Beyond GeLU: The Impact of Activation Functions on the Performance of the PEGASUS-X Model

Gayan Kumarasekara
Department of Computer Science and Engineering
University of Moratuwa
Colombo, Sri Lanka
gayank.21@cse.mrt.ac.lk

Uthayasanker Thayasivam
Department of Computer Science and Engineering
University of Moratuwa
Colombo, Sri Lanka
rtuthaya@cse.mrt.ac.lk

***Abstract*—In the context of abstractive summarisation, activation functions play a crucial role in governing the performance and training stability of large-scale transformer models. This study examines how different activation functions affect the performance of the PEGASUS-X model, an advanced transformer-based summarisation model. In this work, the activation function used in the model's feed-forward layers was replaced and tested across the entire architecture. Four activation functions were evaluated: the standard GELU baseline, ReLU, SiLU, and GELU New. Experiments were carried out on several benchmark datasets, and the summarisation quality was measured using the ROUGE-1, ROUGE-2, and ROUGE-Lsum metrics. The results show that GELU New gives slightly better performance for short text summarisation tasks.**

***Index Terms*—Abstractive Summarisation, PEGASUS-X, Activation Functions, Transformer Models, Deep Learning**

I. Introduction and 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][2][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 summarisation 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 summarisation 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].

While PEGASUS-X gives state-of-the-art results for abstractive summarisation, its architecture can be further optimised to improve summarisation of longer documents. The potential of architectural modifications, such as introducing 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.

In this study, the PEGASUS-X architecture is systematically analysed to assess the impact of different activation functions on summarisation performance. The standard GELU activation function in the model's feed-forward networks is replaced with alternative functions, including ReLU, SiLU, and GELU-new, across the entire architecture. The evaluation is conducted on multiple benchmark datasets to understand how these modifications influence model behaviour and summarisation quality.

II. Related Work

Deep neural networks rely on nonlinear activation functions to learn complex features [8][9]. Early architectures used sigmoidal units such as logistic or tanh, but these saturating functions often caused vanishing gradients during training [10]. The Rectified Linear Unit (ReLU) became widely adopted because it preserves gradients for positive inputs and simplifies optimisation [11]. Numerous variants and alternatives have since been proposed. For example, Leaky ReLU [12] and Parametric ReLU [13] introduce nonzero slopes for negative inputs. ELU [14] and SELU [15] provide smooth exponential behaviour. A notable smooth activation is the Gaussian Error Linear Unit (GELU) [16], which weights inputs by their probability under a Gaussian. GELU was adopted as the default activation function in BERT [17] and related transformers. Other novel functions have been discovered via search or hand-design. For example, Ramachandran et al. [11] used automated architecture search to find Swish, which modestly outperformed ReLU in image classification benchmarks. In NLP-specific evaluations, Blau et al. [10] compare 21 activations across multiple tasks and find that penalised tanh yields very stable gains. These works illustrate that, while there is a wide variety of activation functions, the choice of nonlinearity can significantly affect performance.

Large transformer models rarely revisit their activation functions. As Fang et al. [8] note, transformer-based language models typically fix their nonlinearity a priori and do not re-tune it later. For example, the original transformer used

ReLU in its feed-forward sublayers [18], but BERT and many subsequent models replaced this with GELU. By default, then, architectures like BART [1] or PEGASUS [4] also use GELU for the intermediate layer. There has been some exploration of alternatives. Fang et al. [8] introduce learnable rational activation functions in a BERT-like model, showing that such an activation function can be learned per layer and that a model based on such activation functions outperforms the fixed-GELU baseline on the GLUE and SQuAD benchmarks. However, most works in natural language processing implicitly assume the default activation functions and do not empirically compare different activation functions, in contrast to early neural network research.

Transformer-based summarisation systems such as BART [1], T5 [19] and PEGASUS [4] inherit their nonlinearities from the underlying transformer design, and no study has specifically examined modifying them for summarisation. In other words, existing literature on summarisation focus on objectives and architecture without addressing the choice of activation function. This gap mirrors the general trend noted above. As Fang et al. observe [8], the selection of the activation function is rarely discussed or explored in transformers.

III. Methodology

This section details the experimental design adopted to investigate how different activation functions influence the performance of the PEGASUS-X model in abstractive text summarisation. The approach focuses on systematically modifying the nonlinear components of the model while maintaining identical architectural and training conditions, thereby isolating the effect of activation function choice. The methodology is designed to ensure a fair and reproducible comparison among activation functions. Each configuration uses the same dataset, training hyperparameters, and optimisation strategy, allowing performance differences to be attributed primarily to the activation function. Model training and evaluation are performed on identical computational setups to maintain consistency in gradient dynamics and convergence behaviour.

A. 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 at 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.

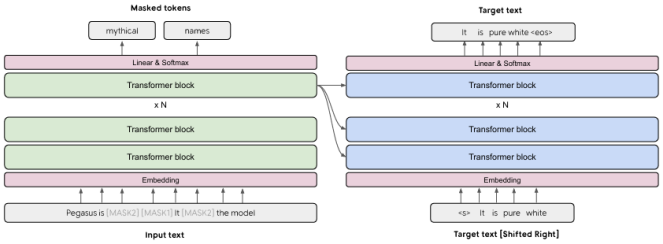

Fig. 1. The architecture of the PEGASUS model which was later extended for the PEGASUS-X model

- **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. The input context length is extended from 512 tokens of the standard PEGASUS model to 16K tokens during fine tuning.
- **Pretraining and fine tuning strategy -** Similar to PEGASUS, PEGASUS-X is pretrained on short sequences of 512 tokens with masked sentence prediction. But it 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 [19] in some cases, while only slightly regressing on short input tasks.

B. Data Acquisition and Preprocessing

To comprehensively evaluate the influence of activation functions on the PEGASUS-X model, experiments were conducted across six benchmark summarisation datasets that vary in length, style, and domain. This diversity ensured that the findings generalise across both short and long-document summarisation settings.

- **GovReport -** The GovReport dataset [7] contains long-form government reports and policy documents summarised into concise executive summaries. With documents averaging around 9,000 tokens and summaries approximately 500 tokens, this dataset evaluates model scalability and the stability of gradient dynamics in long-context settings. It consists of 17,000 training samples, 1,000 validation samples, and 1,000 test samples.
- **CNN/DailyMail -** The CNN/DailyMail dataset [20] consists of online news articles paired with multi-sentence highlights written by journalists. It contains approximately 287,000 training samples, 13,000 validation samples, and 11,000 test samples. Articles average about 760 tokens, while summaries are around 60 tokens long. This dataset primarily measures the model's ability to produce coherent, factual, and moderately abstractive multi-sentence summaries. Due to computational resource

constraints, only a part of the dataset was used to fine tune the model.

- **XSum -** The XSum dataset [21] contains BBC news articles with single-sentence abstractive summaries designed to capture the core message of each article. It comprises roughly 204,000 training samples, 11,000 validation samples, and 11,000 test samples. Documents average 430 tokens, requiring the model to generate concise, information-dense summaries rather than extractive paraphrases.
- **SummScreen -** SummScreen [22] is a dialogue-centric dataset built from television and movie transcripts paired with human-written recaps. It includes approximately 26,000 examples, with transcripts often exceeding 6,000 tokens per episode. This dataset tests the ability of PEGASUS-X to handle extended contexts and conversational input structures, offering insight into how activation functions behave in long-sequence summarisation.
- **QMSum -** The QMSum dataset [23] is a human-annotated benchmark dataset consisting of long transcripts of meetings. The average input length is around 9,100 words. The dataset comprises 1,808 query-summary pairs derived from 232 meetings across academic, product, and committee domains. The reference summaries are relatively short, averaging approximately 70 words. QMSum evaluates a model's ability to handle long-context dialogues and perform focused, information-seeking summarisation.
- **BIGPATENT -** The BIGPATENT dataset [24] is a large-scale collection of U.S. patent documents for abstractive summarisation. It contains approximately 1.3 million records, where the input document is the detailed patent description and the summary is the human-written patent abstract. This dataset is designed to challenge summarisation models by requiring them to handle specialised, long-form technical text with a complex discourse structure. The summaries generally exhibit lower lexical overlap with the source compared to news datasets, promoting highly abstractive generation. Due to computational resource constraints, only a part of the dataset was used to fine tune the model.

To ensure consistent and reproducible data handling across all datasets, a standardised preprocessing pipeline was implemented. The steps were designed to align with PEGASUS-X's tokenisation and long-context processing capabilities.

- Raw text from each dataset was first cleaned to remove HTML tags, escape characters, and extra whitespace. For GovReport, section headers and bullet points were merged into continuous text to maintain coherence during tokenisation.
- Sentences were lowercased to maintain consistency across datasets.
- All datasets were tokenised using the standard PEGASUS tokenizer, configured with a vocabulary size of 96,000 subword units. Tokenisation was consistently applied across the datasets using the Hugging Face `transformers` library, ensuring compatibility with PEGASUS-X's pretraining scheme and shared embeddings across experiments.
- To accommodate PEGASUS-X's extended context capability, dataset-specific maximum input lengths were applied. Inputs shorter than the maximum length were dynamically padded, while longer sequences were truncated from the end, as preliminary trials indicated that critical information typically occurs earlier in documents.
- Official train/validation/test splits provided with each dataset were used to ensure comparability with prior work.
- Each model variant was evaluated on the held-out test split for each dataset. The final scores were averaged over multiple runs to mitigate variance caused by stochastic factors in training.

This evaluation framework facilitated the examination of activation function behaviour across diverse text characteristics, ranging from short, factual summaries to long, multi-paragraph reports. Combined with a unified preprocessing pipeline that enforces consistent tokenisation, truncation, and batching strategies, this design provided a controlled environment in which the influence of activation functions could be assessed independently of the variability of the dataset. This standardisation ensured that the observed performance differences reflect the true impact of activation functions on summarisation quality and training stability.

C. Experimental Configuration

A separate variant of PEGASUS-X was fine tuned for each activation function. The model architecture and most of the hyperparameters were kept constant to isolate the impact of the activation function. Following activation functions were used for experimentation.

- **Rectified Linear Unit -** The Rectified Linear Unit (ReLU) [25] is one of the most widely adopted activation functions in deep learning due to its simplicity and computational efficiency. It is defined as

$$\text{ReLU}(x) = \max(0, x)$$

and introduces non-linearity by setting all negative input values to zero while preserving positive values unchanged. This sparsity property promotes efficient gradient propagation and accelerates convergence during training. However, ReLU suffers from a limitation known as the dying ReLU problem, where neurons can become inactive when their inputs consistently fall below zero, leading to vanishing gradients and limited representational flexibility. Despite this drawback, ReLU remains a strong baseline activation due to its robustness and low computational cost, particularly in large-scale transformer architectures.
- **Gaussian Error Linear Unit -** The Gaussian Error Linear Unit (GELU) is a smooth, probabilistic activation

function that blends the linear and non-linear regimes of neural activations. It is mathematically expressed as

$$\text{GELU}(x) = x \cdot \Phi(x) = \frac{x}{2}\left[1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right]$$

where $\Phi(x)$ denotes the cumulative distribution function of the standard normal distribution, and $\text{erf}(\cdot)$ is the Gaussian error function. Intuitively, GELU scales each input x by the probability that a standard normal random variable is less than x, thus smoothly weighting the activations instead of applying a hard threshold. Unlike ReLU, which deterministically sets negative inputs to zero, GELU allows small negative values to pass through with reduced magnitude, resulting in a more natural gating of information. This probabilistic behaviour has been empirically shown to enhance learning stability and generalisation in large-scale Transformer models such as BERT [17] and PEGASUS [4]. The continuous and differentiable nature of GELU also mitigates abrupt gradient changes, leading to smoother optimisation in deep architectures.

- **GELU New -** The GELU New activation function is an efficient approximation of the original GELU function, designed to reduce computational complexity while maintaining similar nonlinear characteristics. It replaces the error function used in the original GELU with a smooth hyperbolic tangent formulation, expressed as

$$\text{GELU}_{\text{new}}(x) = \frac{1}{2}x\left[1 + \tanh\left(\sqrt{\tfrac{2}{\pi}}\left(x + 0.044715x^3\right)\right)\right]$$

This tanh-based approximation achieves nearly identical activation behaviour to the original GELU but offers faster computation and improved numerical stability, particularly on GPUs and TPUs. In practice, GELU New facilitates more efficient training and inference, making it a preferred choice in modern transformer architectures such as PEGASUS-X, where minor speed optimisations accumulate significantly across deep layers. The smooth curvature of GELU New also contributes to stable gradient flow and consistent convergence behaviour during fine-tuning.

- **Sigmoid Linear Unit -** The Sigmoid Linear Unit (SiLU) [26], also referred to as the Swish activation function, is defined as

$$\text{SiLU}(x) = x \cdot \sigma(x)$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. SiLU introduces a smooth, non-monotonic transformation that retains small negative input values while suppressing larger negative values more strongly. This property provides a compromise between ReLU's sparsity and GELU's smooth probabilistic nature. SiLU enhances gradient flow and supports better information propagation through deep networks by avoiding sharp activation thresholds. As a result, SiLU has demonstrated superior convergence behaviour and improved representational power in transformer-based models compared to traditional piecewise-linear activations.

D. Training Setup

All training experiments were carried out using the Hugging Face `transformers` library in conjunction with PyTorch. Since the link in the GitHub repository to the tokenizer was broken, both the model and tokenizer were sourced directly from the same Hugging Face model repository. This avoided external URL dependencies and ensured that the tokenizer configuration matches the checkpoint. Each variant of PEGASUS-X was fine-tuned on NVIDIA Tesla P100 GPUs for computational efficiency. Table I shows the hyperparameters that were held constant across all runs.

TABLE I
HYPERPARAMETERS THAT WERE USED THROUGHOUT THE EXPERIMENTS

Hyperparameter	Value
Batch Size	8
Epochs	3
Learning Rate	5e-5
Optimiser	AdamW
Weight Decay	0.01
Gradient Clipping	1.0

Some hyperparameters were adjusted according to the datasets, as shown in table II.

TABLE II
MAXIMUM INPUT AND OUTPUT TOKEN LENGTHS FOR EACH DATASET

Dataset	Max Input Tokens	Max Output Tokens
XSum	1024	128
CNN/DailyMail	1024	128
QMSum	16384	256
SummScreen	16384	256
GovReport	12288	1024
Big Patent	16384	256

E. Evaluation Metrics

Model performance was evaluated using the ROUGE metric family [27], which measures n-gram overlap between the generated and reference summaries. Specifically, ROUGE-1, ROUGE-2, and ROUGE-Lsum scores are reported. These metrics were chosen because they are the standard evaluation metrics used for the original PEGASUS-X model [5].

- **ROUGE-1 -** ROUGE-1 measures the overlap of unigrams between the generated summary and the reference summary. It primarily captures the model's ability to reproduce important words that appear in the ground-truth summary. A higher ROUGE-1 score indicates that the summary generated by the system successfully preserves the key lexical content of the original text. It is formally defined as

$$\text{ROUGE-1} = \frac{\sum_{w \in R} \min\left(C_G(w),\, C_R(w)\right)}{\sum_{w \in R} C_R(w)}$$

where $C_G(w)$ and $C_R(w)$ represent the word counts in the generated and reference summaries, respectively.

- **ROUGE-2 -** ROUGE-2 extends the concept of lexical overlap to bigrams, providing a measure of the model's capability to maintain local word order and phrase-level coherence. It reflects how well the generated summary captures the syntactic and semantic flow of the original content. The ROUGE-2 score is calculated as

$$\text{ROUGE-2} = \frac{\sum_{\text{b} \in R} \min(C_G(\text{b}), C_R(\text{b}))}{\sum_{\text{b} \in R} C_R(\text{b})}$$

  Higher ROUGE-2 values indicate that the summariser generates text sequences that closely match the phrasing and local dependencies found in the reference summary.
- **ROUGE-Lsum -** ROUGE-Lsum extends the ROUGE family by evaluating the quality of a generated summary based on the longest common subsequence (LCS) between the generated and reference summaries. Unlike ROUGE-1, which focuses on exact word overlap, ROUGE-Lsum captures the fluency and structural coherence of summaries by considering the sequential order of words. It rewards longer matching phrases that appear in the same order, even if they are not contiguous. The sentence-level ROUGE-L score between a generated summary G and a reference summary R is calculated as

$$\text{ROUGE-Lsum} = \frac{(1+\beta^2) \cdot P_{\text{LCS}} \cdot R_{\text{LCS}}}{R_{\text{LCS}} + \beta^2 \cdot P_{\text{LCS}}}$$

  where $P_{\text{LCS}} = \frac{LCS(G,R)}{|G|}$ denotes the precision and $R_{\text{LCS}} = \frac{LCS(G,R)}{|R|}$ denotes the recall based on the length of the longest common subsequence $LCS(G,R)$. The parameter β (typically set to 1.2) balances recall and precision. For document-level summarisation, ROUGE-Lsum computes the LCS at the sentence level across the entire summary, thereby reflecting how well the generated text preserves both the content and the syntactic structure of the reference summary. A higher ROUGE-Lsum score indicates that the generated summary not only captures the essential content but also maintains a coherent sentence flow similar to human-written summaries.

Together, above metrics serve as the primary quantitative indicators in this study to assess the performance impact of different activation functions on PEGASUS-X across multiple datasets.

IV. Results

The experimental results comparing the impact of different activation functions on the summarisation performance of PEGASUS-X are presented in tables III, IV and V. Each table reports the average ROUGE scores obtained across the six benchmark datasets.

Overall, the GELU activation function consistently achieved higher ROUGE-1, ROUGE-2, and ROUGE-Lsum scores across most datasets. This suggests that its smooth, probabilistic gating mechanism enables more effective modelling of non-linear relationships in abstractive summarisation tasks compared to ReLU and SiLU. Furthermore, the GELU New function showed consistent improvements over GELU on the CNN/DailyMail and XSum datasets, indicating potential benefits for models handling shorter input sequences.

TABLE III
ROUGE-1 SCORES ACROSS DIFFERENT ACTIVATION FUNCTIONS

Dataset	ReLU	GELU	GELU New	SiLU
XSum	37.4	45.8	**46.2**	22.9
CNN/DailyMail	34.8	43.4	**43.9**	25.1
QMSum	26.5	**32.9**	32.4	17.8
SummScreen	27.2	**35.0**	34.5	18.3
GovReport	41.5	**59.3**	58.7	31.8
BIGPATENT	48.4	**61.3**	60.6	30.1

TABLE IV
ROUGE-2 SCORES ACROSS DIFFERENT ACTIVATION FUNCTIONS

Dataset	ReLU	GELU	GELU New	SiLU
XSum	17.2	22.8	**23.1**	10.4
CNN/DailyMail	18.8	21.2	**21.5**	10.2
QMSum	7.4	**9.8**	9.5	2.9
SummScreen	4.8	**8.9**	8.5	2.3
GovReport	20.1	**29.3**	28.7	13.2
BIGPATENT	37.4	**42.6**	41.9	22.9

TABLE V
ROUGE-LSUM SCORES ACROSS DIFFERENT ACTIVATION FUNCTIONS

Dataset	ReLU	GELU	GELU New	SiLU
XSum	29.6	37.6	**38.0**	20.1
CNN/DailyMail	30.5	40.6	**41.2**	21.9
QMSum	17.4	**21.4**	21.0	12.8
SummScreen	13.2	**20.4**	19.9	8.9
GovReport	21.8	**30.9**	30.1	14.6
BIGPATENT	37.4	**50.1**	49.4	23.7

V. Discussion

The results presented in tables III, IV, and V indicate that GELU leads to the highest ROUGE-1, ROUGE-2, and ROUGE-Lsum scores across most benchmark datasets, followed by ReLU and then SiLU. This pattern suggests that the choice of activation function plays a measurable role in shaping the model's representational capacity and generalisation behaviour. The improvements under GELU are most pronounced in datasets containing complex and lengthy textual structures, such as *GovReport* and *SummScreen*, where effective gradient flow and smoother activation transitions are particularly advantageous.

GELU can be viewed as a probabilistic variant of ReLU, where inputs are modulated by the cumulative distribution function of a standard normal distribution. Rather than applying a hard thresholding operation as in ReLU, GELU weights each input by the probability of its significance. Consequently, GELU allows smoother gradient propagation, especially for near-zero activations, leading to more stable optimisation and improved convergence characteristics.

Interestingly, GELU New achieved slightly higher ROUGE scores than GELU on the *CNN/DailyMail* and *XSum* datasets.

These datasets feature comparatively shorter and more homogeneous input sequences, suggesting that the adjusted nonlinearity of GELU New may be particularly beneficial for tasks requiring finer sensitivity to local linguistic variations. This observation highlights the potential of activation function refinements to adapt to dataset-specific characteristics in abstractive summarisation.

From a theoretical point of view, this probabilistic gating mechanism enhances the model's ability to retain subtle semantic variations, which are crucial for abstractive summarisation tasks. Since PEGASUS-X relies on the encoder–decoder attention mechanism to align and compress semantic representations, smoother activation transitions reduce information loss during nonlinear transformations, ultimately yielding more coherent and informative summaries.

ReLU remains computationally efficient and generally robust, but its piecewise linear nature introduces certain limitations. Specifically, the zero-gradient region of ReLU for negative inputs can lead to inactive neurons. While this issue is partially mitigated in deep transformer architectures through layer normalisation, it still results in less expressive feature representations compared to GELU. ReLU demonstrated stable but comparatively lower ROUGE scores, reflecting its tendency to disregard low-activation signals that might otherwise contribute to nuanced linguistic understanding. Nonetheless, its simplicity ensures efficient training and serves as a strong baseline for activation comparisons.

SiLU produced results below ReLU in all cases. Although SiLU introduces a smooth, non-monotonic activation curve, its sigmoid component can cause gradient saturation for large positive or negative values, thereby reducing training dynamics in deeper layers. In PEGASUS-X, which already employs complex attention pathways, this reduced gradient responsiveness may hinder the propagation of useful contextual signals across encoder and decoder stages. While SiLU has shown advantages in certain vision and reinforcement learning tasks, its relatively weaker performance in text summarisation suggests that smoothness alone is insufficient without the adaptive probabilistic behaviour characteristic of GELU.

VI. Conclusion and Future Work

The consistent superiority of GELU across most benchmark datasets underscores the importance of smooth, probabilistically informed activation functions in transformer-based summarisation models. In particular, GELU New achieved the best results on shorter-text datasets such as *CNN/DailyMail* and *XSum*, indicating its potential advantage in capturing fine-grained semantic nuances. These findings highlight that the choice of activation function influences not only the stability of the optimisation, but also the linguistic fidelity and representational capacity of the generated summaries.

Future work will focus on exploring other hyperparameters of PEGASUS-X, such as layer depth, attention head configuration, feed-forward dimensionality, and dropout regularisation. A systematic investigation of these factors may yield deeper insights into the interaction between model architecture and summarisation performance, further advancing the design of efficient and contextually robust transformer models.

References

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

[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. DOI: 10.48550/arXiv.1912.08777.

[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 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 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] H. Fang, J.-U. Lee, N. S. Moosavi, and I. Gurevych, "Transformers with learnable activation functions," in *Findings of the Association for Computational Linguistics: EACL 2023*, Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 2382–2398. DOI: 10.18653/v1/2023.findings-eacl.181.

[9] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, vol. 503, pp. 92–108, 2022. DOI: 10.1016/j.neucom.2022.06.111.

[10] S. Eger, P. Youssef, and I. Gurevych, "Is it time to swish? comparing deep learning activation functions across NLP tasks," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 4415–4424. DOI: 10.18653/v1/D18-1472.

[11] P. Ramachandran, B. Zoph, and Q. V. Le, *Searching for activation functions*, 2017. DOI: 10.48550/arXiv.1710.05941.

[12] A. L. Maas, "Rectifier nonlinearities improve neural network acoustic models," 2013.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," ser. ICCV '15, USA: IEEE Computer Society, 2015, pp. 1026–1034. DOI: 10.1109/ICCV.2015.123.

[14] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, *Fast and accurate deep network learning by exponential linear units (ELUs)*, 2016. DOI: 10.48550/arXiv.1511.07289.

[15] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, Long Beach, California, USA: Curran Associates Inc., 2017, pp. 972–981. DOI: 10.48550/arXiv.1706.02515.

[16] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," *arXiv preprint arXiv:1606.08415*, 2016. DOI: 10.48550/arXiv.1606.08415.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[18] A. Vaswani et al., "Attention is all you need," 2017. DOI: 10.48550/arXiv.1706.03762.

[19] M. Guo et al., "LongT5: Efficient text-to-text transformer for long sequences," in *Findings of the Association for Computational Linguistics: NAACL 2022*, Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 724–736. DOI: 10.18653/v1/2022.findings-naacl.55.

[20] 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.

[21] 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.

[22] M. Chen, Z. Chu, S. Wiseman, and K. Gimpel, "SummScreen: A dataset for abstractive screenplay summarization," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8602–8615. DOI: 10.18653/v1/2022.acl-long.589.

[23] M. Zhong et al., "QMSum: A new benchmark for query-based multi-domain meeting summarization," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online: Association for Computational Linguistics, Jun. 2021, pp. 5905–5921. DOI: 10.18653/v1/2021.naacl-main.472.

[24] 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.

[25] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10, Haifa, Israel: Omnipress, 2010, pp. 807–814.

[26] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks*, vol. 107, pp. 3–11, 2018, Special issue on deep reinforcement learning. DOI: 10.1016/j.neunet.2017.12.012.

[27] 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.