University of Moratuwa

Department of Computer Science and Engineering

Improving mT5 for OPUS-100 Machine Translation through Multilingual Denoising and Domain Adaptation

H. D. E. Maduranga

Index No: 210352R

Project Id: NLP014

Progress Evaluation Report

August 26, 2025

Contents

List of Figures

1 Introduction and Background

1.1 Multilingual Machine Translation

Machine Translation (MT) aims to automatically translate text from one natural language to another. Traditional MT relied on **rule-based** and **statistical methods** earlier stages , but recent advances in **neural machine translation (NMT)** have established large **encoder–decoder Transformer architectures** [1] as the de factor standard.

Multilinguality becomes important in machine translation because of a variety of factors. First, it facilitates cross-lingual transfer, whereby what is acquired from high-resource languages can improve the quality of corresponding low-resource language translations, allowing the model to generalize adequately in the presence of limited data. Multilingual models also render training and deployment affordable, given that a single model can replace multiple language-specific models, reducing maintenance and resource requirements. They also facilitate consistency across languages, particularly when used for multilingual text in global applications. In addition, multilingual models allow for zero-shot and few-shot translation, which offers translation between language pairs with little or no parallel data using indirect transfer through shared representations. Several multilingual models demonstrate the advantages of using multilingual models: **mBART (Multilingual BART)**, a sequence-to-sequence denoising autoencoder that is trained on massive multilingual data for generation and translation tasks; **mT5 (Multilingual T5)**, an extension of the T5 model that was trained on multi-language text-to-text tasks; and **M2M-100**, a Facebook AI massively multilingual NMT model that enables direct translation between 100 languages without pivoting through English.

While single-language NMT systems achieve strong results in **high-resource languages (HRLs)**, they often fail in **low-resource languages (LRLs)** due to data scarcity. Multilingual models attempt to address this by sharing parameters across languages, enabling **cross-lingual transfer**.

1.2 The T5 and mT5 Models

The **T5 (Text-to-Text Transfer Transformer) model** [2] by google introduced a unified framework for natural language processing by reformulating every task in a **text-to-text format**. This means that problems such as translation, summarization, classification, or question answering are all expressed as converting one text string into another. A major contribution of T5 is its **span-corruption pretraining** objective, where spans of text are masked and replaced with sentinel tokens, enabling the model to capture both local and global dependencies more effectively than token-level masking. T5 was pretrained on the **Colossal Clean Crawled Corpus (C4)**, a large-scale and diverse dataset derived from Common Crawl, ensuring broad coverage of language patterns. Furthermore, the model was released in different sizes, ranging from **T5-Small to T5-11B**, and demonstrated consistent improvements in performance with increased scale. Its text-to-text formulation also simplified fine-tuning across downstream tasks, eliminating the need for task-specific architectures and making T5 a versatile and general-purpose NLP framework.

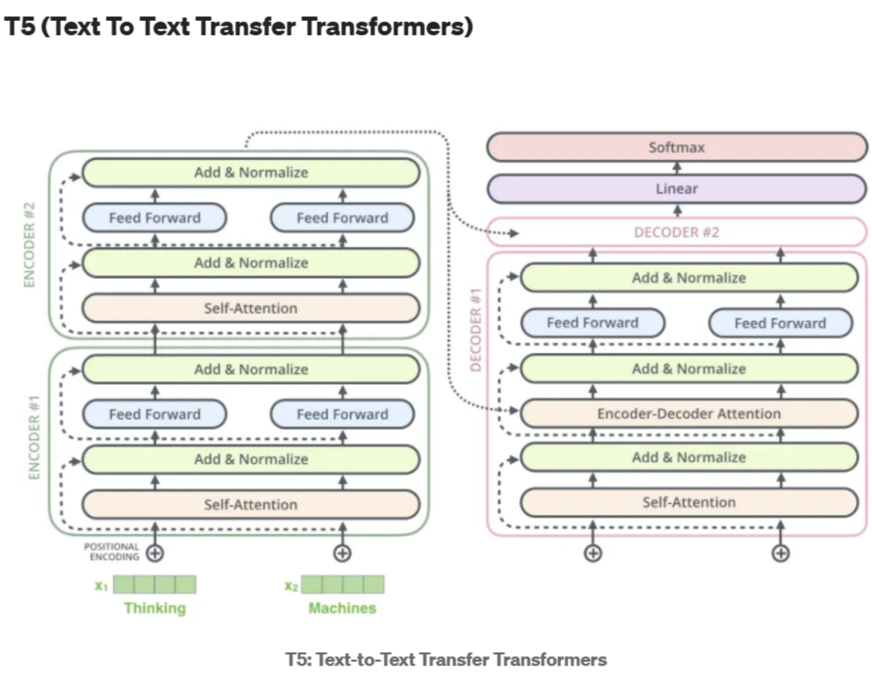

Figure 1: T5 Architecture.

The **mT5 (multilingual T5)** [3] extends T5 to **101 languages**, trained on the **mC4 corpus** (a multilingual variant of Common Crawl) pecifically designed to support a wide spectrum of languages. mT5 provides a strong foundation for multilingual NLP, supporting both **zero-shot** and **many-to-many translation**.

Key features of mT5:

- **Architecture:** same as T5 (encoder–decoder Transformer).
- **Pretraining objective:** multilingual denoising autoencoding (masking spans of text and reconstructing them).
- **Strength:** supports translation across 100+ language pairs without language-specific components ften surpassing existing multilingual baselines on benchmarks like XTREME.
- **Weakness:** performance drops in low-resource pairs, domain mismatch issues (web text vs. specific domains).

1.3 OPUS-100 Dataset

The **OPUS-100 dataset** [4] is a curated multilingual translation benchmark built from OPUS resources. It covers **100 language pairs** with **1M sentence pairs per direction**. The dataset includes predefined training, validation, and test splits, making it suitable for benchmarking multilingual MT.

Advantages:

- Wide linguistic diversity (including low-resource languages).
- Supports many-to-one, one-to-many, and many-to-many settings.
- Balanced sampling, making it more stable than ad-hoc parallel corpora.

Primary use cases:

- Neural Machine Translation (NMT) training and evaluation
- Cross-lingual transfer learning
- Multilingual pre-training for models like `mT5` [3], `mBART` [6], or `mBERT` [7]

1.4 Multilingual Denoising Pre-training

Denoising pre-training is central to T5/mT5. Sentences are corrupted by masking random spans (15% of tokens) and replaced with special <extra_id> tokens. The model learns to reconstruct the original sequence.

Objective: The model learns to denoise by predicting the original text from the corrupted input.

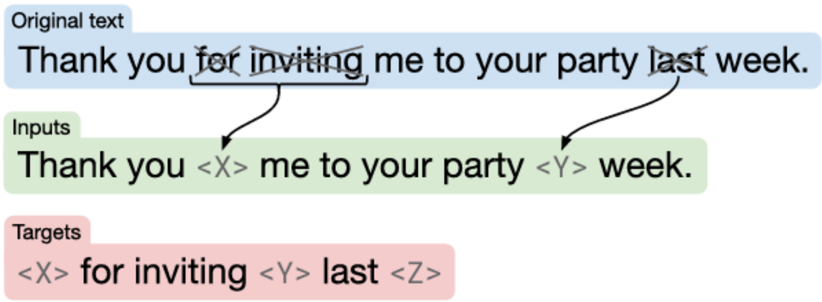

Figure 2: A Corrupted Span Reconstruction Task

In the **multilingual** setting, this helps models:

- Learn shared representations across languages.
- Handle code-switching or mixed-lingual input.
- Transfer knowledge from high-resource to low-resource languages.

For **domain adaptation**, we can use **continued pretraining** [5]:

- **DAPT (Domain-Adaptive Pre-Training):**. In this approach, a pretrained language model is exposed to additional training on domain-specific corpora, allowing it to better capture the vocabulary, style, and semantic nuances of the target domain.
  A common method is **DAPT (Domain-Adaptive Pre-Training)**, where the model is further pretrained on large collections of domain-specific monolingual texts using the same denoising or masked language modeling objective as in the original pre-training. This helps the model adapt its internal representations to domain-relevant patterns, thereby improving performance on downstream tasks within that domain.

2 Motivation

Although mT5 performs well across many multilingual tasks, it faces challenges with low-resource languages (LRLs) and domain-specific translation.

2.1 Domain Mismatch

mT5 is pre-trained on generic web data (mC4), which often differs from real-world domains such as news, medical, or legal text. Domain mismatch can lead to mistranslations, semantic drift, or errors in specialized contexts.

2.2 Role of Multilingual Denoising Pre-training

Pre-training mT5 with span-corruption denoising on OPUS-100 addresses these limitations by:

- Strengthening representations for LRLs to improve translation with limited parallel data.
- Enhancing domain robustness through exposure to diverse multilingual monolingual data.
- Exploiting abundant monolingual data to boost downstream performance without relying solely on parallel corpora.

2.3 Research Relevance

This approach aims to quantify pre-training benefits, analyze hyperparameter effects, and understand how model capacity influences cross-lingual transfer and domain adaptation, contributing to more robust multilingual models.

3 Methodology Outline

3.1 Primary objective

Train an **mT5 model with a multilingual span-corruption (denoising) pre-training objective on OPUS-100** to obtain stronger cross-lingual representations . And evaluate using suitable metrices.
As part of the evaluation, the **unsupervised reconstruction quality** will be measured through validation loss and **perplexity**, which can then be related to improvements in downstream tasks. Furthermore, the study will analyze the model's behaviour on low-resource languages such as Sinhala to better understand the effectiveness of cross-lingual transfer and to highlight the potential gains for languages with limited training data.

3.2 Experimental Workflow

The proposed methodology pipeline is structured as follows :

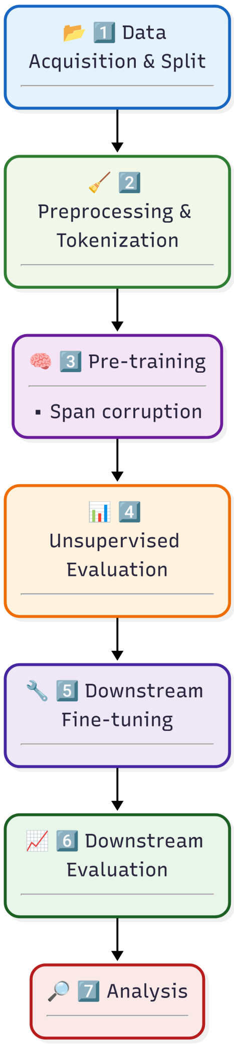

Figure 3: Experimental workflow.

1. **Data acquisition and split:** Obtain the OPUS-100 dataset, create monolingual corpora for pre-training, and prepare parallel sentence pairs for fine-tuning and evaluation.
2. **Preprocessing and tokenization:** Apply language-agnostic tokenization using the mT5 tokenizer along with standard data cleaning steps.
3. **Pre-training:** Train the mT5 model with a span-corruption denoising objective (T5 objective), employing `DataCollatorForT5MLM`.
4. **Unsupervised evaluation:** Measure validation loss and perplexity on held-out monolingual corpora to estimate reconstruction quality.
5. **Downstream fine-tuning:** Fine-tune the pre-trained checkpoint on bilingual translation pairs (e.g., English $\rightarrow$ Sinhala).
6. **Downstream evaluation:** Evaluate translation quality using established metrics such as

BLEU, METEOR, and chrF, and perform statistical significance tests against baseline models.

7. **Analysis:** Conduct controlled ablation studies by varying hyperparameters, model sizes, and data regimes to analyze their effects.

3.3 Dataset and Preprocessing Pipeline

The primary dataset for this study is **Helsinki-NLP/OPUS-100**, a parallel corpus in around 100 languages. Both the parallel sentence pairs (for evaluation and fine-tuning) and the monolingual sides of the corpus (for denoising pre-training) are utilized to support the different stages of the experimental pipeline.For pre-training, a shuffled monolingual corpus is created by extracting all language sides. little data set part set aside as a pre-training validation set for **monitoring loss and perplexity** during training.

The preprocessing stage applies **Unicode normalization, removes control characters and excessive whitespace, and discards empty or very short sentences**. To control sequence length, sentences are optionally filtered to **5–512 tokens**. Finally, all text is tokenized with the mT5 SentencePiece tokenizer (using the fast implementation for efficiency), ensuring a clean and consistent multilingual corpus for pre-training and fine-tuning.

This pipline constructs the **monolingual pre-training corpus** by decomposing each OPUS-100 **parallel sentence pair** into separate monolingual examples. This transformation yields a large multilingual corpus for the **denoising pre-training objective**, exposing the model to diverse linguistic structures and enhancing its ability to learn robust cross-lingual representations.

3.4 Model Architecture and Pre-training Setup

The `google/mt5-small` model was selected for development and controlled experiments because it offers *lower memory requirements*, making it suitable for *training on limited GPU resources*. Its relatively smaller size also enables faster training and iteration cycles. Moreover, due to the feasibility of computer resources, this model can be executed in resource-constrained environments such as Kaggle notebooks, where GPU memory and runtime limitations restrict the use of larger-scale models for pre-training.

3.4.1 Pre-training Objective

The pre-training task is based on a **denoising autoencoder objective**, following the *span-corruption mechanism* used in the T5 and mT5 models. In this approach, contiguous spans of tokens in the input sequence are **masked and replaced** with `sentinel tokens` such as <extra_id_0> and <extra_id_1>. The decoder is then trained to reconstruct these missing spans by generating them in sequence as the target output. This method enables the model to handle **longer contexts** and learn dependencies across segments of text rather than focusing solely on individual token prediction.

The implementation of this process is facilitated by the `DataCollatorForT5MLM`, which automates **sentinel token insertion**, **span masking**, and **decoder label construction**. This collator ensures consistent preprocessing and batching, thereby reducing implementation complexity. A key advantage of this span-corruption denoising strategy is that it enables the model to learn **robust cross-lingual representations**, which is particularly important in multilingual settings. By forcing the model to predict meaningful spans across multiple languages, the approach encourages **transfer learning** and improves **generalization** across both high-resource and low-resource languages.

Several of the **hyperparameters** play critical roles in determining the effectiveness of pre-training:

- **Noise density:** Determines the proportion of tokens that are replaced and masked, controlling how much input context the model must reconstruct.
- **Mean span length:** Specifies the average number of consecutive tokens in each masked span, balancing between short and long sequences so the model learns both local and global dependencies.
- **Target and maximum input lengths:** Restrict the amount of data processed in a single instance, preventing memory overflow during training.

For optimization, the following strategies are employed:

- **Optimizer:** Training is performed using the `Adafactor` optimizer, a memory-efficient algorithm designed for the T5 family of models.
- **Mixed precision training:** Applied where applicable to reduce computational overhead while preserving numerical stability.
- **Batch size and gradient accumulation:** The batch size is tuned based on available GPU memory, with gradient accumulation strategies used to simulate larger effective batch sizes without exceeding hardware limits.

The training process employs the `Adafactor` optimizer with a short warmup phase for stability, limited epochs for development, and extended runs for large-scale pre-training. Regular checkpointing, logging, and early stopping based on validation loss ensure efficient monitoring and prevent overfitting, balancing resource feasibility with scalability.

3.4.2 Unsupervised Evaluation (Post-Pretraining)

The quality of pre-training is assessed using two primary metrics. First, the **validation loss (cross-entropy)** is computed on held-out monolingual data to measure reconstruction accuracy. Second, **perplexity (PPL)** is reported, defined as:

$$\boxed{\text{PPL} = \exp(\text{loss})}$$

where lower values indicate better predictive performance. Perplexity is measured both on a per-language basis and in an aggregated form across all languages.

3.5 Possible Extensions

Although the initial phase of this project will focus primarily on multilingual denoising pre-training and early evaluation, there are several downstream fine-tuning and evaluation enhancements that can be considered if sufficient time and resources are available. These include:

Downstream Fine-Tuning: Machine Translation
A potential extension involves fine-tuning the pre-trained mT5-small model on parallel translation pairs, specifically targeting Low Resource Language translation. This would enable a direct assessment of the effectiveness of denoising pre-training for low-resource machine translation.

To systematically evaluate the benefits of pre-training, the following baselines could be established:

- **Baseline A (Scratch):** `mT5-small` architecture initialized randomly and trained directly on parallel Low Resource Language data, without denoising pre-training.
- **Baseline B (Pretrained):** The Hugging Face `google/mt5-small` checkpoint, which has been pre-trained on large-scale multilingual corpora, fine-tuned for Low Resource Languag translation.

The evaluation involves comparing these setups: a custom pre-trained mT5-small model, a randomly initialized baseline trained on Low Resource Language data, and a pre-trained google/mt5-small fine-tuned checkpoint. Their performance would be assessed using complementary automatic metrics: For evaluation, **BLEU** [8] is commonly used for standardized corpus-level comparison, while **BERTScore** [9] and **METEOR** [10] are employed to capture semantic adequacy and alignment with human judgments. Together, these baselines and metrics provide a robust framework for measuring the **impact of denoising pre-training** .

4 Project Planning

4.1 Key Deliverables

1. **Baseline Model:** Multilingual denoising pre-train the standard **mT5 model** on the **OPUS-100 dataset** and establish benchmark performance using evaluation metrics.
2. **Data Preprocessing Pipeline:** Preprocessed and tokenized OPUS-100 corpus for both monolingual pre-training and parallel fine-tuning.
3. **Fine-tuned Translation Models (Optional):** For low-resource language translation, we employ a fine-tuned model and compare its performance against two baseline models. **Baseline A** is a randomly initialized `mT5-small` model trained from scratch on the EN–SI parallel corpus. **Baseline B** is the off-the-shelf `google/mt5-small` checkpoint, which has already been pre-trained on large-scale multilingual corpora and is subsequently fine-tuned for Low Resource Language translation.
4. **Documentation** prepare comprehensive final research paper covering objectives, methodology, pre-training and fine-tuning experiments, evaluation results, analysis, and conclusions, adhering to formal academic standards . In addition, a short paper for mid-evaluation report will summarize the project concept, preliminary implementation, early results, and technical validation, serving as a checkpoint for mid-term evaluation.

4.2 Resources and Tools

- **Kaggle GPU resources** for model training.
- **Hugging Face Transformers** and **PEFT libraries** for fine-tuning and pre-training.

- **SentencePiece tokenizer** for multilingual subword tokenization.
- **DataCollatorForT5MLM** for handling span-corruption denoising pre-training (automatic insertion of sentinel tokens and label construction).
- **Evaluation tools:** SacreBLEU, chrF++ , NLTK .
- **Datasets: OPUS-100** multilingual parallel corpus for DAPT.
- **Git/GitHub** for version control and reproducibility.

4.3 Risk Management

- **Compute limitations on Kaggle:** Mitigated using PEFT and smaller mT5 variants (mT5-base, mT5-small).
- **Data preprocessing challenges:** Addressed by relying on existing OPUS-100 splits and the Hugging Face datasets library.
- **Evaluation bottlenecks:** Resolved by automating BLEU/BERTScore computation with SacreBLEU for consistency.

5 Project Timeline

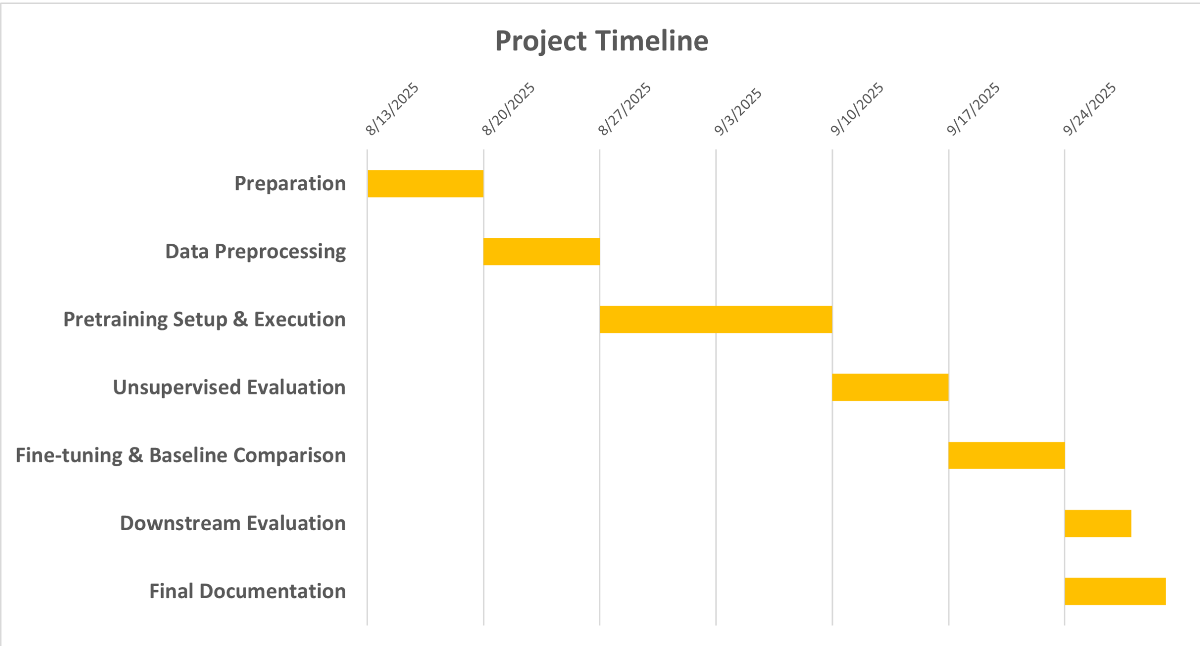

Figure 4: Project Timeline.

6 Conclusion

This project set out to explore the effectiveness of multilingual denoising pre-training for enhancing translation performance, particularly in low-resource languages. By leveraging the OPUS-100 dataset and the mT5 model, the proposed work aims to strengthen cross-lingual representations, reduce the impact of domain mismatch, and improve translation quality where parallel data is scarce. The methodology combines robust preprocessing, controlled experiments with resource-friendly model variants, and evaluation through established metrics such as BLEU, METEOR, and perplexity.

The expected outcome is a more comprehensive understanding of how span-corruption pre-training contributes to cross-lingual transfer, as well as practical insights into optimizing model performance for domain-specific translation. Ultimately, the findings of this project are intended to inform future research on scalable, efficient, and accurate multilingual machine translation systems that are capable of supporting both high-resource and low-resource language communities.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*.

[2] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140), 1-67.

[3] Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., ... & Raffel, C. (2021). mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of NAACL*.

[4] Zhang, B., Williams, P., Titov, I., & Sennrich, R. (2020). Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of ACL*.

[5] Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., & Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of ACL*.

[6] Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., & Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8, 726–742.

[7] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4171–4186.

[8] Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318.

[9] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations (ICLR)*.

[10] Banerjee, S., & Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72.