Progress Report

CS4681 - Advanced Machine Learning

Name : Kumarasekara G.K.
Student Number : 210314E
Project Code : NLP009

1. Project Overview

1.1 Background

Text summarisation is a fundamental task in natural language processing. It aims to condense lengthy documents into concise summaries while preserving key information. Transformer based models [1]–[3], have demonstrated strong performance in abstractive summarisation, largely due to large scale pretraining and encoder–decoder architectures. A notable example is the PEGASUS [4] model, which leverages a self-supervised objective called "Gap Sentences Generation" to pretrain on massive text corpora, showing impressive performance on various summarization tasks. However, its effectiveness is limited by the quadratic complexity of the attention mechanism, which makes it computationally expensive and less effective for very long documents.

To address this, the PEGASUS-X [5] model was introduced as an extension of PEGASUS, specifically designed to handle long input summarization tasks. PEGASUS-X incorporates efficient attention mechanisms and additional pretraining on long inputs, enabling the model to process up to 16K tokens efficiently. PEGASUS-X has achieved state-of-the-art results on several long document summarisation benchmarks, such as arXiv [6], PubMed [6], and GovReport [7].

1.2 Problem Statement

While PEGASUS-X gives state-of-the-art results for abstractive summarisation, its architecture can be optimised further to improve summarisation of longer documents. The potential of architectural modifications, such as introducing new layers or alternative activation functions, remains underexplored. Thus, there is an opportunity to investigate whether lightweight modifications to PEGASUS-X can further improve its efficiency and effectiveness without dramatically increasing model size or computational cost. This research therefore aims to explore targeted modifications to enhance the ability of PEGASUS-X to generate more accurate summaries for longer texts.

1.3 Objectives

The main objective of this project is to explore methods to improve the PEGASUS-X model for summarisation tasks. It can be further broken down into the following goals.

- Experimenting with architectural modifications such as adding layers and changing activation functions
- Evaluating the effect of these changes on summarisation quality measured by standard metrics
- Comparing the modified model against the baseline PEGASUS-X to identify improvements in long document summarisation
- Investigating whether improvements generalise across both long and short summarisation task

2. Literature Review

2.1 Efficient Attention Mechanisms for Long Inputs

Long document summarisation relies on attention mechanisms that scale beyond the quadratic cost of vanilla Transformers. One approach is block-local attention with a few global tokens. For example, Beltagy et al. [3] have used a fixed-size window approach around each token and a small set of global tokens that attend to all positions. This achieves linear memory cost in the input length and retains global information through the special tokens. They also introduce an encoder–decoder variant for long input summarisation, which effectively handles documents up to several thousand tokens.

In contrast, sparse random attention which is used in BigBird [8] combines local windows with randomly chosen tokens and a few global tokens. This pattern preserves expressivity while reducing attention to linear complexity. Experiments have shown that a BigBird encoder with PEGASUS decoder can perform far better performance than a vanilla transformer. LongT5 [9] uses a transient global attention scheme. It injects global interactions without extra inputs by allocating certain query/key positions to attend globally, mimicking local/global patterns. This allows scaling to very long inputs with only modest performance loss.

2.2 Hierarchical and Multi-stage Summarization

Hierarchical transformers explicitly model multi-level document structure. For example, Rohde et al. [10] have designed a hierarchical attention transformer (HAT) with both sentence-level and document-level layers. It first encodes individual sentences and attends among sentence representations to form the summary. Similarly, Zhu et al. [11] propose HMNet for meeting transcripts. There, one transformer processes each turn, and a second transformer runs over the sequence of turn representations. This two-level encoder captures both local utterance context and global meeting context. These hierarchical models often use cross-domain pretraining before fine tuning on meetings, further boosting performance.

A related strategy is multi-stage summarization. Zhang et al. [12] introduce SUMM^N, a split-and-merge pipeline. They first divide the input into segments, generate an intermediate summary for each segment, then concatenate those and run a final summariser to produce the overall summary. This multi-stage approach can handle arbitrarily long input by adding more segments.

2.3 Fine Tuning Strategies

Karotia and Susan [13] have tackled lay language summarisation by first fine tuning general models on domain specific data, and then feeding their outputs into a specialized domain specific model for further fine tuning. They have found that combining general and domain specific summarizers improves performance across relevance, readability and factuality metrics. More broadly, one can first fine tune on a large generic summarization corpus, then on smaller domain specific data. Such multi-stage fine tuning often boosts ROUGE and content accuracy in the target domain.

Extract–then–summarize pipelines also fall under this category. Many works first identify salient sentences from the long input and then summarise only those. This is a form of task decomposition and indirectly a fine tuning strategy. Compared to end-to-end approaches, pipelines can be more interpretable but might miss context. However, Summ^N [12] pipeline exceeds end-to-end baselines on very long inputs.

2.4 Architectural Modifications

BART-LS [14] incorporates pooling layers between blocks to compress local representations before higher-level attention. Such layers can reduce sequence length and inject inductive bias for summarisation. These modifications are complementary to global tokens and long context pretraining. For example, Phang et al. [5] report that adding global tokens, staggered blocks, and long sequence pretraining together substantially improves performance over a baseline PEGASUS.

Most long summarisation models retain standard transformer activations and differ mainly in attention or depth. Some very large models use gated activations to increase capacity without changing the overall architecture, but these are not yet common in summarization specific literature. Instead, adding layers has been more impactful. For instance, HMNet's [11] second turn-level encoder and HAT's [10] sentence-level layers are explicit architectural additions.

3. Methodology

3.1 Baseline Model - PEGASUS-X

PEGASUS-X [5] is an extension of the original PEGASUS [4] summarisation model, which was designed to handle long input sequences of up to 16,384 tokens while remaining efficient in terms of memory and computation. The model is based on an encoder–decoder architecture, but introduces several key modifications.

- **Efficient attention mechanism -** The encoder uses a block-local attention mechanism, where tokens are divided into fixed blocks and attend only within their block. To overcome the limitation of isolated blocks, staggered blocks are introduced so that boundaries shift across layers, allowing information to flow across blocks with minimal cost. In addition, global tokens are added. These are special learnable embeddings that can attend to, and be attended by, all tokens, enabling the model to capture global context efficiently.
- **Architecture adjustments -** The baseline PEGASUS-X introduces very few new parameters compared to PEGASUS which mainly include the global token embeddings and additional LayerNorm layers. Input context length is extended from 512 tokens for the standard PEGASUS model to 16K tokens during fine tuning.
- **Pretraining and fine tuning strategy -** Similar to PEGASUS, the model is pretrained on short sequences of 512 tokens with masked sentence prediction. But PEGASUS-X adds a stage of pretraining with longer inputs of 4096 tokens for 300K steps, which adapts the model for long document tasks. For downstream tasks such as arXiv and GovReport, the model is fine tuned with input lengths of up to 16K tokens.

On long document summarisation benchmarks, PEGASUS-X has achieved state-of-the-art results, outperforming much larger models like LongT5 [9] in some cases, while only slightly regressing on short input tasks.

3.2 Experimental Setup

Experiments will run on cloud GPUs available via Kaggle Notebooks and Google Colab. Main libraries used will be `PyTorch` and `transformers`. The baseline model will be loaded from the Hugging Face `transformers` library. Since the link in the GitHub repository to the tokenizer is broken, both the model and tokenizer will be sourced directly from the same Hugging Face model repository. This avoids external URL dependencies and ensures the tokenizer configuration matches the checkpoint.

3.3 Planned Modifications

The goal of this project is to explore lightweight architectural and training modifications to the PEGASUS-X baseline, focusing on efficiency and stability under longer document summarisation. Unlike large scale pretraining from scratch, the modifications will be applied during fine tuning on long document datasets.

- **Activation function substitution -** Standard PEGASUS-X uses the GeLU [15] activation in feed-forward layers. Recent work suggests that alternative activations such as Swish [16] or SwiGLU [17] improve training stability and gradient flow, especially in deeper networks. Therefore, it is planned to replace GeLU with Swish and SwiGLU in the feed-forward sublayers of the Transformer.
- **Additional intermediate layers -** Adding lightweight bottleneck or projection layers can increase representational power without significant parameter growth. Therefore, it is planned to insert an additional LayerNorm and linear projection between the attention and feed-forward blocks of selected encoder layers.
- **Training optimisation adjustments -** Gradient checkpointing will be enabled for memory savings. Different learning rate schedules will be experimented. Gradient accumulation will be applied to simulate larger batches.

3.4 Evaluation

The performance of PEGASUS-X and its modified variants will be evaluated using widely adopted automatic metrics for summarisation.

- **ROUGE-1** [18] - Measures unigram overlap between system and reference summaries.
- **ROUGE-2** [18] - Measures bigram overlap, capturing short phrase similarity.
- **ROUGE-L** [18] - Measures the longest common subsequence, reflecting fluency and structural similarity.
- **BLEU** [19] - Evaluates precision-oriented n-gram overlap, penalising extraneous content.

- **BERTScore** [20] - Uses contextual embeddings to compute semantic similarity, capturing paraphrasing beyond surface n-grams.

Following PEGASUS-X, ROUGE F1 scores and the geometric mean of ROUGE scores will be mainly reported. BLEU and BERTScore will be included as complementary measures where feasible.

3.5 Datasets

The datasets used for this project are summarised in Table 1. Long document benchmarks are the primary focus, while short document corpora are included for comparison to ensure modifications do not degrade performance on shorter inputs.

Dataset	**Domain**	**Avg. Input Length**
arXiv [6]	Scientific articles	∼6,900 tokens
PubMed [6]	Biomedical research	∼4,700 tokens
GovReport [7]	Government reports	∼8,000 tokens
BigPatent [21]	Patent documents	up to 10,000 tokens
CNN/DailyMail [22]	News articles	∼700 tokens
XSum [23]	News articles	∼430 tokens

Table 1: Datasets considered for fine tuning and evaluation of PEGASUS-X modifications. Long document datasets are the primary focus. Short document datasets will be used for comparison.

3.6 Expected Outcomes

The project is expected to yield the following outcomes.

- **Improved summarisation quality -** The use of alternative activation functions and architectural adjustments are anticipated to improve ROUGE, BLEU, and BERTScore metrics compared to the baseline PEGASUS-X model.
- **Better training stability -** Modifications such as gradient checkpointing, tuned learning rates, and gradient accumulation are expected to reduce instability during long input fine tuning.
- **Enhanced efficiency tradeoffs -** Through experiments with different maximum input lengths, the project will provide insights into the balance between input length, memory usage, and summarisation quality on limited hardware.
- **Empirical comparison of activations -** The evaluation will clarify whether GeLU, Swish, or SwiGLU yields better performance for long document summarisation tasks, contributing to understanding activation choice in transformer models.

- **Reproducible implementation -** All experiments will be conducted using the Hugging Face `transformers` and `datasets` libraries, ensuring reproducibility and accessibility for further research.

3.7 Limitations

Despite careful planning, this project is subject to several limitations.

- **Computational constraints -** Experiments will be limited to GPUs available on Google Colab and Kaggle. This may restrict the ability to train with the maximum supported input length.
- **Time constraints** - The project must be completed within a fixed seven week period. This restricts the number of experiments that can be conducted, limiting the scope of hyperparameter tuning and dataset coverage.
- **No large scale pretraining -** The project does not include full scale pretraining from scratch, which means improvements are restricted to fine tuning modifications applied on top of the pretrained PEGASUS-X model.
- **Restricted dataset coverage -** Due to time and resource constraints, only a subset of long document datasets will be used extensively. Very large datasets such as BigPatent may be excluded or only partially utilised.

4. Project Timeline

The project spans from 14 August 2025 to 5 October 2025. It is structured into three main phases.

1. Preparation (14 - 28 August)
    - Literature Review (14 - 24 Aug) - Reviewing PEGASUS-X and related long document summarisation models.
    - Setting Up the Environment (15 - 18 Aug) - Configuring the development environment and GPU.
    - Understanding the Codebase (18 - 28 Aug) - Exploring PEGASUS-X implementation for later modification.
2. Implementation and Testing (26 Aug - 29 Sep)
    - Testing Baseline PEGASUS-X Model on Datasets (26 - 31 Aug).

- Modifying PEGASUS-X and Training It (28 Aug - 26 Sep) - Introducing architectural changes such as new activation functions.
- Evaluation and Analysis (1 - 29 Sep) - Measuring performance using standard metrics and analysing tradeoffs.

3. Documentation (22 Aug - 5 Oct)

- Creating the Progress Report (22 - 26 Aug).
- Creating the Short Paper (16 - 21 Sep).
- Creating the Final Paper (25 Sep - 2 Oct).
- Submitting the Final Report to a Conference (3 - 5 Oct).

The Gantt chart presented below provides a visual representation of the proposed project plan.

TASK NUMBER	TASK TITLE	START DATE	DUE DATE	DURATION (DAYS)
1	**Preperation**			
1.1	Literature Review	14/08/25	24/08/25	10
1.2	Setting Up the Environment	15/08/25	18/08/25	3
1.3	Understanding the Codebase	18/08/25	28/08/25	10
2	**Implementation and Testing**			
2.1	Testing Baseline Pegasus-X Model on Datasets	26/08/25	31/08/25	5
2.2	Modifying Pegasus-X and Training It	28/08/25	26/09/25	28
2.3	Evaluation and Analysis	01/09/25	29/09/25	28
3	**Documentation**			
3.1	Creating the Progress Report	22/08/25	26/08/25	4
3.2	Creating the Short Paper	16/09/25	21/09/25	5
3.3	Creating the Final Paper	25/09/25	02/10/25	7
3.4	Submitting the Final Report to a Conference	03/10/25	05/10/25	2

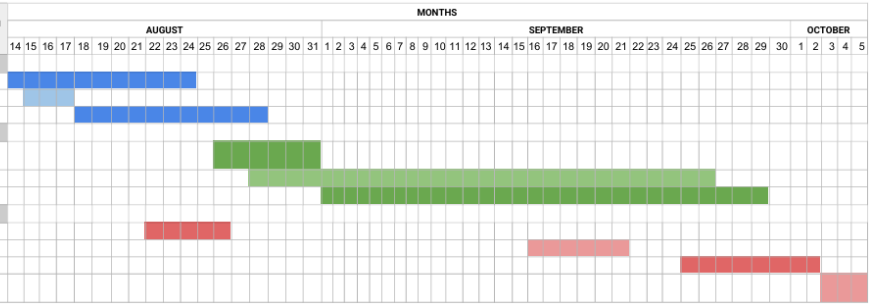

Figure 1: Project Timeline

5. Discussion

This project explores lightweight modifications to PEGASUS-X, focusing on activation functions and small architectural adjustments. The aim is to improve long document summarisation quality while remaining feasible on limited hardware. Compared to large scale approaches such as LongT5 or BigBird, this study prioritises practical efficiency over major architectural redesigns. The discussion therefore centers on the tradeoff between achievable accuracy and constrained resources.

Key limitations of this project will be time and computational resources, which restrict the number of experiments, dataset coverage, and the maximum input lengths that can be tested. Nevertheless, the experiments are expected to provide insights into how small changes influence performance and training stability. Overall, the project contributes towards understanding efficient adaptations of PEGASUS-X and highlights directions for future research on scalable long document summarisation.

References

[1] M. Lewis, Y. Liu, N. Goyal, *et al.*, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019. DOI: 10.48550/arXiv.1910.13461.

[2] Y. Liu, "Fine-tune BERT for extractive summarization," *arXiv preprint arXiv:1903.10318*, 2019. DOI: 10.48550/arXiv.1903.10318.

[3] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020. DOI: 10.48550/arXiv.2004.05150.

[4] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization," in *International conference on machine learning*, PMLR, 2020, pp. 11 328–11 339.

[5] J. Phang, Y. Zhao, and P. Liu, "Investigating efficiently extending transformers for long input summarization," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore: Association for Computational Linguistics, Dec. 2023, pp. 3946–3961. DOI: 10.18653/v1/2023.emnlp-main.240.

[6] A. Cohan, F. Dernoncourt, D. S. Kim, *et al.*, "A discourse-aware attention model for abstractive summarization of long documents," in *Proceedings of NAACL-HLT*, 2018, pp. 615–626. DOI: 10.18653/v1/N18-2097.

[7] L. Huang, D. Wu, P. Wang, *et al.*, "Efficient attentions for long document summarization," in *Findings of ACL*, 2021, pp. 1412–1426. DOI: 10.18653/v1/2021.naacl-main.112.

[8] M. Zaheer, G. Guruganesh, K. A. Dubey, *et al.*, "Big bird: Transformers for longer sequences," *Advances in neural information processing systems*, vol. 33, pp. 17 283–17 297, 2020. DOI: 10.48550/arXiv.2007.14062.

[9] M. Guo, J. Ainslie, D. Uthus, *et al.*, "LongT5: Efficient text-to-text transformer for long sequences," in *Findings of the Association for Computational Linguistics: NAACL 2022*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds., Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 724–736. DOI: 10.18653/v1/2022.findings-naacl.55.

[10] T. Rohde, X. Wu, and Y. Liu, "Hierarchical learning for generation with long source sequences," *arXiv preprint arXiv:2104.07545*, 2021. DOI: 10.48550/arXiv.2104.07545.

[11] C. Zhu, R. Xu, M. Zeng, and X. Huang, "A hierarchical network for abstractive meeting summarization with cross-domain pretraining," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 194–203. DOI: 10.18653/v1/2020.findings-emnlp.19.

[12] Y. Zhang, A. Ni, Z. Mao, *et al.*, "Summ^N: A multi-stage summarization framework for long input dialogues and documents," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds., Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 1592–1604. DOI: 10.18653/v1/2022.acl-long.112.

[13] A. Karotia and S. Susan, "BioLay_AK_SS at BioLaySumm: Domain adaptation by two-stage fine-tuning of large language models used for biomedical lay summary generation," in *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, D. Demner-Fushman, S. Ananiadou, M. Miwa, K. Roberts, and J. Tsujii, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 762–768. DOI: 10.18653/v1/2024.bionlp-1.69.

[14] W. Xiong, A. Gupta, S. Toshniwal, Y. Mehdad, and S. Yih, "Adapting pretrained text-to-text models for long text sequences," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 5566–5578. DOI: 10.18653/v1/2023.findings-emnlp.370.

[15] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," *arXiv preprint arXiv:1606.08415*, 2016. DOI: 10.48550/arXiv.1606.08415.

[16] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," in *International Conference on Learning Representations (ICLR) Workshop*, 2018. DOI: 10.48550/arXiv.1710.05941.

[17] N. Shazeer, "GLU variants improve transformer," *arXiv preprint arXiv:2002.05202*, 2020. DOI: 10.48550/arXiv.2002.05202.

[18] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, Barcelona, Spain: Association for Computational Linguistics, 2004, pp. 74–81.

[19] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proceedings of ACL*, 2002, pp. 311–318. DOI: 10.3115/1073083.1073135.

[20] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," in *International Conference on Learning Representations (ICLR)*, 2020. DOI: 10.48550/arXiv.1904.09675.

[21] E. Sharma, C. Li, and L. Wang, "BIGPATENT: A large-scale dataset for abstractive and coherent summarization," in *Proceedings of ACL*, 2019, pp. 2204–2213. DOI: 10.48550/arXiv.1906.03741.

[22] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of ACL*, 2017, pp. 1073–1083. DOI: 10.18653/v1/P17-1099.

[23] S. Narayan, S. B. Cohen, and M. Lapata, "Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization," in *Proceedings of EMNLP*, 2018, pp. 1797–1807. DOI: 10.18653/v1/D18-1206.