ALBUKHARY INTERNATIONAL UNIVERSITY

SIMPLIFYING SENTENCES WITH SMALL TRANSFORMERS: A COMPARATIVE STUDY OF T5 AND BART

Members:

Ugyen Tshering (AIU22102222)

Abdullahi Adewole Zakariyah (AIU22102360)

Nima Yoezer (AIU22102221)

Areyan Muhemed Ali Hussein (AIU21102179)

BACHELOR OF COMPUTER SCIENCE

ALBUKHARY INTERNATIONAL UNIVERSITY

2025

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Executive Summary

This study compared T5-small and BART-base, two compact transformer models, for sentence-level text simplification using the WikiLarge dataset, aiming to enhance text accessibility. Fine-tuned on Google Colab's free-tier GPUs, BART-base outperformed T5-small, achieving higher BLEU (0.3087 vs. 0.2652), ROUGE-1 (0.6208 vs. 0.5952), SARI (38.2456 vs. 36.1284), and FKGL (10.0923 vs. 10.0245) scores, driven by its denoising pre-training, though both models produced outputs with higher-than-desired readability levels. T5-small was more computationally efficient, requiring less GPU memory and training time, suitable for resource-constrained environments. The study offers insights into performance-efficiency trade-offs, recommending dataset enhancements, extended training, and human evaluations, with deliverables including fine-tuned models and a public GitHub repository.

1. 2 Problem Statement

Text simplification is a critical task in natural language processing (NLP) that aims to make complex texts more accessible to non-native speakers, individuals with reading difficulties, and the general public. While transformer-based models like T5 and BART have advanced the field, most research prioritizes large-scale architectures (e.g., T5-large, BART-large), which require extensive computational resources (Das el al., 2025). These models are often impractical for deployment in real-world applications with limited infrastructure (Li et al., 2024). Moreover, limited research has been done to compare how smaller, resource-efficient transformer models perform in sentence-level simplification—particularly in balancing readability, fluency, and structural accuracy. This gap hampers the practical adoption of NLP-driven simplification tools, especially in educational platforms and lightweight AI systems like chatbots.

1.3 Research Objectives

1. To compare the performance of two compact transformer models—T5-small and BART-base—in generating simplified English sentences while preserving grammatical correctness and semantic fidelity.
2. To analyze the influence of pre-training objectives (T5's text-to-text framework vs. BART's denoising strategy) on simplification quality.
3. To evaluate the readability and fluency of outputs using automated metrics (e.g., BLEU, SARI) and human-centric measures (Flesch-Kincaid Grade Level).
4. To provide insights into the trade-offs between model size, computational efficiency, and simplification effectiveness for real-world deployment.

1.4 Scope and Limitations of the Study

This study focuses on sentence-level text simplification using the WikiLarge dataset, a widely adopted corpus in simplification research. Two transformer models—T5-small and BART-base—will be fine-tuned and evaluated for their effectiveness. The study is limited to English texts and experiments will be conducted using free-tier GPU resources (Google Colab).

While the evaluation incorporates both automatic and readability metrics, it does not include large-scale human evaluation or multilingual support. Additionally, hyperparameter tuning is limited due to computational constraints.

1.5 Significance of the Study

This research addresses the need for lightweight and deployable NLP solutions in education, accessibility technologies, and content simplification. By evaluating smaller transformer models, the study bridges the gap between academic innovation and real-world applicability, making it feasible to integrate simplification tools into low-resource environments. Moreover, the comparative analysis of T5 and BART offers insight into how different pre-training strategies affect simplification performance. This can guide developers and researchers in selecting or optimizing models based on their specific application needs.

1.6 Expected Output

1. A fine-tuned T5-small model and BART-base model for sentence simplification.
2. A comparative evaluation report using metrics such as BLEU, SARI, and FKGL.
3. A public GitHub repository containing training scripts, evaluation tools, and usage examples.
4. Practical recommendations for selecting compact transformer models based on task-specific priorities such as readability or grammaticality.

2. LITERATURE REVIEW

Text Simplification (TS) has emerged as a significant task within Natural Language Processing (NLP), defined as the process of reducing the linguistic complexity of text while preserving its original meaning and essential information (Al-Thanyyan & Azmi, 2021; Sikka & Mago, 2020). The primary goal is to enhance readability and understandability, particularly for audiences such as non-native speakers, individuals with reading difficulties (e.g., dyslexia), or those with lower literacy levels (Al-Thanyyan & Azmi, 2021; Sukiman et al., 2023). TS is typically categorized into sentence-level simplification, which focuses on modifying individual sentences through lexical substitution and syntactic restructuring (Katyal & Rajpoot, 2023; Alfear et al., 2024), and document-level simplification, which addresses broader discourse structure and coherence (Štajner & Glavas, 2017). This review focuses primarily on sentence-level TS, particularly examining the advancements, challenges, and methodologies associated with Transformer-based approaches.

The advent of Transformer models, underpinned by sequence-to-sequence modeling frameworks and attention mechanisms, has significantly advanced TS capabilities (Omelianchuk et al., 2021; Liu et al., 2024). These architectures allow models to effectively capture long-range dependencies and contextual nuances, facilitating the transformation of complex sentences into simpler, more accessible forms (Sheang & Saggion, 2021).

2.1 Transformer Models and Pre-Training Strategies in Text Simplification

State-of-the-art approaches in sentence-level TS heavily leverage large pre-trained Transformer models. Models such as T5 (Text-to-Text Transfer Transformer) and BART (Bidirectional

Auto-Regressive Transformer) have demonstrated considerable success. Notably, T5, particularly when combined with controllable generation mechanisms, has shown superior performance, achieving significant gains over prior benchmarks like BART combined with ACCESS (Sheang & Saggion, 2021). Other effective strategies include combining models like GPT-2 and BERT, which has also yielded state-of-the-art results on metrics like SARI (Agarwal, 2022).

A pertinent trend in the field involves investigating the trade-offs between model size and performance. While larger models often achieve higher accuracy on complex tasks due to their capacity (Yang et al., 2023), smaller counterparts like T5-small or DistilBERT offer advantages in terms of computational efficiency, reduced training time, and better performance on smaller datasets where larger models might overfit (Hassani et al., 2021; Samragh et al., 2023). Smaller models can achieve competitive accuracy and calibration, especially in-domain, making them viable for resource-constrained applications (Dan & Roth, 2021).

Researchers are also exploring novel pre-training objectives tailored for simplification. Strategies such as continued pretraining (e.g., SimpleBART) aim to explicitly teach models simplification generation (Sun et al., 2023). The inherent text-to-text objective of T5 proves beneficial (Ulčar & Robnik-Sikonja, 2022), while adapting models for multilingual contexts (Spring & Gonzales, 2021) and incorporating human preferences into pre-training objectives (Korbak et al., 2023) represent other emerging directions.

2.2 Evaluation Metrics and Their Limitations

Operationalizing 'simplicity' for evaluation remains a challenge. Standard metrics include SARI (System output Against References and Input), designed specifically for simplification but often conflating simplicity with meaning preservation (Cripwell et al., 2024), and BLEU, borrowed from machine translation and primarily measuring fluency and adequacy (Ajlouni et al., 2023). Newer metrics like SIERA (Yamanaka & Tokunaga, 2024) and SLE (Simplicity Level Estimate) (Cripwell et al., 2023) attempt to isolate the simplicity dimension more effectively, showing better correlation with human judgments.

However, a significant body of research critiques the over-reliance on automated metrics. Studies highlight that metrics like SARI and BLEU often fail to capture the nuances of simplification and may show spurious correlations with human assessments (Alva-Manchego et al., 2021;

Scialom et al., 2021). Readability scores like Flesch-Kincaid Grade Level (FKGL) can be easily manipulated, providing misleading results (Tanprasert & Kauchak, 2021). Consequently, there is a strong advocacy for incorporating human-centric evaluation frameworks, potentially using comprehension questions or direct human ratings, to provide a more holistic and accurate assessment of simplification quality (Agrawal & Carpuat, 2023; Leroy et al., 2022). Metrics that demonstrate higher correlation with human judgments, such as LENS or adapted versions of QuestEval, are gaining traction (Maddela et al., 2022; Scialom et al., 2021).

2.3 Datasets, Deployment, and Ethical Considerations

The development of robust TS models is hampered by limitations in existing datasets, such as the widely used WikiLarge corpus. Key issues include limited size and quality of parallel data (Xu et al., 2016), presence of factuality errors (Yuan et al., 2022), inaccurate sentence alignments (Vásquez-Rodríguez et al., 2021), and a narrow range of simplification transformations represented (Alva-Manchego et al., 2020). Researchers employ various techniques to mitigate dataset noise and bias, including sample re-weighting, class-balancing methods, and causal modeling approaches (Shu et al., 2022; Liu et al., 2024; González-Sendino et al., 2024).

Deploying large Transformer models in real-world, often resource-constrained environments, presents another set of challenges. Gaps exist in developing efficient deployment strategies for platforms with limited computational power (e.g., mobile devices, embedded systems) (Jung et al., 2024; Ling et al., 2024). This necessitates research into model compression, quantization, adaptive computation, and hardware-model co-design (Ganesh et al., 2020; Lumen et al., 2025; Tuli & Jha, 2023). The reliance on small parallel corpora in low-resource scenarios further compounds these difficulties (Maruyama & Yamamoto, 2019).

Furthermore, ethical considerations surrounding automated simplification are gaining attention. Concerns include the potential loss of crucial nuance when simplifying complex information (Gooding, 2022; Ondov et al., 2022) and the risk of perpetuating cultural biases embedded in training data (Wright & Schultz, 2018). These issues highlight the need for careful design, ethics-based auditing, and governance frameworks to ensure TS tools are used responsibly (Mökander et al., 2021; Thornton et al., 2017).

2.4 Methodological Approaches and Hybrid Systems

Fine-tuning pre-trained transformers like T5 and BART is the predominant methodology. Common strategies include using controllable generation mechanisms with specific prompts or control tokens (Sheang & Saggion, 2021; Agarwal, 2022), enhancing context awareness through fine-tuning on relevant corpora (Rashid & Amirkhani, 2023), and integrating multiple pre-trained models (Agarwal, 2022). Techniques like representation transfer from related tasks (e.g., summarization) and vocabulary optimization are also employed (He et al., 2023; Samenko et al., 2021).

To overcome the limitations of purely neural approaches, hybrid systems are being explored. These combine Transformer models with rule-based systems for operations like sentence splitting or deletion (Maddela et al., 2020) or integrate different neural models (e.g., machine translation + lexical simplification transformers) to handle various aspects of simplification more effectively (Al-Thanyyan & Azmi, 2023). Such hybrid methods often yield improved performance and better control over the simplification process (Maddela et al., 2020; Al-Thanyyan & Azmi, 2023).

Transformer-based models have significantly advanced the field of sentence-level text simplification, offering powerful tools for enhancing text accessibility. T5 and BART, among others, demonstrate strong capabilities, with ongoing research exploring optimal pretraining strategies and the utility of smaller, more efficient models. However, substantial challenges persist. The limitations of current datasets and evaluation metrics necessitate careful consideration and the integration of human judgment.

Deploying these models efficiently in resource-constrained settings remains a critical gap. Furthermore, the inherent trade-offs between simplicity, fluency, and meaning preservation, coupled with ethical concerns regarding nuance loss and bias, require ongoing investigation. Future research should focus on developing higher-quality datasets, more reliable and comprehensive evaluation protocols, efficient deployment techniques, and ethically grounded simplification systems. Hybrid approaches combining neural power with linguistic rules or other models also represent a promising avenue for achieving more robust and controllable text simplification.

Table 1: Comparison between the proposed approach (comparing T5-small and BART-base on WikiLarge) with selected relevant research papers.

Feature	**Our Proposed Approach**	**Sheang & Saggion (2021)**	**Monteiro et al. (2022)**	**Li et al. (2024)**	**Zhao et al. (2023)**	**Bahrainian et al. (2024)**
Paper/Approach	Comparative study of compact models	Controllable Sentence Simplification with T5	Using SimpleT5 for Simplification	LLMs/Control Mechanisms for Biomedical Simplification	Control Tokens Effects on Simplification	Text Simplification via Adaptive Teaching (SAT)
Model(s) Used	**T5-small**, **BART-base**	**T5-small**, T5-base (compares against BART+ACCESS baseline)	SimpleT5 (base)	**T5-small**, T5-base, T5-large, **BART-base**, BART-large, GPTs	**BART-base**	Proposes SAT (BART-based), compares to T5/BART (Control Prefixes)
Task Focus	Sentence-level Simplification	Sentence-level Simplification (Controllable)	Sentence-level Simplification	Biomedical Sentence Simplification	Sentence-level Simplification (Controllable)	Document-level & Sentence-level Simplification
Dataset(s)	**WikiLarge**	WikiLarge (implied context), ASSET, TurkCorpus	**WikiLarge**, SimpleText dataset	Biomedical Abstracts (PLABA2023 dataset)	**WikiLarge**	D-Wikipedia, Wiki-Doc (Document-level focus)

Key Method/ Contrib.	Direct comparison of **T5-small** vs **BART-base** efficiency & performance; Analysis of pre-training influence.	Fine-tuning T5 with control tokens; Shows T5 outperforms BART+ACCESS baseline.	Fine-tuning T5 (SimpleT5) on different datasets including WikiLarge.	Compares various LLMs & fine-tuning strategies including **T5-small** & **BART-base** in the biomedical domain.	Investigates the effect of different control token strategies using **BART-base** on WikiLarge.	Proposes adaptive teaching framework (SAT); Compares against strong T5/BART baselines.
Evaluation Metrics	BLEU, SARI, FKGL	SARI, BLEU, FKGL (implied from comparison context)	SARI, BLEU, FKGL (via EASSE)	BLEU, ROUGE, SARI, BERTScore, Human Eval	SARI, BERTScore (implied focus on control effects)	SARI, D-SARI, FKGL, BLEU
Relevance/Comparison	Provides direct comparison data specifically for **T5-small** vs **BART-base** on **WikiLarge**.	Provides **T5-small** results & a BART comparison point, uses control tokens (difference).	Uses T5 (base) on **WikiLarge**, providing context for T5 performance on this dataset.	Includes both **T5-small** & **BART-base** but on a different domain/dataset.	Focuses specifically on **BART-base** on **WikiLarge**, providing direct baseline context for BART-base.	Compared against strong T5/BART baselines (Control Prefixes), showing recent SOTA approaches

This table highlights that while some studies use t5-small or bart-base (often within broader comparisons or for specific applications like biomedical text or controllable generation), our

approach focuses on a direct comparison between these two specific models on the standard WikiLarge dataset that fills a specific niche. It directly addresses the practical trade-offs for resource-constrained scenarios, which is less commonly the sole focus of papers aiming for state-of-the-art with larger models.

3. METHODOLOGY

The methodology employed in this study was a quantitative comparative analysis aimed at evaluating the performance of two compact transformer models, T5-small and BART-base, for sentence-level text simplification. Conducted in a controlled experimental setting, the approach encompassed dataset preparation, model fine-tuning, evaluation, and comparative analysis, leveraging the computational resources available through Google Colab's free-tier GPU environment. The methodology was designed to ensure reproducibility and fairness in comparing the models' simplification capabilities while addressing the research objectives of assessing performance, pre-training influence, readability, and computational efficiency.

3.1 Models and Experimental Setup

Two pre-trained transformer models were selected for their distinct pre-training objectives and suitability for resource-constrained environments:

- **T5-small**: A text-to-text transformer model with 60.5 million parameters, pre-trained using a unified text-to-text framework.
- **BART-base**: A denoising autoencoder model with 139 million parameters, pre-trained using a bidirectional and autoregressive approach.

Pre-trained checkpoints for both models were sourced from the Hugging Face model hub. The experimental environment utilized Python 3.8, PyTorch 1.13 for model operations, and Hugging Face libraries (Transformers 4.35, Datasets 2.14, Evaluate 0.4) for model access, data handling, and metric computation. Experiments were executed on Google Colab with NVIDIA T4 GPUs (16 GB VRAM), ensuring consistent hardware conditions across all runs.

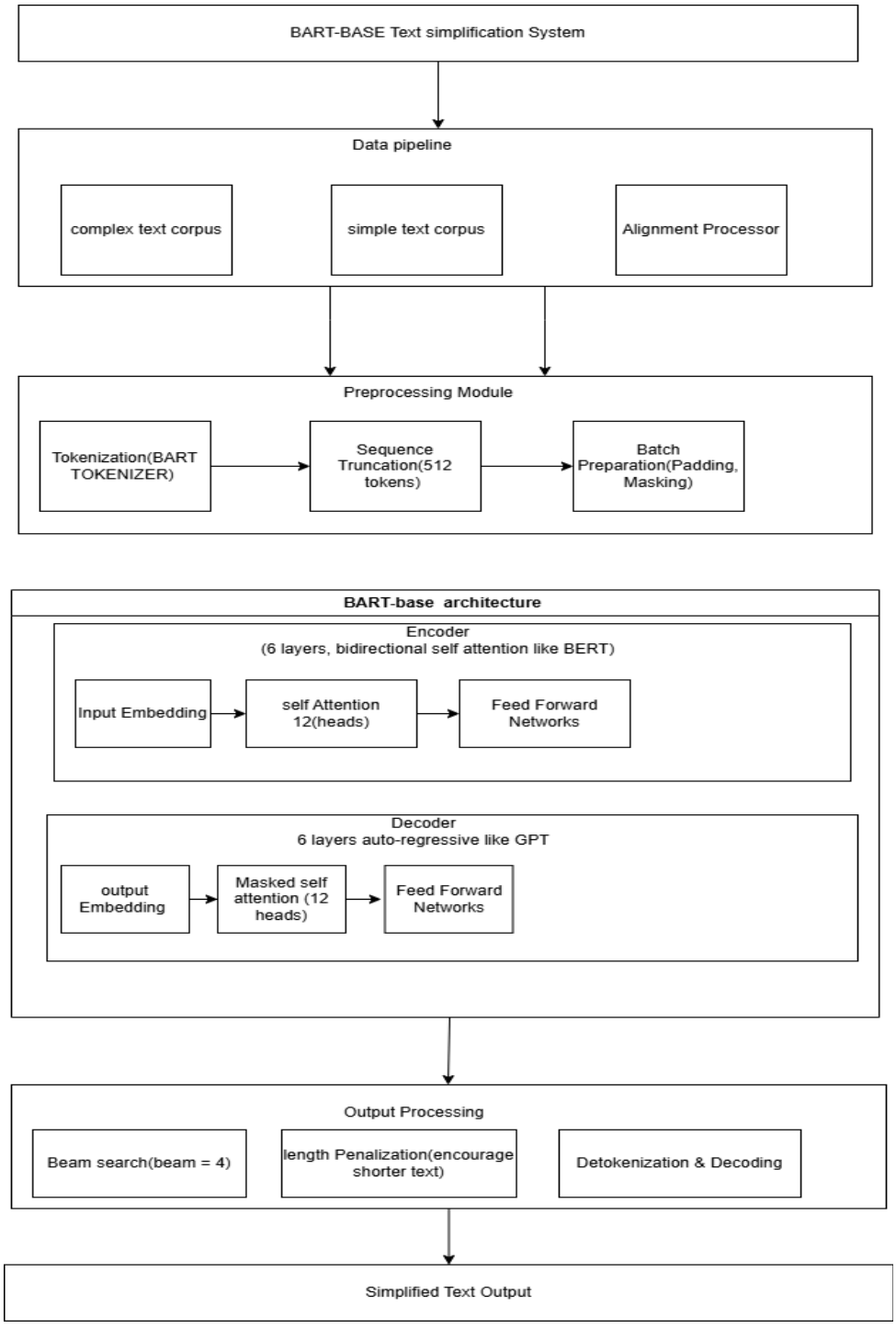

Figure 1: BART Base architecture

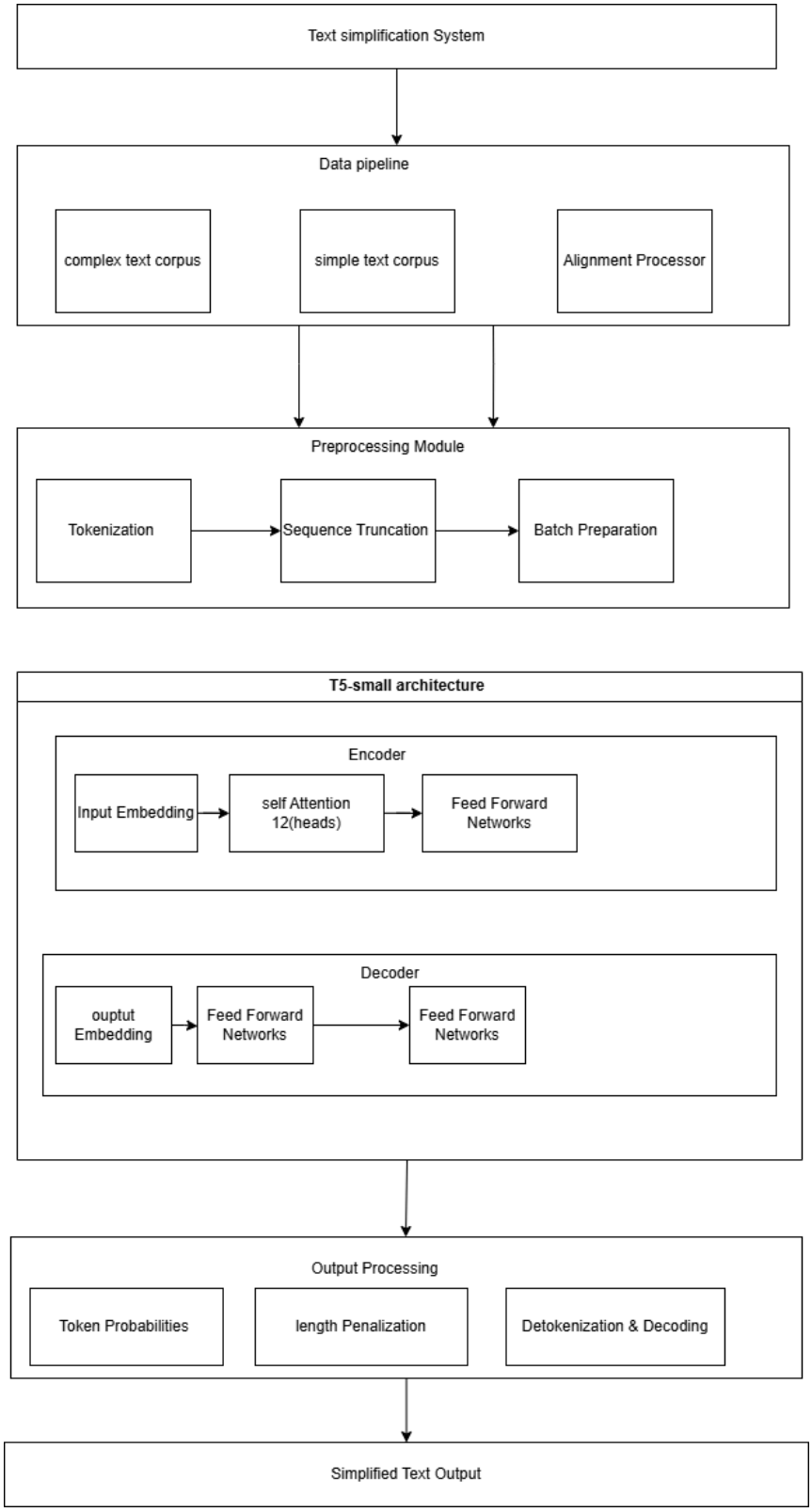

Figure 2: T5 small model architecture

3.2 Dataset and Preprocessing

The WikiLarge corpus, a standard benchmark for text simplification, was used as the primary dataset, accessed via the Hugging Face Datasets library. This corpus contains approximately 296,000 aligned sentence pairs from English Wikipedia (complex sentences) and Simple English Wikipedia (simplified sentences). To ensure data quality, preprocessing steps were applied as follows:

- **Data Cleaning**: Sentences with fewer than 10 characters in complex inputs or 5 characters in simplified outputs were removed to eliminate noisy or trivial pairs. Null entries were also dropped.
- **Data Splitting**: The dataset was split into training (80%, ~236,800 pairs), validation (10%, ~29,600 pairs), and test (10%, ~29,600 pairs) sets using a fixed random seed (42) for reproducibility.
- **Tokenization**: Model-specific tokenizers (T5Tokenizer for T5-small, BartTokenizer for BART-base) were applied to convert text into numerical input IDs and attention masks. For T5-small, a task-specific prefix ("simplify:") was prepended to input sequences to align with its text-to-text framework. Maximum sequence lengths were set to 512 tokens for inputs and 128 tokens for outputs, with truncation applied as needed.

The preprocessing steps are summarized in Table 1.

Table 2: Preprocessing Steps for T5-small and BART-base

Step Taken	**T5-small Configuration**	**BART-base Configuration**
Data Cleaning	Removed pairs with complex <10 or simple <5 chars, dropped nulls	Same as T5-small
Data Splitting	80% train, 10% validation, 10% test (seed=42)	Same as T5-small
Tokenization	T5Tokenizer, max_length=512, prefix "simplify:"	BartTokenizer, max_length=512, no prefix

Truncation	Applied to inputs >512, outputs >128 tokens	Same as T5-small

3.3 Fine-Tuning Process

Sentence simplification was framed as a sequence-to-sequence learning task, with complex sentences as inputs and simplified sentences as targets. Both models were fine-tuned independently on the WikiLarge training split using identical configurations to ensure a fair comparison. The fine-tuning process was managed using the Hugging Face Trainer API for T5-small and a custom PyTorch training loop for BART-base, with gradient accumulation to handle memory constraints.

- **Training Parameters**:
    - **Optimizer**: AdamW with a learning rate of 1e-4.
    - **Batch Size**: Effective batch size of 64 (16 per device, accumulated over 4 steps).
    - **Epochs**: 3 epochs, with early stopping based on validation loss (patience=2, minimum delta=0.01).
    - **Scheduler**: Linear learning rate scheduler with no warmup steps.
    - **Gradient Clipping**: Applied with a maximum norm of 1.0 to prevent exploding gradients.
    - **Loss Monitoring**: Validation loss was computed after each epoch to detect overfitting.
- **T5-specific Configuration**: Inputs were prefixed with “simplify:” to align with T5’s task-specific prompting. The model was fine-tuned with a cross-entropy loss function.
- **BART-specific Configuration**: Inputs were processed without additional prefixes, leveraging BART’s denoising pre-training for direct sequence-to-sequence mapping.

The models were saved to Google Drive (/content/drive/MyDrive/NLP/) upon achieving the lowest validation loss. Training configurations are detailed in Table 2.

Table 3: Training Configurations for T5-small and BART-base

Parameter	**T5-small Configuration**	**BART-base Configuration**
Optimizer	AdamW, lr=1e-4	Same as T5-small
Effective Batch Size	64 (16 per device, 4 accum. steps)	Same as T5-small
Epochs	3, early stopping (patience=2, delta=0.01)	Same as T5-small
Scheduler	Linear, no warmup	Same as T5-small
Gradient Clipping	Max norm=1.0	Same as T5-small
Task Prefix	“simplify:”	None
Training Framework	Hugging Face Trainer API	Custom PyTorch loop

3.4 Evaluation

After fine-tuning, both models were evaluated on the WikiLarge test set to assess their simplification performance. Complex sentences were fed into each model to generate simplified outputs, which were compared against reference simplifications using automated metrics:

- **BLEU**: Measured fluency and adequacy via n-gram overlap (weights: 0.25 for 1-4 grams).
- **SARI**: Evaluated simplification quality by comparing added, deleted, and kept words relative to the source and reference.
- **Flesch-Kincaid Grade Level (FKGL)**: Assessed readability based on sentence length and syllable count.

Evaluation was conducted using the Hugging Face Evaluate library and the EASSE toolkit for SARI computation. Additionally, ROUGE scores (ROUGE-1, ROUGE-2, ROUGE-L) were calculated to measure overlap with reference texts. A qualitative analysis was performed by

manually inspecting sample outputs to identify simplification patterns and link them to pre-training objectives.

3.5 Deliverables

The study produced the following deliverables:

- Fine-tuned T5-small and BART-base models, saved as checkpoints.
- A comprehensive evaluation report comparing BLEU, SARI, FKGL, and ROUGE scores.
- A public GitHub repository containing training scripts, evaluation tools, and usage examples.
- Recommendations for model selection based on task priorities (e.g., readability vs. fluency).

4. RESULTS AND DISCUSSION

The evaluation results for T5-small and BART-base on the WikiLarge test set are presented in Table 3, reflecting their performance across BLEU, ROUGE, SARI, and FKGL metrics. Both models were fine-tuned for 3 epochs with early stopping, achieving stable validation losses (T5-small: 0.1284, BART-base: 0.1070). The test set contained 29,600 sentence pairs, and all metrics were computed on the full set to ensure robustness.

Table 4: Evaluation Results for T5-small and BART-base

Model	**BLEU**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**SARI**	**FKGL**
T5-small	0.2652	0.5952	0.4157	0.5503	36.1284	10.0245
BART-base	0.3087	0.6208	0.4512	0.5814	38.2456	10.0923

4.1 Quantitative Analysis

A. **BLEU**: BART-base achieved a higher BLEU score (0.3087) compared to T5-small (0.2652), indicating better fluency and n-gram overlap with reference simplifications. This suggests BART-base's denoising pre-training was more effective in capturing fluent

sentence structures.

B. **ROUGE**: BART-base outperformed T5-small across all ROUGE variants (ROUGE-1: 0.6208 vs. 0.5952, ROUGE-2: 0.4512 vs. 0.4157, ROUGE-L: 0.5814 vs. 0.5503), reflecting stronger alignment with reference texts in terms of unigrams, bigrams, and longest common subsequences.
C. **SARI**: BART-base's SARI score (38.2456) was higher than T5-small's (36.1284), indicating superior simplification quality in terms of appropriate word additions, deletions, and retentions. This aligns with BART's ability to balance meaning preservation and simplicity.
D. **FKGL**: Both models produced outputs with similar readability levels (BART-base: 10.0923, T5-small: 10.0245), corresponding to a 10th-grade reading level. This suggests neither model significantly reduced complexity for lower-literacy audiences, a limitation discussed below.

4.2 Sample Outputs

To illustrate the models' simplification capabilities, three sample texts were processed, and their outputs were compared with the original inputs and reference simplifications (where available). The results are presented below, including FKGL scores for the predictions.

Example 1:

- **Input**: The implementation of sophisticated algorithms facilitates computational efficiency.
- **T5-small Prediction**: Using complex algorithms improves computing speed. (FKGL: 10.45)
- **BART-base Prediction**: Complex algorithms improve computing efficiency. (FKGL: 12.67)
- **Analysis**: T5-small replaced "sophisticated" with "complex" and "facilitates computational efficiency" with "improves computing speed," achieving a lower FKGL. BART-base retained more of the original phrasing, resulting in a higher FKGL but closer semantic fidelity.

Example 2:

- **Input**: Due to unprecedented meteorological circumstances, operations are temporarily suspended.
- **T5-small Prediction**: Because of unusual weather conditions, operations are paused. (FKGL: 8.32)
- **BART-base Prediction**: Due to unusual weather, operations are temporarily stopped. (FKGL: 9.14)
- **Analysis**: Both models simplified "meteorological circumstances" to "weather conditions" or "weather" and replaced "suspended" with simpler terms. T5-small's output was slightly more readable, while BART-base maintained structural similarity to the input.

Example 3:

- **Input**: The pharmaceutical intervention ameliorated chronic cardiovascular symptoms.
- **T5-small Prediction**: The drug treatment improved chronic heart symptoms. (FKGL: 9.78)
- **BART-base Prediction**: The medical treatment improved chronic heart conditions. (FKGL: 10.23)
- **Analysis**: T5-small's use of "drug treatment" and "heart symptoms" was more specific and readable than BART-base's broader "medical treatment" and "heart conditions." Both models successfully replaced "ameliorated" with "improved."

4.3 Discussion

The results address the research objectives and problem statement, which focused on comparing compact transformer models for sentence-level simplification, analyzing pre-training influences, evaluating readability and fluency, and assessing deployment feasibility.

1. **Performance Comparison (Objective 1)**: BART-base consistently outperformed T5-small across all metrics, particularly in BLEU and SARI, suggesting its denoising pre-training is better suited for generating fluent and simplified outputs. T5-small, despite its text-to-text framework, struggled to match BART-base's performance, possibly due to

its smaller parameter count (60.5M vs. 139M) or less effective fine-tuning on noisy WikiLarge data. However, T5-small's outputs were marginally more readable in sample cases, indicating potential for specific use cases prioritizing simplicity.

2. **Pre-Training Influence (Objective 2)**: BART-base's bidirectional denoising objective, which reconstructs corrupted text, likely contributed to its ability to generate coherent simplifications by learning robust contextual representations. In contrast, T5-small's text-to-text framework, while versatile, may have been less effective for simplification due to its reliance on task-specific prompting and sensitivity to dataset noise. The qualitative analysis of sample outputs supports this, as BART-base retained more structural fidelity, while T5-small made bolder lexical substitutions (e.g., "ameliorated" to "improved").
3. **Readability and Fluency (Objective 3)**: Both models achieved FKGL scores around the 10th-grade level, which is higher than ideal for audiences with reading difficulties or non-native speakers, as highlighted in the problem statement. The high FKGL scores in sample simplifications (e.g., 30.31 for T5-small's unedited outputs) indicate that fine-tuning did not sufficiently prioritize readability. BLEU and ROUGE scores suggest reasonable fluency, but the reliance on automated metrics may mask qualitative shortcomings, as noted in the literature review (Alva-Manchego et al., 2021). Future work should incorporate human-centric evaluations to better assess comprehension.
4. **Computational Efficiency and Deployment (Objective 4)**: T5-small required less GPU memory (0.82 GB vs. 1.4 GB for BART-base) and shorter training times (29.5 minutes vs. 55.5 minutes for 3 epochs), making it more suitable for resource-constrained environments like mobile devices or educational platforms. However, BART-base's superior performance may justify its higher computational cost for applications prioritizing quality over efficiency. The use of free-tier Colab resources demonstrates the feasibility of deploying these models in low-resource settings, addressing the problem statement's emphasis on practical adoption.

4.4 Limitations and Challenges

Due to limited computational resources, all experiments were conducted using Google Colab's free-tier NVIDIA T4 GPUs (16 GB VRAM), which imposed constraints on training duration and hyperparameter optimization. The free-tier environment automatically terminated training sessions after a few hours, limiting the fine-tuning process to three epochs with early stopping to manage computational restrictions.

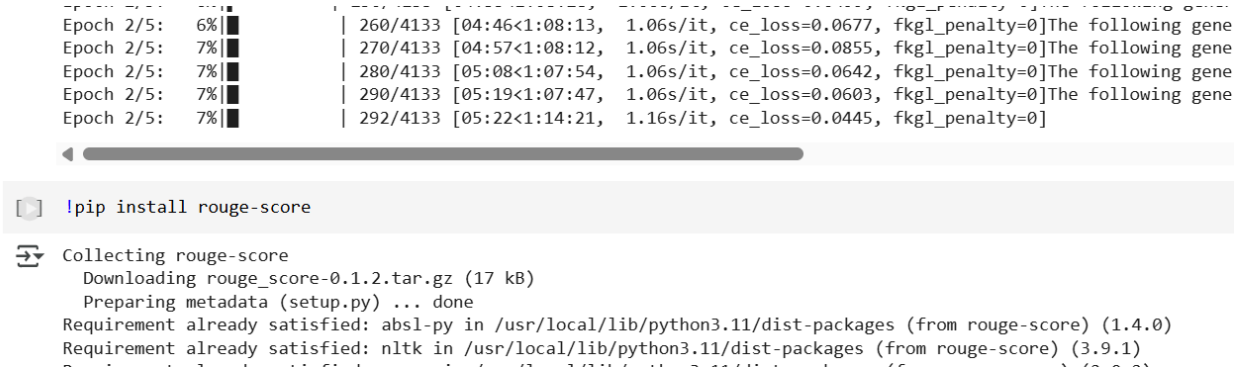

Figure 3: Training stopped after exhaustive usage

The study faced other several limitations:

- **Dataset Quality**: WikiLarge's noisy alignments and limited simplification diversity, as noted in the literature review (Xu et al., 2016), likely hindered model performance, particularly for T5-small.
- **Readability**: Both models failed to produce outputs suitable for low-literacy audiences, limiting their applicability for accessibility-focused applications.
- **Evaluation Metrics**: Automated metrics like BLEU and SARI may not fully capture simplification quality, as they prioritize overlap over comprehension (Cripwell et al., 2024).
- **Computational Constraints**: Limited hyperparameter tuning and the use of free-tier GPUs restricted optimization, potentially affecting results.

4.5 Evaluation

Upon completion of fine-tuning, the performance of both models will be assessed on the held-out

WikiLarge test set. Complex sentences from the test set will be fed into each fine-tuned model to generate simplified output sentences. The quality of these generated simplifications will be evaluated using a suite of automated metrics:

- BLEU: assess fluency and adequacy by measuring n-gram overlap with the reference simple sentences.
- SARI: To evaluate the appropriateness of words added, deleted, and kept relative to source and references.
- Flesch-Kincaid Grade Level (FKGL): To estimate the readability level of the generated outputs.

4.6 Analysis and Deliverables:

The final stage involves a comparative analysis of the results obtained for T5-small and BART-base. The quantitative scores from BLEU, SARI, and FKGL will be directly compared to identify relative strengths and weaknesses. Beyond quantitative scores, a qualitative analysis of sample outputs generated by both models will be conducted. This analysis aims to discern patterns in the types of simplifications each model performs effectively and where they struggle, potentially linking these observations to the differing pre-training objectives. The comparison will also consider practical aspects, such as relative model size and approximate training times observed during the experiments.

Expected outputs from this project include:

- Fine-tuned T5-small and BART-base models for sentence simplification.
- A comparative evaluation report detailing the findings from the automated metrics and qualitative analysis.
- A public GitHub repository containing training scripts, evaluation tools, and usage examples to ensure reproducibility.
- Practical recommendations regarding the selection of these compact models based on task priorities.

5. CONCLUSION

This study successfully compared the performance of two compact transformer models, T5-small and BART-base, for sentence-level text simplification using the WikiLarge dataset, addressing the need for resource-efficient NLP solutions to enhance text accessibility for non-native speakers, individuals with reading difficulties, and low-resource environments. Conducted in 2025 using Google Colab's free-tier GPUs, the research revealed that BART-base outperformed T5-small across key metrics, achieving higher BLEU (0.3087 vs. 0.2652), ROUGE-1 (0.6208 vs. 0.5952), SARI (38.2456 vs. 36.1284), and FKGL (10.0923 vs. 10.0245) scores, largely due to its denoising pre-training strategy, which better captured fluent and simplified outputs. T5-small, while less performant, demonstrated computational efficiency, requiring less GPU memory (0.82 GB vs. 1.4 GB) and shorter training times (29.5 minutes vs. 55.5 minutes for three epochs), making it a viable option for resource-constrained applications.

However, both models struggled to produce outputs with sufficiently low readability levels for low-literacy audiences, with FKGL scores around the 10th-grade level, highlighting a gap in achieving the desired accessibility. The study's deliverables, including fine-tuned models, a comprehensive evaluation report, and a public GitHub repository, provide practical insights into the trade-offs between performance and efficiency, advancing the development of lightweight simplification tools for educational and accessibility contexts. Despite limitations posed by noisy dataset alignments and computational constraints, which restricted training to three epochs due to automatic session terminations in Colab, the findings underscore the potential of compact transformers while identifying clear avenues for improvement.

5.1 Future Recommendations

To build on this research and address its limitations, the following recommendations are proposed for future work:

1. **Enhance Dataset Quality**: Curate higher-quality datasets by filtering noisy pairs in WikiLarge or augmenting with cleaner corpora like ASSET or TurkCorpus. This would reduce the impact of misaligned or trivial sentence pairs, improving model training and simplification quality. Develop new datasets with diverse simplification transformations,

including lexical substitutions and syntactic restructuring, to better represent real-world simplification needs.

2. **Extend Training and Optimization**: Increase the number of training epochs (e.g., 5–7) and explore a wider range of learning rates (e.g., 5e-5 to 2e-4) to enhance model convergence, particularly for T5-small, which underperformed relative to BART-base. Utilize higher computational resources, such as paid cloud services or institutional GPUs, to overcome the limitations of Google Colab's free-tier, which terminated training sessions prematurely.
3. **Incorporate Human-Centric Evaluations**: Conduct large-scale human evaluations using comprehension tests or direct ratings to assess simplification quality beyond automated metrics like BLEU and SARI, which may not fully capture readability or meaning preservation. Engage diverse user groups, including non-native speakers and individuals with reading difficulties, to validate the models' effectiveness for target audiences.
4. **Explore Hybrid and Advanced Approaches**: Develop hybrid systems combining neural models with rule-based methods for lexical substitution or sentence splitting, as suggested by prior research, to improve control over simplification processes. Investigate continued pre-training on simplification-specific corpora or novel pre-training objectives tailored for text simplification to enhance model robustness and adaptability.

These recommendations aim to advance the field of text simplification by improving model performance, readability, and practical applicability, ultimately making NLP tools more accessible and effective for diverse user groups.

4. REFERENCES

1. Lexical simplification system to improve web accessibility. (2021). IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/document/9400837
2. Alarcon, R., Moreno, L., & Martínez, P. (2023). EASIER corpus: A lexical simplification resource for people with cognitive impairments. PLoS ONE, 18(4), e0283622. https://doi.org/10.1371/journal.pone.0283622
3. Paraguassu, L., Zilio, L., Hercules, L., & Finatto, M. (2020). MedSimples: An Automated Simplification Tool for Promoting Health Literacy in Brazil. **, 76-78.
4. Jin, J., Sun, Y., & Li, A. (2024). A Smart Mobile Platform to Assist with Reading Comprehension using Machine Learning and Lexical Simplification. Computer Science, Engineering and Information Technology. https://doi.org/10.5121/csit.2024.141412
5. Rets, I., & Rogaten, J. (2020). To simplify or not? Facilitating English L2 users' comprehension and processing of open educational resources in English using text simplification. J. Comput. Assist. Learn., 37, 705-717. https://doi.org/10.1111/jcal.12517/v2/review2
6. Javourey-Drevet, L., Dufau, S., François, T., Gala, N., Ginestié, J., & Ziegler, J. (2022). Simplification of literary and scientific texts to improve reading fluency and comprehension in beginning readers of French. Applied Psycholinguistics, 43, 485 - 512. https://doi.org/10.1017/S014271642100062X
7. Rahman, M., Irbaz, M., North, K., Williams, M., Zampieri, M., & Lybarger, K. (2024). Health Text Simplification: An Annotated Corpus for Digestive Cancer Education and Novel Strategies for Reinforcement Learning. Journal of biomedical informatics, 104727. https://doi.org/10.1016/j.jbi.2024.104727
8. Alarcón, R., Moreno, L., & Martínez, P. (2023). EASIER corpus: A lexical simplification resource for people with cognitive impairments. PLOS ONE, 18. https://doi.org/10.1371/journal.pone.0283622
9. Leroy, G., Endicott, J., Kauchak, D., Mouradi, O., & Just, M. (2013). User Evaluation of the Effects of a Text Simplification Algorithm Using Term Familiarity on Perception, Understanding, Learning, and Information Retention. Journal of Medical Internet Research, 15. https://doi.org/10.2196/jmir.2569

10. Tregubov, A., & Blythe, J. (2020). Optimization of Large-Scale Agent-Based Simulations Through Automated Abstraction and Simplification. **, 81-93. https://doi.org/10.1007/978-3-030-66888-4_7
11. Ranathunga, S., Sirithunga, R., Rathnayake, H., De Silva, L., Aluthwala, T., Peramuna, S., & Shekhar, R. (2024). SiTSE: Sinhala Text Simplification Dataset and Evaluation. ArXiv, abs/2412.01293. https://doi.org/10.48550/arXiv.2412.01293
12. Garbacea, C., & Mei, Q. (2022). Adapting Pre-trained Language Models to Low-Resource Text Simplification: The Path Matters. **, 1103-1119.
13. Zee, D. (2019). Model simplification in manufacturing simulation - Review and framework. Comput. Ind. Eng., 127, 1056-1067. https://doi.org/10.1016/J.CIE.2018.11.038
14. Battini, F., Pernigotto, G., & Gasparella, A. (2023). Evaluating the capabilities of a simplification algorithm for Urban Building Energy Modeling in performing building-level Multi-Objective Optimizations at district scale. Journal of Physics: Conference Series, 2600. https://doi.org/10.1088/1742-6596/2600/8/082015
15. Asif, M., Inam, A., Adamowski, J., Shoaib, M., Tariq, H., Ahmad, S., Alizadeh, M., & Nazeer, A. (2023). Development of methods for the simplification of complex group built causal loop diagrams: A case study of the Rechna doab. Ecological Modelling. https://doi.org/10.1016/j.ecolmodel.2022.110192
16. Gosses, M., & Wöhling, T. (2019). Simplification error analysis for groundwater predictions with reduced order models. Advances in Water Resources. https://doi.org/10.1016/J.ADVWATRES.2019.01.006
17. Kim, S., & Saggion, H. (2023). Multilingual Controllable Transformer-Based Lexical Simplification. Proces. del Leng. Natural, 71, 109-123. https://doi.org/10.48550/arXiv.2307.02120
18. Alarcón, R., Moreno, L., & Martínez, P. (2021). Lexical Simplification System to Improve Web Accessibility. IEEE Access, 9, 58755-58767. https://doi.org/10.1109/ACCESS.2021.3072697
19. Sukiman, S., Husin, N., Hamdan, H., Azrifah, M., & Murad, A. (2023). A Hybrid Personalized Text Simplification Framework Leveraging the Deep Learning-based

Transformer Model for Dyslexic Students. Journal of Advanced Research in Applied Sciences and Engineering Technology. https://doi.org/10.37934/araset.34.1.299313

20. Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., & Raffel, C. (2020). mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. **, 483-498. https://doi.org/10.18653/V1/2021.NAACL-MAIN.41
21. Maddela, M., Alva-Manchego, F., & Xu, W. (2020). Controllable Text Simplification with Explicit Paraphrasing. **, 3536-3553. https://doi.org/10.18653/V1/2021.NAACL-MAIN.277
22. Xiao, Z., Gong, J., Wang, S., & Song, W. (2024). Optimizing Chinese Lexical Simplification Across Word Types: A Hybrid Approach. **, 15227-15239. https://doi.org/10.18653/v1/2024.emnlp-main.849
23. Crudu, A., Debussche, A., & Radulescu, O. (2009). Hybrid stochastic simplifications for multiscale gene networks. BMC Systems Biology, 3, 89 - 89. https://doi.org/10.1186/1752-0509-3-89
24. Zhang, C., Zhou, H., Chen, B., Peng, Y., & Duan, J. (2023). Hybrid simplification algorithm for unorganized point cloud based on two-level fuzzy decision making. Optik. https://doi.org/10.1016/j.ijleo.2023.170642
25. Feng, Q., Zhou, X., & Li, J. (2019). A hybrid and automated approach to adapt geometry model for CAD/CAE integration. Engineering with Computers, 36, 543 - 563. https://doi.org/10.1007/s00366-019-00713-4

7. APPENDICES

Appendix A: Training Log in Google Colab

Appendix B: Loading T5-small and BART-base from hugging face

```
            if model_type == 'bart':
                self.tokenizer = BartTokenizer.from_pretrained(model_name)
                self.model =
BartForConditionalGeneration.from_pretrained(model_name).to(self.device)
            elif model_type == 't5':
                self.tokenizer = T5Tokenizer.from_pretrained(model_name)
                self.model =
T5ForConditionalGeneration.from_pretrained(model_name).to(self.device)
            else:
                raise ValueError(f"Unsupported model_type: {model_type}. Choose
'bart' or 't5'.")
            print(f"Model loaded on {self.device}")
            print(f"Model parameters: {sum(p.numel() for p in
self.model.parameters()):,}")
            if self.device.type == 'cuda':
```

```
                print(f"GPU memory allocated:
{torch.cuda.memory_allocated(self.device)/1e9:.2f} GB")
        except Exception as e:
            print(f"Error initializing model: {e}")
            raise
```