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 summarization, the critical role of activation functions in governing the performance and convergence of large-scale transformer architectures necessitates a detailed examination. In this study, the impact of various activation functions on the PEGASUS-X model has been investigated. While standard transformer models often rely on a limited set of non-linearities, the potential benefits derived from smoother alternatives require quantification. To this end, the function embedded within the model's feed-forward networks was systematically replaced across the entire architecture. Four candidate activation functions were evaluated: the established GELU baseline, alongside other functions such as ReLU and SiLU. Experimental evaluations were performed using Several datasets. The resulting summarization quality was quantified using standard metrics, such as ROUGE-1 and ROUGE-2 scores. The comparative analysis generated from this work serves to clearly describes the architectural and performance trade-offs associated with each activation choice. These findings provide empirical evidence that the selection of the activation function is a critical hyperparameter, directly impacting the representational power and convergence of large-scale summarization models, offering a clear guideline for future PEGASUS-X fine-tuning strategies.**

***Index Terms*—Abstractive Summarization, 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]–[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].

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 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.

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 optimization [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 in BERT [17] and related Transformers. Other novel functions have been discovered via search or hand-design. For instance, Ramachandran et al. [11] used automated architecture search to find Swish, which modestly outperformed ReLU on image-classification benchmarks. In NLP-specific evaluations, Blau et al. [10] compare 21 activations across multiple tasks and find that penalized tanh yields very stable gains. These works illustrate that while a wide variety of activations exists, the choice of nonlinearity can significantly affect performance.

Large transformer models rarely revisit their activation functions. As Fang et al. (2022) note, Transformer-based language models typically fix their nonlinearity a priori and do not re-tune it later [8]. 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 or PEGASUS 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 GLUE and SQuAD benchmarks. However, most work in natural language processing implicitly assumes the default activation functions and does not empirically compare different activations, in contrast to early neural network research.

Transformer-based summarization 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 summarization. In other words, existing summarization papers focus on objectives and architecture without addressing the activation function choice. This gap mirrors the general trend noted above. As Fang et al. observe [8], activation function selection is "seldom discussed or explored" in Transformers. No connected prior work that evaluates ReLU vs. GELU or other activations specifically in a summarization model could be found.

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 fair and reproducible comparison across 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 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.

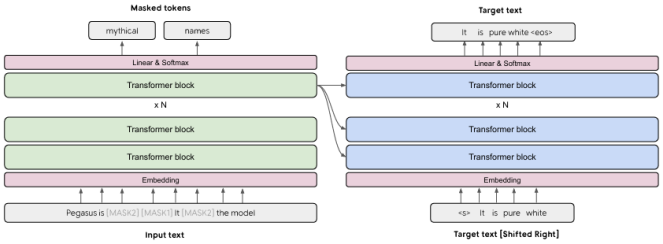

Fig. 1. The architecture of the PEGASUS model which was later extended for PEGASUS-X model

- **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. 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 [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 four benchmark summarisation datasets that vary in length, style, and domain: CNN/DailyMail, XSum, SummScreen, and GovReport. This diversity ensures 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 under 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] comprises online news articles paired with multi-sentence highlights written by journalists. It contains approximately 287,000 training examples, 13,000 validation samples, and 11,000 test instances. 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.

- **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 examples, 11,000 validation samples, and 11,000 test instances. 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 4,000 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.

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, escaped characters, and extraneous 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, and Unicode normalisation was applied to unify accented characters.
- All datasets were tokenised using the standard PEGASUS tokenizer, configured with a vocabulary size of 96,000 subword units. Tokenisation was applied consistently across 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 padded dynamically, 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.

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 dataset variability. Such standardisation ensured that observed performance differences reflect the true impact of activation functions on summarisation quality and training stability.

C. Experimental Configuration

For each activation function, a separate variant of PEGASUS-X was instantiated. The model architecture and hyperparameters, except for the activation layer, 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) [23] 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

$$\text{GELU}(x) = x \cdot \Phi(x)$$

  where $\Phi(x)$ denotes the cumulative distribution function of the standard normal distribution. Unlike ReLU, which deterministically zeroes out negative inputs, GELU weights the input by the probability that it should be activated, allowing for a smoother transition between active and inactive states. This probabilistic gating mechanism captures richer input dynamics, which has been empirically shown to improve the learning stability and generalisation of large language models, including BERT [17] and PEGASUS [4]. The continuous and differentiable nature of GELU also mitigates abrupt gradient changes, enabling more stable optimisation in deep architectures.
- **Sigmoid Linear Unit -** The Sigmoid Linear Unit (SiLU) [24], 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 negatives 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 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 were held constant across all runs.

TABLE I
HYPERPARAMETERS THAT WERE USED THROUGHOUT THE EXPERIMENTS

Hyperparameter	Value
Learning Rate	2×10^{-5}
Batch Size	8
Epochs	3
Optimiser	AdamW
Weight Decay	0.01
Gradient Clipping	1.0

E. Evaluation Metrics

Model performance was evaluated using the ROUGE metric family, which measures n-gram overlap between the generated and reference summaries. Specifically, ROUGE-1 and ROUGE-2 scores will be reported.

- **ROUGE-1 -** ROUGE-1 [25] 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 system-generated summary 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 [25] 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.

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 table II and table III. Each table reports the average ROUGE scores obtained across the four benchmark datasets.

TABLE II
ROUGE-1 SCORES ACROSS DIFFERENT ACTIVATION FUNCTIONS

Dataset	ReLU	GELU	SiLU
SummScreen	27.2	**35.0**	18.8
GovReport	41.5	**59.3**	31.8
CNN/DailyMail	34.8	**43.4**	25.1
XSum	37.4	**45.8**	22.9

TABLE III
ROUGE-2 SCORES ACROSS DIFFERENT ACTIVATION FUNCTIONS

Dataset	ReLU	GELU	SiLU
SummScreen	4.8	**8.9**	2.3
GovReport	20.1	**29.3**	13.2
CNN/DailyMail	18.8	**21.2**	10.2
XSum	17.2	**22.8**	10.4

Overall, the GELU activation function consistently achieved higher ROUGE-1 and ROUGE-2 scores across all datasets, suggesting that its smooth, probabilistic gating function better captures non-linear relationships in summarisation tasks compared to ReLU and SiLU.

V. DISCUSSION

The results presented in tables II and III indicate that GELU leads to the highest ROUGE-1 and ROUGE-2 scores on every benchmark dataset, 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 that contain 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 version of ReLU, where inputs are modulated by the cumulative distribution function of a standard normal distribution. This means that instead of performing a hard thresholding operation as in ReLU, GELU weights each input by the probability of its significance. Consequently, GELU allows a smoother gradient propagation,

especially for near-zero activations, leading to more stable optimisation and better convergence characteristics.

From a theoretical standpoint, 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, ReLU's zero-gradient region 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, reducing training dynamics in deep 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 its smoothness alone is insufficient without the adaptive probabilistic behaviour characteristic of GELU.

VI. Conclusion and Future Work

The consistent superiority of GELU across all tested datasets highlights the importance of smooth, probabilistically informed activation functions in transformer-based summarisation models. This study demonstrates that nonlinearity design is not merely a training stabilisation choice but also a decisive factor influencing representational richness, convergence behaviour, and the overall linguistic fidelity of generated summaries.

For future work, the experimental scope can be expanded by incorporating a wider range of benchmark datasets to further validate the observed trends across diverse domains and text lengths. Additionally, investigating emerging activation variants such as GELU-new and other adaptive nonlinearities may provide deeper insights into how activation functions affect contextual understanding and summarisation performance. Such extensions would contribute to a more comprehensive understanding of the role of activation functions in large-scale transformer architectures like PEGASUS-X.

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] 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*, A. Vlachos and I. Augenstein, Eds., 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: https://doi.org/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*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds., 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. arXiv: 1710.05941 [cs.NE].

[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. arXiv: 1511.07289 [cs.LG].

[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.

[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)*, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

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

[19] 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.

[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)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds., Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8602–8615. DOI: 10.18653/v1/2022.acl-long.589.

[23] 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.

[24] 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.

[25] 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.