# Automatic Summarization of Long Documents (Student Abstract)

**Naman Chhibbar**[1], **Jugal Kalita**[2]

[1]Indian Institute of Technology, Hyderabad
[2]University of Colorado, Colorado Springs
naman.iith@gmail.com, jkalita@uccs.edu

**Abstract**

A vast amount of text is added to the internet daily, making utilization and interpretation of textual data complex and cumbersome. As a result, automatic text summarization is crucial for extracting relevant information, saving precious time. Although many transformer models excel in summarization, they are constrained by their input size, preventing them from processing texts longer than their context size. This study introduces several novel algorithms that allow any LLM to efficiently overcome its input size limitation, effectively utilizing its full potential without any architectural modifications. We test our algorithms on texts with more than 70,000 words, and our experiments show a significant increase in BERTScore with competitive ROUGE scores.

## 1 Introduction

Due to the ever-increasing amount of online textual data, document summarization has become crucial for the efficient and accurate extraction of relevant information. Large Language Models (LLMs) based on transformers have shown outstanding abilities in many NLP tasks, including document summarization. Recent developments have demonstrated remarkable improvements in the relevancy and coherence of summaries generated by such LLMs.

However, long document summarization, which involves removing redundancies and makes reading long texts concise and efficient, remains a major challenge. One of the significant limitations in the transformer architecture is limited context size, stemming from the quadratic memory and computational complexity of the attention mechanism (Du et al. 2023). This constraint hinders extracting relevant information from extensive texts where summarization is valuable to overcome the time, effort, and interpretative issues posed by complex and large documents.

We experiment with novel approaches to address the input size limitations of transformers. The methods introduced do not include any architectural modifications to the model used for summarization and can be incorporated into any existing summarization pipeline. We believe that these methods can effectively utilize the full potential of any existing LLM by capturing information from crucial aspects of the document. Although our experiments focus on summarization, we hypothesize that our methods can be applied to NLP tasks that require processing long texts.



## 2 Problem Statement

Our goal is to distill a long document such that it fits within the context size of the model while retaining important information. In our experiments, we use documents with lengths up to **seventeen times** the context size of the model and aim to reduce the summary length to about 400 words or less, preserving maximal information and coherence.

## 3 Methodology

In this section, we introduce the three algorithms used in our experiments. Two of the three algorithms start by segmenting the text into smaller, contiguous, and exhaustive parts. This is done by using a sentence tokenizer to separate sentences and then grouping them to form the segments.

### 3.1 Central Truncation

In this method, we truncate the text from the middle; that is, we keep parts from the start and the end of the document. This time-efficient method is an adaptation of Worsham and Kalita (2018) and Sun et al. (2019), both of which pertain to long text classification. We introduce a new hyperparameter to control the fraction of the text to be taken from the head.

### 3.2 Document Skimming

Our second method is inspired by the speed reading strategy called skimming (Dhillon, Herman, and Syafryadin 2020), which involves skipping over less important parts of the text for efficiency. This method starts by segmenting the document into smaller segments and then uniformly samples the segments, meaning each segment has an equal probability of being selected. This ensures that the model is exposed to all parts of the document while preserving efficiency. We also experiment with removing redundant segments before and after sampling to prevent the model from repeating itself. While removing redundant segments before sampling requires processing the whole document, it ensures better utilization of the context size of the model. Whereas removing redundant segments after sampling is computationally efficient, it may not use the context size to its full extent. To alleviate this, we over-sample the segments beforehand.

### 3.3 Summarization with Keyword Extraction

The last method is based on extracting keywords from the text to help choose the segments intelligently. We use LDA (Blei, Ng, and Jordan 2003) with a single topic to extract topic words (keywords) from the text. These keywords are concatenated with space as a delimiter to form a single sentence. This sentence is compared against the segments using a sentence transformer using cosine similarity to gain scores for each segment. Segments with the highest scores are selected for summarization.

## 4 Experiments

We conduct our experiments on the GovReport (Huang et al. 2021) and the BigPatent (Sharma, Li, and Wang 2019) datasets with maximum word counts of 73,815 and 71,027, respectively. We use ROUGE-1 (R-1), ROUGE-2 (R-2), and ROUGE-L (R-L) scores (Lin 2004) and BERTScore (Zhang et al. 2019) to evaluate the generated summaries. Our baselines include: Unlimiformer (Bertsch et al. 2023) in which the decoder only attends to the tokens picked by k-Nearest-Neighbour tokens in the input, Hepos (Huang et al. 2021) in which the decoder only attends to $n/s_h$ tokens in the input, where $n$ is the input length and $s_h$ is the number of decoder heads, PEGASUS-X (Phang, Zhao, and Liu 2022) with staggered block-local attention, LLaMA-7B (Chen, Cong, and Lv 2022) with positional interpolation, and BigBird-Pegasus (Zaheer et al. 2020). The models we used for summarization are BART (Lewis et al. 2020), LongT5 (Guo et al. 2021), and GPT-3.5 Turbo (Brown et al. 2020). Context sizes of the models are provided in parentheses in the tables.

We could not obtain BERTScores for some baselines due to code unavailability or computational limitations.

| Model | R-1 | R-2 | R-L | BERTScore |
|---|---|---|---|---|
| BART /w Unlimiformer (1,024) | 53.4 | 22.5 | 22.5 | 66.0 |
| PRIMERA w/ Unlimiformer (4,096) | 56.5 | 24.8 | 26.3 | 67.7 |
| Hepos (10,240) | 51.34 | 19.09 | **48.73** | - |
| PEGASUS-X /w Staggered Block-Local Attention (16k) | 60.3 | **30.0** | 31.5 | - |
| LLaMA-7B /w Positional Interpolation (15k) | 60.0 | 28.0 | 29.5 | - |
| Summarization /w Extraction /w GPT-3.5 Turbo (4,096) | **61.99** | 18.52 | 38.46 | **86.20** |
| Central truncation /w LongT5 (4,096) | 46.20 | 4.38 | 38.27 | 82.19 |
| Skimming /w post-sampling removal /w LongT5 (4,096) | 46.76 | 4.56 | 39.61 | 81.96 |

Table 1: Automatic evaluation on GovReport dataset. The best metric in each category is highlighted in **bold**.

## 5 Future Work

We find that segmentation is a crucial step in the pipeline and can influence the output summary significantly; ensuring the uniformity of the length of the segments while preserving coherence is essential. We encourage future work to experiment with different segmenters. Future work can also focus on extending the "Summarization with Keyword Extraction" method. One possibility is experimenting with different ways of using keywords and extraction algorithms.

| Model | R-1 | R-2 | R-L | BERTScore |
|---|---|---|---|---|
| BigBird-Pegasus (16k) | **60.64** | **42.46** | **50.01** | - |
| Skimming w/ pre-sampling removal w/ GPT-3.5 Turbo (4,096) | 27.40 | 3.31 | 21.25 | **82.62** |
| Central truncation w/ GPT-3.5 Turbo (4,096) | 27.77 | 3.09 | 20.56 | 82.57 |
| Skimming w/ post-sampling removal w/ GPT-3.5 Turbo (4,096) | 26.16 | 2.13 | 20.21 | 82.40 |

Table 2: Automatic evaluation on BigPatent dataset. The best metric in each category is highlighted in **bold**.

## 6 Conclusion

Our experiments show that "Document Skimming" with post-sampling removal (of redundant segments) performs well and is efficient. The "Central Truncation" method also shows good results, which shows that simple methods can also be effective in long document summarization. "Document Skimming" with pre-sampling removal and "Summarization with Keyword Extraction" achieve the best results but are computationally expensive.

Our experiments show a significant increase in BERTScore compared to Unlimiformer. This shows that our pipelines can efficiently utilize details in long texts. Even though our ROUGE-2 scores are lower than the baselines, ROUGE-1 and ROUGE-L scores are competitive. Since BERTScore is better at capturing semantic similarity, we highlight the use of BERTScore compared to ROUGE scores. Hence, we hypothesize that our pipelines generate better summaries even though ROUGE scores are not the highest. It should also be noted that the models used in our experiments have much smaller context sizes than the baselines, indicating that our algorithms have a greater potential if used with larger models.

## Acknowledgements

All work herein reported is supported by the National Science Foundation under Grant No. 2349452. Any opinion, finding, or conclusion in this study is that of the authors and does not necessarily reflect the views of the National Science Foundation.

## References

Bertsch, A.; Alon, U.; Neubig, G.; and Gormley, M. 2023. Unlimiformer: Long-Range Transformers with Unlimited Length Input. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 35522–35543. Curran Associates, Inc.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null): 993–1022.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell,

A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chen, X.; Cong, P.; and Lv, S. 2022. A long-text classification method of Chinese news based on BERT and CNN. *IEEE Access*, 10: 34046–34057.

Dhillon, B. P. S.; Herman, H.; and Syafryadin, S. 2020. THE EFFECT OF SKIMMING METHOD TO IMPROVE STUDENTS' ABILITY IN READING COMPREHENSION ON NARRATIVE TEXT. *Linguists: Journal Of Linguistics and Language Teaching*, 6(1): 77–88.

Du, J.; Jiang, J.; Zheng, J.; Zhang, H.; Huang, D.; and Lu, Y. 2023. Improving Computation and Memory Efficiency for Real-world Transformer Inference on GPUs. *ACM Trans. Archit. Code Optim.*, 20(4).

Guo, M.; Ainslie, J.; Uthus, D.; Ontanon, S.; Ni, J.; Sung, Y.-H.; and Yang, Y. 2021. LongT5: Efficient text-to-text transformer for long sequences. *arXiv preprint arXiv:2112.07916*.

Huang, L.; Cao, S.; Parulian, N.; Ji, H.; and Wang, L. 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*, 1419–1436. Online: Association for Computational Linguistics.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880. Online: Association for Computational Linguistics.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics.

Phang, J.; Zhao, Y.; and Liu, P. J. 2022. Investigating efficiently extending transformers for long input summarization. *arXiv preprint arXiv:2208.04347*.

Sharma, E.; Li, C.; and Wang, L. 2019. BIGPATENT: A Large-Scale Dataset for Abstractive and Coherent Summarization. In Korhonen, A.; Traum, D.; and Màrquez, L., eds., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2204–2213. Florence, Italy: Association for Computational Linguistics.

Sun, C.; Qiu, X.; Xu, Y.; and Huang, X. 2019. How to fine-tune bert for text classification? In *Chinese computational linguistics: 18th China national conference, CCL 2019, Kunming, China, October 18–20, 2019, proceedings 18*, 194–206. Springer.

Worsham, J.; and Kalita, J. 2018. Genre Identification and the Compositional Effect of Genre in Literature. In Bender, E. M.; Derczynski, L.; and Isabelle, P., eds., *Proceedings of the 27th International Conference on Computational Linguistics*, 1963–1973. Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Zaheer, M.; Guruganesh, G.; Dubey, K. A.; Ainslie, J.; Alberti, C.; Ontanon, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33: 17283–17297.

Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.