Are Large Language Models Capable of domain-specific Text Summarization?

Anonymous EMNLP submission

Abstract

Abstractive text summarization and several state-of-the-art summarization models have gained considerable interest in recent years. All these models, however, are usually benchmarked against a general-purpose corpus, and their performance on domain-specific text summarization is yet to be determined. This paper presents an overview of some representative large language models (LLMs) based on the research gaps they address and then categorizes them based on their usability guidelines and design principles. We also selected three open-source text summarization datasets, chosen based on their domain complexity, providing a unified framework for assessing various LLMs in specialized domains. We evaluate contemporary models against the selected datasets while trying to optimize each model for the best performance using their usability guidelines. Our experiments show that PEGASUS-X, an Efficient Transformer fine-tuned on a 16K context window outperforms all other LLMs including direct inference on GPT 3.5. Additionally, we observed that increasing the context window only slightly increases the model performance and corroborates the fact that bigger models do perform better. This study serves as a crucial resource for researchers aiming to develop and compare large language models for domain-specific abstractive summarization.

1 Introduction

Abstractive Text Summarization has been an active research area in the past years, and while state-of-the-art models can produce human competitive summaries, they are more suitable for general-purpose text. The performance of these models deteriorates when tested on a domain-specific text summarization task. One common explanation is the shift in the dataset distribution as most of the large language models (LLMs) are pre-trained on general-purpose corpora such as C4 (Raffel et al., 2020a), and hence do not fully comprehend the fine-grained linguistic details and concepts of a niche area such as the medical, scientific, or legal domain.

Apart from the domain-adaptation capabilities, an additional challenge in abstractive summarization is the associated large document size (Afzal et al., 2023). Most of the text that needs to be summarized is large in size, and basic text summarization models cannot handle it because of the input size limitation of 512 or 1024 tokens. A simple workaround has been truncating the input text, leading to a loss in context size that hinders the model's performance. At this time, GPT-3.5 [1] offers a 16K token context window, and GPT-4 (OpenAI, 2023) up to a 32K context window. However, both of these models are closed-domain and only accessible through an API.

Over the years, several models suitable for the abstractive text summarization task have been released, each following a different design principle and usability guidelines. Firstly, we had the transformer-based Seq2Seq models like T5 (Raffel et al., 2020b) and BART (Lewis et al., 2019), depicting a classic encoder-decoder architecture while being pre-trained on a large corpus and later fine-tuned on a smaller domain-specific dataset. Despite showing great performance, these models still suffer from the quadratic complexity emerging from the self-attention matrix and are thus limited to handling only 512 or 1024 tokens, respectively. An initial attempt to reduce the quadratic complexity was illustrated in the architectures employed by the Efficient Transformers (Tay et al., 2022) family. Longformer-Encoder-Decoder (Beltagy et al., 2020) or BigBirdPegasus (Zaheer et al., 2021) with a sparse self-attention matrix scaled the input length up to 4096 tokens. However, the most recent architectures like LongT5 (Guo et al., 2022) and Pegasus-X (Phang et al., 2022), utiliz-

[1] https://platform.openai.com/docs/models/gpt-3-5

ing the same approach, scaled the input text length limitation up to 16K tokens, while still, mostly, preserving model performance.

While there is no denying the above models' abilities, their performance on domain-specific data and in general their domain-adaptation capabilities are yet to be evaluated. This paper intends to evaluate one representative model of each class on their domain-specific text summarization capabilities while taking into account their usability guidelines such as fine-tuning or direct inference. Nevertheless, given the recent surge in the number of LLMs, we felt it to be appropriate to take several models into consideration, differing in model size, context size, and overall architecture. In general, vanilla Seq2Seq models such as BART, BigBirdPegasus, and PEGASUS-X are meant to be fine-tuned on a downstream task. On the other hand, GPT-like models are more suitable for direct inference or in-context learning approaches (Brown et al., 2020).

Additionally, we propose a set of datasets against which we evaluate our models, providing a standard benchmark to evaluate model performance on domain-specific summarization. We select these datasets based on their large document size and the specificity of the textual domain represented. We further elaborate on this benchmark in section 4. Through our experiments, we tried to answer the following two theoretical questions:

1. Does allowing more text as input improve the quality of the generated summary for the domain-specific text summarization task?

2. Are ChatGPT-like LLMs, that are not meant to be fine-tuned, able to perform competitively on a domain-specific summarization task?

Finally, we present a taxonomy in which we categorize text summarization models into standard Encoder-Decoder Transformer models, Efficient Transformers, and GPT-like models (LLMs) with billions of parameters. We compare the performance between these categories by experimenting with some representative models as explained in section 5.

2 Background

2.1 Quadratic Complexity of Transformers

Since the introduction of the original Transformer architecture by Vaswani et al. (2017), its attention mechanism has become a cornerstone for numerous state-of-the-art natural language processing models, since it represents a vast increase in performance and efficiency compared to the traditional LSTMs (Hochreiter and Schmidhuber, 1997). However, despite how successful these models have become, they maintain quadratic complexity in the attention module, leading to severe computational challenges when working with large documents pervasive in our environment (e.g. books, research articles, and legal documents, among others).

2.2 Large Language Models

The history of LLMs showcases a steady and remarkable evolution. Their capabilities have significantly expanded over time due to increased model size, larger datasets, and a plethora of algorithmic innovations. The groundbreaking work by Vaswani et al. (2017) presented the Transformer model, which introduced the self-attention mechanism, enabling models to consider long-range dependencies in text and initiating a new era in natural language processing. These models are trained with the simple objective of predicting the next word given a specific context, which quite surprisingly is sufficient to promote quite impressive reasoning and writing abilities, provided that enough scale is in play.

This realization led to an escalating trend towards larger models. Work like GPT-4 (OpenAI, 2023) and PaLM (Chowdhery et al., 2022) expanded on Transformer's capabilities, being trained on enormous text corpora and showcasing impressive performance on a broad set of natural language understanding and generation tasks. They showed remarkable zero-shot and few-shot learning capabilities, leading to a paradigm shift in how we approach task-specific training, foregoing fine-tuning task-specific models and instead relying on a larger, general, language model.

2.3 Efficient Transformers

On the other hand, the original Transformer architecture has issues scaling to larger token counts due to the novel attention mechanism itself. To address this, researchers have proposed a plethora of efficient models which aim to reduce the quadratic nature of attention to a linear basis. Furthermore, they can be roughly clustered (Tay et al., 2022) based on their optimization approaches which can differ quite substantially. Some noteworthy examples include making clever use of memory access patterns with FLASH attention (Dao et al., 2022), explicitly learning attention patterns (Tay et al., 2020a; Kitaev et al., 2020), computing a low-rank representation of the attention matrix Choromanski et al., 2022; Wang et al., 2020 and the computation of fixed local and/or global attention patterns (Zhu et al., 2021; Beltagy et al., 2020; Zaheer et al., 2021).

Naturally, these differ in implementation complexity and hardware compute efficiency, making the standalone evaluation of their performance troublesome. Regardless, released attempts at benchmarking (Zhang et al., 2022; Xiong et al., 2022b) these optimizations show a key takeaway: local attention modules with fixed or almost fixed attention patterns, which focus on computing attention against adjacent tokens, have overshadowed some of the more complex attention patterns listed above which attempt to approximate the global attention matrix. This suggests that the information present in the neighboring tokens is mostly sufficient to achieve strong performance in downstream tasks.

Furthermore, when considering contemporary models, we can effectively verify which optimizations have withstood the test of time by observing which of them persist in the efficient adaptations of previously well-received models such as PegasusX (Phang et al., 2022), BART-LS (Xiong et al., 2022a), LongT5 (Guo et al., 2022).

Not surprisingly, these "proven" optimizations coincide with most of the attention benchmark findings (see, for example, Phang et al. (2022) and its staggered block-wise attention mechanism similar to the aforementioned fixed attention patterns). Following this conclusion, our model selection, discussed in a later section, attempts to reflect the attention module timeline discussed here.

2.4 Transfer Learning

Since it takes lots of time and hardware resources to train a large language model, Transfer Learning allows us to reuse the pre-trained model weights for specific tasks/domains instead of starting from scratch. In general, this paper explores Transfer Learning from a domain-adaptation point of view. This is possible in the form of continued pre-training of the existing weights, fine-tuning a few selected layers for a new task/domain, or through in-context learning which tries to localize and identify the relevant embedding space by using the additional context from the prompt. In addition, since we are focusing on domain-specific language, we will further evaluate how model performance differs when the model is tasked to summarize documents with a lexical corpus different from what is available in its pre-training process, compared to the performance observed after the fine-tuning procedure. Moreover, recent work (Hu et al., 2021; Mao et al., 2022) has been successful at exploring a more parameter-efficient method of domain adaptation which we would like to explore, but leave as a future work direction, sticking to the traditional approach with the hyperparameters detailed in Appendix A.

3 Related Work

Benchmarking LLMs is not a novel idea, however, after a thorough literature review, we found existing publications either to be too broad for our intended goal or focused on a parallel aspect. Furthermore, to the best of our knowledge, these models have not been benchmarked on a domain-specific text summarization task, thus we intend to evaluate if these models are suited for those who are dependent on the specificity of their data and its overall length. This paper should provide a uniform overview of what models perform best in this scenario. We will proceed to mention some of the publications that inspired our work.

Long Range Arena (LRA) (Tay et al., 2020b). Widely accepted as a significant contribution, particularly due to the growing number of efficient transformer models being introduced and the need to assess their performance. Although LRA is extensive, we feel that it is lacking in the sense that it only covers datasets related to general reasoning tasks, such as the hierarchical mathematical reasoning dataset ListOps (Nangia and Bowman, 2018) and image classification using the CIFAR-10 dataset (Krizhevsky, 2009). Additionally, the benchmark only covers the encoder-based model. While this is helpful in capturing the models' general scope of understanding and generalizing, it fails to focus on the language generation capabilities of the models, which is our main concern.

SCROLLS (Shaham et al., 2022). The Benchmark, focusing on the overall Natural Language Generation capabilities of LLMs, is the most similar to our research. It attempts to benchmark the performance of Efficient Transformers in tasks similar to the ones used in pre-training, such as span corruption from the original T5 model (Raffel et al., 2020b). While the SCROLLS paper focuses on a variety of tasks, we focus only on the summarization task, as it holds relevance for several industry-related use cases. Additionally, the SCROLLS benchmark evaluates only the Efficient Transformers with long-range capabilities, whereas we also include the latest LLMs which have surged in popularity.

An Examination of Large Language Models (Zhao et al., 2023). A survey following the development and significance of large language models (LLMs). Tracing the progression from statistical language models to today's sophisticated LLMs, it aligns with the historic relevance and evolution of our study. The survey places emphasis on the unanticipated emerging capabilities of LLMs, such as in-context learning, which are non-existent in their smaller counterparts, aligning with our attempt to study how increased size improves summarization performance.

4 Benchmark

4.1 Datasets

To evaluate the performance of each model and how it varies given different context lengths, we have selected three datasets given the specificity of their domains and overall general features. Furthermore, below is a brief summary of each, along with a detailed length analysis in Table 1.

arXiv (Cohan et al., 2018). Based on scientific articles from the arXiv platform, this dataset uses abstracts as a reference summary which ensures high-quality human-written summaries. In addition, as articles are often long and come from a complex lexical domain, they present themselves as an ideal medium for the long-range context transformer evaluation we intend to accomplish.

PubMed (Cohan et al., 2018). Similarly to arXiv, PubMed focuses on the scientific domain, albeit with a much narrower scope, focusing only on medical publications. All in all, we include it in the benchmark despite sharing the same structure with arXiv, in the sense that we also aim to evaluate these models' domain-adaptation ability.

GovReport (Huang et al., 2021). Stemming from the reports of government meetings, GovReport is an interesting addition to the benchmark as both the summaries and original texts are significantly longer than the other datasets, as observed in table 1. Moreover, per the authors, GovReport summaries source the relevant bigrams from a larger portion of the original text compared to the other datasets, further enabling our analysis of the relationship between model performance and encoding length.

Dataset	# Doc	# W	# Sum W
arXiv	215,913	6029.9	272.7
PubMed	133,215	3049.9	204.4
GovReport	19,466	9409.4	553.4

Table 1: **Dataset Size Analysis**. Where relevant, averages are reported for each dataset. **# Doc** refers to the number of documents, **# W** and **# Sum W** refers to the number of words in the original text and summaries, respectively.

4.2 Preprocessing and filtering

In order to ensure quality and consistency, we reproduce the SCROLLS (Shaham et al., 2022) preprocessing procedure by removing samples meeting the following criteria:

1. The summary text is longer than half of the original text.

2. The original text is a thousand times longer than the summary.

3. The summary exists verbatim in the original text.

Additionally, and as is to be expected, this removed only a small number of samples given the datasets' inherent quality and prefiltering performed by their authors. Nonetheless, further details on the number of removed samples can be found in table 2, where we can verify that at most 4% of the samples were removed, a small enough percentage that we argue the datasets' overall characteristics were maintained.

	# Samples		
Dataset	Train	Del	% Del
arXiv	203,037	6253	3%
PubMed	119,924	4439	4%
GovReport	17,517	63	0.4%

Table 2: **Preprocessing statistics**. We report the number of samples in the training split of the dataset before and after the preprocessing procedure, along with the percentage of samples removed.

4.3 Models

As per the motivation given in the background and related work sections, and given the large number of tokens in our datasets, we have chosen models able to handle these samples efficiently. Moreover, we think our selection should reflect the release timeline of these new architectures to illustrate progress and the expressiveness of the benchmark.

With these thoughts in mind, we have chosen BART (Lewis et al., 2019) as a baseline model and compared it with BigBirdPegasus (Zaheer et al., 2021) and PegasusX (Phang et al., 2022), both possessing long-range capabilities. Additionally, we compare these representative models with state-of-the-art LLMs including LLaMA (Touvron et al., 2023) and its derivatives vicuna, chatGPT with GPT 3.5 (OpenAI., 2022) as the backbone and lastly Falcon (Almazrouei et al., 2023). Since all of these models are much different in size and architecture, we tried to optimize each model to be the best version of itself while following the usability guidelines. We discuss all these models in their respective subsections below, but we have also summarized the models in Figure 1.

4.3.1 BART

Lewis et al. (2019) is a combination of two ideas and architectures that followed the original transformer proposal. For the encoder, it makes use of a BERT-style (Devlin et al., 2019) procedure, obtaining embeddings by reconstructing masked-out tokens in the input sentence. Meanwhile, the decoder segment is identical to the GPT-like decoder found in most LLMs.

Furthermore, due to its early popularity as a summarization model for short-form text like news articles in XSUM (Narayan et al., 2018), we felt it was natural to include it as a baseline for the evaluation of other contemporary models.

4.3.2 BigBirdPegasus

Zaheer et al. (2021) appears as a modification of the attention module proposed by Ainslie et al. (2020) with the inclusion of randomness in the attention pattern, allowing select tokens to randomly attend to others. Furthermore, as demonstrated theoretically by the authors, this pattern serves as an approximation to the full attention matrix while preserving linearity with respect to the input size.

Moreover, the model itself is akin to a Pegasus model, the differentiating factor remains the special attention module introduced here. We choose to include BigBirdPegasus due to it being one of the first models in the efficient transformer class that claimed state-of-the-art results when it was first published.

4.3.3 PegasusX

Phang et al. (2022) perform an extensive investigation of how to best adapt transformer models to long sequence data. Among other issues, they investigate whether an adaptation is more successful by performing additional pretraining over large documents, only using these large documents for pretraining or disregarding them entirely until finetuning for downstream tasks, finding that these models benefit from further pretraining even if it's only for a relatively small portion of the training samples.

Furthermore, the authors suggest a variation of the local attention architecture pattern we have discussed before: by padding the blockwise attention by half a block in every other layer, they effectively can introduce dependencies between blocks that would otherwise be self-contained while not increasing the implementation complexity. Together with the global tokens, this attention architecture allows the model to perform competitively in both short and long-sequence summarizations.

4.3.4 GPT-3.5

A major revelation in the current LLM landscape is the instruction fine-tuning approach that led to the explosion in popularity of the ChatGPT[2] platform and its model predecessor, InstructGPT (Ouyang et al., 2022). By leveraging Reinforcement Learning from Human Feedback (RLHF), as introduced in Ziegler et al. (2020), these models can follow arbitrary instructions, making them suitable for a downstream summarization task. Nevertheless, this model has a large performance bottleneck in its small context length, allowing it to encode only up to 4k tokens.

In this publication, we are using the version based on GPT-3.5, since we have not been given access to the larger and more powerful GPT-4 version. Although the architecture of this model is private and we cannot accurately compare it to models of the same size, we felt that its inclusion in our evaluation suite is natural as it represents the best contemporary capabilities of (assumed) reasonably sized models.

[2] https://chat.openai.com/

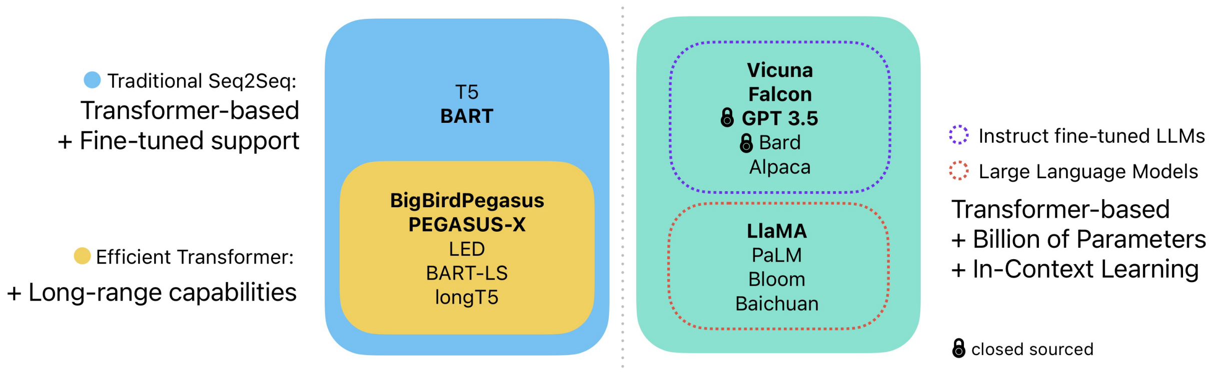

Figure 1: A taxonomy over some representative LLMs suitable for a Text Summarization task where the bold text indicates the models included in our experiments.

4.3.5 LLaMa and Derivatives

The LLaMa (Touvron et al., 2023) family of language models was introduced as a competing foundational LLM to the GPT family. We provide evaluation data on the 7 and 13 billion parameter versions to further demonstrate different summarization performances across different model sizes.

Moreover, a direct comparison to GPT-3.5 and the remaining Seq2Seq models would be unfair given the lack of any instruction-fine-tuning on the LLaMa models. To this effect, we also evaluate Vicuna (Chiang et al., 2023), a model derived from LLaMa by fine-tuning it on data collected from user conversations with the ChatGPT platform, a method that has proven incredibly effective at instruction-fine-tuning. Other reasonable options for instruction-fine-tuned LLaMa derivates might as well be Alpaca (Taori et al., 2023) and WizardLM (Xu et al., 2023), which are derived from different fine-tuning datasets. We choose Vicuna since it promises better performance on reasoning benchmarks such as MMLU (Hendrycks et al., 2021), HellaSwag (Zellers et al., 2019), and the AI2 Reasoning Challenge (Clark et al., 2018).

Also, as is the case with the above model, LLaMa is only capable of handling up to 2K tokens of context, making it extremely handicapped in a long-document summarization situation.

4.3.6 Falcon

Falcon-40B (Almazrouei et al., 2023) is a new entry into the LLM space. It does not bring breakthrough innovations when compared to LLaMa, however, it demonstrates impressive comprehensive abilities, even outperforming LLaMa's 65B version on the benchmarks described above.

Their differences come mostly from the training data used. This model has been trained on a portion of the RefinedWeb (Penedo et al., 2023) dataset augmented with curated text inspired by The Pile (Gao et al., 2020), while LLaMa uses a dataset which, albeit detailed in the original publication, has not been publicly released.

Finally, for evaluation, we use the instruction fine-tuned version of Falcon with both 7 and 40 billion parameters, which, akin to the above model, suffers from a limited 2k tokens context window.

4.4 Metrics

While there has been much discussion on the appropriateness of the Rouge (Lin, 2004) score for automatic evaluation of summarization systems (Fabbri et al., 2021; Graham, 2015; Ng and Abrecht, 2015), mostly due to it being n-gram based and thus not dealing properly with different expressions conveying the same sentiment, it is still the most (and only) reported metric in new model publications and benchmarks.

This is mostly due to the lack of superior alternatives with METEOR (Banerjee and Lavie, 2005) and BLEU (Papineni et al., 2002) suffering from the same n-gram-based fate of failing to capture paraphrases. On the other hand, the recently proposed BERTScore (Zhang et al., 2020) avoids this problem by computing embedding similarity between generated and original texts.

Nevertheless, according to the findings in Koto et al. (2021), the correlation between BERTScore and human evaluation of generated summaries for English text is similar to Rouge. As a result, we have opted to focus on the established Rouge, rather than BERTScore. We report both the obtained ROUGE-1, ROUGE-2, ROUGE-L scores and the geometric mean between ROUGE-1, ROUGE-2, and ROUGE-L, similar to the procedure in other publications.

5 Experiments

As proposed, we evaluate the above models on the previously described datasets. With respect to the models, we first create a distinction between the models that are meant to be fine-tuned and the ones that are to be used out of the box.

In the section below, we provide technical details and model configurations related to fine-tuning and inference.

5.1 Fine-tuning

Given the input size limitations, the vanilla Seq2seq BART is fine-tuned on its maximum input context of 1024. The Efficient transformer BigBirdPegasus is fine-tuned to its maximum input length of 4096 tokens. PEGASUS-X, which supports up to 16384 tokens is fine-tuned on 4096 tokens as well as 8192 tokens to evaluate the effect of longer context on the abstract summarization task. We fine-tuned all the Seq2Seq models for a number of epochs dependent on dataset size and convergence level. Further details can be found in Appendix A. After fine-tuning, we perform inference and use the corresponding ROUGE score for the final evaluation.

5.2 Inference

In order to evaluate the models' performance, we run inference in a Seq2Seq fashion after the fine-tuning procedure for the Efficient Transformer class.

Inference in the LLM models is not trivial since the usual fine-tuning is too computationally demanding and the usual in-context learning paradigm is not suited for the summarization task. Even a single document doesn't fit in the whole context window, making it impossible to provide an example sample. Given the above reasoning, we decide to evaluate these LLMs by prompting them to summarize the provided content appropriately. More details can be found in Appendix A.

6 Results and Discussion

As explained in the experiments section, we distinguish models that should be fine-tuned and those that present good results as-is. By fine-tuning BART, BigBirdPegasus, and PEGASUS-X with different configurations, we have obtained different versions of the models for our evaluation purposes. We also make use of the original model weights without any fine-tuning for analysis. For the remaining LLMs that were meant to be used out-of-the-box, we performed direct inference.

Additionally, we have reported the sample summaries generated by some of the models for the same input text in Appendix B. While we use the ROUGE score as the main indicator of performance, this appendix section provides some additional insight into the model's performance than the one provided by automatic evaluation.

We report results with both ROUGE-1, ROUGE-2, ROUGE-L and the geometric mean of ROUGE-{1,2,L} for all models evaluated with the three datasets detailed previously. While we discuss the key findings from our experiments in the later part of this section, the results are summarized in Table 3.

Efficient Transformers remain competitive via fine-tuning: from a bird's eye view, it is clear that the Efficient Transformers, namely BigBird-Pegasus, and PEGASUS-X, are clear winners as they consistently perform better in terms of ROUGE scores. These are impressive results given the much smaller size and computational requirements of these models, as compared to the state-of-the-art LLMs. Furthermore, as evident in Appendix B, the summaries generated by PEGASUS-X and BigBird-Pegasus, essentially the seq2seq models fine-tuned on the same domain, produce summaries that are more in line with the technical language of the paper. Whereas the ones generated by LLMs like chatGPT use simpler words in the summaries. However, we cannot neglect the additional effort and costs required due to the need for fine-tuning over a specific dataset, as models without fine-tuning perform much worse than their fine-tuned counterparts. Nevertheless, for an industrial or production setting, a smaller model like an Efficient Transformer might be a better choice.

Longer Context Windows have their downsides: for the models that support larger context windows such as PEGASUS-X and GPT-3.5, scaling the context window to 16k does increase their ROUGE scores, albeit only marginally in most cases. A possible explanation for this phenomenon is that the relevant text for a high-quality summarization isn't evenly distributed in the source document, thus further context has diminishing returns. Furthermore, given the fact that increasing the context window length directly increases the training/inference time as well as memory requirements, we can argue that in light of the marginally better ROUGE scores, for resource-constrained environments and particular dataset distributions, scaling the input length may not be the ideal choice.

Bigger Models do perform better: while it is a known fact in the LLM community that bigger models perform better up to a certain degree, we confirm this to be the case in our limited experiment set. We compare two of the most prominent open-source models, LLaMa (7b vs 13b) and Falcon (7b vs 40b) and, as expected, the larger variant performs better in both cases. Additionally, GPT-3.5 outperforms both Falcon and LLaMa models. While the exact size of GPT-3.5 is unknown, we do know that GPT-3 has 175B parameters and therefore assume the 3.5 variant to be, at least, bigger than Falcon's 40B parameters.

GPT-3.5 outperforms other LLMs: among all the LLMs in our domain-specific text summarization study, GPT-3.5 with a 16k context window seems to perform the best in terms of ROUGE score. Although we used only a portion of the full datasets, given the use of random sampling (more details in Appendix A), reported scores should be indicative of model performance on the overall datasets. Concluding, while the others are competitive, this model emerges as a strong and versatile option for summarization applications, despite the privacy concerns related to its closed-source nature.

Limitations

Despite our best attempt to provide an overview of LLMs with regard to their ability to understand domain-specific text, several dimensions of the study could not be explored. A major cause for this is the hardware restrictions. Although we had access to high-quality hardware, its availability was scarce, forcing us to use only one or two GPUs at a time. This limitation made it so we could not test the larger LLMs which promise the best overall performance in other tasks than summarization.

Another hindrance from the lack of hardware availability: we intended to evaluate performance using the latest domain-adaptation methods, such as adapters (Houlsby et al., 2019) and LORAs (Hu et al., 2021) that make it possible to fine-tune these large models on downstream tasks. Exploring this paradigm would be ideal since the usual LLM in-context learning is impossible for long-document summarization: the size of the documents makes it so even one document is hard to fit in the predefined model context length, therefore providing more examples for guidance is impossible.

On the other hand, we also would like to include GPT-4 (OpenAI, 2023) as the latest and greatest LLM but its (current) exclusive API access and large associated costs were prohibitive. Together with its maximum 32k context length and human-level comprehensive abilities, we imagine this model to have very competitive performance with the finetuned Seq2Seq models, all without the need for an expensive training step and for deploying several models for various downstream tasks. This is illustrated by the impressive performance of GPT-3.5 with a 16k context length.

Finally, we mention the lack of expressiveness in the ROUGE metric which is not ideal for an abstractive summarization setting. We have mentioned before how it is a poor proxy of human perception of summarization quality, which is shown by the high ROUGE scores of the standard BART model without any fine-tuning. Inspecting the model's outputs, we notice how often it simply repeats the original text. This coincidentally is similar to summaries, given that the introduction section usually provides a reasonable overview of the text. In the future, we hope to leverage new metrics that are more in line with what humans perceive as high-quality summaries. Additionally, we also wish to study the effectiveness of these automatic evaluation scores by using human evaluation as a baseline.

Ethics Statement

Throughout our experiments, we strictly adhere to the ACL Code of Ethics. Since we used already established open-source benchmark datasets, the concern of privacy does not apply. Furthermore, since no additional data was collected or stored, and no human annotators were used in the experiment, we minimized the risk of prejudice. Through our finetuning strategies, no additional bias was introduced into the models, other than what might already be part of the model weights or the benchmark dataset. The goal of the research was to evaluate the text summarization capabilities of existing models. The results and discussions in this paper are meant to further promote research in the area of domain-specific language modeling with an over-arching goal of bridging the gap between academia and application. All training scripts and trained models will be made available to the research community.

Model	Size	Tuned	Input	Datasets		
				PubMed	**arXiv**	**GovReport**
Classical Transformers						
BART	140m	×	1024	33.99 / 23.57 / 23.57 - 23.57	34.36 / 34.36 / 34.36 - 23.57	49.46 / 49.46 / 49.46 - 23.57
BART	140m	✓	1024	13.72 / 0.37 / 5.59 - 3.05	15.68 / 0.43 / 6.15 - 3.46	10.65 / 0.05 / 4.76 - 1.37
Efficient Transformers						
BigBirdPegasus	577m	×	4096	23.57 / 5.57 / 15.08 - 12.55	24.51 / 5.61 / 15.82 - 12.96	27.45 / 7.73 / 15.78 - 14.97
BigBirdPegasus	577m	✓	4096	45.11 / 19.67 / 27.51 - 29.00	43.09 / 16.77 / 26.32 - 26.69	48.61 / 20.47 / 24.76 - 29.10
PEGASUS-X	569m	✓	4096	44.77 / 19.38 / 27.41 - 28.76	45.05 / 18.14 / 27.18 - 28.11	52.91 / 23.30 / 25.55 - 32.00
PEGASUS-X	569m	✓	8192	46.95 / 22.00 / 29.37 - 31.19	46.48 / 19.42 / 28.23 - 29.43	55.55 / 25.45 / 28.05 - 34.10
PEGASUS-X	569m	×	16384	2.67 / 0.23 / 2.44 - 1.14	5.77 / 0.85 / 5.06 - 2.92	6.89 / 0.86 / 5.27 - 3.14
PEGASUS-X*	569m	✓	16384	**51.00 / 24.7 / 46.6 - 38.9**	**50.00 / 21.8 / 44.6 - 36.5**	**60.30 / 30.00 / 31.50 - 38.5**
Large Language Models						
LLaMA	7b	×	2048	9.45 / 0.55 / 6.50 - 3.24	13.02 / 0.56 / 8.75 - 3.99	14.35 / 1.43 / 8.09 - 5.50
LLaMA	13b	×	2048	21.31 / 4.79 / 10.36 - 10.18	34.57 / 11.14 / 20.32 - 19.9	31.42 / 6.16 / 11.98 - 13.24
Instruction Fine-tuned Large Language Models						
Falcon	7b	×	2048	36.53 / 9.54 / 18.23 - 18.52	34.40 / 10.43 / 18.23 - 18.70	27.32 / 3.60 / 11.96 - 10.56
Vicuna	13b	×	2048	30.48 / 9.03 / 16.29 - 16.48	39.24 / 15.77 / 22.68 - 24.12	31.19 / 8.68 / 15.06 - 15.97
Falcon	40b	×	2048	29.68 / 8.15 / 17.13 - 16.06	32.65 / 10.32 / 17.50 - 18.06	44.89 / 10.63 / 16.56 - 19.92
GPT-3.5	-	×	4096	42.88 / 15.19 / 23.44 - 24.81	42.80 / 14.34 / 22.40 - 23.95	38.53 / 14.88 / 19.51 - 22.36
GPT-3.5**	-	×	16384	43.34 / 15.81 / 23.95 - 25.41	43.76 / 15.30 / 23.43 - 25.03	39.93 / 16.07 / 20.24 - 23.51

Table 3: ROUGE scores of all models in the format ROUGE-1 / ROUGE-2 / ROUGE-L - geometric mean of ROUGE-{1,2,L} computed in inference across all three benchmark datasets. * implies that results have been taken from the original Pegasus-X publication. ** implies that only a portion of each dataset was used.

Acknowledgements

References

Anum Afzal, Juraj Vladika, Daniel Braun, and Florian Matthes. 2023. Challenges in domain-specific abstractive summarization and how to overcome them. pages 682–689.

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. Etc: Encoding long and structured inputs in transformers.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. 2022. Rethinking attention with performers.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge.

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.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating Summarization Evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling.

Yvette Graham. 2015. Re-evaluating automatic summarization with BLEU and 192 shades of ROUGE. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 128–137, Lisbon, Portugal. Association for Computational Linguistics.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. Longt5: Efficient text-to-text transformer for long sequences.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer.

Fajri Koto, Jey Han Lau, and Timothy Baldwin. 2021. Evaluating the efficacy of summarization evaluation across languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 801–812, Online. Association for Computational Linguistics.

Alex Krizhevsky. 2009. Learning multiple layers of features from tiny images.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

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.

Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Wen tau Yih, and Madian Khabsa. 2022. Unipelt: A unified framework for parameter-efficient language model tuning.

Nikita Nangia and Samuel R. Bowman. 2018. Listops: A diagnostic dataset for latent tree learning.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization.

Jun-Ping Ng and Viktoria Abrecht. 2015. Better summarization evaluation with word embeddings for rouge.

OpenAI. 2022. Gpt-3.5 (version 3.5).

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only.

Jason Phang, Yao Zhao, and Peter J. Liu. 2022. Investigating efficiently extending transformers for long input summarization.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020a. Exploring the limits of transfer learning with a unified text-to-text transformer.

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

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022. Scrolls: Standardized comparison over long language sequences.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. 2020a. Sparse sinkhorn attention.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2020b. Long range arena: A benchmark for efficient transformers.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. 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.

Wenhan Xiong, Anchit Gupta, Shubham Toshniwal, Yashar Mehdad, and Wen tau Yih. 2022a. Adapting pretrained text-to-text models for long text sequences.

Wenhan Xiong, Barlas Oğuz, Anchit Gupta, Xilun Chen, Diana Liskovich, Omer Levy, Wen tau Yih, and Yashar Mehdad. 2022b. Simple local attentions remain competitive for long-context tasks.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2021. Big bird: Transformers for longer sequences.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence?

Jun Zhang, Shuyang Jiang, Jiangtao Feng, Lin Zheng, and Lingpeng Kong. 2022. Cab: Comprehensive attention benchmarking on long sequence modeling.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models.

Chen Zhu, Wei Ping, Chaowei Xiao, Mohammad Shoeybi, Tom Goldstein, Anima Anandkumar, and Bryan Catanzaro. 2021. Long-short transformer: Efficient transformers for language and vision.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences.

A Training Details

A.1 Training

The fine-tuning procedure was done by leveraging 2 Nvidia A100-80GB GPUs, relying on the HuggingFace *Transformers* (Wolf et al., 2020) and Microsoft *Deepspeed*[3] libraries for distributed training. Furthermore, we plan on releasing the fine-tuned models along with the codebase used in our study.

Moreover, hyperparameters for the above training run are described in Table 4, and the configuration for Deepspeed Stage 2 can be found in Table 5. In this setting, all values set to *auto* are automatically filled by the HuggingFace Trainer according to the user-provided or default values if none are set.

A.2 Inference

For inference, we rely on a single Nvidia A100-80GB which is capable of handling our models in the `bfloat16` format. The one exception is Falcon-40B which required loading the model in an 8bit quantized fashion utilizing the *bitsandbytes* (Dettmers et al., 2022) library, we consider possible performance losses due to this approach mostly insignificant as the obtained ROUGE scores lie in the expected range. The GPT-3.5 model was evaluated using the API made available from OpenAI[4], where we utilized the latest snapshot available, in this case, *gpt-3.5-turbo-0613* dated to June 13th, 2023.

As was the case for the training procedure, we sample a maximum of 256 tokens for the arXiv and PubMed datasets, while scaling to 1024 tokens for the GovReport dataset, as is standard procedure in other contemporary publications.

Also, regarding the prompt utilized to perform inference on the open-source LLM models, we show it in Figure 2. In order to make sure the *"SUMMARY: "* portion is included in the context window, we always force it as a suffix as the last input tokens.

Finally, for the GPT model, we used the system message, *"You are an expert at summarization. Proceed to summarize the following text"*, followed by the maximum portion of the source document able to fit in the model's context window. What's more, with the 16k context-length experiments we could only experiment with a sample of the overall datasets. We set the randomization seed in PyTorch to 1337 to ensure our experiments are reproducible and promptly select a 10% slice of the full dataset for analysis.

[3] `https://github.com/microsoft/DeepSpeed`
[4] `https://platform.openai.com/`

B Sample Output

Summaries for the sample (idx 42) from the test split of pubmed dataset.

PEGASUS-X fine-tuned on 8192 tokens:
multiple endocrine neoplasia (men) type 2a and b syndromes are rare but significant endocrine diseases that are increasingly encountered by pediatricians. mtc is a relatively slow - growing tumor but, if metastasized or relapsed, it becomes very aggressive causing more than 13% of all thyroid cancer - related mortality. mtc is usually diagnosed in the childhood, and the majority of mtc diagnosed in the childhood is hereditary. mtc diagnosed during childhood almost always results from a dominantly inherited or de novo activating mutations in the ret proto - oncogene, which encodes the ret receptor tyrosine kinase. advances in predictive genetic testing for ret mutations have enabled early diagnosis of hereditary men syndromes and prophylactic thyroidectomy in presymptomatic patients to prevent mtc. in this review, we discuss the etiology of pediatric mtc and currently available therapeutic modality for the cancer. \n

GPT 3.5 direct inference on 4192 tokens:

Thyroid cancer is a common type of cancer that accounts for about 1% of all cancers. Medullary thyroid cancer (MTC) is a type of thyroid cancer that is slow-growing but can become aggressive if it spreads or relapses. It is more common in children and is usually hereditary. MTC is caused by mutations in the RET proto-oncogene. Early diagnosis of hereditary MTC through genetic testing has enabled preventive measures such as thyroidectomy in presymptomatic patients. The main treatment for MTC is surgical removal of the thyroid, and adjuvant treatment options are limited. However, newer targeted therapies such as vandetanib and cabozantinib have shown promise in clinical trials. Prophylactic thyroidectomy at an early age is recommended for children with hereditary MTC to reduce the risk of developing aggressive MTC.

Dataset	Batch Size	Learning Rate	Epochs	Input Tokens	Gen. Tokens	Beam Size
			BART			
arXiv	128	8e-4	4	1024	256	1
PubMed	128	8e-4	4	1024	256	1
GovReport	128	8e-4	8	1024	1024	1
			BigBirdPegasus			
arXiv	64	8e-4	4	4096	256	1
PubMed	64	8e-4	4	4096	256	1
GovReport	64	8e-4	8	4096	1024	1
			PEGASUS-X			
arXiv	64	8e-4	4	4096 / 8192	256	1
PubMed	64	8e-4	4	4096 / 8192	256	1
GovReport	64	8e-4	8	4096 / 8192	1024	1

Table 4: Hyperparameters for fine-tuning models on selected datasets

Key	Value
bf16.enabled	auto
optimizer.type	AdamW
optimizer.params.lr	auto
optimizer.params.betas	auto
optimizer.params.eps	auto
optimizer.params.weight_decay	auto
scheduler.type	WarmupLR
scheduler.params.warmup_min_lr	auto
scheduler.params.warmup_max_lr	auto
scheduler.params.warmup_num_steps	auto
zero_optimization.stage	2
zero_optimization.offload_optimizer.device	cpu
zero_optimization.offload_optimizer.pin_memory	true
zero_optimization.allgather_partitions	true
zero_optimization.allgather_bucket_size	2e8
zero_optimization.overlap_comm	true
zero_optimization.reduce_scatter	true
zero_optimization.reduce_bucket_size	2e8
zero_optimization.contiguous_gradients	true
gradient_accumulation_steps	auto
gradient_clipping	auto
steps_per_print	2000
train_batch_size	auto
train_micro_batch_size_per_gpu	auto
wall_clock_breakdown	false
zero_allow_untested_optimizer	true

Table 5: Deepspeed configuration parameters. This table directly maps to a JSON file.

```
You are an expert at summarization. Proceed to summarize the following text:

TEXT: {source_document}

SUMMARY:
```

Figure 2: Prompt used to evaluate the Large Language Models on the summarization datasets.

BigBird-Pegasus fine-tuned on 4192 tokens:

thyroid cancer (mtc) is the most common endocrine neoplasia. mtc is characterized by the development of a highly penetrant, autosomal dominant, and highly aggressive form of mtc. the most common genetic mutation in mtc is the ret mutation. the men2a subtype is characterized by the early development of an aggressive form of mtc in all affected individuals, typically during the first year of life. the men2b subtype is characterized by the early development of an aggressive form of mtc in all affected individuals, typically during the first year of life. the men2a subtype, accounting for 9095% of the men type 2 cases, is a highly penetrant, autosomal dominant endocrine tumor syndrome characterized by the development of mtc in > 90% of ret mutation carriers. the men2b subtype, accounting for approximately 510% of the men type 2 cases, is characterized by the early development of an aggressive form of mtc in all affected individuals, typically during the first year of life. the men2a subtype, accounting for 9095% of the men type 2 cases, is closely associated with men2a, demanding a genetic screening for men2 \n'