PRIMER: Pyramid-based Masked Sentence Pre-training for Multi-document Summarization

Wen Xiao[†*]   **Iz Beltagy**[‡]   **Giuseppe Carenini**[†]   **Arman Cohan**[‡§]

[†]University of British Columbia, Vancouver, Canada
[‡]Allen Institute for AI, Seattle, WA, USA
[§]Paul G. Allen School of Computer Science & Engineering, University of Washington

{xiaowen3,carenini}@cs.ubc.ca, {beltagy,armanc}@allenai.org

Abstract

Recently proposed pre-trained generation models achieve strong performance on single-document summarization benchmarks. However, most of them are pre-trained with general-purpose objectives and mainly aim to process single document inputs. In this paper, we propose PRIMER, a pre-trained model for multi-document representation with focus on summarization that reduces the need for dataset-specific architectures and large amounts of fine-tuning labeled data. Specifically, we adopt the Longformer architecture with proper input transformation and global attention to fit for multi-document inputs, and we use Gap Sentence Generation objective with a new strategy to select salient sentences for the whole cluster, called Entity Pyramid, to teach the model to select and aggregate information across a cluster of related documents. With extensive experiments on 6 multi-document summarization datasets from 3 different domains on the zero-shot, few-shot and full-supervised settings, our model, PRIMER, outperforms current state-of-the-art models on most of these settings with large margins.[1]

1 Introduction

Multi-Document Summarization is the task of generating a summary from a cluster of related documents, which is useful in many scenarios, e.g. summarizing all the reviews for a restaurant, summarizing a series of news, or summarizing related papers in the same domain.

State-of-the-art approaches to multi-document summarization are mostly graph-based (Liao et al., 2018; Li et al., 2020; Pasunuru et al., 2021), leveraging graph neural networks to connect information between the documents, or hierarchical (Liu and Lapata, 2019a; Fabbri et al., 2019; Jin et al.,

---

[*]Work mainly done during internship at AI2.

[1]The code and pretrained models will be released at https://github.com/allenai/PRIMER
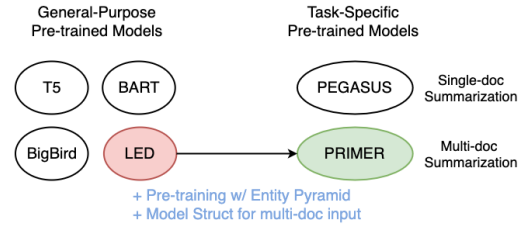

Figure 1: Prior work on pre-training mostly focuses on single document representation and does not target a specific downstream task. In contrast, PRIMER is a pre-trained model for multi-document processing with focus on summarization.

2020), building intermediate representations of individual documents and then aggregating information across. While effective, these models either require domain-specific additional information e.g. Abstract Meaning Representation (Liao et al., 2018), or discourse graphs (Christensen et al., 2013; Li et al., 2020), or use dataset-specific, customized architectures, making it difficult to leverage pre-trained language models. Simultaneously, recent pre-trained language models (typically encoder-decoder transformers) have shown the advantages of pre-training and transfer learning for generation tasks and achieved strong performance on summarization (Raffel et al., 2020; Lewis et al., 2020; Beltagy et al., 2020; Zaheer et al., 2020). Yet, existing pre-trained models either use single-document pre-training objectives (e.g. masked language modeling (Raffel et al., 2020), autoregressive language modeling (Radford et al., 2019), or other denoising objectives (Lewis et al., 2020)), or use encoder-only models that do not work for generation tasks like summarization (e.g., CDLM, Caciularu et al., 2021).

Therefore, we argue that these pre-trained models are not necessarily the best fit for multi-document summarization. Alternatively, we propose a simple pre-training approach for multi-
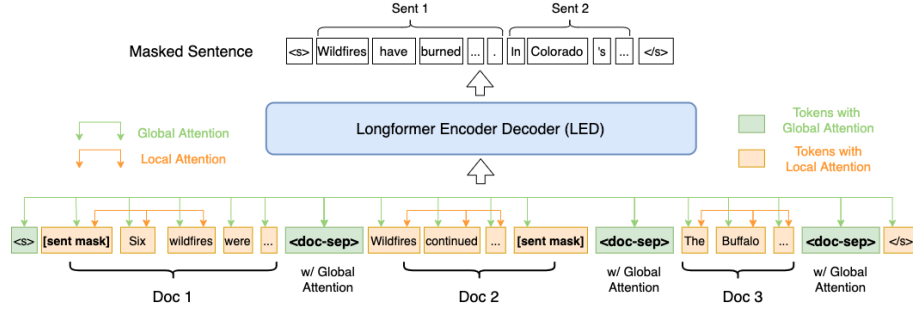
Figure 2: Model Structure of PRIMER. Each document ends with a document separator token with global attention on it, and all the other tokens (except `<s>`) have local attention only. (For better visualization, the window size of local attention in the figure is $w = 3$, we use the default window size $w = 512$ in practice.) The selected sentences are replaced with [sent mask], and the model is trained to recover the masked sentences in the output.

document summarization, reducing the need for dataset-specific architectures and large amounts of fine-tuning labeled data (See Figure 1 to compare with other pre-trained models). Our method is designed to teach the model to identify and aggregate salient information across a "cluster" of related documents during pre-training. Specifically, our approach uses the Gap Sentence Generation objective (Zhang et al., 2020), i.e. masking out several sentences from the input document, and recovering them in order in the decoder. However, we propose a novel strategy for masking sentences which we call, Entity Pyramid, inspired by the Pyramid Evaluation method (Nenkova and Passonneau, 2004). With Entity Pyramid, we mask salient sentences in the entire cluster then train the model to generate them, encouraging it to find important information across documents and aggregate it in one summary.

We conduct extensive experiments on **6** multi-document summarization datasets from **3** different domains. We show that despite its simplicity, PRIMER achieves superior performance compared with prior state-of-the-art pre-trained models, as well as dataset-specific models in both few-shot and full fine-tuning settings. PRIMER performs particularly strong in zero- and few-shot settings, significantly outperforming prior state-of-the-art up to **5** Rouge-1 points with as few as 10 examples. Our contributions are summarized below:

• We propose PRIMER, the first pre-trained generation model for multi-document inputs with focus on summarization. PRIMER relies on Entity Pyramid, a novel pre-training strategy that trains the model to select and aggregate salient information from documents.

• We extensively evaluate PRIMER with 6 datasets

from 3 different domains for the zero-shot, few-shot and fully-supervised setting. We show that PRIMER outperforms current state-of-the-art models on most of these evaluations with large margins.

2 Model

In this section, we discuss our proposed model PRIMER, a new pre-trained general model for multi-document summarization. Unlike prior work, PRIMER minimizes dataset-specific modeling by simply concatenating a set of documents and processing them with a general efficient encoder-decoder transformer model (§2.1). The underlying transformer model is pre-trained on an unlabeled multi-document dataset, with a new entity-based sentence masking objective to capture the salient information within a set of related documents (§2.2).

2.1 Model Architecture and Input Structure

Prior work on multi-document summarization mainly employs dataset-specific and often complex customized architectures to capture and connect information from within a set of documents (§5). Instead, our goal is to minimize dataset-specific modeling to leverage general pre-trained transformer models for the multi-document task and make it easy to use in practice. Therefore, to summarize a set of related documents, we simply concatenate all the documents in a single long sequence, and process them with an encoder-decoder transformer model. Since the concatenated sequence is long, instead of more standard encoder-decoder transformers like BART (Lewis et al., 2020) and T5 (Raffel et al., 2020), we use the Longformer-Encoder-Decoder (LED) Model (Beltagy et al., 2020), an efficient transformer model with linear complexity

Document #1
Wildfires have burned across tens of thousands of acres of parched terrain in Colorado, spurring thousands of evacuations ...(0.107)..., residents have sought shelter in middle schools, and local officials fear tourists usually drawn to the region for the summer may not come.
Document #2
... *In Colorado's southwest, authorities have shuttered the San Juan National Forest in southwestern Colorado and residents of more than 2,000 homes were forced to evacuate.(0.187)* No homes had been destroyed ... "Under current conditions, one abandoned campfire or spark could cause a catastrophic wildfire, ..., with human life and property," said San Juan National Forest Fire Staff Officer Richard Bustamante...
Document #3
The Buffalo Fire west of Denver is ... Several wildfires in Colorado have prompted thousands of home evacuations ...(0.172)... Nearly 1,400 homes have been evacuated in Summit County, Colorado, ...(0.179)... "Under current conditions, one abandoned campfire or spark could cause a catastrophic wildfire, ... , with human life and property," said Richard Bustamante, SJNF forest fire staff officer ...
Entities with High Frequency
Colorado, 416, Tuesday, Wildfires, San Juan National Forest,...

Figure 3: An example on sentence selected by Principle strategy and Entity Pyramid strategy. Text in red are the sentences with the highest Principle ROUGE scores. We label the most frequent entity 'Colorado' in the source documents in blue, followed by the Pyramid ROUGE scores in parenthesis. The final selected sentence by Entity Pyramid strategy is in *italic*. which is a better pseudo-summary than the ones selected by the Principle strategy.

with respect to the input length.[2] LED sparsifies the full attention matrix in transformers by combining a sliding window local attention and global attention on few locations of interest, thereby reducing the quadratic complexity with respect to length to linear. LED uses this sparse attention mechanism in the encoder self-attention side while using the full attention on decoder and cross-attention.[3]

When concatenating, we add special document separator tokens (<doc-sep>) between the documents to make the model aware of the document boundaries (Figure 2). We also assign global attention to these tokens which the model can use to share information across documents (Caciularu et al., 2021) (see §4 for ablations of the effectiveness of this input structure and global attention).

---

[2]We use LED and not other efficient transformers like Bigbird-PEGASUS (Zaheer et al., 2020) for two reasons, the first is that BigBird's global attention can't be assigned to individual tokens in the middle of the sequence, which is important for the representation of long documents as shown in Caciularu et al. (2021). Second, because pre-trained checkpoints are available for LED, while BigBird-PEGASUS released the already fine-tuned checkpoints.

[3]Since the length of the summary is short, using full attention on decoder and cross-attention is not computationally prohibitive.

2.2 Sentence (Un)Masking for Multi-document Summarization

While pre-trained transformers with general autoregressive or autoencoding language modeling objectives (Raffel et al., 2020; Lewis et al., 2020) perform well across a variety of tasks, utilizing pre-training objectives that are closer to the downstream summarization task have been shown to provide further gains. In particular, PEGASUS (Zhang et al., 2020) introduces Gap Sentence Generation as a pre-training objective where sentences are masked from the input and the model is tasked to generate the masked sentences. Following PEGASUS, we use the Gap Sentence Generation objective, but introduce a new masking strategy that is designed for multi-document summarization.

Specifically, we select and mask out m summary-like sentences from a set of related documents (we also call this set a "cluster") that we want to summarize, i.e. every selected sentence is replaced by a single token [sent-mask] in the input, and train the model to generate the concatenation of those sentences as a "pseudo-summary" of the cluster (See Figure 2). This is close to abstractive summarization because the model needs to reconstructs the masked sentences using the information in the rest of the documents.

The critical point is how to select sentences that best summarize or represent the cluster of multiple related documents. Zhang et al. (2020) use three strategies - Random, Lead (first m sentences), and "Principle". The Principle strategy is to select sentences greedily based on their importance. As a proxy for importance in the case of single-document input, an importance score is computed by the ROUGE score between each sentence, s_i, and the rest of the document ($D/\{s_i\}$), i.e. $\text{Score}(s_i) = \text{ROUGE}(s_i, D/\{s_i\})$. Intuitively, this assigns a high score to the sentences that have a high overlap with the other sentences.

However, we argue that a naive extension of such strategy to multi-document summarization would be problematic since multi-document inputs typically contain a large amount of redundant information, and such strategy would prefer exact match between sentences, resulting in selection of less representative information. For example, as shown in Figure 3, the cluster contains three news articles talking about a wildfire happened in Corolado, and the pseudo-summary of this cluster should be related to the location, time and consequence of the

wildfire, but with the Principle strategy, the non-salient sentences quoting the words from an officer are assigned the highest score, as the exact same sentence appeared in two out of the three articles.

To address this limitation, we propose a new masking strategy inspired by the Pyramid Evaluation framework (Nenkova and Passonneau, 2004), which aims to select sentences that best represent the entire cluster of input documents.

2.2.1 Entity Pyramid Masking

We first provide a brief description of the pyramid evaluation framework and then discuss our proposed masking method.

Pyramid Evaluation The Pyramid Evaluation method (Nenkova and Passonneau, 2004) is based on the intuition that relevance of a unit of information can be determined by the number of reference (i.e. gold standard) summaries that include it. The unit of information used by the Pyramid method is the Summary Content Unit (SCU); words or phrases that represent single facts. These SCUs are first identified by human annotators in each reference summary, and they receive a score proportional to the number of reference summaries that contain them. A Pyramid Score for a candidate summary is then the normalized mean of the scores of the SCUs that it contains. One advantage of the Pyramid method is that it directly assesses the content information of the candidate summaries.

Entity Pyramid Masking Inspired by the way to measure the content relevance in the Pyramid Evaluation, we hypothesize that a similar idea could be applied for the multi-document summarization. Specifically, for a cluster with multiple related documents, the more documents a unit of information appears in, the more salient that information should be to the cluster. Therefore, it should be considered for inclusion in the pseudo-summary in our masked sentence generation objective. However, the SCUs in the original Pyramid Evaluation are human-annotated, which is not feasible for large scale pre-training. As a proxy, we explore leveraging information expressed as named entities, since they are key building blocks in extracting information from text about events/objects and the relationships between their participants/parts (Jurafsky and Martin, 2009). Following the Pyramid framework, we use the entity frequency in the cluster as a proxy for saliency. Concretely, as shown in Fig.4, we have the following three steps to select

---

Algorithm 1 Entity Pyramid Sentence Selection

Input: Document cluster
Input: List of entities w/ frequency > 1. N length of the list
Input: m number of sentences to select
Output: List of sentences to mask
1: $E \leftarrow$ sort entities by frequency, descending
2: $selected = []$
3: **for** $i \leftarrow 1$ **to** $|E|$ **do**
4: $\quad current\_sent = \emptyset$
5: $\quad max = 0$
6: $\quad$ **for** doc in $cluster$ **do**
7: $\quad\quad$ **for** $sent$ in doc **do**
8: $\quad\quad\quad$ **if** E_i in $sent$ **then**
9: $\quad\quad\quad\quad$ **if** $Score(sent) > max$ **then**
10: $\quad\quad\quad\quad\quad current\_sent = sent$
11: $\quad\quad\quad\quad$ **end if**
12: $\quad\quad\quad$ **end if**
13: $\quad\quad$ **end for**
14: $\quad$ **end for**
15: $\quad selected.append(current\_sent)$
16: $\quad$ **if** $|selected| == m$ **then**
17: $\quad\quad$ Break
18: $\quad$ **end if**
19: **end for**
20: **Return** $selected$

---

salient sentences in our masking strategy:

1. *Entity Extraction*. We extract all the named entities from the cluster using NER tools.[45]

2. *Entity Pyramid Estimation*. We then build an Entity Pyramid for estimating the saliency of entities based on their document frequency, i.e. the number of documents each entity appears in.

3. *Sentence Selection*. Similar to the Pyramid evaluation framework, we identify salient sentences with respect to the cluster of related documents. Algorithm 1 shows the sentence selection procedure. As we aim to select the entities better representing the whole cluster instead of a single document, we first remove all entities from the Pyramid that appear only in one document. Next, we iteratively select entities from top of the pyramid to bottom (i.e., highest to lowest frequency), and then we select sentences in the document that include the entity as initial candidate set. Finally, within this candidate set, we find the most representative sentences to the cluster by measuring content overlap of the sentence w.r.t documents other than the one it appears in. This final step supports the goal of our pre-training objective, namely to reconstruct sentences that can be recovered using information from other documents in the cluster, which encourages the model to better connect and aggregate information across multiple documents. Following Zhang et al. (2020) we use ROUGE scores (Lin, 2004) as a

---

[4]We use Spacy https://spacy.io/.
[5]Note that entity information is used only at pre-training time. This is unlike some of prior work that utilize additional information (like coref, discourse, or AMR) at fine-tuning and inference time.
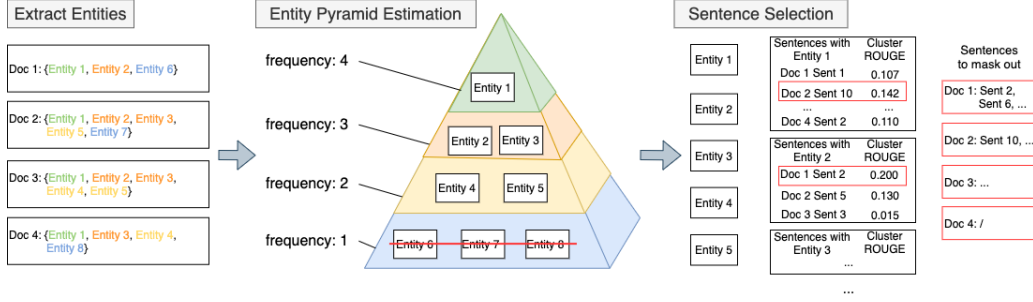
Figure 4: The Entity Pyramid Strategy to select salient sentences from the cluster. In the Entity Extraction step, all the entities (Entity 1 - Entity 8) are extracted from each document in the cluster, then the pyramid is built based on the number of documents that each entity appears in Pyramid Estimation step. In the Sentence Selection step, the most representative sentence is chosen based on Cluster ROUGE for each entity with frequency > 1, e.g. Sentence 10 in Document 2 is chosen for Entity 1. Finally the selected sentences are masked out and the model learns to recover the sentences in the pre-training.

proxy for content overlap. For each sentence s_i, we specifically define a Cluster ROUGE score as
$$Score(s_i) = \sum_{\{doc_j \in C, s_i \notin doc_j\}} \text{ROUGE}(s_i, doc_j)$$
Where C is the cluster of related documents.

Note that different from the importance heuristic defined in PEGASUS (Zhang et al., 2020), Entity Pyramid strategy favors sentences that are representative of more documents in the cluster than the exact matching between fewer documents. As shown in Table 3, instead of the quoted words, our strategy select the most representative sentences in the cluster with high frequency entities.

3 Experiments

In this section we first briefly describe our experimental setup, then discuss the zero and few-shot performance comparisons and finally present results for the full data fine-tuning setting compared with the state-of-the-art.

3.1 Experimental Setup

Implementation Details We use the Longformer-Encoder-Decoder (LED) (Beltagy et al., 2020) large as our model initialization, which has the same architecture as BART (Lewis et al., 2020), except new position embeddings by repeatedly copying BART's 1K position embeddings 4 times to modeling long sequences and using sparse local attention. The length limits of input and output are 4096 and 1024, respectively, with sliding window size as $w = 512$ for local attention in the input.

As the multi-document summarization task has a higher compression ratio, defined as $len(Summary)/len(SourceInput)$, (e.g. 12%

Dataset	#Examples	#Doc/C	Len_{src}	Len_{summ}
Newshead (2020)	360K	3.5	1734	-
Multi-News (2019)	56K	2.8	1793	217
Multi-Xscience (2020)	40K	4.4	700	105
Wikisum* (2018)	1.5M	40	2238	113
WCEP-10 (2020)	10K	9.1	3866	28
DUC2004 (2005)	50	10	5882	115
arXiv (2018)	214K	5.5	6021	272

Table 1: The statistics of all the datasets we explore in this paper. *We use subsets of Wikisum (10/100, 3200) for few-shot training and testing only.

for Multi-News dataset and 15% for Multi-Xscience dataset), we use 15% as ratio of masked sentences for generation. In addition to this 15% masked sentences, following PEGASUS (Zhang et al., 2020), we also copy an additional 15% of the input sentences to the output without masking them in the input. This allows the model to also learn to copy information from the source directly and found to be useful by Zhang et al. (2020).

The model is pre-trained for 100K steps with early stopping, batch size 16, optimized using Adam (Kingma and Ba, 2015) with learning rate of $3e-5$ following Beltagy et al. (2020), with 10K warmup steps and linear decay.

Pre-training Corpus For pre-training, we use the *Newshead* dataset (Gu et al., 2020), a relatively large resource, where each instance includes a set of related documents about a specific news event. Note that this dataset does not have any ground-truth summaries. The articles about each event are automatically clustered based on the content similarity within a time window. The statistics of the dataset can be found in Table 1.

Evaluation Datasets For evaluation, we use a wide range of multi-document summarization datasets from various domains (News, Wikipedia, and Scientific literature) as well as one single document summarization dataset, to comprehensively assess the effectiveness of our approach across a variety of domains. Below we briefly discuss these datasets (see Table 1 for dataset statistics).

Multi-News (Fabbri et al., 2019): A multi-document dataset with summaries written by professional editors from the newser.com.

Wikisum (Liu* et al., 2018) Each summary is a Wikipedia article, and the source documents are either citations in the reference section or the Web Search results of section titles.[6] In our experiments, we use the data crawled by Liu and Lapata (2019a).

WCEP (Gholipour Ghalandari et al., 2020) is built based on news events from Wikipedia Current Events Portal and the references are obtained similar to Wikisum. There are at most 100 documents within each cluster in the original dataset, thus we remove all the duplicates and only keep up to 10 documents for each cluster based on the relevance score in the original dataset, which is similar with the WCEP-10 variant in the original paper.

Multi-X-Science (Lu et al., 2020) a multi-document summarization dataset created from scientific articles, the summaries are paragraphs of related work section, while source documents include the abstracts of the query and referred papers.

DUC benchmarks (Dang, 2005) include multi-document summarization datasets in the news domain, with 10-30 documents and 3-4 human-written summaries per cluster. Since these datasets are small, we use them primarily for few-shot evaluation. We use DUC2003 for training (only one of the reference summaries for each document is used for training) and DUC2004 as test.

ArXiv (Cohan et al., 2018) is a single document summarization dataset in the scientific paper domain. Each document is a scientific paper, and the summary is the corresponding abstract. As each scientific paper consists of multiple sections, we treat each section as a separate document within a cluster in our experiments. This is to evaluate our model's effectiveness on summarizing single documents having multiple sections.

Evaluation metrics Following previous works (Zhang et al., 2020), we use ROUGE scores (R-1, -2 and -L), which is the standard evaluation metric, to evaluate the downstream task of multi-document summarization.[7]

3.2 Zero- and Few-shot Evaluation

While achieving strong results, many existing works in adapting pre-trained models for summarization require large amounts of fine-tuning data, which is often impractical for new domains. In contrast, since our pre-training strategy is mainly designed for multi-document summarization, we expect that our approach can quickly adapt to new datasets without the need for significant fine-tuning data. To test this hypothesis, we first provide evaluation results in zero and few-shot setting where the model is provided with no, or only a few (10 and 100) training examples. Obtaining such a small number of examples should be viable in practice for new data.

Comparison To better show the utility of our pre-trained models, we compare our model with three state-of-the-art pre-trained generation models listed below. These pre-trained models have been shown to outperform dataset-specific models in summarization (Lewis et al., 2020; Zhang et al., 2020), and because of pre-training, they are expected to also work well in the few-shot settings. As there is no prior work doing few-shot and zero-shot evaluations on all the datasets we consider, and also the results in the few-shot setting might be influenced by sampling variability (especially with only 10 examples) (Bragg et al., 2021), we run the same experiments for the compared models five times with different random seeds (shared with all the models), with the public available checkpoints .[8]

BART (Lewis et al., 2020) an encoder-decoder transformer model pre-trained on the objective of reconstructing the corrupted documents in multiple ways, e.g. Token Deletion, Text Infilling, Sentence Rotation and etc.

PEGASUS (Zhang et al., 2020) a pre-trained model designed for abstractive summarization as downstream task, especially for the single document input. It is trained on the objective of Gap

---

[6]Due to the large size of the dataset, we evaluate all the models on the first 3200 data in the test set. And in the few-shot experiments, we randomly choose few examples (10 or 100) from the training set and validation set.

[7]We use the implementation of ROUGE from Google, https://github.com/google-research/google-research/tree/master/rouge, with the default settings for the stemmer.

[8]All the checkpoints are from HuggingFace. https://huggingface.co/models

Models	Multi-News(256)			Multi-XSci(128)			WCEP(50)			WikiSum(128)			arXiv(300)			DUC2004 (128)		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
PEGASUS★(Zhang et al., 2020)	36.5	10.5	18.7	-	-	-	-	-	-	-	-	-	28.1	6.6	17.7	-	-	-
PEGASUS (our run)	32.0	10.1	16.7	27.6	**4.6**	15.3	**33.2**	**12.7**	**23.8**	24.6	5.5	15.0	29.5	7.9	17.1	32.7	**7.4**	17.6
BART (our run)	27.3	6.2	15.1	18.9	2.6	12.3	20.2	5.7	15.3	21.6	5.5	15.0	29.2	7.5	16.9	24.1	4.0	15.3
LED (our run)	17.3	3.7	10.4	14.6	1.9	9.9	18.8	5.4	14.7	10.5	2.4	8.6	15.0	3.1	10.8	16.6	3.0	12.0
PRIMER (our model)	**42.0**	**13.6**	**20.8**	**29.1**	4.6	**15.7**	28.0	10.3	20.9	**28.0**	**8.0**	**18.0**	**34.6**	**9.4**	**18.3**	**35.1**	7.2	**17.9**

Table 2: Zero-shot results. The models in the first block use the full-length attention ($O(n^2)$) and are pre-trained on the single document datasets. The numbers in the parenthesis following each dataset indicate the output length limit set for inference. PEGASUS★ means results taken exactly from PEGASUS (Zhang et al., 2020), where available.

Sentence Generation on C4 (Raffel et al., 2020) and Hugenews datasets (Note that the pre-training data size in PEGASUS is magnitudes larger than ours). As it is only evaluated on one multi-document summarization dataset (Multi-news), we rerun the model on all the datasets. To verify the quality of our reproduction, the average ROUGE scores of our re-run model vs. (the ones reported on the paper) with 10 examples and 100 examples fed are 23.81 ± 0.79 vs. (24.13) and 25.86 ± 0.41 vs. (25.48), with minor differences plausibly resulting from different samplings.

Longformer Encoder-Decoder (LED) (Beltagy et al., 2020) is the initial state of our model before pre-training. The parameters of LED are inherited from the BART model, and to enable the model to deal with longer input, the position embeddings are repeatedly copied from BART's 1K position embeddings. It is different from our model with respect to both pre-training and input structure (document separators and global attentions), with global attention on the (<s>) token only and no document separators.

Similar to Pasunuru et al. (2021), the inputs of all the models are the concatenations of the documents within the clusters (in the same order), each document is truncated based on the input length limit divided by total number of documents so that all documents are represented in the input [9]. To preserve the same format as the corresponding pre-trained models we set the inference length limit of input and output for BART and PEGASUS exactly as their pre-trained settings (i.e. 512/256 and 1024/1024 respectively) on all of the datasets (Except for the zero-shot experiments, the details can be found in Sec.3.3).[10] We use the same length limit as our

model for the LED model, i.e. 4096/1024 for input and output respectively, for all the datasets.

3.3 Zero-Shot Results

For zero-shot[11] abstractive summarization experiments, since the models have not been trained on the in-domain datasets, the lengths of generated summaries mostly depend on the pre-trained settings. Thus to better control the length of generated summaries and for fair comparison between all models, following Zhu et al. (2019), we set the length limit of the output at inference time to the average length of gold summaries.[12] Exploring other approaches to controlling length at inference time (e.g., Wu et al., 2021) is an orthogonal direction which we leave for future work.

Table 2 shows the performance comparison among all the models. Results indicate that our model achieves substantial improvements compared with all the three baselines on most of the datasets. As our model is pre-trained on clusters of documents with longer input and output, the benefit is stronger on the dataset with longer summaries, e.g. Multi-News and arXiv. Comparing PEGASUS and BART models, as the objective of PEGASUS is designed mainly for summarization tasks, not surprisingly it has relatively better performances across different datasets. In addition, the gap between PEGASUS and BART is larger on the news datasets (e.g. Multi-News, WCEP, and DUC2004). The reason behind this may be that PEGASUS is pre-trained mainly on news datasets, while BART is pre-trained on a book corpus and Wikipedia, in addition to the news corpus. Interestingly, LED underperforms other models, plausibly since part

---

[9] pilot experiments show that simple truncation results in inferior performance, which is also in line with Pasunuru et al. (2021)

[10] Regarding the length limit of inputs for PEGASUS, we do experiments with 512, 1024, 4096 on Multi-News dataset, and the model with length limit 512 achieves the best performance,

thus we use the setting for all the datasets. It is not applicable with BART, as PEGASUS uses sinusoidal position embedding, while BART use learned position embedding.

[11] For clarity, by zero-shot we mean using no examples in the target dataset to train the model

[12] In practice, it is fair to assume we know the approximate length of the expected summary depending on the task/domain.
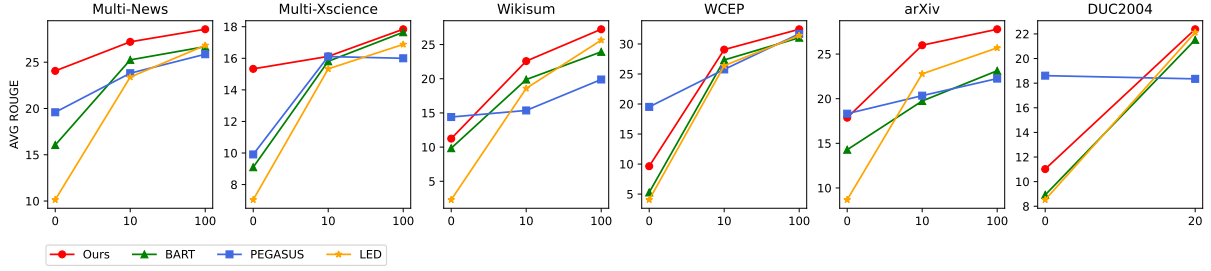
Figure 5: The AVG ROUGE scores (ROUGE-1, ROUGE-2 and ROUGE-L) of the pre-trained models with 0, 10 and 100 training data. All the results of few-shot experiments (10 and 100) are obtained by the average of 5 random runs (with the same set of seeds shared by all the models).

of the position embeddings (1k to 4k) are not pre-trained. Encouragingly, our model performs the best, demonstrating the benefits of our pre-training strategy for multi-document summarization, while also being able to work with long sequences.

3.4 Few Shot Evaluation

Compared with the strict zero-shot scenario, few-shot experiments are closer to the practical scenarios, as it is arguably affordable to label dozens of examples for almost any application.

We fine-tune all of the four models on different subsets with 10 and 100 examples, and the results are shown in Figure 5.[13] Since ROUGE-1, -2 and -L show the same trend, we simply show the average of the three metrics in the figure for brevity.[14] To show the generality, all the results of few-shot experiments are the average over 5 runs[15].

The result of each run is obtained by the 'best' model chosen based on the ROUGE scores on a randomly sampled few-shot validation set with the same number of examples as the training set, which is similar with Zhang et al. (2020). Note that their reported best models have been selected based on the whole validation set (which may give PEGASUS some advantage over ours). Nevertheless we argue that sampling few-shot validation sets as we do here is closer to real scenarios.

Our model outperforms all baselines on all of the datasets with 10 and 100 examples demonstrating the benefits of our pre-training strategy and input structure. Comparing the performances of our model with the different number of training data fed in, our model converges faster than other models with as few as 10 data-points. PEGASUS

Datasets	Prev. SOTA			PRIMER		
	R1	R2	RL	R1	R2	RL
Multi-News	49.2	19.6	24.5	**49.9**	**21.1**	**25.9**
Multi-XScience	**34.1**	6.8	—	31.9	**7.4**	—
WCEP	35.4	15.1	25.6	**46.1**	**25.2**	**37.9**
arXiv	46.6	19.6	41.8	**47.6**	**20.8**	**42.6**

Table 3: Fully supervised results. The results of previous SOTA are taken from Pasunuru et al. (2021) for Multi-News, Lu et al. (2020) for Multi-XScience[16], Hokamp et al. (2020) for WCEP, and Beltagy et al. (2020) for arXiv.

and BART have a similar performance, with BART being slightly better than PEGASUS, which may result from the larger length limit of the inputs. LED, on the other hand, get the most performance jump from zero- to few-shot setting. However, on all the multi-document summarization datasets, it still interestingly delivers a comparable performance with the other two models, even though it allows for longer input.

3.5 Fully Supervised Evaluation

To show the advantage of our pre-trained model when there is abundant training data, we also train the model with the full training set.[17] Table 3 shows the performance comparison with previous state-of-the-art.[18], along with the results of previous SOTA. Different from the zero-/few-shot experiments, since we use the same train/valid/test set as in those prior works, we report results as presented

---

[13]The hyperparameter settings can be found in §A.1

[14]The detailed ROUGE scores can be found in Table 5 in the Appendix.

[15]The results with variance can be found in Figure 7 in Appendix B

[16]We couldn't determine the version of ROUGE-L used in Lu et al. (2020) for the evaluation of Multi-XScience, and we couldn't get the code from the authors because the evaluation code is not available anymore. As a results, our ROUGE-L results are not comparable and we don't report it for Multi-XScience.

[17]The hyperparameter settings can be found in §A.2

[18]Due to the lack of computational resources, we do not train the model on Wikisum.
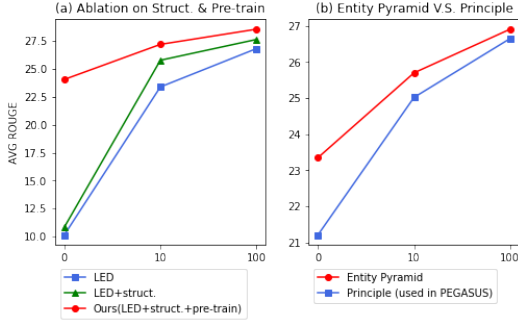
Figure 6: Ablation study with the few-shot setting on the Multi-News dataset regarding to (a) input Structure (`<doc-sep>` tokens between documents and global attention on them) and pre-training, (b) different pre-training strategy (with LED-base model).

in the corresponding original papers.

We observe that PRIMER achieves state-of-the-art results on Multi-News, WCEP, and arXiv, while slightly underperforming the prior work on Multi-XScience (R-1). On Multi-XScience clusters have less overlapping information which is a bit different than the pre-training setting of PRIMER. The source documents in this dataset are the abstracts of all the research publications cited in the related work paragraphs, which might be less similar with each other and the target related work. Remarkably, PRIMER achieves better performance than the LED model (State-of-the-art) on the arXiv dataset while using a sequence length 4x shorter (4K in PRIMER v.s. 16K in LED), further showing that the pre-training and input structure of our model not only works for multi-document summarization but can also effectively improve single-document summarization for long documents.

4 Ablation Study

Pre-training and input structure We conduct ablation studies on the Multi-News dataset in the few-shot setting, to validate the contribution of each component in our pre-trained models. Results are illustrated in Figure 6 (a). While the pre-training contributes the most in the zero-shot setting, the input structure (`<doc-sep>` tokens between documents and global attention on them) helps more in the case of 10 examples, but when there are 100 training examples, the contributions of each component are about the same.

Strategy Comparison: Principle V.S. Entity Pyramid To verify the effectiveness of our pre-training strategy, i.e. Entity Pyramid, we do an additional ablation study with the base model on the

Multi-News dataset. We prepare the pre-training data by following two different strategies, i.e. the Entity Pyramid strategy proposed in this paper, and the Principle strategy used in PEGASUS (Zhang et al., 2020). Then, we train the models on the two pre-training data with all the other settings remaining the same (e.g. length limit of input and output, pre-training dataset, input structure, as well as the document separators). Finally, we run the same experiments under zero- and few-shot scenario on the Multi-News dataset as in Sec.3.2, and the results are shown in Figure 6 (b). The model pre-trained with our novel Entity Pyramid strategy shows a clearly improved performance on the Multi-News dataset under few-shot scenarios.

5 Related Work

Neural Multi-Document Summarization Neural multi-document summarizerization models can be roughly categorized into two classes, graph-based models (Liao et al., 2018; Li et al., 2020; Pasunuru et al., 2021) and hierarchical models (Liu and Lapata, 2019a; Fabbri et al., 2019; Jin et al., 2020). The graph-based models typically leverage additional information, e.g. contextual representation (Yasunaga et al., 2017), AMR (Liao et al., 2018), or discourse structure (Christensen et al., 2013; Li et al., 2020), to build a graph for the documents within the cluster that connects information across, and use graph neural networks to encode this graph. However, extracting such auxiliary information often requires additional models both at training and inference time, makes such models dataset soecific and less general, and limits their utility in practice. Hirarchical models are another class of successful models for multi-document summarization, examples of which include multi-head pooling and inter-paragraph attention architectures (Liu and Lapata, 2019a), MMR-based attention models (Fabbri et al., 2019; Mao et al., 2020), and attention across representations of different granularity (words, sentences, and documents) (Jin et al., 2020). Prior work has also shown the advantages of customized optimization in multi-document summarization (e.g., RL Su et al., 2021). Such models are often dataset-specific and difficult to develop and adapt to other datasets or tasks.

Pre-trained Models for Summarization Pre-trained language models have been successfully applied to summarization, e.g., BERTSUM (Liu and Lapata, 2019b) for extractive summarization

and BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) for abstractive summarization. Instead of regular language modeling objectives, PEGA-SUS (Zhang et al., 2020) introduced a pre-training objective with focus on summarization, using Gap Sentence Generation, where the model is tasked to generate summary-worthy sentences. Outside of summarization, (Levine et al., 2020) show the benefits of using alternative masking (PMI-based strategy) instead of random masking in language modeling. Pre-training on title of HTMLs have been recently shown to be useful for few-shot summarization as well (Aghajanyan et al., 2021). However, it can only work with short summaries, as the titles of a webpage are usually short and highly compressed. All the aforementioned models are pre-trained on single-document datasets with a single document as input. Goodwin et al. (2020) evaluate three state-of-the-art models (BART, PEGA-SUS, T5) on several multi-document summarization datasets with low-resource settings, showing that despite the large improvement on single-document summarization, highly abstractive multi-document summarization remains challenging. Efficient pre-trained transformers (e.g., Longformer (Beltagy et al., 2020) and BigBird (Zaheer et al., 2020)) that can process long sequences have been also proven successful in summarization, typically by the ability to process long inputs, connecting information across the entire document. CDLM (Caciularu et al., 2021) is a follow up work for pre-training the Longformer model in cross-document setting using global attention on masked tokens during pre-training. However this model only addresses encoder-only tasks and not suitable for generation. In this work, we show how efficient transformers can be pre-trained using a task-inspired pre-training objective for multi-document summarization.

6 Conclusion and Future Work

In this paper, we propose PRIMER a pre-trained model with focus on the downstream task of multi-document summarization. PRIMER employs a new input structure for multi-document generation tasks and processes them with an efficient long-context transformer. It also uses a newly proposed pre-training objective based on masking salient sentences. PRIMER outperforms prior state-of-the-art pre-trained and dataset-specific models on 6 summarization datasets from 3 different domains, on zero, few-shot as well as full fine-tuning setting. Careful ablations demonstrate that both the input

structure and our pre-training strategy are critical to performance improvements.

Although PRIMER can be served as a zero-shot summarizer, we can only control the output length of generated summaries at inference time by specifying a length limit during decoding. Exploring a controllable generator in the future, in which the desired length can be injected as part of the input is a natural future direction. Besides the summarization task, we would like to explore using PRIMER for other generation tasks with multiple documents as input, like multi-hop question answering.

References

Armen Aghajanyan, Dmytro Okhonko, Mike Lewis, Mandar Joshi, Hu Xu, Gargi Ghosh, and Luke Zettlemoyer. 2021. HTLM: hyper-text pre-training and prompting of language models. *CoRR*, abs/2107.06955.

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

Jonathan Bragg, Arman Cohan, Kyle Lo, and Iz Beltagy. 2021. Flex: Unifying evaluation for few-shot nlp. *arXiv preprint arXiv:2107.07170*.

Avi Caciularu, Arman Cohan, Iz Beltagy, Matthew E. Peters, Arie Cattan, and Ido Dagan. 2021. Cross-document language modeling. *ArXiv*, abs/2101.00406.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2013. Towards coherent multi-document summarization. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1173, Atlanta, Georgia. Association for Computational Linguistics.

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.

Hoa Trang Dang. 2005. Overview of duc 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12.

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.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

Demian Gholipour Ghalandari, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim. 2020. A large-scale multi-document summarization dataset from the Wikipedia current events portal. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1302–1308, Online. Association for Computational Linguistics.

Travis Goodwin, Max Savery, and Dina Demner-Fushman. 2020. Flight of the PEGASUS? comparing transformers on few-shot and zero-shot multi-document abstractive summarization. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5640–5646, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Xiaotao Gu, Yuning Mao, Jiawei Han, Jialu Liu, You Wu, Cong Yu, Daniel Finnie, Hongkun Yu, Jiaqi Zhai, and Nicholas Zukoski. 2020. Generating representative headlines for news stories. In *Proceedings of The Web Conference 2020*, WWW '20, page 1773–1784, New York, NY, USA. Association for Computing Machinery.

Chris Hokamp, Demian Gholipour Ghalandari, Nghia The Pham, and John Glover. 2020. Dyne: Dynamic ensemble decoding for multi-document summarization. *CoRR*, abs/2006.08748.

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.

Dan Jurafsky and James H. Martin. 2009. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Prentice Hall, Upper Saddle River, N.J.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Yoav Levine, Barak Lenz, Opher Lieber, Omri Abend, Kevin Leyton-Brown, Moshe Tennenholtz, and Yoav Shoham. 2020. Pmi-masking: Principled masking of correlated spans. *arXiv preprint arXiv:2010.01825*.

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.

Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. 2020. Leveraging graph to improve abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6232–6243, Online. Association for Computational Linguistics.

Kexin Liao, Logan Lebanoff, and Fei Liu. 2018. Abstract Meaning Representation for multi-document summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1178–1190, Santa Fe, New Mexico, USA. 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.

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. 2019a. 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.

Yang Liu and Mirella Lapata. 2019b. Text summarization with pretrained encoders. 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 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yao Lu, Yue Dong, and Laurent Charlin. 2020. Multi-XScience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8068–8074, Online. Association for Computational Linguistics.

Yuning Mao, Yanru Qu, Yiqing Xie, Xiang Ren, and Jiawei Han. 2020. Multi-document summarization with maximal marginal relevance-guided reinforcement learning. *arXiv preprint arXiv:2010.00117*.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 145–152, Boston, Massachusetts, USA. 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.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Andy Su, Difei Su, John M Mulvey, and H Vincent Poor. 2021. Pobrl: Optimizing multi-document summarization by blending reinforcement learning policies. *arXiv preprint arXiv:2105.08244*.

Chien-Sheng Wu, Linqing Liu, Wenhao Liu, Pontus Stenetorp, and Caiming Xiong. 2021. Controllable abstractive dialogue summarization with sketch supervision. *arXiv preprint arXiv:2105.14064*.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, Vancouver, Canada. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. *CoRR*, abs/2007.14062.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Chenguang Zhu, Ziyi Yang, Robert Gmyr, Michael Zeng, and Xuedong Huang. 2019. Make lead bias in your favor: Zero-shot abstractive news summarization. *arXiv: Computation and Language*.

A Hyperparameters in Few-shot and Full Supervised Experiments

A.1 Few-shot Experiments

We use Adam as the optimizer with linear scheduled learning rate $3e-5$ for BART, LED and our model, and use the default optimization settings of the few-shot experiments from Zhang et al. (2020), i.e. AdaFactor optimizer with scheduled learning rate $5e-4$. For all the experiments with 10 examples, the batch size is 10, the models are trained for 200 steps, with warm-up as 20 steps. For the experiments with 100 examples, we use the same batch size, with the total step and warm-up step set to be 1000 and 100, respectively.

A.2 Fully Supervised Experiments

We use Adam as the optimizer with linear scheduled learning rate $3e-5$, and batch size as 16 for all the datasets in the full supervised experiments. The number of steps and warm-up steps are set based on the size of the datasets. The details can be found in Table 4

Dataset	Total Steps	Warmup Steps
Multi-News	25k	2.5k
Multi-XScience	20k	2k
WCEP	5k	.5k
arXiv	40k	4k

Table 4: Details of total steps and warm-up steps used in the Full Supervised experiments.

B Detailed Results in Few-shot Setting

The exact ROUGE scores in Figure 5 are shown in Table 5. And the results with variance is shown in Figure 7

C Detailed Analysis on Fully Supervised Experiments

To show the advantage of our pre-trained model when there is sufficient data, we also train the model with the full training set, and the results can be found in Table 6-9[19], along with the results from previous works. Differently from the zero-/few-shot experiments, here we report the state-of-the-art results on different datasets, as they were presented in the corresponding original papers. Since we use the same train/valid/test set as in those prior

---

[19]Due to the lack of computational resources, we do not train the model on Wikisum.

Model	0 Examples	10 Examples	100 Examples
Multi-News			
PEGASUS	31.97/10.06/16.74	39.02/12.10/20.32	42.99/13.50/21.10
BART	26.10/8.98/13.06	42.30/13.74/19.71	44.23/14.77/21.02
LED	16.60/4.78/9.05	38.86/12.48/18.82	44.45/14.85/21.16
Ours	**39.09/13.91/19.19**	**44.02/15.54/22.03**	**46.01/16.76/22.91**
Multi-Science			
PEGASUS	**27.33**/4.77/**15.04**	28.14/4.68/**15.49**	28.01/4.09/15.89
BART	15.21/3.49/8.61	27.80/4.74/14.90	31.17/5.32/16.45
LED	11.79/2.47/6.86	26.57/4.05/15.36	29.46/4.85/16.32
Ours	26.90/**4.98**/14.09	**28.36**/4.73/15.29	**31.25/5.43/16.84**
Wikisum			
PEGASUS	**23.67/5.37/14.17**	23.44/6.44/16.21	28.50/9.83/21.33
BART	15.80/4.60/9.13	28.95/9.88/20.80	32.97/13.81/25.01
LED	8.70/2.34/5.78	26.53/9.30/19.95	34.15/16.03/26.75
Ours	17.79/5.02/10.90	**31.10/13.26/23.39**	**36.05/17.85/27.81**
WCEP			
PEGASUS	**27.69/10.85/20.03**	35.60/14.84/26.84	42.09/19.93/33.04
BART	7.11/3.41/5.32	37.46/15.82/28.70	41.34/19.19/32.58
LED	5.69/2.19/4.32	36.29/15.04/27.80	41.83/19.46/32.92
Ours	13.50/5.30/10.11	**38.97/17.55/30.64**	**42.96/20.53/33.87**
arXiv			
PEGASUS	**29.76**/7.94/**17.27**	33.10/8.52/19.40	36.38/9.55/20.83
BART	23.26/7.57/12.01	32.53/8.70/17.98	37.62/10.78/20.99
LED	13.94/3.76/8.35	36.51/11.16/20.68	41.00/13.74/22.34
Ours	29.14/**8.64**/15.82	**41.13/13.81/23.02**	**43.42/15.85/24.07**

Table 5: Detailed ROUGE scores (R-1/R-2/R-L) on all the datasets in the few-shot setting (corresponds to Figure 5)

works, we can perform a fair comparison , without re-running all those extremely time-consuming experiments .

Overall, our model achieves state-of-the-art on Multi-News (see Table 6 , WCEP dataset (see Table 8) and arXiv dataset (see Table 9).

Models	ROUGE-1	ROUGE-2	ROUGE-L
PEGASUS (Zhang et al., 2020)	47.52	18.72	24.91
BART-Long-Graph (Pasunuru et al., 2021)	49.03	19.04	24.04
BART-Long-Graph(1000) (Pasunuru et al., 2021)	49.24	18.99	23.97
BART-Long(1000) (Pasunuru et al., 2021)	49.15	19.50	24.47
Ours	**49.94**	**21.05**	**25.85**

Table 6: ROUGE scores of the previous models and our fully supervised model on the Multi-News dataset. The results of PEGASUS is from Zhang et al. (2020), and the other results are from Pasunuru et al. (2021)

Multi-News The experiment results on Multi-News dataset can be found in Table 6. Specifically, the PEGASUS model (Zhang et al., 2020) is pre-trained on a large-scale single-document dataset with the Gap Sentence Generation objective, which is the same as ours, but with a different masking strategy, BART-Long (Pasunuru et al., 2021) uses the same model structure as ours , and BART-Long-Graph (Pasunuru et al., 2021) additionally has discourse graph injected. Comparing the results with the BART-Long model, our model is around 1 ROUGE point higher, which may result from either better model structure or pre-training.
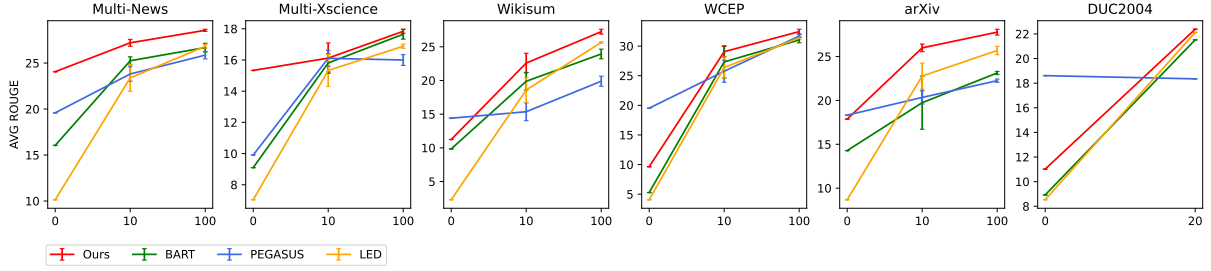
Figure 7: The AVG ROUGE scores (ROUGE-1, ROUGE-2 and ROUGE-L) of the pre-trained models with 0, 10 and 100 training data with variance. All the results of few-shot experiments (10 and 100) are obtained by the average of 5 random runs (with the same set of seeds shared by all the models). We only train the models on DUC2003 for one time, thus the variance is 0.

Interestingly, in one of the ablation studies in Pasunuru et al. (2021), they find that the BART-Long model achieves its best performance with the length limit of 1000, and no further improvement is found when the length limit is greater than that. Thus we may conclude the gap between the performances is mainly from our design on the model, i.e. the document separators, proper global attention as well as the pre-training on a multi-document dataset.

Models	R1	R2	RL*
LEAD	27.46	4.57	18.82
BERTABS	31.56	5.02	28.05
BART	32.83	6.36	26.61
SCIBERTABS	32.12	5.59	29.01
SOTA(Pointer Generator)	**34.11**	6.76	30.63
LEAD(ours)	26.49	4.26	14.70
Ours	31.93	**7.37**	18.02

Table 7: ROUGE scores of the previous models and our fully supervised model on the Multi-Xscience dataset. All the results are from Lu et al. (2020). * The ROUGE-L is not comparable as we have different settings on the settings of evaluation, see the gap between LEAD and LEAD(ours).

Models	R1	R2	RL
BERTREG (Gholipour Ghalandari et al., 2020)	35.0	13.5	25.5
SUBMODULAR+ABS(Gholipour Ghalandari et al., 2020)	30.6	10.1	21.4
DynE (Hokamp et al., 2020)	35.4	15.1	25.6
Ours	**46.08**	**25.21**	**37.86**

Table 8: ROUGE scores of the previous models and our fully supervised model on the WCEP dataset.

WCEP As for the WCEP dataet, BERTREG (Gholipour Ghalandari et al., 2020) is a Regression-based sentence ranking system with BERT embedding, which is used as extractive summarization method, while Submodular+Abs is a simple

two-step abstractive summarization model with a submodular-based extractive summarizer followed by a bottom-up abstractive summarizer (Gehrmann et al., 2018). DynE is a BART-based abstractive approach, which is to ensemble multiple input, allowing single document summarization models to be directly leveraged on the multi-document summarization task. Our model outperforms all the models by a large margin, including the SOTA model DynE, and it may indicate that the plain structure is more effective than purely ensembling the output of single documents.

Models	R1	R2	RL
PEGASUS (1K)	44.21	16.95	38.83
Bigbird-PEGASUS (3k)	46.63	19.02	41.77
LED(4K)	44.40	17.94	39.76
LED(16K)	46.63	19.62	41.83
Ours(4k)	**47.58**	**20.75**	**42.57**

Table 9: ROUGE scores of the previous models and our fully supervised model on the arXiv dataset. The result of PEGASUS and BigBird-PEGASUS are from (Zaheer et al., 2020), and the results of LED are from (Beltagy et al., 2020). The number in the parenthesis indicates the length limit of the input.

arXiv In addition to the experiments on multi-document summarization datasets, we also compare our fully supervised model with previous works on the arXiv dataset, with each section treated as a single document. All the models to be compared with are based on pre-trained models, and Bigbird-PEGASUS and LED utilize the pre-training of PEGASUS (Zaheer et al., 2020) and BART (Lewis et al., 2020), respectively. However, both Bigbird and LED apply more efficient attentions, which make the models able to take longer

input (3k for BigBird, 4K and 16k for LED). Our model has a better performance than all the models, including LED(16K), which allows for the input 4 times longer than ours. It is worth mentioning that LED(4K) has the same structure as our model, with the same length limit of the input, and with the pre-training on multi-document datasets, our model is more than 3 ROUGE point better than it, which shows that the strategy not only works for multi-document summarization but can also effectively improve single-document summarization for long documents.