Efficient Summarization of Long Documents Using Hybrid Extractive-Abstractive Method

Weichao CHEN[†] Mizuho IWAIHARA[‡]

Graduate School of Information, Production and Systems, Waseda University
2-7 Hibikino, Wakamatsu-ku, Kitakyushu, Fukuoka, 808-0135 Japan
E-mail: †vico_chen@akane.waseda.jp, ‡iwaihara@waseda.jp

Abstract Automatic summarizers based on pre-trained language models (PLMs) have achieved great success in the field of short text summarization. However, the complexity of the self-attention mechanism on which PLMs rely grows quadratically with the input length, thus limiting its application to long document summarization. As a solution, we investigate an efficient hybrid long document summarization method that enables compression and summarization of original long documents by combining content selection and abstractive summarization. Long documents will first be extracted for salient sentences and reorganized into compressed documents, the length of which is within the limits of the PLMs, and finally generate the final summary by abstractive summarization. Experiments show that the proposed method can achieve good summarization performance for general long documents in single-GPU or low-resource scenarios.

Keyword Natural language processing, Long document summarization, Pretrained language models

1. Introduction

A summary is a concise and comprehensive version of the original document. A summary can help readers quickly understand the main content of the whole article. The task of generating a summary from the original document is called text summarization, which can be further divided into extractive and abstractive summarizations [1]. Extractive summarization methods, such as text ranking and clustering, work by selecting and compiling parts of the original text, while abstractive summarization generates a new summary that captures the main ideas and concepts in the original text, where passages not appearing in the original document may be generated. Abstractive summarization has the advantage of smoothness and fluency, while keeping the content information, so our discussion will mainly focus on abstractive summarization [6].

Typical abstractive summarization algorithms use seq2seq-based models such as transformers. These algorithms have had great success in the field of short text summarization. However, the self-attention mechanism of transformer-based models has a quadratic level of space complexity, leading to a dramatic increase in the computational consumption of these models as the number of input words increases. So, usually these models have a limit on the number of input tokens, such as 1024 tokens limit in BART [10]. For relatively long documents, such as press releases, scientific literature, specialized reports, they often have lengths that far exceed these limits, making it impossible to use the entire content as input of these models. In most cases we simply truncate long documents to a limited length for input into these models. This approach is usually effective, but obviously will lose semantic information of the truncated part, so the results obtained by this approach are not comprehensive [9].

Existing research has focused on adapting PLMs to long documents. The main long document adaptation methods can be broadly divided into two categories: Efficient self-attention mechanisms and hybrid summarization.

Efficient self-attention. In terms of reason analysis, pre-trained language models cannot be applied to long documents mainly because of the computational cost of full self-attention in the transformer structure. The most straightforward solution is to improve the self-attention mechanism to make it more efficient. Currently, the most popular approach is to replace the full attention with sparse attention mechanisms such as local attention and sliding windows, which allows to focus on specific parts of the input while also reducing computational complexity. Models that use this approach include Longformer [3] and BigBird [14]. For example, Longformer incorporates sliding window attention, dilated sliding windows, and global + sliding windows, which enables the processing of long documents with more than 4096 tokens in length. These approaches have shown reasonable performance on long document summarization tasks.

Hybrid summarization (content selection + abstractive summarization). Hybrid summarization is another important approach for long document summarization that combines content selection and abstractive summarization. This approach does not change the structure of the original pre-trained language model (PLM); instead, it solves the long document problem by changing the input of the PLMs. In this approach, long documents are first processed through a content selection process, where the most beneficial sentences or paragraphs for generating a summary are selected and reorganized into a new, compressed document that is long enough for downstream summarization within the limits of the PLMs. For example, Bajaj et al. [2] investigated a combined BERT [4] and BART model for long document summarization, using BERT as a salient sentence classifier to filter the source document sentences and BART as a summarizer for the summarization task. The main advantage of this approach is that it preserves the semantic information while being efficient.

Despite the recent efforts on efficient attention mechanisms, there are still potential limitations to their use in summarizing long documents. For example, the reduced complexity of these mechanisms may come at the cost of summary quality to the original document. Additionally, models using efficient attention mechanisms, such as Longformer and BigBird, may still struggle with high resource consumption and difficulty in capturing the overall context of the document. In contrast, the hybrid model is much lighter and more reliable. An example is the two-stage model, CogLTX [5] proposed by Ding et al. which achieves SOTA results on long document classification and Q&A tasks by using key sentence selection and downstream models. This approach can be applied not only to the above domains but also provide a paradigm for long document summarization.

Based on the successful precedent of CogLTX for long document classification and Q&A tasks, in this paper, we further explore its application to long document summarization. Our extension of CogLTX to long document summarization can be summarized in 3 parts —
(1) we utilize BART to process the downstream summarization task in place of BERT;
(2) a compression mechanism based on the combination of similarity scoring and latent intervention is introduced;
(3) more initialization methods specific to the summarization task are utilized in the model.

In summary, the main contribution of this paper is to propose a hybrid summarization model, which combines extractive and abstractive summarizations by conducting co-training of the content selection model BERT and the abstractive summarization model BART. From the results, we can conclude that this model has good summarization results for both general long documents and extreme long documents. Compared to BigBird and Longformer, which rely on GPU clusters, our model effect has a 1.5 ROUGE scores improvement and outperforms the other models in BERTScore. Impressively, it has low resource consumption for processing documents of any length.

2. Related Work

Long document summarization using hybrid models is an emerging research topic, which is highly related to abstractive summarization and extractive content selection methods such as sentence similarity calculation.

2.1 Bi-Encoder & Cross-Encoder & SimCSE

In content selection using extractive models, sentence pair similarity calculation is a popular method. Traditional methods include Tf-Idf and TextRank, but more recent deep learning-based methods such as Bi-Encoder and Cross-Encoder tend to be more accurate for similarity calculation.

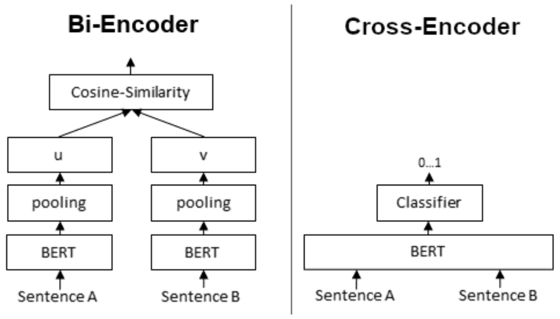

Figure 1. Structures of Bi-Encoder and Cross-Encoder

Bi-Encoder is a simple encoder architecture using BERT with separate encoders but shared parameters for input and output sequences, while Cross-Encoder combines the sequences into a single sequence representation. The key difference between these architectures is how the pretrained language model is finetuned, which can affect model performance and properties [13].

SimCSE [7] is particularly effective for learning sentence embeddings that capture semantic similarity, as demonstrated by its SOTA performance on a variety of benchmark tasks, including semantic textual similarity, natural language inference, and text classification. One of the advantages of SimCSE is that it does not require any labeled data and can be trained on large amounts of unlabeled text. It also has a simple architecture that can be trained efficiently on a single GPU.

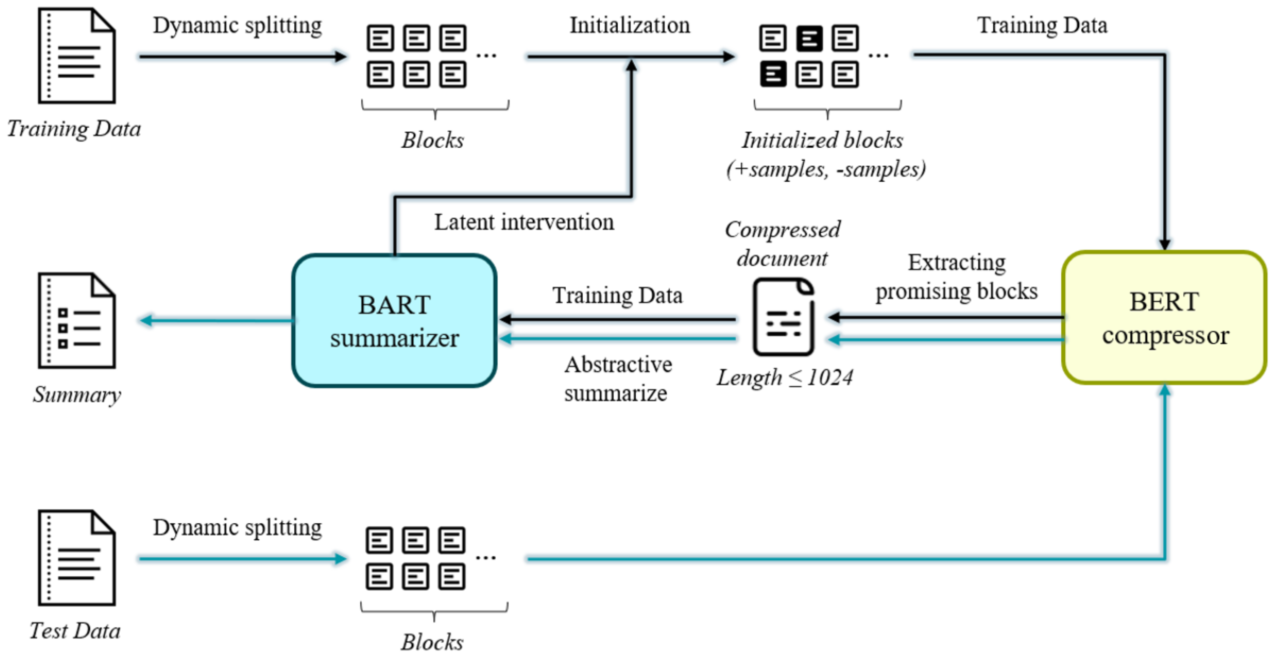

Figure 2. Overall structure of our proposed model

2.2 BART

BART (Bidirectional Encoder Representations from Transformers) is a sequence-to-sequence language model that has been used for tasks such as text generation, translation, and summarization. It is a transformer-based model that uses attention mechanisms to process long input sequences and generate output sequences. It is trained on large amounts of data using an auto-regressive language modeling objective.

The architecture of BART is a combination of BERT as the encoder and GPT [12] as the decoder which allows capturing long-range dependencies and contextual information effectively. This has been shown by its state-of-the-art results on a variety of NLP tasks such as question answering, and language understanding. BART is highly flexible and can be fine-tuned for different tasks and languages.

2.3 CogLTX

CogLTX [5] is a two-stage model that partitions long documents into blocks, scores each block with a judge model, and then combines the high-scoring blocks for training downstream tasks. The model utilizes two critical mechanisms, one is MemRecall, which works by splitting documents into blocks, connecting each block with available key information, and obtaining the average similarity score of the blocks by a scoring machine, then "forgetting" the blocks that did not score well, and repeating the process multiple times to complete multi-step reasoning. The second one is the joint training of two BERTs, that is why this model has two-stage processing. In summary, this model has shown outstanding performance on classification and Q&A tasks over long documents, that inspire us to think about the possibilities on long document summarization task.

3. Methodology

Figure 2 shows the overall structure of our proposed model. It is important to emphasize that our model discards the notion of "natural guidance" in CogLTX because there is no natural prompt for summarization task. We use a combination of similarity scoring and latent intervention to implement an unsupervised process. More importantly, we have introduced BART and utilized a new compression mechanism to expand the compression size to 1024. The main process of the model is as follows.

For the training process, long documents are first dynamically split into blocks that represent the minimum granularity of compression. These blocks are then initialized to determine the initial relevance. The positive and negative sample blocks based on the relevance are used as training data to train the BERT scorer, so that we can obtain the score of each block of the original long document. Based on these scores, it is then possible to determine which blocks in the source document are the most promising, and the highest scoring blocks are selected and reorganized in their order in the original document to produce compressed documents that are shorter than 1024 -- suitable to be processed directly by BART. The compressed documents are then used as input to the downstream summarization task for normal abstractive summarization training. Finally, through a "latent intervention" method, the blocks of the compressed document are again checked to confirm that they are useful for generating summaries and reducing the loss of BART summarizer. The results of the latent intervention are used as feedback for the next epoch training.

For the inference process, the long document is similarly dynamically split into blocks. These blocks are scored by BERT through a dynamic scoring method. The top-K blocks with the highest scores are selected and reorganized into compressed documents in their order in the document. The compressed document is passed to the BART summarizer to produce the final summary.

3.1 Dynamic splitting

Dynamic splitting is a method for dividing long documents with smaller granularity. The length limit L_b of the block determines the minimum granularity of the model input; too small a block may cause semantic information to be fragmented and scattered, while too large a block may cause detailed semantic information to be lost. This method is based on the cost of text splitting. For a long document, the document is first segmented based on punctuation, and if a piece of text exceeds L_b, it is truncated and segmented, resulting in a significant cost. Initial segmentation has variable block lengths, a better approach is to dynamically aggregate the smaller blocks again, ensuring that there is the smallest possible sum of costs without the aggregated blocks exceeding L_b in length. In this way we can obtain the most comprehensive information about the text block.

3.2 Initialization

Initialization represents the pre-processing of blocks split from a long document with the purpose of determining the relevance of the blocks before the training process begins. The relevance reflects the importance of the block in the long document and represents the degree of influence of this block on the summary's generation. The simplest and most efficient way to determine the relevance of a block is by computing the semantic similarity between the target summary and each block.

Unlike traditional word frequency-based methods such as Tf-Idf, the deep learning-based Cross-Encoder, Bi-Encoder and SimCSE methods have the advantage of generalizing semantic information more accurately, making them better suited for sentence pair similarity calculation. In our proposed model, we use each of them to compute the similarity between each block of a long document and the corresponding summary. The blocks with the highest similarity are considered as "relevant" while the rest are "irrelevant" by default.

3.3 BERT and BART training

After obtaining the initialized positive and negative sample blocks, we can construct the dataset used to train BERT. Here BERT serves as a classifier to score the blocks. To ensure the integrity of the learning content, we use a dataset that consists of short spans randomly extracted from long documents, along with other spans made up of positive and negative samples.

After collecting BERT scores for each block of the long document, the highest scoring blocks are collected, reorganized in their original order within the long document, which is referred to as document compression in this paper and represents the end of the first stage. The length of the compressed document is less than the maximum length of 1024 limited by BART, but it contains the best information of the original document, so it can be used to perform the summarization task on behalf of the original document. The second stage is fine-tuning BART using the compressed document set.

3.4 Latent intervention

Blocks that are initially marked as "relevant" do not always maintain their status, in fact, they can be pulled down after each training epoch. This "rechecking" process is called "latent intervention". Latent intervention works according to the contribution of the block to the generated summary. Specifically, if a block in a compressed document is discarded causing an increase in the loss of the BART summarizer, this means that we better not lose it and it is relevant. Conversely, if a block in a compressed document is discarded causing a decrease in the loss of the BART summarizer, this means that the block played a bad role in the generation of the summary, and it is probably irrelevant. This allows us to reposition the relevance of these blocks based on the results of the latent intervention in order to correct our initial judgments and better help the training of the next epoch. This idea is known as unsupervised training in CogLTX.

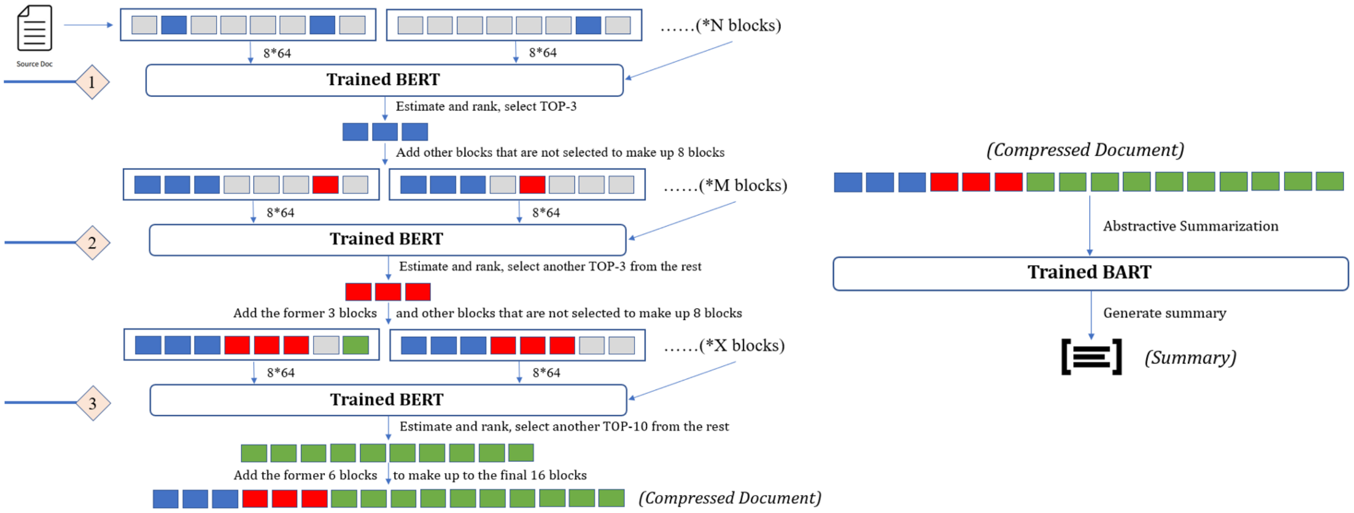
Source Doc
……(*N blocks)
8*64
8*64
1
Trained BERT
Estimate and rank, select TOP-3
Add other blocks that are not selected to make up 8 blocks
……(*M blocks)
8*64
8*64
2
Trained BERT
Estimate and rank, select another TOP-3 from the rest
Add the former 3 blocks
and other blocks that are not selected to make up 8 blocks
……(*X blocks)
8*64
8*64
3
Trained BERT
Estimate and rank, select another TOP-10 from the rest
Add the former 6 blocks
to make up to the final 16 blocks
(Compressed Document)
(Compressed Document)
Abstractive Summarization
Trained BART
Generate summary
(Summary)

Figure 3. The inference process of our proposed model.

3.5 Inference

With a trained BERT compressor and BART summarizer, long documents can be applied the intended compression process and summarization process after being split into blocks. As shown in Figure 3, the compression process is based on a multi-layer dynamic compression mechanism. Specifically, the number of blocks contained in the compressed document is limited to a maximum of N_b due to the limitation of the BART input, where N_b also represents the size of the working memory. It is well known that in the human brain, the data in working memory can be cleared, updated, or added at any time. Multi-layer dynamic compression is based on such a principle. The compressed document is filled step by step through multiple memory iteration cycles, each of which dynamically updates and retains a certain number of the most valuable memories based on the prediction scores. Only three cycles are demonstrated in the figure, with the number of memory blocks retained in each cycle being 3, 3, and 10, corresponding to the blue, red, and green blocks in the figure, respectively.

The abstractive summarization process is relatively simple, and the length of the compressed document obtained by multi-layer dynamic compression is kept within the maximum limit of BART, which can directly generate a target summary corresponding to the original long document. This is shown in the right part of Figure 3.

4. Datasets and Experiments

A good long document summarization model should be adaptable to documents of various lengths. In order to fully evaluate the model's summarization performance, we selected datasets of different length classes for training.

Table 1. A comparison of Pubmed, arXiv and GovReport datasets. $\#(D,S)$ means the number of document-summary pairs, Avg.$|D|$ and Avg.$|T|$ represents the average length of documents and summaries in each dataset.

Dataset	$\#(\boldsymbol{D},\boldsymbol{S})$	**Avg.**$|\boldsymbol{D}|$	**Avg.**$|\boldsymbol{S}|$
Pubmed	133215	3000	215
arXiv	215913	6000	300
GovReport	19463	9000	500

PubMed is a database of medical literature, including abstracts and full-text articles from a wide range of medical journals. The articles in this dataset may be suitable for long document summarization in the field of medicine. arXiv contains electronic preprints of scientific papers in a variety of fields, including mathematics, computer science, physics, biology, finance, and statistics. These papers can be quite long and suitable for long document summarization in the field of scientific research. GovReport [8] is a database of government reports, including reports from federal agencies, congressional committees, and independent organizations. These reports can vary in length and suitable for long document summarization in the field of government research and policy. In summary, these three datasets have their own specific domains. They are also different in terms of the length that we care most about.

4.1 Evaluation Metrics

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a widely used evaluation metric for summarization tasks. It compares an automatically generated summary to a reference summary and calculates the overlap between the two. There are several variations of ROUGE, including ROUGE-N, which measures the overlap between the summary and the reference in terms of n-grams (sequences of n words), and ROUGE-L, which measures the longest common subsequence between the summary and the reference. ROUGE scores are typically used to evaluate the performance of summarization systems and to compare the quality of different summaries. However, ROUGE scores are calculated based on word frequencies, and it has its limitations in reflecting the true semantic content and information coherence. Therefore, we also utilize BERTScore [16] which can better compare and evaluate summaries at the semantic level.

BERTScore is a metric for evaluating the quality of text generation tasks, including summarization. It is based on the BERT language model and measures the similarity between the generated text and the reference text using a combination of precision and recall. BERTScore is designed to be less sensitive to the length of the generated text than other evaluation metrics, such as ROUGE, which can be useful when evaluating summaries of different lengths. BERTScore has been shown to be effective at evaluating the quality of summaries and other types of generated text.

4.2 Baseline Models

We chose three types of widely used methods in the field of long document summarization as baseline models for comparison.

The first type of methods is unsupervised extractive summarization methods, which mainly includes the most representative traditional word frequency statistics method Tf-Idf, as well as classical BERT-based extractive summarization methods. The summaries generated by these methods are derived from the sentence reorganization extracted from the source documents.

The second type is supervised abstractive end-to-end methods based on BART or PEGASUS [15]. This type of approach differs from the two-stage model proposed in our paper, which emphasizes end-to-end models. Models using efficient self-attention such as Longformer and Bigbird are also included in this category.

The third type is hybrid summarization based on content selection and abstractive summarization, where LoBART [11] is a recognized representative work in this area, and our work also belongs to this type. We will focus on this type of comparison.

4.3 Experiment Settings

The experiments were implemented on a standalone environment with a GeForce RTX 3090. Block size L_b is set to 64 and the corresponding number N_b of blocks in working memory is 16. For SimCSE initialization, we did unsupervised training of SimCSE by selecting domain-specific datasets to extract sentences as training data. We chose RoBERTa, a variant of BERT, as the model for the compressor. The batch size of the compressor was set to 32 and the learning rate was set to 2e-5. Since the training dataset of RoBERTa needed to be sufficiently large to contain every block in the original long document, we used data enhancement by sampling the same document multiple times for the training data. The number of samples was set to 5. The batch size of the summarizer BART was set to 2 and the learning rate was set at a relatively small 1e-5. In the latent intervention stage, the relevance of text blocks was rechecked based on a threshold value that increased the relevance of blocks only when the BART loss difference exceeded $t_{up} = 0.15$ and decreased the relevance of blocks when it was less than $t_{down} = -0.07$. The training of the compressor and summarizer was set to 6 epochs.

In the inference stage, we set three layers of dynamic compression with the number of reserved blocks in each layer being 3, 3, and 10, respectively.

5. Experimental Results

For different initialization strategies, we conducted comparative experiments, and the results are shown in Table 2. Here we also introduce Rouge Similarity, which uses the ROUGE scores of the sentence pairs as their similarity scores. The experiment is based on the partial GovReport dataset with 5 epochs training. It can be seen that the unsupervised trained SimCSE shows a slight advantage in the final summarization results, so in the following experiments, we use the unsupervised trained SimCSE as the default initialization strategy.

Table 2. different initialization strategies comparison

Model	R-1	R-2	R-L
Rouge similarity	51.44	18.12	25.53
Bi-Encoder	50.58	18.01	24.69
Cross-Encoder	50.75	18.16	25.06
SimCSE (trained)	**51.82**	**18.22**	**27.33**

Table 3. Experimental results and comparison with other baseline models

Datasets	Pubmed				arXiv				GovReport			
	R-1	**R-2**	**R-L**	**BERTScore**	**R-1**	**R-2**	**R-L**	**BERTScore**	**R-1**	**R-2**	**R-L**	**BERTScore**
Unsupervised Extractive												
Tf-Idf	/	/	/	/	34.4	9.5	28.5	0.822	/	/	/	/
BERT	/	/	/	/	35.1	9.8	29.0	0.819	/	/	/	/
Supervised Abstractive – End-to-End												
BART-only	42.1	17.5	37.1	/	41.3	15.3	36.8	0.846	52.8	19.1	49.9	/
Longformer	46.9	20.2	42.8	/	46.7	19.6	41.8	0.865	/	/	/	/
BigBird	46.3	20.6	42.3	/	46.2	19.0	41.5	0.855	**56.8**	**22.6**	**53.8**	/
Supervised Abstractive – Hybrid summarization												
LoBART	**48.0**	**20.9**	**43.5**	/	**48.6**	**20.1**	**42.2**	0.845	/	/	/	/
Our Proposed Model												
Ours	46.8	20.3	35.7	0.863	47.9	19.5	35.6	**0.868**	55.3	21.9	41.6	0.865

Table 3 shows the experimental results of our proposed model on three different long document datasets. As seen in Table 3, we observed a 0.5 R-1 improvement when compared to BigBird on the Pubmed dataset, and this result is also close to the result of Longformer. For the arXiv dataset, our model has a relatively 1.7 / 1.2 R-1 improvement when compared to BigBird and Longformer. For the GovReport dataset which has the longest average length, our model also achieves a good result in R-1 and R-2 scores, it is slightly lower than the result of BigBird. For the Pubmed and arXiv datasets, LoBART has a relatively high ROUGE score, but for the arXiv dataset, our model has the highest BERTScore.

However, our model has a relatively poor performance in terms of R-L scores compared to other models, as evidenced by the fact that our model generates summaries that have a small common subsequence with the target summary. This can be explained by the characteristics of the content selection mechanism. Content selection can pick out the most salient sentences in the original document, however, the selected sentences are often discrete, so it will cause the reduction of R-L. The decrease of R-L corresponds to the increase of BERTScore, and how to ensure the balance between the two is also a key point to be considered in our future work.

6. Conclusion and Future Work

For the long document problem that is often encountered when using traditional pre-trained language models for text summarization, we propose a hybrid long document summarization method. Compared with general extractive summarization models and efficient self-attention based long document adaptation mechanisms, our proposed model can achieve a balance in resource consumption and model performance, which is especially important for implementing long document summarization in single-GPU or low resource scenarios. Also, as an advantage of the hybrid summarization model, our model has an impressive performance in handling documents of any length, and even outperforms SOTA on specific domains. As an additional result, the summaries we generated are more closely aligned with human judgment (See Appendix Table 5), as demonstrated by the fact that our model obtained a higher BERTScore than other baseline models.

For future work, we tend to further improve the performance of the model, especially when dealing with extremely long documents. As a further addition to the CogLTX series of studies, we are hopeful to implement model migration and corresponding long-document adaptation in other domains including keyphrase extraction and generation, multi-label text classification, etc., to address the long-document problem faced by these domains as well.

References

[1] Abualigah, Laith, et al. "Text summarization: a brief review." Recent Advances in NLP: the case of Arabic language (2020): 1-15.

[2] Bajaj, Ahsaas, et al. "Long Document Summarization in a Low Resource Setting using Pretrained Language Models." arXiv preprint arXiv:2103.00751 (2021).

[3] Beltagy, Iz, Matthew E. Peters, and Arman Cohan. "Longformer: The long-document transformer." arXiv preprint arXiv:2004.05150 (2020).

[4] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[5] Ding, Ming, et al. "Cogltx: Applying bert to long texts." Advances in Neural Information Processing Systems 33 (2020): 12792-12804.

[6] El-Kassas, Wafaa S., et al. "Automatic text summarization: A comprehensive survey." Expert systems with applications 165 (2021): 113679.

[7] Gao, Tianyu, Xingcheng Yao, and Danqi Chen. "Simcse: Simple contrastive learning of sentence embeddings." arXiv preprint arXiv:2104.08821 (2021).

[8] Huang, Luyang, et al. "Efficient attentions for long document summarization." arXiv preprint arXiv:2104.02112 (2021).

[9] Koh, Huan Yee, et al. "An Empirical Survey on Long Document Summarization: Datasets, Models, and Metrics." ACM computing surveys 55.8 (2022): 1-35.

[10] Lewis, Mike, et al. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." arXiv preprint arXiv:1910.13461 (2019).

[11] Manakul, Potsawee, and Mark JF Gales. "Long-span summarization via local attention and content selection." arXiv preprint arXiv:2105.03801 (2021).

[12] Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

[13] Reimers, Nils, and Iryna Gurevych. "Sentence-bert: Sentence embeddings using siamese bert-networks." arXiv preprint arXiv:1908.10084 (2019).

[14] Zaheer, Manzil, et al. "Big bird: Transformers for longer sequences." Advances in Neural Information Processing Systems 33 (2020): 17283-17297.

[15] Zhang, Jingqing, et al. "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization." International Conference on Machine Learning. PMLR, 2020.

[16] Zhang, Tianyi, et al. "Bertscore: Evaluating text generation with bert." arXiv preprint arXiv:1904.09675 (2019).

Appendix

Table 4. Sample summaries for arXiv dataset

Target Summaries	Generated Summaries
additive models play an important role in semiparametric statistics . this paper gives learning rates for regularized kernel based methods for additive models . these learning rates compare favourably in particular in high dimensions to recent results on optimal learning rates for purely nonparametric regularized kernel based quantile regression using the gaussian radial basis function kernel , provided the assumption of an additive model is valid . additionally , a concrete example is presented to show that a gaussian function depending only on one variable lies in a reproducing kernel hilbert space generated by an additive gaussian kernel , but does not belong to the reproducing kernel hilbert space generated by the multivariate gaussian kernel of the same variance . * key words and phrases . * additive model , kernel , quantile regression , semiparametric , rate of convergence , support vector machine .	This paper discusses the importance of additive models in semiparametric statistics and presents learning rates for regularized kernel-based methods for these models. The learning rates are compared favorably, particularly in high dimensions, to recent results on optimal learning rates for purely nonparametric regularized kernel-based quantile regression using the Gaussian radial basis function kernel, provided the assumption of an additive model is valid. The paper also provides a concrete example to demonstrate that a Gaussian function depending only on one variable lies in a reproducing kernel Hilbert space generated by an additive Gaussian kernel but does not belong to the reproducing kernel Hilbert space generated by the multivariate Gaussian kernel of the same variance. The key terms and phrases discussed in the paper include additive model, kernel, quantile regression, semiparametric, rate of convergence, and support vector machine.
the effect of a random phase diffuser on fluctuations of laser light (scintillations) is studied . not only spatial but also temporal phase variations introduced by the phase diffuser are analyzed . the explicit dependence of the scintillation index on finite - time phase variations is obtained for long propagation paths . it is shown that for large amplitudes of phase fluctuations , a finite - time effect decreases the ability of phase diffuser to suppress the scintillations .	The paper examines the impact of a random phase diffuser on fluctuations of laser light, specifically scintillations. Both spatial and temporal phase variations caused by the diffuser are analyzed. The paper derives the explicit relationship between the scintillation index and finite-time phase variations for long propagation paths. The study shows that for large amplitudes of phase fluctuations, a finite-time effect reduces the ability of the phase diffuser to suppress the scintillations.