Leveraging Locality in Abstractive Text Summarization

Yixin Liu[1], Ansong Ni[1], Linyong Nan[1], Budhaditya Deb[2], Chenguang Zhu[2], Ahmed H. Awadallah[2], Dragomir Radev[1]
[1]Yale University, [2]Microsoft Research
{yixin.liu, ansong.ni, linyong.nan, dragomir.radev}@yale.edu
{Budha.Deb, chezhu, hassanam}@microsoft.com

Abstract

Despite the successes of neural attention models for natural language generation tasks, the quadratic memory complexity of the self-attention module with respect to the input length hinders their applications in long text summarization. Instead of designing more efficient attention modules, we approach this problem by investigating if models with a *restricted* context can have competitive performance compared with the memory-efficient attention models that maintain a global context by treating the input as an entire sequence. Our model is applied to individual *pages*, which contain parts of inputs grouped by the *principle of locality*, during both encoding and decoding stages. We empirically investigated three kinds of localities in text summarization at different levels, ranging from sentences to documents. Our experimental results show that our model can have better performance compared with strong baseline models with efficient attention modules, and our analysis provides further insights of our locality-aware modeling strategy.

1 Introduction

Neural abstractive summarization (Rush et al., 2015; Nallapati et al., 2016) is mainly formulated as a sequence-to-sequence (Sutskever et al., 2014) (Seq2Seq) problem. Neural attention models, e.g., Transformers (Vaswani et al., 2017), have been widely used for Seq2Seq tasks, allowing effective modeling of various dependencies in input and output sequences. However, the *self-attention* module in such models introduces a quadratic memory growth with respect to the input sequence length. Consequently, for long-text summarization datasets,[1] recent works (Beltagy et al., 2020; Kitaev et al., 2020; Zaheer et al., 2020) have explored using *efficient attention* to reduce the memory footprint while still maintaining the same **global context** of a full-attention model – every input token can receive information from all the other input tokens. However, efficient attention is an approximation of full attention and can have inferior performance compared with its counterpart (Kitaev et al., 2020). To investigate an alternative memory-efficient modeling approach, we argue that models with a *restricted* context, where each token only receives a subset of tokens as its context during the entire computation, can be competitive to efficient attention models if they can effectively leverage **localities** in the text summarization task.

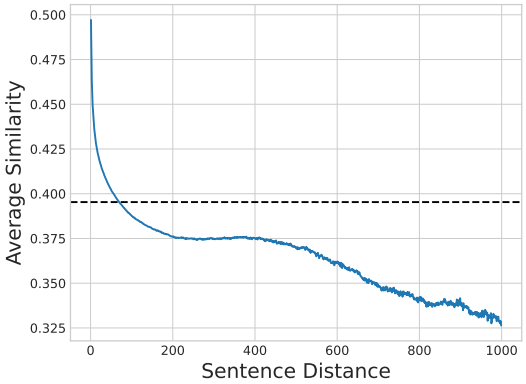

Figure 1: Intrinsic spatial locality in the arXiv dataset. The X-axis represents the distance of two sentences in source documents measured by the difference of their locations (indexes). Y-axis represents the average semantic similarity calculated by the cosine similarity between sentence embeddings, which are generated by a pre-trained sentence embedding model (Gao et al., 2021). The dash line shows the average similarity.

Locality, or the principle of locality, is one of the fundamental principles of virtual memory systems (Denning, 2005), and exists in a wide range of domains (Koopman et al., 2013; Fonseca et al.,

[1]For example, the average input document length of arXiv dataset (Cohan et al., 2018) is more than 8000 tokens.

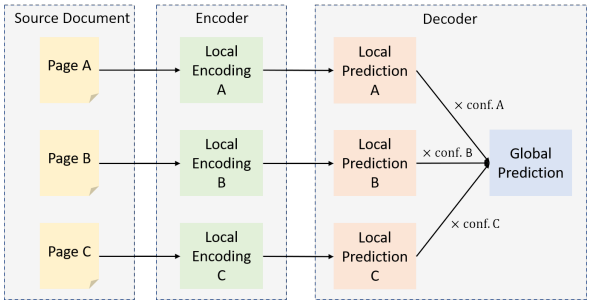

Figure 2: Model architecture. Our model views the source document as a number of *non-overlapping pages*, and the final output is a weighted combination of local predictions on the individual pages.

2003; Zamanian et al., 2015).[2] A famous example of locality is the *spatial locality* in computer memory systems – data units that are stored closely in the disk are likely to be accessed during a short time period by a computer process, therefore it is beneficial to read a block of data as a *page* in the memory instead of reading only one data unit at a time. Such patterns also exist in text summarization task. For example, on the arXiv dataset, we observe an *intrinsic* spatial locality in source documents – two sentences are more semantically similar when their locations in the document are closer (Fig. 1). This observation supports the inductive bias of *window attention* (Beltagy et al., 2020; Zaheer et al., 2020), which allows each token interact with its neighboring tokens within the window size.

We introduce a framework of leveraging localities for text summarization, which aims to reduce the memory complexity of full-attention models while still maintain a competitive performance. Specifically, instead of viewing the input document as an entire sequence, we represent an input document as a number of *pages* which are constructed according to the principle of locality (Fig. 2). Each of these pages is encoded independently by the encoder of our abstractive model, and the decoder makes *local* predictions over each page along with *local* confidence scores of its predictions, which are used to combine the local predictions into final outputs. In this framework, tokens in different pages never directly interact with each other during both encoding and decoding, which highlights the role of **localities** in text summarization task. In contrast, efficient attention models still share the assumption that all tokens in the entire source sequence have to interact with each other because (1) *global tokens* or *overlapping* window attention maintain a *global* context during encoding; (2) the encoder-decoder attention takes the source document embeddings as an entire sequence during decoding.

Using the proposed framework, we are able to investigate several different localities in text summarization task: (1) *spatial* locality or *sequential* locality – neighboring sentences are grouped into the same (non-overlapping) page; (2) *discourse* locality – different sections in a scientific paper may cover different aspects, therefore they are viewed as different pages (Cohan et al., 2018); (3) *document* locality – for multi-document summarization, each document in a document cluster can be viewed as an individual page (Jin and Wan, 2020). Our approach also has other advantages: (1) Unlike most of the efficient attention models, our model can be directly initialized from pre-trained models (e.g. BART (Lewis et al., 2020)) with a small overhead,[3] which requires no further pre-training; (2) It reduces the overall complexity of encoder self-attention to a *linear* relationship with the input document length. We empirically demonstrate that our model has better performance than strong baseline models built upon various efficient-attention modules on several summarization datasets. Furthermore, we conduct detailed analyses on different modeling options for our framework, shedding lights on its broader usages.

2 Preliminaries

Abstractive summarization models aim to generate an appropriate summary of an input document. Given a pair of an input document D and a reference summary S, the standard training algorithm of a neural abstractive summarization model g adopts the cross-entropy loss, which requires the model to predict the next token of the reference summary given the input document and the prefix of the reference summary before the current token:

$$\mathcal{L}_{xent} = -\sum_{i=1}^{l} \log p_{g_\theta}(s_i | D, S_{<i}; \theta), \quad (1)$$

where θ is the trainable parameters of the model g, p_{g_θ} is the predicted probability over the vocabulary, l is the length of the summary S, $\{s_1, \cdots, s_i, \cdots, s_l\}$ are tokens in S, $S_{<i}$ denotes

[2] A formal definition of locality coined by Denning (1980) is: "The concept that a program favors a subset of its segments during extended intervals (phases) is called locality."

[3] We use a single linear layer for predicting the confidence scores based on the decoder hidden states.

the partial reference sequence $\{s_0, \cdots, s_{i-1}\}$ and s_0 is a pre-defined start token.

Encoder-Decoder Model The encoder-decoder model formulates abstractive summarization as a Seq2Seq task,

$$h_i = \text{Decoder}(\text{Encoder}(D), S_{<i}), \quad (2)$$

where h_i is the hidden representation. The generation probability is

$$p_{g_\theta}(\cdot|D, S_{<i}; \theta) = \text{softmax}(L_{vocab}(h_i)), \quad (3)$$

where L_{vocab} is a linear projection layer.

Neural Attentions and Its Limitations Neural attention module is essential to the success of Transformers (Vaswani et al., 2017) and large pre-trained language models (Radford et al., 2019; Lewis et al., 2020; Zhang et al., 2020) for language generation tasks such as machine translation or text summarization. Given a query matrix Q, a key matrix K and a value matrix V, the output of the dot-product attention is:

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T)V. \quad (4)$$

To compute Eq. 4 in a parallel manner, it requires $\mathcal{O}(l_Q \cdot l_K)$ memory space to store the intermediate result of QK^T where l_Q and l_K are the length of Q and K respectively. This becomes a bottleneck of the *self-attention* module for long input documents, where Q, K, V are coming from the same input D, and the space complexity becomes $\mathcal{O}({l_D}^2)$, where l_D is the length of the input document and can be very large (e.g. more than 10000 tokens).

3 Locality-aware Abstractive Text Summarization

To avoid the quadratic growth of memory with respect to the length of the input document in neural attention models, we introduce a different view for modeling the input documents. Specifically, instead of viewing the input document as an entire text sequence, we view it as a series of *non-overlapping pages* with a fixed maximum length:

$$D := \{P_1, \cdots, P_i, \cdots, P_n\}, \quad (5)$$

where P_i is the i-th page and n is the number of pages. We hypothesize that with the **principle of locality**, the abstractive summarizer can make local predictions about the output summary based on *individual* pages without having each input token interact with the entire input document:

$$h_i^{(j)} = \text{Decoder}(\text{Encoder}(P_j), S_{<i}), \quad (6)$$

where $h_i^{(j)}$ is the *local* hidden state of the i-th token of the summary given the j-th page. Apart from the hidden state, we also require the decoder to predict a confidence score of its local prediction:

$$c_{ij} = L_{conf}(h_i^{(j)}), \quad (7)$$

where L_{conf} is a linear layer projecting the hidden state $h_i^{(j)}$ to a scalar. The confidence scores are normalized:

$$\hat{c}_{ij} = \frac{\exp(c_{ij})}{\sum_{k=1}^{n} \exp(c_{ik})}, \quad (8)$$

and used to combine the local hidden states for predicting the final output:

$$p_{g_\theta}(\cdot|D, S_{<i}; \theta) = \text{softmax}(L_{vocab}(\sum_{j=1}^{n} \hat{c}_{ij} \cdot h_i^{(j)})). \quad (9)$$

Fine-tuning from Pre-trained Models Our model can be direct initialized from a pre-trained language model (e.g. BART (Lewis et al., 2020)) except for an additional linear layer L_{cong} (Eq. 7). The cross-entropy loss (Eq. 1) with label smoothing (Szegedy et al., 2016) is used for training.

Space Complexity Our model has a linear space complexity with respect to the length of input documents. Specifically, given a pre-defined maximum page length L_{page}, a document of which the length is l_D will be split into at most $\lceil \frac{l_D}{L_{page}} \rceil$ pages. The space complexity of the encoder self-attention for one page is $\mathcal{O}(L_{page}^2)$, and the complexity for all pages is

$$\mathcal{O}(L_{page}^2 \cdot \lceil \frac{l_D}{L_{page}} \rceil) = \mathcal{O}(L_{page} l_D). \quad (10)$$

When $l_D \gg L_{page}$, the complexity is $\mathcal{O}(l_D)$.[4]

Localities in Abstractive Summarization We mainly explore three types of localities for abstractive summarization, which provide the principles of splitting an input document or document cluster (in the case of multi-document summarization) into different *pages*.

[4] In practice, the page size L_{page} can be large (e.g. 512 tokens). However, we note that sparse attention models can also use window attention with large sizes (e.g. Longformer uses 512 tokens).

(1) **Spatial Locality**: in the most direct form, an input document can be sequentially split into different pages, of which the underlying intuition is that neighboring sentences are likely to focus on the same topic. Under this setting, each document will be equally split in to n_p pages, which is a pre-defined number.
(2) **Discourse Locality**: long documents usually have hierarchical discourse structures, and different discourse units at the same level can have very different focuses. For example, a scientific paper usually have different sections serving different purposes (e.g. introduction, related work, etc.), and this discourse structure can be a useful inductive bias (Cohan et al., 2018). Under this setting, each discourse unit (e.g. a section in a scientific paper) is viewed as a page.
(3) **Document Locality**: for multi-document summarization, we can view each single document in the document cluster as a page. Previous work (Jin and Wan, 2020) has shown that multi-document summarization can benefit from single-document summarization model by first summarizing each document then combining the predictions.

4 Related Work

4.1 Efficient Attention Models

To reduce the quadratic memory growth of neural attention models with respect to the input length, various methods have been proposed, of which the most important and commonly used building blocks are window attention (Beltagy et al., 2020; Zaheer et al., 2020) and low-rank approximation (Liu* et al., 2018; Wang et al., 2020; Peng et al., 2021; Choromanski et al., 2021).

Window attention provides each input token with a restricted context because each token can only receive information from its neighboring tokens that locate in the same window. However, multi-layer models with **overlapping** window attention (Beltagy et al., 2020; Zaheer et al., 2020; Manakul and Gales, 2021; Guo et al., 2021) can still maintain a *global* context as it resembles graph attention networks (Veličković et al., 2018) with a *connected* graph. On the other hand, **non-overlapping** window attentions (local attentions) with *fixed* windows (Liu* et al., 2018; Zhao et al., 2020; Pietruszka et al., 2020) have a *restricted* context as tokens in different windows cannot interact with each other. Instead of using fixed windows throughout the model, using window attentions with **learnable** patterns (Kitaev et al., 2020; Tay et al., 2020; Huang et al., 2021) offer more flexibility and windows can be dynamically constructed at different layers of the model, which allows the model to have a larger context. In addition, head-wise sparse attention (Qiu et al., 2020; Huang et al., 2021) is another method of reducing memory usage while preserving the global context.

Comparing with these methods, our model has a distinct feature in that we maintain a **local** context of the input tokens at both encoding and decoding stages. Zhao et al. (2020) proposed a similar block-wise encoder-decoder attention module which only uses a subset of input tokens (blocks) for a certain decoding stage. However, our method differs from theirs because our model dynamically combines the local predictions based on all the individual *pages* into the final output (Eq. 9).

4.2 Hierarchical Summarization Models

Hierarchical attention (Yang et al., 2016) models aim to utilize the inherent structure of documents as an important inductive bias. For text summarization, Ling and Rush (2017) proposes a coarse-to-fine structure consisting of word-level and chunk-level attentions. Cohan et al. (2018); Xu et al. (2020a); Dong et al. (2021) introduce discourse-aware attentions at the level of document sections or elementary discourse units. Related work also use a similar structure that computes both token-level and sentence-level attentions for abstractive (Rohde et al., 2021), extractive (Xiao and Carenini, 2019; Ruan et al., 2022) and unsupervised (Xu et al., 2020b) summarization.

Hierarchical models have also been widely used for multi-document summarization. The hierarchical attentions can locate at the sentence level (Fabbri et al., 2019), paragraph level (Liu and Lapata, 2019), and document level (Zhang et al., 2018; Jin and Wan, 2020; Jin et al., 2020). Cao and Wang (2022) introduces a hierarchical summarization task which emphasizes the structure of input documents. Ernst et al. (2021) first clusters the propositions of the source documents, then generates summaries based on the proposition clusters.

The multi-stage method of text summarization (Chen and Bansal, 2018; Xu and Durrett, 2019; Pilault et al., 2020) also has a hierarchical structure. In particular, Zhang et al. (2021) first generates a coarse summary for each part of the input document, then further summarizes the generated sum-

Datasets	# Examples			Avg. Tokens	
	Train	Valid	Test	Doc.	Sum.
arXiv	203K	6.4K	6.4K	8154.3	197.8
PubMed	120K	6.7K	6.6K	3983.6	261.3
GovReport	17.5K	973	974	10726.1	681.6
MultiNews	45.0K	5.6K	5.6K	2526.4	277.2

Table 1: Datasets Statistics. We report the average number of tokens generated by the BPE tokenizer (Sennrich et al., 2016) used by BART (Lewis et al., 2020) on the *validation* set. For MultiNews dataset, we report the *sum* of lengths of the individual source document in a document cluster as it is a multi-document dataset.

maries. Mao et al. (2021) first extracts sentences from the source documents, and the generation stage is based on the marginal generation probability conditioned on the selected sentences.

Our method introduces **pages** as a new, unified abstraction for hierarchical models which can be instantiated as sentence clusters, scientific paper sections, and entire documents in the case of multi-document summarization. Furthermore, unlike the previous work, our model emphasizes the role of locality by preventing explicit interactions among different units at the higher levels of the hierarchy.

5 Experiments

5.1 Experimental Settings

Datasets We mainly use four datasets in our experiments. The datasets statistics are in Tab. 1.

`arXiv` and `PubMed` are two scientific paper summarization datasets introduced by Cohan et al. (2018).[5] The abstracts of the scientific papers are used as the summaries of the main contents of those papers.

`GovReport`[6] (Huang et al., 2021) is a long document summarization dataset collected from government reports, which are published by U.S Government Accountability Office and Congressional Research Service.

`MultiNews`[7] (Fabbri et al., 2019) is a multi-document news summarization dataset. Its new articles and human-written summaries are collected from the website newser.com.

Baselines We use the following top-performing models as baselines for comparison.

(1) **LED** (Longformer Encoder-Decoder) (Beltagy et al., 2020) is an encoder-decoder model with sparse encoder self-attention module.
(2) **HEPOS** (Huang et al., 2021) combines both efficient encoder self-attention and encoder-decoder attention in its encoder-decoder architecture.
(3) **PRIMERA** (Xiao et al., 2021) shares the same architecture as **LED**, but has task-specific pre-training for multi-document summarization.
(4) **HAT-BART** (Rohde et al., 2021) is built upon BART (Lewis et al., 2020) while it has additional hierarchical layers for sentence-level interactions. It uses *full* attentions instead of sparse attentions.

Implementation Details We use BART[8] as the backbone of our model, and we initialize our model from a checkpoint pre-trained on CNN/DailyMail (Hermann et al., 2015; Nallapati et al., 2016) dataset except for the linear layer computing the confidence scores (Eq. 7). We use Adam optimizer (Kingma and Ba, 2015) with learning rate scheduling as follows:

$$lr = 2 \times 10^{-3} \min(\text{step}^{-0.5}, \text{step} \cdot \text{warmup}^{-1.5}). \quad (11)$$

warmup is the number of warmup steps, which is set to 10000. step is the number of update steps taken so far. We select the model checkpoints based on their performance on the validation set, which is evaluated by the cross-entropy loss (Eq. 1). Our models are trained on one NVIDIA A6000 GPU, and it takes around 5-25 hours (depending on the size of the dataset) for one training epoch. All models converged in 10 epochs. We use ROUGE (Lin, 2004) as the automatic evaluation metric for performance comparison. More specifically, we report the F1 score of ROUGE-1/2/L in our experiments.

We name our model as **PageSum** for the following experiments.

5.2 Exp-I: Spatial Locality

We first investigate the case of *spatial locality*, where the sentences in the source document are sequentially split into different *pages* with the same number of sentences. The default maximum number of tokens for a page (Eq. 6) in our model is 1024. We set the number of pages to be either 7 or 20 in this experiment, and the maximum number of input tokens is 7168 or 20480 respectively.

We report the model performance[9] in Tab. 2 on `arXiv`, `PubMed`, `GovReport` datasets. We

[5] https://github.com/armancohan/long-summarization
[6] https://github.com/luyang-huang96/LongDocSum
[7] https://github.com/Alex-Fabbri/Multi-News

[8] It contains around 400M parameters.
[9] For fair comparison, apart from the results reported in

System	arXiv			PubMed			GovReport		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
LED* (4096)	44.40	17.94	39.76	-	-	-	-	-	-
LED* (16384)	46.63	19.62	41.83	-	-	-	-	-	-
LED‡ (16384)	48.10	19.78	43.08	46.93	19.88	42.73	59.42	26.53	56.63
HEPOS* (7168)	48.24	20.26	41.78	48.12	21.06	42.72	55.00	21.13	51.67
HEPOS* (10240)	47.87	20.00	41.50	47.93	20.74	42.58	56.86	22.62	53.82
PRIMERA* (4096)	47.60	20.80	42.60	-	-	-	-	-	-
PRIMERA‡ (4096)	47.65	20.76	43.19	-	-	-	-	-	-
HAT-BART* (3072)	46.68	19.07	42.17	**48.36**	**21.43**	37.00	-	-	-
PageSum (7168)	**49.72**	**21.06**	**44.69**	48.24	21.06	**44.26**	59.05	26.37	56.22
PageSum (20480)	-	-	-	-	-	-	**59.91**	**27.20**	**57.07**

Table 2: System performance comparison for *spatial locality*. R-1/2/L are the ROUGE-1/2/L F-1 scores respectively. The numbers in the parentheses after the system names indicate the maximum input length (tokens). *: results reported in the original papers. ‡: results from our own evaluation script (and own checkpoints).

have the following observations. (1) **PageSum** achieves compatible or better ROUGE scores on all three long text summarization datasets compared with the baseline models that leverage sparse or efficient attention modules. (2) On `Pubmed`, **HAT-BART** achieves slightly better performance than **PageSum** while having a much smaller maximum length of input. There can be two reasons for this result. First, HAT-BART uses *full* attentions instead of *sparse* or *local* attentions. Second, the average length of `Pubmed` is relatively short, which restricts the potential benefits of having a longer maximum input length. (3) On `GovReport`, increasing the maximum length of input for **PageSum** helps to improve the model performance.

5.3 Exp-II: Discourse Locality

We use `arXiv` dataset to explore another locality principle – *discourse locality*. Specifically, we view each section of the input document as an individual *page*. The maximum number of tokens for one *page* is still 1024, however, unlike in §5.2, here we allow each example to have different numbers of pages because documents in `arXiv` can have different numbers of sections. We set the maximum number of pages to be 8 in this experiment, which is slightly more than the number in §5.2 to compensate the fact that some sections may have much fewer tokens than 1024. For each page, we concatenate the name of the section and the content together as the input.

System	R-1	R-2	R-L
LED* (4096)	44.40	17.94	39.76
LED* (16384)	46.63	19.62	41.83
LED‡ (16384)	48.10	19.78	43.08
HEPOS* (7168)	48.24	20.26	41.78
HEPOS* (10240)	47.87	20.00	41.50
PRIMERA* (4096)	47.60	20.80	42.60
PRIMERA‡ (4096)	47.65	20.76	43.15
HAT-BART* (3072)	46.68	19.07	42.17
PageSum-Spatial (7168)	49.72	21.06	44.69
PageSum-Discourse (8192)	**49.84**	**21.19**	**44.89**

Table 3: System performance comparison for *discourse locality* on `arXiv`. R-1/2/L are the ROUGE-1/2/L F-1 scores respectively. The numbers in the parentheses after the system names indicate the maximum input length (tokens). **PageSum-Spatial** is PageSum with spatial locality, while **PageSum-Discourse** is with discourse locality. *: results reported in the original papers. ‡: results from our own evaluation script.

The results are in Tab. 3, showing that PageSum with *discourse locality* achieves higher ROUGE scores than PageSum with *spatial locality*. While the improvement is marginal, we note that with discourse locality, PageSum can also generate more coherent summaries. For this aspect, following Bommasani and Cardie (2020), we evaluate the *semantic coherence* of generated summaries using the next sentence prediction (NSP) task introduced in BERT (Devlin et al., 2019). Specifically, we use a pre-trained BERT model[10] to predict the probability (p_{BERT}) of one sentence $S^{(i-1)}$ in the summary

the original papers, we additionally used public-available checkpoints of LED from Hugging Face's Transformers (Wolf et al., 2020) on `arXiv` ('allenai/led-large-16384-arxiv') and `PubMed` ('patrickvonplaten/led-large-16384-pubmed') to generate the summaries and used our own script for evaluation. The difference of the performance between the original result and the result of our own evaluation script is likely because the original implementation uses window-attention with 512 tokens while the HF implementation uses 1024 tokens.

[10]We use the checkpoint ('bert-large-uncased') from Hugging Face's Transformers (Wolf et al., 2020).

reference	random	spatial	discourse
0.9800	0.9543	0.9734	0.9798

Table 4: Semantic coherence (Eq. 12) of summaries on arXiv. **reference** is the reference summary. **random** is a random oracle which randomly shuffles the sentences in the reference summary. **spatial** is PageSum with spatial locality while **discourse** is with discourse locality. PageSum with discourse locality has significantly higher ($p < 0.01$) coherence than PageSum with spatial locality.

System	R-1	R-2	R-L
PRIMERA*	49.90	21.10	25.90
PRIMERA‡	50.29	21.2	46.23
BART-Long-Graph*	49.24	18.99	23.97
PageSum-Spatial	49.03	19.10	44.73
PageSum-Document	**51.17**	**21.39**	**46.88**

Table 5: System performance comparison for *document locality* on MultiNews. R-1/2/L are the ROUGE-1/2/L F-1 scores respectively. **PageSum-Spatial** is PageSum with spatial locality, while **PageSum-Document** is with document locality. *: results reported in the original papers. ‡: results from our own evaluation script.

S being followed by the next sentence $S^{(i)}$:

$$\mathrm{SC}(S) = \frac{\sum_{i=2}^{N_S} p_{\mathrm{BERT}}(S^{(i)}|S^{(i-1)})}{N_S - 1}, \qquad (12)$$

where N_S is the number of sentences in the summary. Tab. 4 shows the average semantic coherence of summaries. The summaries generated by PageSum with discourse locality have higher semantic coherence, suggesting that grouping the sentences based on discourse structures help to generate more well-structured summaries.

5.4 Exp-III: Document Locality

For multi-document summarization, we evaluate PageSum with *document locality* on MultiNews, where we view each document in the document cluster as a page. The other experiment setting is the same as in §5.3. Apart from the baseline systems in §5.1, we additionally include another model BART-Long-Graph (Pasunuru et al., 2021) for comparison, which is specifically designed for multi-document summarization and achieves top performance on MultiNews. The results are shown in Tab. 5.[11] PageSum also achieves strong performance in this setting, outperforming the previous state-of-the-art models. We also note that PageSum with *document locality* achieves much better performance than its counterpart with *spatial locality*, suggesting the importance of choosing the suitable locality for a specific task.

[11] We notice a large difference between ROUGE-L scores reported by the original paper and calculated using our evaluation script for PRIMERA. The reason might be the usage of difference versions of ROUGE-L score: we use the summary-level ROUGE-L score which is the default choice of the standard ROUGE Perl script, but there is also a sentence-level ROUGE-L score introduced in Lin (2004).

Page Size	#Pages	R-1	R-2	R-L
128	32	47.67	18.76	42.82
256	16	48.29	19.32	43.38
512	8	**48.82**	19.80	**43.85**
1024	4	48.66	**19.90**	43.74

Table 6: Performance comparison of different *page sizes* on arXiv. **Page Size** denotes the number of tokens in one page. **#Pages** denotes the number of pages. R-1/2/L are the ROUGE-1/2/L F-1 scores respectively.

System	R-1	R-2	R-L
arXiv			
Global-Decoding	48.57	**19.92**	43.71
PageSum-Spatial	**48.66**	19.90	**43.74**
MultiNews			
Global-Decoding	48.75	19.03	44.48
PageSum-Document	**51.17**	**21.39**	**46.88**

Table 7: Comparison of page-wise decoding and global decoding on arXiv and MultiNews. R-1/2/L are the ROUGE-1/2/L F-1 scores respectively.

5.5 Analysis

We analyze several important aspects of our method to gain further insights.

Page Size To investigate how the maximum length of a *page* would affect the model performance, we conduct experiments with different page sizes on arXiv. For fair comparison, we first truncate each document in arXiv to 4096 tokens, then split the document into different pages based on the page size. The results are shown in Tab. 6. We observe that increasing the page size generally helps to improve the model performance. However, the model performance stops increasing after the page size reaches 512 tokens.

Page-wise v.s. Global Decoding Both the encoder and decoder in PageSum are designed to follow the principle of locality. Specifically, the

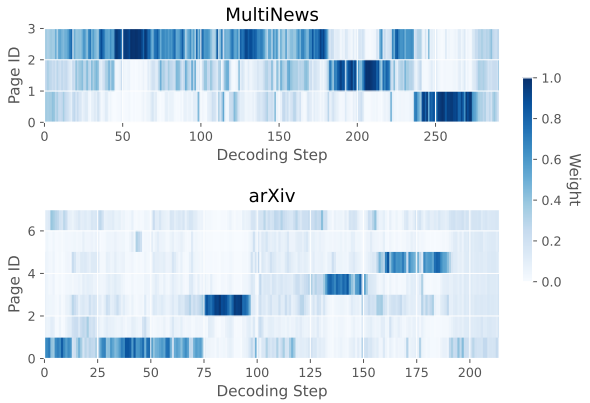

Figure 3: Visualization of importance scores of different pages at each decoding step on `MultiNews` and `arXiv` datasets. Darker colors represent higher importance scores.

decoder in PageSum first makes *local* predictions based on each encoded page (Eq. 6), which are later combined into final predictions. An alternative approach is to directly make *global* predictions based on the entire input document – the encoded pages are concatenated as a single sequence, which serves as the input to the decoder. We compare this option with our modeling strategy in Tab. 7.[12] The results show that on `arXiv`, page-wise decoding with *spatial locality* has a similar performance compared with global decoding. On the other hand, *document locality* on `MultiNews` is proven to be a very useful inductive bias because PageSum with *document locality* has a large improvement over the model with global decoding.

Visualizing Locality The confidence scores calculated by PageSum's decoder (Eq. 7) can be interpreted as the importance scores of different *pages* at each decoding step. That is, a page associated with a higher score will contribute more to the decision at the current step. Fig. 3 depicts how the importance scores changed during the decoding of the *reference summaries* on `MultiNews` and `arXiv` using two examples. We observe two phenomenons: (1) *space locality* – at each decoding step usually only a subset of pages are making large contributions to the current prediction; (2) *time locality* – PageSum's decoder tends to focus on the similar subset of pages at neighboring decoding steps.

[12]On `arXiv`, we compare the models with this setting: 4 pages, 1024 tokens for each page.

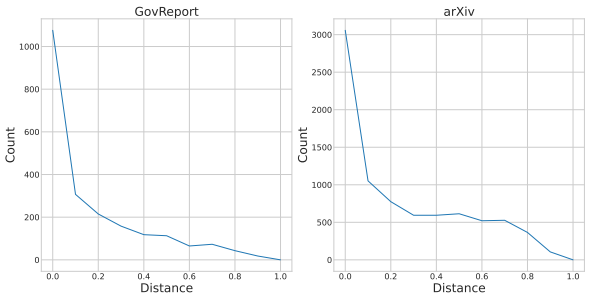

Figure 4: Number of interdependent sentences with different distances on `GovReport` and `arXiv` datasets. X-axis represents the ratio of sentence distances normalized by the number of sentences in the entire document.

Fusion Sentence	ED issued a notice of *proposed rulemaking* in **late 2018**, after revoking some of its previous guidance to schools in 2017.
14th Source Sentence	And ED recently issued another notice of *proposed rulemaking*, after having revoked some of its prior guidance to schools in 2017.
410th Source Sentence	On **November 29, 2018**, ED issued a notice of *proposed rulemaking* in the Federal Register.
PageSum Output	ED recently issued another notice of proposed rulemaking, after having revoked some of its prior guidance to schools in 2017.

Table 8: Case Study on `GovReport` about long-distance dependencies. Both 14th and 410th sentences contribute to the same reference sentence. PageSum's output fails to capture this long-distance dependency.

5.6 Case Study: Long-Distance Dependencies

A global context can be much more important in the presence of long-distance dependencies for text summarization models (Fernandes et al., 2019; Xu et al., 2020a), and PageSum can have difficulties handling those dependencies. To study this phenomenon, we first develop a method of identifying the long-distance dependencies then analyze the effect of the identified dependencies.

We leverage the notion of *sentence fusion* (Barzilay and McKeown, 2005) to investigate reference-based sentence-level dependencies. Specifically, following Lebanoff et al. (2019a,b), we define a **fusion sentence** in the *reference summary* to be a sentence that has significant overlaps with two or more sentences[13] in the *source document*. Then, we define two sentences $\hat{s}_1, \hat{s}_2$ in the source document D to be **interdependent** if they have the most significant contribution to a fusion sentence h:

$$(\hat{s}_1, \hat{s}_2) := \max_{(s_i, s_j), s_i, s_j \in D} \text{ROUGE}_{\text{Recall}}(h, s_i \oplus s_j). \quad (13)$$

[13]We focus on the case of two sentences.

More details can be found in Appendix A.

We found that our model can fail to capture long-distance dependencies where two interdependent sentences are far away from each other. We show such an example in Tab. 8, where the 14th sentence and 410th sentence in the source document both contribute to the same fusion sentence. PageSum's output only captures the information in the 14th sentence, but fails to take the 410th sentence into consideration. However, the impact of the potential failures is restricted. Specially, as shown in Fig. 4, most of the interdependent sentences are close to each other, and there are much fewer interdependent sentence pairs with long distances.

6 Discussions and Conclusions

We empirically investigate three kinds of localities in abstractive text summarization by using them as important inductive biases in our model. Using a new abstraction of viewing the input document as a series of *pages*, our model emphasizes the role of locality in both encoding and decoding stages because it ensures that tokens in different input pages never directly interact in both encoder self-attention and encoder-decoder attention. The experimental results show that our model has strong performance and follows the principle of locality. In addition, we also show that it is important to select the suitable kind of localities for different specific application scenarios.

While our model can achieve on-par or better performance compared with the models that aim to maintain a global context for the input tokens, we note that our model may fail to capture long-distance dependencies in the documents because of its inductive biases. However, the fact that our model has competitive performance comparing with the state-of-the-art models equipped with efficient or sparse attention modules suggests that those models may fall short of their designing objectives. Therefore, for the future work, we call for more rigorous examinations of the memory-efficient abstractive summarization models that aim to capture global features (e.g. long-distance dependencies) and maintain a global input context.

References

Regina Barzilay and Kathleen R. McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Rishi Bommasani and Claire Cardie. 2020. Intrinsic evaluation of summarization datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8075–8096, Online. Association for Computational Linguistics.

Shuyang Cao and Lu Wang. 2022. Hibrids: Attention with hierarchical biases for structure-aware long document summarization. *ArXiv*, abs/2203.10741.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686, Melbourne, Australia. Association for Computational Linguistics.

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*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Peter J. Denning. 2005. The locality principle. *Commun. ACM*, 48(7):19–24.

P.J. Denning. 1980. Working sets past and present. *IEEE Transactions on Software Engineering*, SE-6(1):64–84.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yue Dong, Andrei Mircea, and Jackie Chi Kit Cheung. 2021. Discourse-aware unsupervised summarization for long scientific documents. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1089–1102, Online. Association for Computational Linguistics.

Ori Ernst, Avi Caciularu, Ori Shapira, Ramakanth Pasunuru, Mohit Bansal, Jacob Goldberger, and Ido Dagan. 2021. A proposition-level clustering approach for multi-document summarization. *ArXiv*, abs/2112.08770.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Patrick Fernandes, Miltiadis Allamanis, and Marc Brockschmidt. 2019. Structured neural summarization. In *International Conference on Learning Representations*.

R. Fonseca, V. Almeida, M. Crovella, and B. Abrahao. 2003. On the intrinsic locality properties of web reference streams. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, volume 1, pages 448–458 vol.1.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mandy Guo, Joshua Ainslie, David C. Uthus, Santiago Ontañón, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2021. Longt5: Efficient text-to-text transformer for long sequences. *ArXiv*, abs/2112.07916.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

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*, pages 1419–1436, Online. Association for Computational Linguistics.

Hanqi Jin and Xiaojun Wan. 2020. Abstractive multi-document summarization via joint learning with single-document summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2545–2554, Online. Association for Computational Linguistics.

Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. Multi-granularity interaction network for extractive and abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6244–6254, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

Hilda Koopman, Dominique Sportiche, and Edward Stabler. 2013. *An introduction to syntactic analysis and theory*. John Wiley & Sons.

Logan Lebanoff, John Muchovej, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019a. Analyzing sentence fusion in abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 104–110, Hong Kong, China. Association for Computational Linguistics.

Logan Lebanoff, Kaiqiang Song, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019b. Scoring sentence singletons and pairs for abstractive summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2175–2189, Florence, Italy. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Jeffrey Ling and Alexander Rush. 2017. Coarse-to-fine attention models for document summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 33–42, Copenhagen, Denmark. Association for Computational Linguistics.

Peter J. Liu*, Mohammad Saleh*, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations*.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Potsawee Manakul and Mark Gales. 2021. Long-span summarization via local attention and content selection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6026–6041, Online. Association for Computational Linguistics.

Ziming Mao, Chen Henry Wu, Ansong Ni, Yusen Zhang, Rui Zhang, Tao Yu, Budhaditya Deb, Chenguang Zhu, Ahmed Hassan Awadallah, and Dragomir Radev. 2021. Dyle: Dynamic latent extraction for abstractive long-input summarization. *ArXiv*, abs/2110.08168.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Ramakanth Pasunuru, Mengwen Liu, Mohit Bansal, Sujith Ravi, and Markus Dreyer. 2021. Efficiently summarizing text and graph encodings of multi-document clusters. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4768–4779, Online. Association for Computational Linguistics.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah Smith, and Lingpeng Kong. 2021. Random feature attention. In *International Conference on Learning Representations*.

Michal Pietruszka, Łukasz Borchmann, and Lukasz Garncarek. 2020. Sparsifying transformer models with trainable representation pooling.

Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Chris Pal. 2020. On extractive and abstractive neural document summarization with transformer language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9308–9319, Online. Association for Computational Linguistics.

Jiezhong Qiu, Hao Ma, Omer Levy, Wen-tau Yih, Sinong Wang, and Jie Tang. 2020. Blockwise self-attention for long document understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2555–2565, Online. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Tobias Rohde, Xiaoxia Wu, and Yinhan Liu. 2021. Hierarchical learning for generation with long source sequences. *CoRR*, abs/2104.07545.

Qianqian Ruan, Malte Ostendorff, and Georg Rehm. 2022. Histruct+: Improving extractive text summarization with hierarchical structure information. *ArXiv*, abs/2203.09629.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, Los Alamitos, CA, USA. IEEE Computer Society.

Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. 2020. Sparse Sinkhorn attention. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9438–9447. PMLR.

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*, volume 30. Curran Associates, Inc.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *ArXiv*, abs/2006.04768.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu,

Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2021. PRIMER: pyramid-based masked sentence pre-training for multi-document summarization. *CoRR*, abs/2110.08499.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3011–3021, Hong Kong, China. Association for Computational Linguistics.

Jiacheng Xu and Greg Durrett. 2019. Neural extractive text summarization with syntactic compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3292–3303, Hong Kong, China. Association for Computational Linguistics.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020a. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, Online. Association for Computational Linguistics.

Shusheng Xu, Xingxing Zhang, Yi Wu, Furu Wei, and Ming Zhou. 2020b. Unsupervised extractive summarization by pre-training hierarchical transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1784–1795, Online. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Kumar 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 *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc.

Erfan Zamanian, Carsten Binnig, and Abdallah Salama. 2015. Locality-aware partitioning in parallel database systems. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, page 17–30, New York, NY, USA. Association for Computing Machinery.

Jianmin Zhang, Jiwei Tan, and Xiaojun Wan. 2018. Adapting neural single-document summarization model for abstractive multi-document summarization: A pilot study. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 381–390, Tilburg University, The Netherlands. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Yusen Zhang, Ansong Ni, Ziming Mao, Chen Henry Wu, Chenguang Zhu, Budhaditya Deb, Ahmed Hassan Awadallah, Dragomir Radev, and Rui Zhang. 2021. Summ^N: A multi-stage summarization framework for long input dialogues and documents. *ArXiv*, abs/2110.10150.

Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Seal: Segment-wise extractive-abstractive long-form text summarization. *ArXiv*, abs/2006.10213.

A Long-Distance Dependencies

We define two sentences $\hat{s}_1, \hat{s}_2$ in the source document D to be **interdependent** if they have the most significant contribution to a fusion sentence h in the reference summary:

$$(\hat{s}_1, \hat{s}_2) := \max_{(s_i, s_j), s_i, s_j \in D} \text{ROUGE}_{\text{Recall}}(h, s_i \oplus s_j). \quad (14)$$

where we use ROUGE Recall to measure the sentence contribution by reviewing h as the reference. We also define two filtering rules:

$$\text{ROUGE}(h, s) > t_1, \quad (15)$$

$$\text{ROUGE}(h, \hat{s}_1 \oplus \hat{s}_2) - \text{ROUGE}(h, s) > t_2, \quad (16)$$

where $s \in \{\hat{s}_1, \hat{s}_2\}$. t_1 and t_2 are two threshold values which are set to 20 and 10 respectively based on our empirical observations. Eq. 15 ensures that each sentence has a non-trivial overlap with the fusion sentence, while Eq. 16 ensures that each sentence has a unique contribution.