ScaleFormer: Span Representation Cumulation for Long-Context Transformer

Jiangshu Du*
jdu25@uic.edu
University of Illinois Chicago
Chicago, IL, USA

Wenpeng Yin
wenpeng@psu.edu
The Pennsylvania State University
University Park, PA, USA

Philip Yu
psyu@uic.edu
University of Illinois Chicago
Chicago, IL, USA

Abstract

The quadratic complexity of standard self-attention severely limits the application of Transformer-based models to long-context tasks. While efficient Transformer variants exist, they often require architectural changes and costly pre-training from scratch. To circumvent this, we propose ScaleFormer (**S**pan Representation **C**umul**a**tion for **L**ong-Cont**e**xt Trans**former**)—a simple and effective plug-and-play framework that adapts off-the-shelf pre-trained encoder-decoder models to process long sequences without requiring architectural modifications. Our approach segments long inputs into overlapping chunks and generates a compressed, context-aware representation for the decoder. The core of our method is a novel, parameter-free fusion mechanism that endows each chunk's representation with structural awareness of its position within the document. It achieves this by enriching each chunk's boundary representations with cumulative context vectors from all preceding and succeeding chunks. This strategy provides the model with a strong signal of the document's narrative flow, achieves linear complexity, and enables pre-trained models to reason effectively over long-form text. Experiments on long-document summarization show that our method is highly competitive with and often outperforms state-of-the-art approaches without requiring architectural modifications or external retrieval mechanisms.

CCS Concepts

• **Computing methodologies** → **Natural language generation**; **Artificial intelligence**; **Natural language processing**;

Keywords

Long-context transformers, Pre-trained language models, Overlapping segmentation, Sliding window attention

ACM Reference Format:

Jiangshu Du, Wenpeng Yin, and Philip Yu. 2025. ScaleFormer: Span Representation Cumulation for Long-Context Transformer. In *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region (SIGIR-AP 2025), December 7–10, 2025, Xi'an, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3767695.3769513

*Work Done Prior to Amazon

SIGIR-AP 2025, Xi'an, China

ACM ISBN 979-8-4007-2218-9/2025/12
https://doi.org/10.1145/3767695.3769513

1 Introduction

The advent of Transformer-based pre-trained language models (PLMs) has marked a paradigm shift in Natural Language Processing (NLP) [21]. However, the Transformer's foundational self-attention mechanism scales quadratically with input sequence length, creating a substantial barrier to its application in domains that rely on long-form text, such as summarizing scientific articles, analyzing legal documents, or processing entire books [1, 2, 25].

To address this long-context barrier, researchers have developed numerous efficient Transformer variants, often based on sparse attention patterns that reduce complexity from quadratic to linear or near-linear. While effective, these models typically introduce custom architectural changes that require expensive pre-training from scratch. As a result, a critical research direction has emerged: adapting existing powerful models for long-context tasks. This raises a central research question: *How can we adapt existing PLMs to handle long-form text without incurring the prohibitive cost of pre-training new models from scratch?*

The landscape of solutions is diverse. One major line of work involves fundamental **architectural modifications** to create efficient Transformers, using techniques like sparse attention [1, 25] or recurrence [6]. Another line focuses on **hierarchical pipelines** that first extract salient information before processing it [20]. A third—and most relevant to our work—centers on **plug-and-play frameworks** designed to wrap around existing PLMs [4, 8, 10, 19]. Prominent examples include SLED [10], which employs a "fusion-in-decoder" strategy [11] by concatenating all encoded chunks, and Unlimiformer [2], which augments the model with a retrieval mechanism that accesses relevant parts of the long input from a k-nearest neighbor (kNN) index.

While effective, these plug-and-play methods place the full burden of discerning the document's overall structure and narrative flow on the decoder, which must infer these relationships from a long sequence of concatenated chunk representations. In this work, we introduce ScaleFormer (Span Representation Cumulation for Long-Context Transformer)—a novel framework that offers an alternative approach by explicitly encoding structural information into the input representations before they reach the decoder.

Our key innovation lies in an efficient and parameter-free fusion scheme. Instead of passing all encoded tokens from every segment to the decoder, we represent each segment using only its "boundaries"—the hidden states of its leftmost and rightmost tokens. These boundary representations are then dynamically enriched through our novel span representation cumulation mechanism. This technique provides each segment with positional awareness

by combining its left boundary with a summary of all preceding text and its right boundary with a summary of all succeeding text. This approach achieves linear complexity while enabling off-the-shelf PLMs to perform effectively on long-context tasks.

Our primary contributions are as follows:

(1) We propose `ScaleFormer`, a novel and efficient framework for applying pre-trained encoder-decoder models to long sequences with linear complexity, without requiring any modifications to the model's internal architecture.
(2) We introduce a directional boundary fusion mechanism that provides each segment with cumulative, position-aware context, explicitly encoding the document's structure into each chunk's representation.
(3) We conduct extensive experiments on long-document summarization tasks, demonstrating that our method achieves highly competitive performance against strong baselines.

2 Related Work

Transformer models have been adapted in various ways to handle long sequences by reducing quadratic self-attention costs. Sparse-attention approaches like Longformer [1] and BigBird [25] limit each token's attention to local windows or selected global tokens. Other efficient attention approximations include Reformer's hashing-based method [13], Routing Transformers' clustering [18], and low-rank or kernel methods such as Linformer [22] and Performer [5]. More recent work continues to push these boundaries, with novel architectures like Infini-attention proposing methods to scale to theoretically infinite contexts by integrating a compressive memory directly into the attention mechanism [16]. These models often require architecture changes and expensive pre-training. In contrast, our method leverages existing pre-trained Transformers without modifying their internal attention mechanism, enabling the efficient reuse of these powerful models.

Retrieval-based methods like RAG [15] and REALM [7] retrieve relevant passages before processing them. The Fusion-in-Decoder (FiD) [11] strategy processes passages independently and fuses the encoded representations in the decoder, inspiring various approaches in QA and summarization. Recent chunk-based methods, notably SLED [10], segment long inputs and encode chunks separately before decoding. Another parallel line of work adapts decoder-only LLMs using similar chunking principles; for instance, CEPE [24] also processes long inputs in parallel chunks but requires inserting new cross-attention modules into a frozen decoder to integrate the context. While our method shares SLED's philosophy of leveraging pre-trained Transformers through chunking, we differ fundamentally in how chunk representations are managed. Instead of using entire chunk encodings, we select only boundary tokens to create compressed representations that are then infused with explicit directional context, a feature absent in prior work.

Another related approach, Unlimiformer [2], extends input lengths by offloading decoder attention to an external kNN index. This allows for extremely long contexts but adds complexity. In contrast, our model operates within the standard Transformer framework without additional indexing, offering a simpler solution. Overall, our method balances efficiency and simplicity by avoiding architectural modifications, extensive retraining, and external retrieval structures.

3 The `ScaleFormer` Framework

We propose `ScaleFormer`, a simple yet effective framework for applying existing PLMs to long-sequence tasks. The core idea is to segment long inputs into overlapping chunks and create a compact, fused representation for each segment that captures both its local content and its structural position within the overall document.

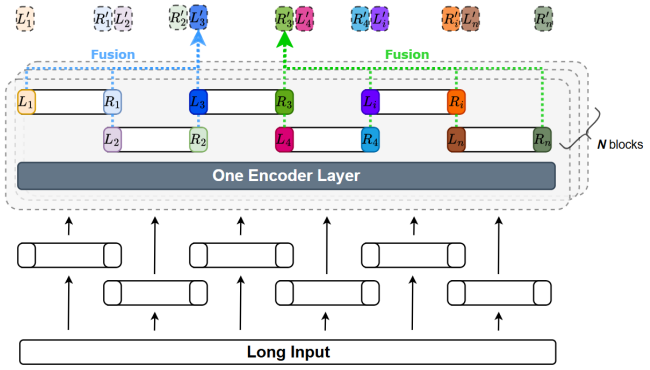

Figure 1: Overview of the `ScaleFormer` framework. A long input document is segmented into overlapping chunks, each encoded independently. Boundary token representations (Left and Right) are extracted and fused with directional context. The Left boundary of a chunk is fused with context from prior chunks, and the Right boundary is fused with context from subsequent chunks, providing structural awareness.

As illustrated in Figure 1, the `ScaleFormer` framework proceeds in several steps. First, a long input document is tokenized and partitioned into multiple overlapping segments. Second, each chunk is processed independently by the encoder. Third, from each sequence of hidden states, we extract the representations corresponding to the first few and last few tokens—the Left (L) and Right (R) boundaries. Fourth, these boundary vectors undergo our novel span representation cumulation step. Finally, the resulting fused representations from all chunks are concatenated and passed to the decoder.

3.1 Overlapping Segment Encoding

Formally, let an input sequence of N tokens be denoted as $X = \{x_1, x_2, \ldots, x_N\}$. We partition X into C overlapping segments $S_1, S_2, \ldots, S_C$. Each segment S_i has a fixed length L, which fits within the model's maximum context window. The overlap between consecutive segments S_i and S_{i+1} is a fixed number of tokens, O.

Each segment S_i is then encoded independently using the pre-trained encoder, M_{enc}, to produce a sequence of top-layer hidden states $H_i \in \mathbb{R}^{L \times d}$, where d is the hidden dimension of the model:

$$H_i = M_{\text{enc}}(S_i) \tag{1}$$

From each H_i, we extract the left boundary $L_i \in \mathbb{R}^{k \times d}$ (first k tokens) and the right boundary $R_i \in \mathbb{R}^{k \times d}$ (last k tokens). This process is inherently parallelizable and scales linearly with the input length N.

3.2 Span Representation Cumulation

After the parallel encoding of segments, the primary challenge is to present this information to the decoder in a way that preserves the document's global structure. A naive approach of simply concatenating the encoded chunks (e.g., $[H_1, H_2, ..., H_C]$) poses a significant challenge for the decoder. This method presents it with a long, structurally undifferentiated sequence of hidden states where the original relationships between chunks are lost.

Consequently, the decoder's cross-attention mechanism is burdened with a difficult dual task: it must first attempt to infer the document's original structure and narrative flow from this flat sequence, and only then can it identify and synthesize the most salient information for summarization. This inefficient use of the model's representational capacity can lead to errors in the final output, such as a lack of coherence, incorrect temporal ordering, or a failure to capture a main thesis developed across the entire document.

Our ScaleFormer is designed specifically to alleviate this burden. Instead of forcing the decoder to infer structure, we explicitly inject it into each segment's representation before the decoding stage. By enriching each chunk's boundaries with cumulative information about its preceding and succeeding context, we provide explicit signals about its position and role, allowing the decoder to focus its resources on generating a high-quality, globally-aware summary.

Boundary Extraction: For each encoded segment H_i, we extract its boundary representations. These are the hidden states corresponding to the first k tokens, which we call the left boundary $L_i \in \mathbb{R}^{k \times d}$, and the last k tokens, the right boundary $R_i \in \mathbb{R}^{k \times d}$. These boundaries serve as proxies for the contextualized information at the edges of each segment, which are crucial for linking information across chunks.

Directional Context: For each segment S_i, we compute a backward-looking context vector, $\text{ctx}_i^{\text{back}}$, and a forward-looking context vector, $\text{ctx}_i^{\text{fwd}}$.

The **backward context** for segment i is the average of its own left boundary L_i and all boundary vectors (L_j and R_j) from preceding segments ($j < i$):

$$\text{ctx}_i^{\text{back}} = \frac{1}{2i-1}\left(L_i + \sum_{j=1}^{i-1}(L_j + R_j)\right) \quad \text{for } i > 1 \tag{2}$$

The **forward context** for segment i is the average of its own right boundary R_i and all boundary vectors from succeeding segments ($j > i$):

$$\text{ctx}_i^{\text{fwd}} = \frac{1}{2(C-i)+1}\left(R_i + \sum_{j=i+1}^{C}(L_j + R_j)\right) \quad \text{for } i < C \tag{3}$$

Here, C represents the total number of chunks. For the first chunk's backward context and the last chunk's forward context, the context vector is simply the local boundary itself.

Weighted Fusion: The final fused representations, L'_i and R'_i, are created by blending the local boundary vector with its corresponding directional context vector, controlled by a hyperparameter $\alpha \in [0, 1]$.

$$L'_i = \alpha \cdot L_i + (1-\alpha) \cdot \text{ctx}_i^{\text{back}} \tag{4}$$

$$R'_i = \alpha \cdot R_i + (1-\alpha) \cdot \text{ctx}_i^{\text{fwd}} \tag{5}$$

3.3 Middle Token Sampling

While boundary vectors are effective for capturing inter-segment context, they may not fully represent the unique content within each segment. To enhance the local representation, we also sample m "middle" tokens, $M_i \in \mathbb{R}^{m \times d}$, from the interior of each encoded chunk H_i. The purpose of these middle tokens is to provide the decoder with direct, unaltered snapshots of the segment's core content. This strategy further improves performance by providing richer local information while maintaining a highly compressed overall representation. This sampling can be seen as a form of text compression, conceptually similar to recent work in prompt compression. For instance, LLMLingua [12, 17] compresses prompts by using a small language model to identify and remove less informative tokens. Although our current implementation employs random sampling, we plan to explore more effective token-selection methods in future work.

3.4 Final Decoder Input

The final sequence of hidden states passed to the decoder M_{dec} is the concatenation of the fused boundary representations and the sampled middle tokens from all segments:

$$H_{\text{final}} = \text{concat}(L'_1, M_1, R'_1, L'_2, M_2, R'_2, \ldots, L'_C, M_C, R'_C) \tag{6}$$

The length of this sequence is $C \times (2k + m)$, which is substantially smaller than the $C \times L$ tokens used by a naive fusion-in-decoder approach like SLED.

3.5 Complexity Analysis

The computational complexity of our ScaleFormer framework scales favorably with the input sequence length N. The encoding step involves C independent forward passes through the encoder on sequences of length L. Since $C \approx N/(L-O)$, the total complexity for encoding is proportional to $(N/L) \times O(L^2) = O(N \cdot L)$. As L is a fixed constant (the model's maximum context size), the encoding complexity is linear with respect to the input length, i.e., $O(N)$. The directional context calculation involves a single pass over the C chunk boundaries, which is computationally negligible. The decoder's complexity is dominated by the length of its input, $C \times (2k + m)$. Since C is proportional to N, and k, m are small constants, the effective complexity of our entire framework is **linear** with respect to the input sequence length N.

4 Experiments

4.1 Experimental Setup

Tasks and Datasets: We evaluate our framework on three long-document summarization benchmarks:

- **SummScreen** [3]: Comprises TV show transcripts and their summaries, with an average input length of approximately 9,000 tokens.
- **GovReport** [9]: A dataset of lengthy government reports, where inputs are truncated to 16,384 tokens for evaluation consistency.
- **BookSum** [14]: A challenging dataset for narrative summarization of books, with extremely long documents averaging over 140,000 tokens.

Table 1: BART-base summarization results on dev and test sets. R-1/2/L are ROUGE scores; BS is BERT-Score. Best scores in each column are in bold. For tied scores, both are bolded.

	Method	SummScreen				GovReport				BookSum			
		R-1	R-2	R-L	BS	R-1	R-2	R-L	BS	R-1	R-2	R-L	BS
Dev	Standard (Trunc.)	30.0	6.5	17.7	56.7	47.7	18.5	22.3	64.0	35.3	8.8	13.9	70.2
	SLED	34.2	8.2	19.2	58.8	55.5	24.8	25.8	66.9	-	-	-	-
	Mem. Trans.	32.8	7.6	19.3	57.7	55.8	25.6	26.9	67.7	-	-	-	-
	Unlimiformer	35.0	8.3	19.6	58.4	**57.4**	**26.4**	27.9	68.2	35.6	**11.1**	15.0	71.1
	`ScaleFormer` (Ours)	34.3	8.1	19.3	58.6	56.1	25.2	27.5	67.6	35.7	9.7	14.7	70.9
	`ScaleFormer` + Middle	**34.7**	**9.2**	**20.1**	**58.9**	**57.4**	26.0	**28.1**	**68.3**	**36.7**	10.9	**15.8**	**71.5**
Test	Standard (Trunc.)	29.7	6.2	17.7	56.3	48.7	19.2	22.8	64.3	36.4	7.6	15.3	70.5
	SLED	32.7	7.9	19.1	58.4	54.7	24.4	25.4	67.0	-	-	-	-
	Mem. Trans.	32.7	7.4	19.2	57.4	55.2	25.1	26.4	67.5	35.6	6.4	14.6	70.9
	Unlimiformer	**34.7**	8.5	19.9	58.5	56.6	**26.3**	27.6	68.2	37.3	6.7	15.2	71.3
	`ScaleFormer` (Ours)	33.3	8.3	19.2	58.7	56.0	25.1	26.8	67.5	37.1	7.1	15.7	71.3
	`ScaleFormer` + Middle	34.2	**9.0**	**20.3**	**59.0**	**57.0**	25.7	**27.7**	**68.4**	**39.2**	**9.3**	**16.3**	**71.9**

Backbone Models and Baselines: To demonstrate its generalizability, we apply our method to **BART-base** and **T5-base**. We compare against a suite of strong baselines representing different approaches to long-context processing:

- **Standard (Truncation):** This baseline truncates the input to the model's maximum context window (1024 tokens). It serves as a lower bound, showing performance without any long-context adaptation.
- **SLED** [10]: A chunk-based method that encodes overlapping segments and concatenates all resulting hidden states for the decoder. It serves as our most direct baseline as it also uses a "fusion-in-decoder" strategy, but crucially, it results in a much longer input sequence for the decoder and lacks an explicit mechanism for encoding document structure.
- **Memorizing Transformers** [23]: Enhances the Transformer with an approximate k-nearest-neighbor (kNN) memory, allowing attention layers to access more context than fits in the standard window.
- **Unlimiformer** [2]: A powerful retrieval-based method that offloads the decoder's cross-attention to a kNN index built from all encoder hidden states.

Evaluation Metrics: Following prior work, we report performance using ROUGE-1, ROUGE-2, and ROUGE-L, and BERT-Score on both development and test datasets.

Implementation Details: Our `ScaleFormer` method uses a chunk size (L) of 1024 and an overlap (O) of 150 tokens. We use $k = 1$ for boundary tokens. The fusion ratio α was set to 0.5 based on performance on the dev set. We evaluate two variants:

- **`ScaleFormer`**: Our core model, which uses only the directional fusion mechanism on boundary tokens. The decoder receives a compressed sequence of fused left and right boundary representations from each chunk.
- **`ScaleFormer` + Middle**: Our full model, which augments the fused boundary tokens with $m = 300$ randomly sampled "middle" tokens from each chunk's interior. This variant is designed to balance global structural awareness with richer local content.

4.2 Main Results

We present our findings on both development and test sets, comparing our two variants—`ScaleFormer` (fusion only) and `ScaleFormer` + Middle—against all baselines.

BART-base Results: As shown in Table 1, `ScaleFormer` framework demonstrates strong performance. The base `ScaleFormer` outperforms SLED on the test sets of SummScreen and GovReport. On the GovReport test set, for instance, `ScaleFormer` achieves a ROUGE-1 score of 56.0, surpassing SLED's 54.7. The addition of middle token sampling (`ScaleFormer` + Middle) provides a substantial boost, establishing our method as state-of-the-art on nearly all benchmarks. On the extremely long-context BookSum dataset, `ScaleFormer` + Middle achieves a ROUGE-1 of 39.2 on the test set, decisively outperforming all baselines including Unlimiformer (37.3), and sets new state-of-the-art scores across all four metrics. On the GovReport test set, it achieves the highest ROUGE-1 (57.0), ROUGE-L (27.7), and BERT-Score (68.4).

T5-base Results: To further validate our framework, we applied it to T5-base. The results, presented in Table 2, confirm the consistency of our approach. The performance trends mirror the BART-base experiments: `ScaleFormer` is a clear improvement over the standard truncation baseline and outperforms SLED. Once again, `ScaleFormer` + Middle outperforms both the standard truncation baseline and SLED across all metrics on the test set.

5 Ablation and Analysis

To understand the contribution of each component in our framework and analyze its sensitivity, we conduct ablation studies on the SummScreen development set using BART-base. Unless specified otherwise, hyperparameters are kept at their default values ($L = 1024$, $O = 150$, $m = 300$, $\alpha = 0.5$). We report ROUGE-L, with results presented in Figure 2.

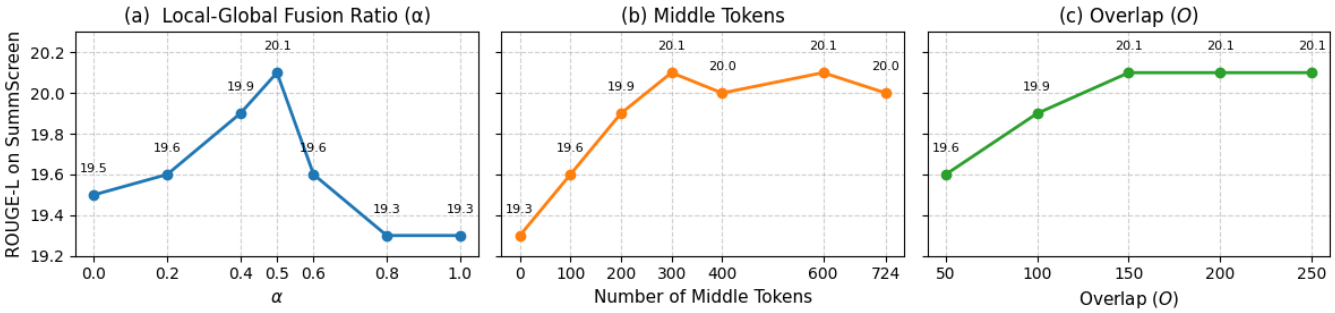

Figure 2: Ablation study results on the SummScreen dev set. We analyze the impact of (a) the directional fusion ratio α, (b) the number of middle tokens sampled (m), and (c) the chunk overlap size (O). Performance is measured in ROUGE-L.

Table 2: T5-base summarization results on dev and test sets. Best scores are in bold.

	Method	SummScreen			GovReport		
		R-1	R-2	R-L	R-1	R-2	R-L
Dev	Standard (Trunc.)	22.2	3.7	15.3	32.8	11.7	20.2
	SLED	25.3	5.0	16.6	47.0	20.2	25.2
	ScaleF. (Ours)	25.5	5.1	16.9	47.9	20.8	26.4
	ScaleF. + Middle	**26.0**	**5.9**	**17.3**	**48.7**	**21.4**	**27.7**
Test	Standard (Trunc.)	21.4	3.6	15.0	33.2	12.1	20.4
	SLED	24.5	4.6	16.5	46.6	20.1	25.1
	ScaleF.(Ours)	24.7	4.7	16.7	47.0	20.3	25.4
	ScaleF. + Middle	**25.4**	**5.1**	**17.1**	**47.9**	**21.2**	**27.4**

5.1 Effect of the Directional Fusion Ratio (α)

The core of our method is the weighted fusion between local boundaries and their directional context, controlled by α. To validate its effectiveness, we analyze performance while varying α from 0.0 to 1.0.

- An α of **1.0** represents a **local-only** model where boundaries are not fused with any directional context, relying only on local information.
- An α of **0.0** represents a **context-only** model where local boundary information is replaced entirely by the directional average.

As shown in Figure 2(a), the model performs poorly at both extremes. A purely local-only model ($\alpha = 1.0$) and a near-context-only model ($\alpha \approx 0.0$) yield low ROUGE-L scores. Performance steadily increases as local and directional contexts are mixed, peaking at $\alpha = 0.5$ with a ROUGE-L of 20.1. This confirms our central hypothesis that a balance between segment-specific details and cumulative document context is critical for strong performance.

5.2 Impact of Middle Token Sampling

We analyze the utility of including m "middle" tokens from each chunk. Figure 2(b) shows a clear benefit. The model without any middle tokens ($m = 0$) starts at a ROUGE-L of 19.3. Performance sharply increases as more local content is provided, reaching a peak of 20.1 at $m = 300$. Adding more tokens beyond this point does not yield further improvements, indicating that 300 tokens provide a sufficient snapshot of local content. This justifies our choice of $m = 300$ for our primary model.

5.3 Analysis of Chunk Overlap

The overlap size (O) between consecutive chunks is critical for ensuring smooth context propagation between segments. In Figure 2(c), performance improves as the overlap increases from 50 to 150 tokens, with the ROUGE-L score rising from 19.6 to 20.1. Beyond this point, increasing the overlap provides no additional benefit while increasing computational overhead. We therefore use $O = 150$ as it achieves the best performance-cost trade-off.

6 Conclusion

In this paper, we introduced `ScaleFormer`, a simple, effective, and parameter-free framework for adapting pre-trained encoder-decoder models to long-context tasks. By segmenting long inputs and fusing chunk boundary representations with directional context, our method provides the model with an explicit understanding of the document's structure, a critical feature often lost in standard chunking approaches. Our method achieves linear complexity without any architectural changes to the underlying model. Extensive experiments on long-document summarization demonstrate that our proposed `ScaleFormer`, especially when augmented with middle token sampling, achieves state-of-the-art or highly competitive results across multiple benchmarks and backbone models.

Acknowledgments

The authors appreciate the reviewers for their insightful comments and suggestions. This work is supported in part by NSF under grants III-2106758, and POSE-2346158

References

[1] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *CoRR* abs/2004.05150 (2020). arXiv:2004.05150 https://arxiv.org/abs/2004.05150

[2] Amanda Bertsch, Uri Alon, Graham Neubig, and Matthew Gormley. 2023. Unlimiformer: Long-Range Transformers with Unlimited Length Input. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 35522–35543. https://proceedings.neurips.cc/paper_files/paper/2023/file/6f9806a5adc72b5b834b27e4c7c0df9b-Paper-Conference.pdf

[3] Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2022. SummScreen: A Dataset for Abstractive Screenplay Summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 8602–8615. doi:10.18653/v1/2022.acl-long.589

[4] Yaofo Chen, Zeng You, Shuhai Zhang, Haokun Li, Yirui Li, Yaowei Wang, and Mingkui Tan. 2025. Core Context Aware Transformers for Long Context Language Modeling. arXiv:2412.12465 [cs.CL] https://arxiv.org/abs/2412.12465

[5] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. 2021. Rethinking Attention with Performers. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Ua6zuk0WRH

[6] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Anna Korhonen, David Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, Florence, Italy, 2978–2988. doi:10.18653/v1/P19-1285

[7] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval Augmented Language Model Pre-Training. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 3929–3938. https://proceedings.mlr.press/v119/guu20a.html

[8] Michael Günther, Isabelle Mohr, Daniel James Williams, Bo Wang, and Han Xiao. 2025. Late Chunking: Contextual Chunk Embeddings Using Long-Context Embedding Models. arXiv:2409.04701 [cs.CL] https://arxiv.org/abs/2409.04701

[9] Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient Attentions for Long Document Summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, Online, 1419–1436. doi:10.18653/v1/2021.naacl-main.112

[10] Maor Ivgi, Uri Shaham, and Jonathan Berant. 2023. Efficient Long-Text Understanding with Short-Text Models. *Transactions of the Association for Computational Linguistics* 11 (2023), 284–299. doi:10.1162/tacl_a_00547

[11] Gautier Izacard and Edouard Grave. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty (Eds.). Association for Computational Linguistics, Online, 874–880. doi:10.18653/v1/2021.eacl-main.74

[12] Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. LLMLingua: Compressing Prompts for Accelerated Inference of Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 13358–13376. doi:10.18653/v1/2023.emnlp-main.825

[13] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The Efficient Transformer. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rkgNKkHtvB

[14] Wojciech Kryscinski, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2022. BOOKSUM: A Collection of Datasets for Long-form Narrative Summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 6536–6558. doi:10.18653/v1/2022.findings-emnlp.488

[15] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 9459–9474. https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf

[16] Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. 2024. Leave No Context Behind: Efficient Infinite Context Transformers with Infini-attention. arXiv:2404.07143 [cs.CL] https://arxiv.org/abs/2404.07143

[17] Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. LLMLingua-2: Data Distillation for Efficient and Faithful Task-Agnostic Prompt Compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 963–981. doi:10.18653/v1/2024.findings-acl.57

[18] Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. 2021. Efficient Content-Based Sparse Attention with Routing Transformers. *Transactions of the Association for Computational Linguistics* 9 (2021), 53–68. doi:10.1162/tacl_a_00353

[19] Ning Shang, Li Lyna Zhang, Siyuan Wang, Gaokai Zhang, Gilsinia Lopez, Fan Yang, Weizhu Chen, and Mao Yang. 2025. LongRoPE2: Near-Lossless LLM Context Window Scaling. arXiv:2502.20082 [cs.CL] https://arxiv.org/abs/2502.20082

[20] Sandeep Subramanian, Raymond Li, Jonathan Pilault, and Christopher J. Pal. 2019. On Extractive and Abstractive Neural Document Summarization with Transformer Language Models. *CoRR* abs/1909.03186 (2019). arXiv:1909.03186 http://arxiv.org/abs/1909.03186

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[22] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-Attention with Linear Complexity. *CoRR* abs/2006.04768 (2020). arXiv:2006.04768 https://arxiv.org/abs/2006.04768

[23] Yuhuai Wu, Markus Norman Rabe, DeLesley Hutchins, and Christian Szegedy. 2022. Memorizing Transformers. In *International Conference on Learning Representations*. https://openreview.net/forum?id=TrjbxzRcnf-

[24] Howard Yen, Tianyu Gao, and Danqi Chen. 2024. Long-Context Language Modeling with Parallel Context Encoding. 2588–2610. https://doi.org/10.18653/v1/2024.acl-long.142

[25] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: transformers for longer sequences. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 1450, 15 pages.