combines an autoregressive decoder with a bidirectional encoder, has demonstrated excellent paraphrasing performance in a variety of domains [12]. However, the practical deployment of such large models is limited by the computational cost of fully fine-tuning them. In order to address this, [13] applied QLoRA fine-tuning to BART, which allowed for effective adaptation in resource-constrained environments by lowering memory requirements by up to 75% while maintaining comparable performance.

Assessing the quality of paraphrases is still a difficult task. Our study does not use sophisticated metrics like BERTScore, which use contextual embeddings to better capture semantic similarity. Rather, we rely on widely recognized traditional evaluation metrics like ROUGE and BLEU, which provide a useful balance between simplicity and effectiveness despite their limitations in fully capturing semantic nuances [14, 15]. This decision facilitates reproducibility and clarity in evaluating paraphrasing performance by enabling us to conduct a reliable and consistent quantitative comparison of various models.

Despite the progress of models like T5, PEGASUS, and BART, there is still a lack of comprehensive, systematic comparisons of their paraphrasing abilities, especially under parameter-efficient fine-tuning regimes and across varying text granularities from sentences to paragraphs. Our work addresses this gap by conducting a thorough evaluation of several LLMs, including T5, PEGASUS, PaLM, and BART with QLoRA balancing generation quality with computational efficiency, and providing practical insights for the deployment of paraphrasing systems in real-world NLP applications.

3. Methodology

In this section, we present a methodical approach designed to tackle the difficulties associated with transformer-based architectures for automatic paraphrase generation. In order to produce fluid and semantically accurate paraphrases at the sentence and paragraph levels, the suggested framework makes use of three cutting-edge models: T5, PEGASUS, and BART enhanced with QLoRA. In addition to putting these models into practice, our strategy focuses on carrying out a thorough comparative analysis to identify each model's advantages and disadvantages. In order to accomplish this, we employ a multi-phase procedure that starts with meticulous data preparation, which includes large-scale dataset collection, noise filtering, and normalization, guaranteeing input quality and linguistic diversity. In order to maximize performance while reducing computational overhead, we use parameter-efficient training techniques in the model initialization and fine-tuning stage that follows. Lastly, we use robust evaluation metrics, specifically ROUGE and BLEU, to quantitatively evaluate the generated paraphrases' overall fluency, lexical richness, and semantic preservation. This approach, which is summed up in Figure 1, forms the basis for our experimental analysis and emphasizes the main contributions of this work: a methodical comparison of top LLMs for paraphrasing and practical insights into their suitability for real-world NLP tasks.

There are three primary steps in the workflow, as shown in Figure 1. In order to maintain semantic fidelity, the dataset acquisition and preprocessing stage first guarantees both scale and quality by putting together a variety of paraphrase pairs and using strict cleaning, normalization, and tokenization. Second, T5, PEGASUS, and BART-QLoRA are implemented in a parameter-efficient manner during the model initialization and fine-tuning stage, allowing for high-quality paraphrase generation while lowering computational overhead. Lastly, the evaluation stage measures fluency, semantic preservation, and lexical richness by combining automated metrics (ROUGE-1, ROUGE-2, ROUGE-L, and BLEU) with focused human evaluation. In addition to enabling a methodical comparison of various transformer-based architectures, this integrated pipeline shows the usefulness of using QLoRA for memory-efficient fine-tuning, providing useful information for paraphrasing applications in the real world.

Therefore, by combining extensive dataset preprocessing, parameter-efficient fine-tuning with QLoRA, and thorough evaluation at the sentence and paragraph levels, our methodology offers a novel and effective framework for paraphrase generation. This method not only makes it possible to compare top LLMs fairly and methodically, but it also shows useful methods for striking a balance between computational efficiency and high-quality paraphrase output, proving the uniqueness and usefulness of our contribution for practical NLP applications.

Figure 1
Overview of the methodological workflow for paraphrase generation

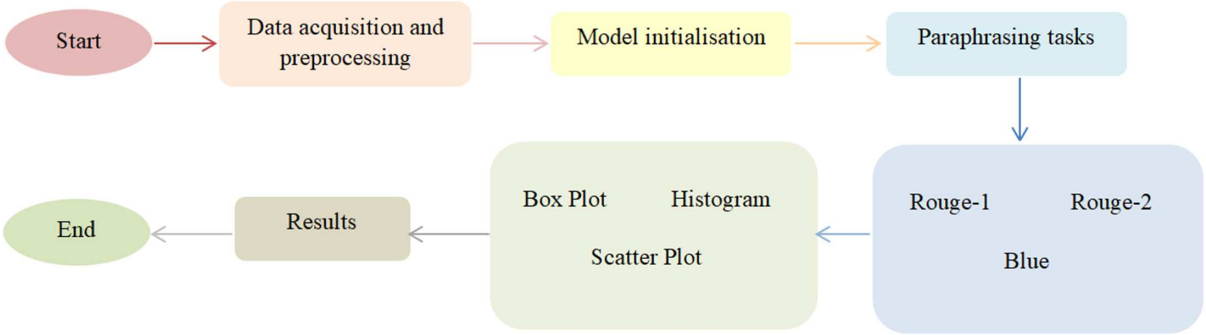


Table 1
Training configuration and hyperparameters

	Models		
Parameter	T5-small	PEGASUS-xsum	BART-QLoRA
Learning rate	3e-5	3e-5	3e-5
Batch size	32	32	16
Number of epochs	5	5	6
Max sequence length	128 (sent.)/256 (para.)	128/256	128/256
Optimizer	AdamW ($\beta1 = 0.9$, $\beta2 = 0.999$)	AdamW	AdamW
Scheduler	Linear warmup with decay	Same	Same
Dropout	0.1	0.1	0.1

3.1. Technical training details

We offer thorough technical specifications for the training and fine-tuning of every model in order to guarantee reproducibility and rigor. Using PyTorch and Hugging Face Transformers, the experiments were carried out on a Google Colab Pro equipped with a Tesla T4 GPU (16 GB VRAM) and 16 GB RAM. Table 1 summarizes important hyperparameters and fine-tuning techniques.

We used 4-bit quantization for BART-QLoRA with a rank $r = 16$ and LoRA alpha of 32, which resulted in a memory reduction of about 75% without compromising model performance. By utilizing parameter-efficient adaptation, fine-tuning greatly reduced computational costs without sacrificing the quality of the paraphrase.

To ensure a balanced evaluation framework, the dataset mentioned in Section 3.2 was divided into 80% training, 10% validation, and 10% testing. SentencePiece (T5), Pegasus tokenizer, and BART tokenizer were used for tokenization, respectively. To ensure consistency across inputs, preprocessing steps included case normalization, punctuation cleaning, and duplicate removal.

3.2. Dataset acquisition and preprocessing

We assembled a large-scale paraphrase corpus consisting of approximately one million sentence pairs collected from multiple widely used paraphrase datasets. The main sources include the Quora Question Pairs dataset, the Microsoft Research Paraphrase Corpus, and the Paraphrase Adversaries from Word Scrambling dataset, which are commonly used benchmarks in paraphrase generation and semantic similarity research [16–18]. These datasets contain pairs of semantically equivalent sentences that exhibit substantial lexical and syntactic variation while preserving semantic meaning.

In order to ensure the dataset's consistency and dependability, we put in place a strict preprocessing pipeline. In order to lessen superficial textual variability, duplicate pairs and entries with missing values were systematically eliminated. Next, punctuation and whitespace were thoroughly normalized. Table 2 provides a succinct overview of the dataset properties and preprocessing procedures.

We chose a representative subset of 500,000 pairs to fine-tune the transformer-based models due to computational limitations. Furthermore, the other architecture received a smaller share of 50,000 examples. This method allowed for a fair comparison of the models under consistent conditions by preserving the original dataset's diversity while maintaining a balanced trade-off between performance and computational efficiency.

Table 2
Dataset composition and preprocessing pipeline

Aspect	Description
Total dataset size	≈ 1,000,000 sentence pairs
Subset for fine-tuning	500,000 sentence pairs
Data sources	Multiple high-quality paraphrase corpora
Preprocessing steps	Removal of duplicates and missing values, normalization of punctuation and whitespace

3.3. Transformer model selection for high-quality paraphrasing

Our approach purposefully uses three complementary transformer-based architectures, such as T5, PEGASUS, and BART, optimized with QLoRA to rigorously handle paraphrase generation across sentence- and paragraph-level inputs. This trio strikes a balance between parameter-efficient adaptation under constrained compute (BART + QLoRA), pretraining specifically focused on abstractive rewriting (PEGASUS's gap-sentence objective), and generative flexibility (T5's unified text-to-text paradigm). The architectural details, fine-tuning procedures, and design justifications for each model are described in detail in the ensuing subsections, laying the groundwork for a controlled and repeatable comparative analysis using ROUGE and BLEU.

We chose T5-small because it strikes a balance between performance and computational efficiency, allowing for quick experimentation while utilizing its strong text-to-text capabilities. PEGASUS-xsum was selected because of its pretraining goal designed for abstractive rewriting, which enables efficient handling of paraphrases at the sentence and paragraph levels. Notably, QLoRA was used to fine-tune BART, introducing a parameter-efficient adaptation strategy that is rarely used in paraphrase generation. This set of models, which includes memory-efficient fine-tuning, specialized abstractive pretraining, and unified text-to-text processing, forms a novel benchmarking framework that thoroughly assesses multi-model performance on extensive, high-quality paraphrase datasets.

3.3.1. T5 model

Because of its extremely adaptable text-to-text formulation of language problems, the Text-to-Text Transfer Transformer (T5) has become one of the most significant architectures in contemporary NLP. T5 reframes a variety of NLP tasks, such as translation, summarization, question answering, and text classification, within a single sequence-to-sequence paradigm in contrast to conventional task-specific models. This cohesive method greatly enhances generalization across various linguistic phenomena and makes task adaptation much simpler [19].

T5's enormous pretraining on the C4 corpus (Colossal Clean Crawled Corpus), which contains billions of tokens, and subsequent fine-tuning on downstream tasks are largely responsible for its success. The model is especially well-suited for paraphrase generation tasks that demand both semantic fidelity and lexical diversity because it makes use of a transformer-based encoder–decoder architecture that is optimized for handling long-context dependencies and complex semantic relationships.

In this study, we evaluate T5's capacity to generate coherent and meaning-preserving paraphrases for machine-generated sentences using a refined version of T5 (T5-small). This version was selected to take advantage of the T5 family's strong generalization capabilities while striking a balance between performance and computational efficiency. To ensure reproducibility and compliance with cutting-edge implementation techniques, we employ the Hugging Face Transformers library for model loading, tokenization, and text preprocessing. The T5 model's encoder–decoder structure, shared vocabulary embeddings, and task prefixing mechanism—all of which together allow for flexible adaptation to paraphrasing scenarios—are highlighted in Figure 2. We intend to examine how T5's unified text-to-text formulation compares to other transformer-based models by incorporating it into our comparative framework. This will provide useful information for the creation of superior automated paraphrase generation systems.

3.3.2. PEGASUS model

With a focus on abstractive text rewriting and summarization, the PEGASUS model is a cutting-edge transformer-based architecture created especially for sequence-to-sequence generation tasks [20]. In contrast to generic language models, PEGASUS presents a novel pretraining goal known as "gap-sentence generation," in which the model is trained to reconstruct key sentences based on the remaining context after they are masked. PEGASUS is especially useful for paraphrasing lengthy text spans that go beyond the sentence level because of this pretraining technique, which allows it to capture long-range dependencies, discourse-level semantics, and contextual relevance.

The transformer encoder–decoder framework, which is optimized for producing fluid and contextually accurate text, is the foundation of PEGASUS's architecture. It has proven to be highly competitive when compared to other LLMs like T5, and it has shown state-of-the-art performance across several abstractive summarization benchmarks, particularly in tasks requiring high content coverage and semantic preservation. Its capacity to produce lexically diverse, coherent, and meaningful rewrites has been repeatedly validated by evaluation metrics such as ROUGE.

To guarantee reproducibility and conformity with commonly used implementation techniques, we utilize the pretrained "google/pegasus-xsum" variant in this study, which can be accessed through the Hugging Face Transformers library. This version can adjust to the subtleties of our target task because it has been refined on a representative subset of our paraphrase

Figure 2
T5 model architecture

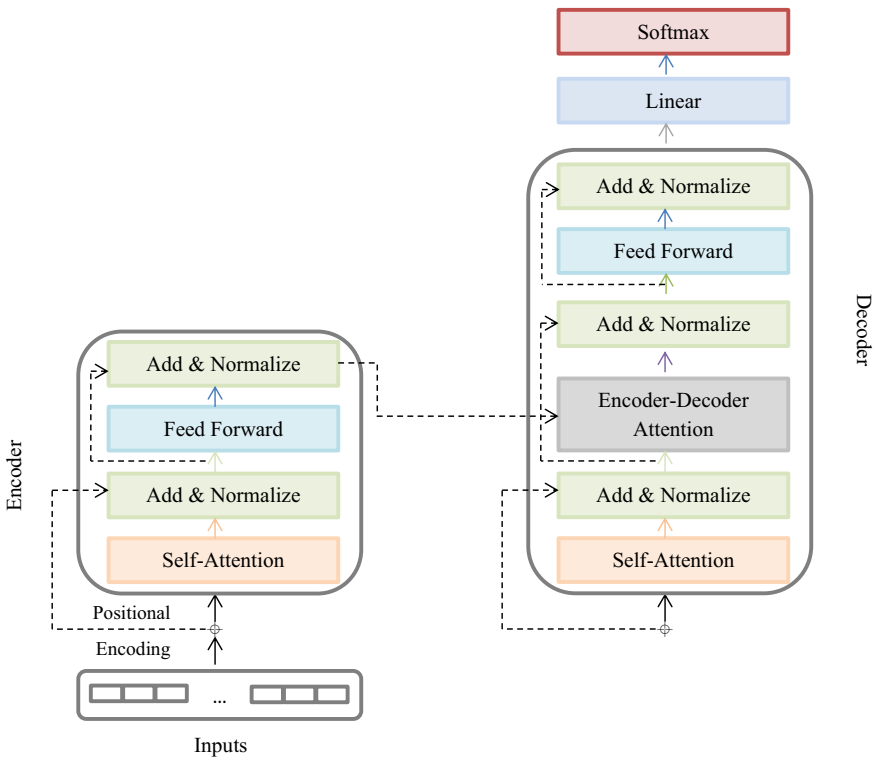
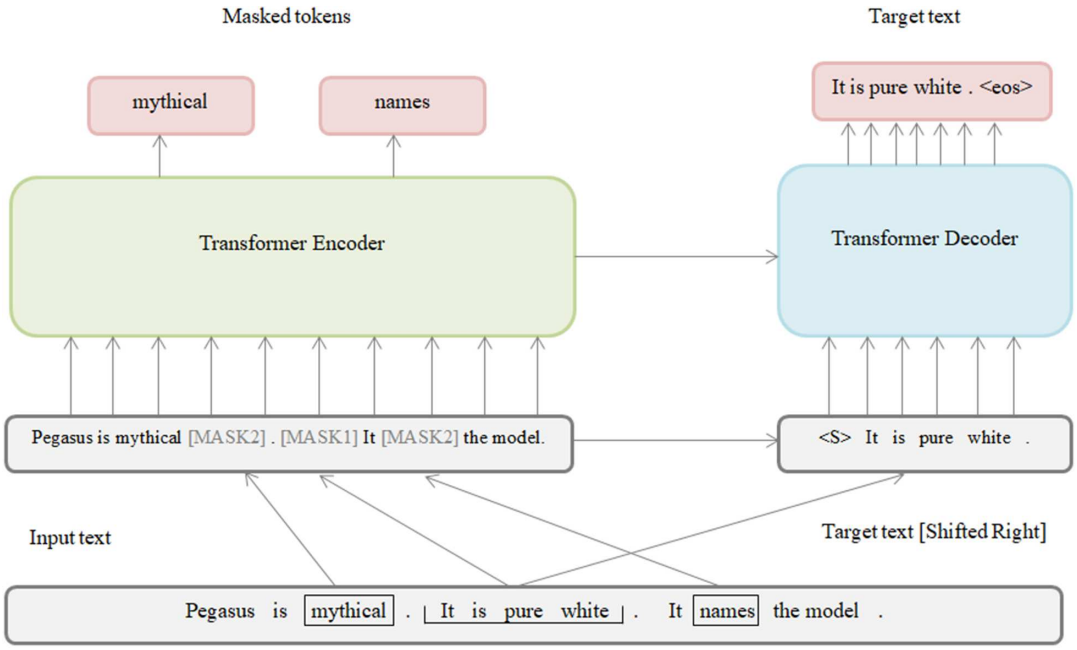


Figure 3
PEGASUS model architecture



dataset. The library's built-in modules handle tokenization and text preprocessing, allowing for a smooth integration into our experimental framework. The encoder–decoder pipeline and gap-sentence pretraining mechanism, which together support the PEGASUS model's robust capacity to produce high-quality paraphrases, are highlighted in Figure 3. We expect to gain a better understanding of how specialized pretraining strategies affect paraphrase generation performance by incorporating PEGASUS into our evaluation framework and comparing its efficacy to T5 and BART-QLoRA.

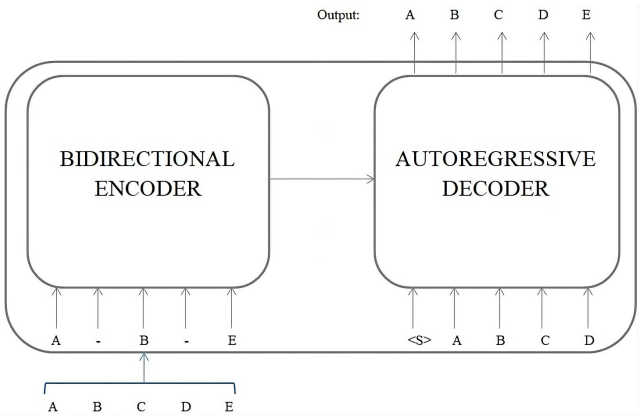
3.3.3. BART model

A potent sequence-to-sequence model called the BART combines the advantages of an autoregressive decoder modeled after GPT and a bidirectional encoder akin to Bidirectional Encoder Representations from Transformers (BERT) [21]. Because of its hybrid architecture, BART can produce coherent, fluid text while efficiently capturing contextual information from both past and future tokens. In a variety of generative NLP tasks, such as machine translation, summarization, and paraphrasing, it has shown competitive performance.

However, complete BART fine-tuning is computationally costly and requires a large amount of memory and processing power, which may restrict its use in practical situations. Our study makes use of QLoRA, a novel method created to drastically cut training time and memory footprint without sacrificing model quality, to overcome this difficulty. In order to accomplish this, QLoRA introduces low-rank adaptation matrices and quantizes model weights to reduce precision. This allows for effective parameter updates while preserving the majority of the pretrained knowledge contained in BART.

In this paper, we use the Hugging Face Transformers library to access the facebook/bart-base checkpoint, and we use QLoRA-based fine-tuning on our carefully selected paraphrase dataset. Even with limited computational resources, this configuration enables us to effectively modify BART for paraphrase generation tasks while preserving competitive performance on par with full-scale fine-tuning techniques. To maintain consistency with the other models in our evaluation, tokenization and text preprocessing are managed within the same framework. We investigate the trade-offs between computational efficiency and generative quality by integrating BART-QLoRA into our experimental pipeline, offering insightful information about how parameter-efficient adaptation strategies can increase the accessibility of large transformer architectures for paraphrasing applications. The BART architecture and the incorporation of QLoRA modules in the fine-tuning procedure are depicted in Figure 4.

Figure 4
BART model architecture



3.4. Metrics assessment

Assessing the effectiveness of transformer-based models for this task requires evaluating the quality of automatically generated paraphrases. It is crucial to use objective and repeatable evaluation metrics that capture both semantic fidelity and linguistic fluency in the generated text, as this is the main focus of our study. In this work, we quantitatively compare the outputs of T5,

PEGASUS, and BART-QLoRA using ROUGE-N and BLEU, two well-known metrics in natural language generation (NLG) research. By assessing lexical overlap, n-gram precision, and content preservation, these metrics offer complementary viewpoints on model performance, allowing for a reliable and impartial comparison of the paraphrasing abilities of the chosen transformer architectures [22, 23].

When ROUGE-N and BLEU are used together, they offer a strong and complementary framework for assessing paraphrase generation models. BLEU prioritizes accuracy and fluency, rewarding outputs that are lexically and syntactically aligned with high-quality references, whereas ROUGE-N emphasizes recall-oriented content preservation through n-gram overlap. By using both metrics, we can capture various aspects of text quality and make sure that the evaluation takes into account readability, grammatical accuracy, and semantic fidelity in addition to lexical similarity [24, 25]. Comparison of transformer-based models for automatic paraphrasing is made possible by this dual-metric approach, which enhances the objectivity and reproducibility of our experimental analysis.

We quantitatively assess the generated paraphrases using ROUGE-N and BLEU metrics, which capture both linguistic fluency and content preservation. While BLEU assesses precision and sequence-level agreement, which reflect the overall quality and readability of the paraphrases, ROUGE-N measures n-gram overlap to determine how well the essential information of the source text is retained. When combined, these metrics offer a solid and trustworthy framework for evaluating model performance in the creation of paraphrases.

3.4.1. ROUGE-N score

One of the most important metrics for assessing NLG tasks is the ROUGE family. In particular, ROUGE-N measures the amount of n-gram overlap between a candidate text generated by a model and a collection of reference texts, acting as a stand-in for lexical fidelity and content preservation. A contiguous sequence of n words is referred to as an n-gram. For example, ROUGE-1 assesses unigram matches, which capture basic word-level similarity, whereas ROUGE-2 takes into account bigram matches, which more accurately reflect local syntactic coherence. Although they are frequently more sensitive to surface-level variations, higher-order ROUGE scores (such as ROUGE-3 and ROUGE-4) can capture more complex phrase-level consistency.

The metric provides complementary insights into the quality of the generated text by calculating precision, recall, and F1-score. Recall measures how much of the reference content is recovered by the candidate, while precision measures how much of the candidate output overlaps with the reference. By combining the two, the F1-score provides a fair assessment of content adequacy.

Because it enables us to objectively evaluate whether transformer-based models like T5, PEGASUS, and BART-QLoRA are able to preserve crucial semantic information from the source text while generating lexically diverse paraphrases, ROUGE-N is especially pertinent to our research. The ROUGE-1 computation is formalized by equation (1), which shows how n-gram matches affect precision, recall, and F1-score:

$$ROUGE-1 = \frac{No.\ of\ overlapping\ unigrams\ in\ ref\ and\ generated\ summary}{Total\ No.\ of\ unigrams\ in\ ref.\ summary} \quad (1)$$

We can empirically benchmark the paraphrasing abilities of various models by utilizing ROUGE-N, going beyond purely qualitative assessments and guaranteeing an impartial, repeatable, and broadly comparable assessment of their performance.

3.4.2. BLEU score

The BLEU score is one of the most established and widely adopted metrics for assessing the quality of machine-generated text, originally introduced for evaluating machine translation systems. BLEU measures how closely a system-generated output resembles one or more human-produced reference texts, providing a quantitative estimate of fluency and adequacy in the generated content.

The precision of n-gram matches between the candidate text and reference texts, which measures how well the model replicates significant word sequences found in high-quality human outputs, is the fundamental basis of BLEU. The metric can assess both lexical choices and local syntactic coherence because it is usually computed across multiple n-gram sizes (such as unigram, bigram, trigram, and four-gram). BLEU includes a brevity penalty to ensure that the score does not favor incomplete or truncated outputs in order to prevent rewarding excessively short predictions.

Thus, the BLEU score remains a common benchmark for NLG applications, such as paraphrase generation, despite being initially created for translation tasks. In this study, we use BLEU to evaluate whether transformer-based models (T5, PEGASUS, and BART-QLoRA) can generate paraphrased outputs that are grammatically correct and lexically aligned with high-quality references. The metric is especially helpful for assessing the fluency and lexical appropriateness of paraphrased text because it emphasizes precision-oriented similarity, providing a complementary viewpoint to ROUGE-N. The BLEU computation is formalized in equation (2), which shows how the brevity penalty and cumulative n-gram matches are added to determine the final score:

$$BLUE = BP \times \sqrt[N]{Precision_1 \times Precision_2 \times \ldots \times Precision_N} \quad (2)$$

$$BP = \begin{cases} 1 \ si\ c > r \\ e^{\left(1-\frac{r}{c}\right)} \ si\ c \leq r \end{cases} \quad (3)$$

Where *c is the lenght of the candidate text; r is the length of the refernce text*.

We guarantee a thorough, repeatable, and multifaceted evaluation of paraphrasing quality across various transformer architectures by incorporating BLEU into our evaluation framework in conjunction with ROUGE-N.

4. Results and Discussions

The empirical results of our research on transformer-based paraphrase generation are presented in this section. The goal is to evaluate and contrast T5, PEGASUS, and BART-QLoRA's capacity to generate lexically diverse, fluent, and semantically faithful paraphrases in a range of input scenarios. To ensure reproducibility and fairness in the comparison, we start by describing the experimental setup, including the datasets, fine-tuning configurations, and evaluation protocols used. We then present a thorough analysis of the findings, emphasizing

Table 3
Experimental setup configuration

Component	Description specification
Programming language	Python 3.10
Main libraries	Hugging Face Transformers, PyTorch, NLTK, Pandas, NumPy, SciPy, Scikit-learn
Models evaluated	T5-small, PEGASUS (google/pegasus-xsum), BART (facebook/bart-base) with QLoRA
Preprocessing tasks	Tokenization, punctuation normalization, stop-word removal, lowercasing
Training strategy	Fine-tuning with Adam optimizer, parameter-efficient adaptation for BART via QLoRA
Evaluation metrics	ROUGE-1 and BLEU scores
Goal of setup	Provide a scalable, reproducible, and fair experimental framework for comparison

each model's relative advantages and disadvantages based on ROUGE-N and BLEU scores, which are backed by qualitative observations. This analysis advances our understanding of automated text rewriting systems by providing insightful information about how architectural decisions and fine-tuning techniques affect paraphrase quality.

4.1. Experiments setup

We used the Python programming language to implement the suggested paraphrase generation framework in our experimental setup, utilizing the Hugging Face Transformers library to access and refine three cutting-edge models: T5, PEGASUS, and BART-QLoRA. To ensure clean and consistent input sequences, the preprocessing pipeline was created using the Natural Language Toolkit for tokenization, case normalization, punctuation cleaning, and stop-word removal. Pandas, NumPy, SciPy, and Scikit-learn enabled scalable and effective data processing by supporting dataset handling, feature engineering, and performance analytics. PyTorch was used for fine-tuning, and QLoRA was incorporated into BART to enable parameter-efficient training, significantly lowering memory usage without sacrificing model quality. Using a Tesla T4 GPU (16 GB VRAM), 16 GB RAM, and a 2.2 GHz CPU, experiments were carried out on Google Colab Pro, which provided the computational power required for extensive analyses. This configuration guaranteed scalability, reproducibility, and fairness, enabling a controlled comparison of transformer-based models under the same circumstances. The experimental setup and primary toolkits used are described in Table 3.

All implementation details and fine-tuning scripts are publicly available at GitHub[1], ensuring full reproducibility of the experiments.

4.2. Experimental comparison of transformer models

Significant performance differences between the three transformer-based models are revealed by the experimental evaluation. A thorough comparison of each model's ROUGE-1, ROUGE-2, ROUGE-L, and BLEU scores is shown in Table 4. These metrics offer a thorough evaluation of lexical richness, fluency, and semantic preservation.

According to the results, T5 consistently receives the highest scores for all metrics (ROUGE-1: 0.6384, ROUGE-2: 0.5121, ROUGE-L: 0.6210, BLEU: 0.3127), demonstrating superior paraphrase quality in terms of both n-gram overlap and sentence-level similarity. Although it is effective, PEGASUS scores are marginally lower, indicating that it may generate paraphrases with less lexical overlap or slight semantic deviations. The balanced trade-off between lexical accuracy and fluency is reflected in BART's intermediate performance.

Table 4
Average ROUGE and BLEU scores for each model

Models	ROUGE-1	ROUGE-2	ROUGE-L	BLUE
T5	0.6384	0.5121	0.6210	0.3127
PEGASUS	0.5825	0.4250	0.5742	0.2584
BART	0.6025	0.4387	0.5921	0.2874

Although ROUGE and BLEU primarily capture surface-level similarity, the consistent ranking of models across multiple metrics provides indirect evidence of semantic preservation and fluency. For instance, the higher ROUGE-2 scores of T5 suggest that bigram sequences, which are important for meaning and context, are better preserved compared to PEGASUS and BART. Overall, these results demonstrate that T5 is the most effective model for producing semantically faithful and fluent paraphrases within our experimental setup.

Figure 5 shows a grouped bar chart of ROUGE-1, ROUGE-2, ROUGE-L, and BLEU scores across all models to better illustrate performance differences. The superior semantic fidelity and fluency of T5 and the economic performance of BART-QLoRA are highlighted in this visualization.

T5 consistently scores highest on all metrics, demonstrating its strong ability to produce paraphrases that are both fluent and semantically faithful, especially for longer inputs. Notable improvements over PEGASUS are +9.6% in ROUGE-1, +20% in ROUGE-2, and +21% in BLEU. These findings show that T5 successfully maintains lexical diversity and contextual coherence while preserving semantic content.

Using its gap-sentence pretraining goal, PEGASUS achieves competitive results (ROUGE-1: 0.5825; BLEU: 0.2584), although it occasionally generates paraphrases with minor semantic omissions or less lexical richness. Despite using fewer computational resources, BART-QLoRA performs well (ROUGE-1: 0.6025; BLEU: 0.2874), highlighting the usefulness of parameter-efficient adaptation with QLoRA.

Example paraphrases produced by each model for a sample sentence are shown in Table 5. This demonstrates variations in lexical variation and semantic preservation.

T5 outputs show greater lexical diversity while maintaining the entire semantic meaning. While BART-QLoRA efficiently generates coherent paraphrases using fewer computational resources, PEGASUS sometimes leaves out details.

[1] https://github.com/cherradii/Paraphrasage

Figure 5
Comparative examination of T5, PEGASUS, and BART-QLoRA using key metrics

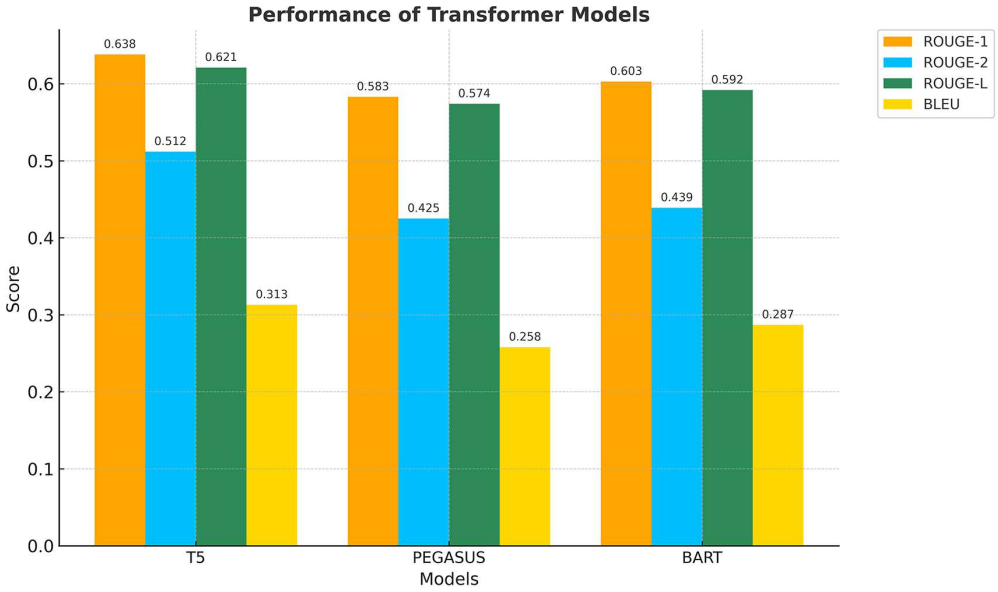


Table 5
Example paraphrases generated by different models

Original Sentence	T5 output	PEGASUS output	BART output
The study focuses on automatic text paraphrasing	This research investigates automatic text paraphrase generation	This work is about automatic paraphrasing of texts	The research concentrates on automatic text paraphrasing

In practical paraphrase generation tasks, where semantic fidelity, fluency, and computational cost must be balanced, this presents our contribution as a step forward in both methodological rigor and empirical performance, providing useful insights for transformer architecture selection.

5. Conclusion and Future Perspectives

In order to address the crucial task of automatic paraphrase generation, this study assessed the capacity of three transformer-based models—such as T5, PEGASUS, and BART-QLoRA—to generate rewrites that are coherent, fluent, and preserve meaning across inputs at the sentence and paragraph levels. Our research hypothesis was validated by the experimental results, which showed that T5 consistently achieves the best performance, especially when it comes to maintaining semantic fidelity and producing a variety of natural expressions, while BART-QLoRA emerges as a competitive, resource-efficient alternative, and PEGASUS exhibits strong capabilities with minor content coverage limitations. By offering a repeatable multi-model benchmark and practical guidance for choosing architectures that strike a balance between computational efficiency and quality in practical text rewriting applications, these findings improve the state of the art. In order to develop more reliable, context-aware, and human-aligned paraphrase generation systems, future research should build on this foundation by incorporating advanced semantic similarity metrics (such as BERTScore), exploring controllable and hybrid paraphrasing strategies, expanding evaluations to multilingual and domain-specific datasets, and integrating human-in-the-loop assessments.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in GitHub at https://github.com/cherradii/Paraphrasage/blob/main/datasets_paraphrases.csv.zip.

Author Contribution Statement

Mohamed Cherradi: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Writing – original draft, Writing – review & editing, Visualization, Supervision. **Hajar El Mahajer:** Conceptualization, Methodology, Software, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration.

References

[1] Zhou, J., & Bhat, S. (2021). Paraphrase generation: A survey of the state of the art. In *Proceedings of the 2021 Conference*

on Empirical Methods in Natural Language Processing, 5075–5086. https://doi.org/10.18653/v1/2021.emnlp-main.414

[2] Lan, W., & Xu, W. (2018). Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In *Proceedings of the 27th International Conference on Computational Linguistics*, 3890–3902.

[3] Natsir, A. H., Hidayah, I., & Adji, T. B. (2023). Deep learning in paraphrase generation: A systematic literature review. In *2023 IEEE 7th International Conference on Information Technology, Information Systems and Electrical Engineering*, 118–123. https://doi.org/10.1109/ICITISEE58992.2023.10405123

[4] Yadav, V., Tang, Z., & Srinivasan, V. (2024). PAG-LLM: Paraphrase and aggregate with large language models for minimizing intent classification errors. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2569–2573. https://doi.org/10.1145/3626772.3657959

[5] Tsai, Y. C., & Lin, F. C. (2023). Paraphrase generation model integrating transformer architecture, part-of-speech features, and pointer generator network. *IEEE Access*, *11*, 30109–30117. https://doi.org/10.1109/ACCESS.2023.3260849

[6] Ge, D., & Gao, J. (2025). Paraphrase discrimination model for data augmentation based on paraphrase text generation. In *2025 International Conference on Advances in Electrical Engineering and Computer Applications*, 601–607. https://doi.org/10.1109/AEECA65693.2025.00110

[7] Zheng, T., & Dai, L. (2025). Large language model enabled multi-task physical layer network. *IEEE Transactions on Communications*, *74*, 307–321. https://doi.org/10.1109/TCOMM.2025.3626010

[8] Palivela, H. (2021). Optimization of paraphrase generation and identification using language models in natural language processing. *International Journal of Information Management Data Insights*, *1*(2), 100025. https://doi.org/10.1016/j.jjimei.2021.100025

[9] Zhang, J., Zhao, Y., Saleh, M., & Liu, P. (2020). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, 11328–11339.

[10] Widjaja, D., Fustian, T., Lucky, H., & Suhartono, D. (2022). Performance comparison of improved common sequence to sequence paraphrasing models. In *2022 3rd International Conference on Artificial Intelligence and Data Sciences*, 299–304. https://doi.org/10.1109/AiDAS56890.2022.9918704

[11] Jyoti, D., Srivastava, J., & Mahato, D. P. (2025). Implementing T5 for text summarization: An algorithmic approach. In *2025 International Conference on Information Networking*, 648–652. https://doi.org/10.1109/ICOIN63865.2025.10992766

[12] Liu, X., Lei, W., Lv, J., Zhou, J., & Raedt, L. D. (2022). Abstract rule learning for paraphrase generation. In *International Joint Conference on Artificial Intelligence*, 4273–4279. https://doi.org/10.24963/ijcai.2022/593

[13] Goswami, J., Prajapati, K. K., Saha, A., & Saha, A. K. (2024). Parameter-efficient fine-tuning large language model approach for hospital discharge paper summarization. *Applied Soft Computing*, *157*, 111531. https://doi.org/10.1016/j.asoc.2024.111531

[14] Shen, L., Liu, L., Jiang, H., & Shi, S. (2022). On the evaluation metrics for paraphrase generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 3178–3190. https://doi.org/10.18653/v1/2022.emnlp-main.208

[15] Perełkiewicz, M., Dadas, S., & Poświata, R. (2025). SMCLM: Semantically Meaningful Causal Language Modeling for autoregressive paraphrase generation. *IEEE Access*, *13*, 119197–119214. https://doi.org/10.1109/ACCESS.2025.3585679

[16] Chowdhury, J. R., Zhuang, Y., & Wang, S. (2022). Novelty controlled paraphrase generation with retrieval augmented conditional prompt tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, *36*(10), 10535–10544. https://doi.org/10.1609/aaai.v36i10.21297

[17] Aguilar, J., Salazar, C., Velasco, H., Monsalve-Pulido, J., & Montoya, E. (2020). Comparison and evaluation of different methods for the feature extraction from educational contents. *Computation*, *8*(2), 30. https://doi.org/10.3390/computation8020030

[18] Yang, Y., Zhang, Y., Tar, C., & Baldridge, J. (2019). PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 3687–3692. https://doi.org/10.18653/v1/D19-1382

[19] Wang, M., Xie, P., Du, Y., & Hu, X. (2023). T5-based model for abstractive summarization: A semi-supervised learning approach with consistency loss functions. *Applied Sciences*, *13*(12), 7111. https://doi.org/10.3390/app13127111

[20] Mukhtar, S., Primadani, C. C., Lee, S., & Jung, P. (2023). A comparison of summarization methods for duplicate software bug reports. *Electronics*, *12*(16), 3456. https://doi.org/10.3390/electronics12163456

[21] Suryadevara, L. S. R., Lagadapati, N., Velamala, V. S. K., Potti, Y. L. S., & Masum, M. (2025). BARTNet: Context-aware deep learning framework for BART ridership forecasting. In *2025 IEEE Conference on Artificial Intelligence*, 825–828. https://doi.org/10.1109/CAI64502.2025.00147

[22] Zhou, C., Qiu, C., Liang, L., & Acuna, D. E. (2025). Paraphrase identification with deep learning: A review of datasets and methods. *IEEE Access*, *13*, 65797–65822. https://doi.org/10.1109/ACCESS.2025.3556899

[23] Lemesle, Q., Chevelu, J., Martin, P., Lolive, D., Delhay, A., & Barbot, N. (2025). Paraphrase generation evaluation powered by an LLM: A semantic metric, not a lexical one. In *Proceedings of the 31st International Conference on Computational Linguistics*, 8057–8087.

[24] Jayawardena, L., & Yapa, P. (2024). Parameter efficient diverse paraphrase generation using sequence-level knowledge distillation. In *2024 5th International Conference on Advancements in Computational Sciences*, 1–12. https://doi.org/10.1109/ICACS60934.2024.10473289

[25] Kazemnejad, A., Salehi, M., & Baghshah, M. S. (2020). Paraphrase generation by learning how to edit from samples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6010–6021. https://doi.org/10.18653/v1/2020.acl-main.535

How to Cite: Cherradi, M., & Mahajer, H. E. (2026). Transformer-Based Approaches in Paraphrasing Texts. *Artificial Intelligence and Applications*. https://doi.org/10.47852/bonviewAIA62027294