MDCure: A Scalable Pipeline for Multi-Document Instruction-Following

Gabrielle Kaili-May Liu[1] **Bowen Shi**[1] **Avi Caciularu**[2]
Idan Szpektor[2] **Arman Cohan**[1]

[1]Yale University [2]Google Research
{kaili.liu, arman.cohan}@yale.edu

Abstract

Multi-document (MD) processing is crucial for LLMs to handle real-world tasks such as summarization and question-answering across large sets of documents. While LLMs have improved at processing long inputs, MD contexts still present unique difficulties, including management of inter-document dependencies, redundancy, and incoherent structures. To address this challenge, we introduce MDCure, a scalable and effective instruction data generation framework to enhance the MD capabilities of LLMs without the computational cost of pre-training or reliance on human-annotated data. MDCure generates high-quality synthetic MD instruction data over sets of articles via targeted prompts. We also introduce MDCureRM, a cost-effective, MD-specific reward model to score and filter generated data based on their training utility for MD settings. MDCure is compatible with open- and closed-source models in addition to policy optimization methods such as PPO, enabling even small open-source models to surpass proprietary LLMs as strong generators of high-quality MD instruction data without further data filtering. With MDCure, we fine-tune a wide variety of LLMs up to 70B parameters in size from the FlanT5, Qwen2, and LLAMA3.1 model families. Extensive evaluations on a wide range of MD and long-context benchmarks spanning various tasks and domains show MDCure consistently improves performance over pre-trained baselines and base models by up to 75.1%.

1 Introduction

With the rapid expansion of bodies of online texts in domains of science, finance, education, news, and more, multi-document (MD) processing becomes a critical aspect of LLMs' ability to understand and reason over large quantities of text information, with applications such as (query-focused) multi-document summarization (MDS) (Xiao et al., 2022; Giorgi et al., 2023), multi-hop question-answering (MHQA) (Yang et al., 2018), and cross-document reasoning (Cattan et al., 2021b).

While LLMs can now process tens to hundreds of thousands of tokens, they struggle with the unique demands of multi-document understanding and reasoning: finding and aggregating information across diverse documents, resolving contradictions, handling redundant information, bridging information gaps, and constructing coherent narratives (Hosseini et al., 2024; Lior et al., 2024; Tang et al., 2024; Wang et al., 2024d). Current approaches to improving MD capabilities primarily rely on continued pre-training of long-context models on related document sets (Caciularu et al., 2021; Yasunaga et al., 2022; Caciularu et al., 2023; Peper et al., 2024). However, such methods are not extensible to broader tasks and require large amounts of pre-training data.

We propose MDCure, an effective and scalable framework for creating high-quality synthetic datasets designed to prioritize MD capabilities of LLMs and adapt LLMs for MD settings. MDCure builds upon recent work showing the value of small-scale, high-quality supervised data for efficiently improving LMs' downstream task performance (Zhou et al., 2023; Sachdeva et al., 2024; Tirumala et al., 2024), in addition to addressing the challenges of data scarcity and complexity in MD tasks. While curating organic, human-created training datasets might seem plausible, such approaches are often prohibitively expensive (Cobbe et al., 2021a; Yue et al., 2024) and suffer from limited scope, failing to capture complex reasoning patterns involved in MD settings (Srivastava et al., 2023; Chen et al., 2024c). Synthetic data leveraging frontier LLMs is becoming increasingly essential in post-training of open-source models (Abdin et al., 2024; Lambert et al., 2024). Our proposed MDCure is the first framework for creation of high-quality and MD-focused post-training datasets. Not only does MDCure achieve strong

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 29258–29296
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

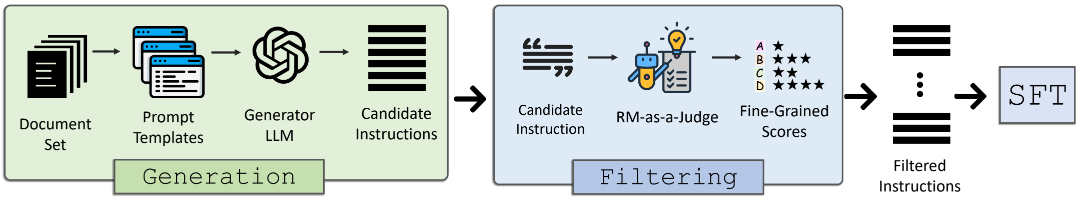


Figure 1: The MDCure pipeline generates diverse multi-document instructions, filters them using fine-grained scoring from MDCureRM, and tunes a base LLM to enhance its multi-document capabilities.

cross-task and cross-domain generalization in long-context MD settings, it also offers practical value due to its compatibility with both cost-effective commercial LLMs (e.g., GPT-3.5) and moderately-sized open-source LLMs (e.g., LLAMA3.1-70B).

MDCure divides the process to automatically curate high-quality MD instruction data into two phases. During *Generation*, zero-shot prompt templates are used to create complex, cross-text instruction-following prompt-response pairs from a set of related documents provided as input context. During *Filtering*, a multi-objective, MD-specific reward model, MDCureRM, is trained to assess the generated instructions based on targeted criteria to ensure high-quality, diverse, MD-focused samples. We find that MDCureRM is pivotal to improving the constitution of the resulting instruction dataset. Versus frontier LLMs, MDCureRM overcomes limitations on the tradeoff between cost and quality, offering greater enhancements to MD data quality while also reducing filtering costs. Moreover, as we show, MDCureRM seamlessly integrates with policy optimization methods such as PPO to iteratively improve synthetic MD instruction data generation policies, enabling even small open-source models such as LLAMA3.1-8B-Instruct to surpass SOTA proprietary models as strong generators of high-quality MD instruction data without further filtering or data curation.

We showcase the effectiveness of MDCure through extensive experiments on 6 MD and long-context benchmarks in over 6 content domains. To analyze the impact of MDCure at scale, we create MDCure instruction datasets of size 12K, 36K, and 72K to fine-tune FlanT5 (Chung et al., 2022), Qwen2-Instruct (Yang et al., 2024), and LLAMA3.1-Instruct (Dubey et al., 2024) models up to 70B parameters in size. Our results show that MDCure enables models to consistently achieve superior performance versus existing MD pre-training approaches, and to generalize well across diverse tasks and texts. Further, LLMs post-trained using our MDCure datasets consistently outperform prior instruction-following LLMs and strong long-context baselines, in both zero- and few-shot settings, with gains of up to **75.1%** average improvement. Our contributions are as follows:

- MDCure, a novel framework for obtaining high-quality, MD-focused post-training datasets to improve MD instruction-following capabilities of LLMs, which achieves strong results with both open-source and proprietary cost-effective LLMs.
- MDCureRM, a novel evaluator reward model specifically designed for the MD setting, to quality-filter MD instruction data more successfully and inexpensively versus proprietary LLMs.
- A suite of instruction data complementary to collections such as FLAN (Longpre et al., 2023) for improving MD task performance of any LLM.[1]

2 Related Work

Multi-Document Modeling. A common approach for MD modeling with LLMs is to perform flat concatenation of all input documents to process with a long-context model (e.g., Longformer (Beltagy et al., 2020), CoLT5 (Ainslie et al., 2023)). However, such models lack sufficient skill at synthesizing cross-document information (Caciularu et al., 2022; Wolhandler et al., 2022; Chen et al., 2023; Hosseini et al., 2024). More recently, pre-training-based approaches have proven valuable and involve (continual) pre-training of long-context LMs on instances formulated over sets of related documents using masking and infilling objectives (Zhang et al., 2020; Caciularu et al., 2021; Xiao et al., 2022; Yasunaga et al., 2022). Other notable recent works leverage LLMs to better perform targeted content selection for MDS and QMDS (Caciularu et al., 2023; Kurisinkel and chen, 2023; Kurisinkel and Chen, 2023; Li et al., 2023a; Peper et al., 2024). In contrast, our MDCure is, to our knowledge, the first systematic framework for high-quality MD-

[1] We release our code at https://github.com/yale-nlp/MDCure.

specific instruction generation with LLMs to overcome the computational expense of pre-training, the limitations of heuristic-based data generation, and limited generalization of MD performance.

Synthetic Instruction Generation. Instruction tuning (IT) aligns LMs for diverse tasks via supervised instruction-answer pairs but is hindered by (1) the difficulty of obtaining high-quality instructions (Song et al., 2024; Ziegler et al., 2024), and (2) unavailability of high-quality open-source instruction data (Lambert et al., 2024). To address this, recent works have focused on leveraging proprietary LLMs for synthetic data generation (Honovich et al., 2023; Wang et al., 2023; Chen et al., 2024a). Most of these efforts target general *single-document* tasks. A few, such as Chen et al. (2024c); Lupidi et al. (2024), target MHQA by generating and merging single-hop and two-hop QA pairs. While effective, these approaches focus on QA without generalizing to broader MD tasks and rely on complex frameworks. In contrast, MDCure targets MD tasks broadly and enables cross-task generalization, while providing a simpler, more scalable solution to generate complex MD instruction data. As synthetic data can be noisy, data selection using LLMs can be useful in filtering low quality data (Yue et al., 2024; Ziegler et al., 2024). Recent approaches also leverage fine-grained rewards for task-specific data selection (Wang et al., 2024c). Building on this, we introduce MDCureRM, a multi-objective MD-specific reward model that refines instruction data effectively and cost-efficiently and which serves well both as a standalone judge and in RLAIF (Bai et al., 2022; Lee et al., 2024) settings. Overall, with our MDCure framework, we construct and release open-source, high-quality instruction-following datasets that enables significant improvements on MD tasks. As we show, MDCure is not dependent on any specific LLM, works well with both open and proprietary models, and generalizes to many tasks.

3 MDCure

MDCure (Fig. 1) utilizes a two-step approach to produce high-quality MD instruction samples over sets of related documents. These samples can be used via IT to improve base LLMs' MD abilities.

3.1 Generation Phase

During Generation, MDCure performs targeted construction of candidate instructions for a given set of documents by sampling from a set of carefully tailored zero-shot templates (App. B.2) to prompt a generator LLM. Adopting a principled approach, we design our prompt templates to encourage cross-document reasoning by requiring answers that synthesize information from multiple documents, with wide template variety ensuring diverse task formulations, ranging from single-word answers to detailed summaries. This yields synthetic MD data that reflects real-world complexity and encourages LLMs to leverage deep connections among texts to strengthen cross-document comprehension skills. Versus Caciularu et al. (2023), our instructions are more diverse and open-ended. Qualitative examples and diversity analysis of our MDCure instructions[2] are provided in App.s B.5 and B.6, respectively. A systematic analysis of the effect of different prompt approaches on the quality of resulting generations is provided in §A.1.

Dataset Source Selection. MDCure's generation focuses on constructing candidate instructions from sets of documents, which may be sourced from existing datasets of related texts or created by clustering general text corpora. In our experiments, we use topically related document sets from the news domain to model cross-text relationships, building on prior work that shows training with related document sequences can enhance model performance (Caciularu et al., 2021, 2023; Shi et al., 2023). Nonetheless, our generation procedure is equally functional over sets of unrelated documents, which may be utilized to target creation of instruction samples concerning processing of conflicting or irrelevant cross-document information. We base our generated instructions on the NewSHead dataset (Gu et al., 2020). Additional details are in App. B.1.

Generator Model. MDCure is not generally dependent on any specific generator LLM. In combination with our proposed filtering procedure (§3.2), any strong instruction-following LLM can be used to construct effective instruction data. We utilize GPT-3.5-Turbo in experiments to balance generation quality and cost. We also show the compatibility and comparable efficacy of MDCure with use of open-source generator LLAMA3.1-70B in §5.4.[3]

[2] Details regarding instruction format are in App. B.4.

[3] Other cost efficient LLMs such as GPT-4o-mini or Gemini 1.5 Flash and open-source LLMs such as LLAMA3.3-70B can also be used; our generation pipeline was done before the release of these more cost-efficient models.

3.2 Filtering Phase

The generation prompts, while carefully designed, can still yield noisy or unsuitable instructions for MD settings. To address this, we apply a model-based filtering phase to remove low-quality, non-factual, or single-document instructions. Inspired by recent work on fine-grained RMs (Wu et al., 2023; Wang et al., 2024b,c), we train a fine-grained, *MD-specific* reward model, MDCureRM, to evaluate each instruction-answer pair based on six different criteria. These criteria capture both the overall quality of the instruction-response pairs and their effectiveness in handling multi-document content and are depicted in Fig. 2. This fine-grained RM-as-a-Judge mechanism refines the final curated instruction set to yield high-quality training data. We analyze the criticality of each MDCureRM scoring criterion and the effect of their relative weightings via ablation studies in App. A.2. A meta-evaluation to analyze the alignment of MDCureRM with human annotations versus GPT-3.5-Turbo is provided in App. E, where we observe agreements in line with LLM evaluation literature (Liu et al., 2024a). §5.5 further explores the efficacy of using reward signals provided by MDCureRM to train a custom MD instruction generator model via PPO without the need for post-generation filtering.

Training Setup. We adopt a multi-objective reward modeling framework (Wu et al., 2023; Wang et al., 2024b) to train MDCureRM. Traditionally, reward models for LLM alignment are trained with Bradley-Terry loss (Bradley and Terry, 1952) on pairwise data with binary preference annotations. However, we include the fine-grained information present in multi-objective ratings rather than binarizing the data, found to be effective in Wang et al. (2024c). Each training example for MDCureRM consists of a prompt, response, and a 6-dimensional rating vector, with each dimension corresponding to a specific reward objective. We apply multi-objective regression to learn from these ratings, utilizing a regression loss function.

Data. Training data for MDCureRM is obtained using an LLM-based pipeline without requiring human annotation. We first use GPT-4o-mini (OpenAI et al., 2024) and Mistral-7B-Instruct-v0.2 (Jiang et al., 2023) to generate MD instruction-answer pairs of varying quality.[4] This results in roughly 20,000 data points. We then prompt GPT-4o to assign scores for each sample according to the six criteria, to be utilized as annotations to train the reward model. The specific prompt used to do so can be found in App. B.8. The target reward scores are normalized to the range $[0, 1]$. Although using GPT-4o directly as the RM is also possible, it is prohibitively costly to scale. Training MDCureRM ensures both cost-effectiveness and scalability for filtering large quantities of long-context data.

`Fine-Grained Scoring Criteria`

- **Context Integration**: How well does the instruction synthesize information from multiple documents?
- **Inter-Document Relationships**: Does the instruction help understand relationships between documents?
- **Complexity**: Does the instruction challenge critical thinking and synthesis from multiple sources?
- **Relevance**: Is the instruction aligned with the document content?
- **Coherence & Factuality**: Is the response logical, factual, and well-supported by the documents?
- **Creativity**: How diverse or creative is the instruction in question type and format?

Figure 2: Fine-grained scoring criteria utilized by MDCureRM for RM-as-a-Judge evaluation of candidate instruction quality in the Filtering phase of MDCure.

Implementation. MDCureRM utilizes the Llama3-8B architecture, with weights initialized from a Bradley-Terry reward model (Xiong et al., 2024) trained from Llama3-8B-Instruct (Dubey et al., 2024). The original output layer is replaced with a linear regression layer, outputting a 6-dimensional rating. The regression head processes the final-layer hidden states. We use Mean Squared Error (MSE) between target and predicted rewards as the loss function, freezing the base model during training, following Wang et al. (2024c). We discuss alternative architectures and the design of MDCureRM in App. C.2.

Data Filtering. During inference, MDCureRM generates a 6-element rating for each candidate instruction. These scores are re-scaled to a 1-5 range and combined in a weighted average (details in App. B.9). The top N highest-scoring samples are selected as the final instructions.

4 Experimental Setup

We conduct extensive experiments to evaluate the efficacy of MDCure for improving multi-document modeling capabilities of LLMs. In particular, we provide evidence for the following:

- The scalability of MDCure as we vary the size of the generated instruction data and the size of the

[4] We select these models due to their strong instruction-following capabilities, while considering the generation costs.

base LLM being post-trained.

- The importance of MDCureRM filtering for curating high-quality MD instruction data.
- The strong cross-task and cross-domain generalization and superior results achieved by MDCure over pre-trained and long-context baselines.
- The ability for MDCure data to excel over existing synthetic MD *pre-training* datasets.
- The adaptability of MDCure for few-shot and extended-context MD instruction data generation.
- The compatibility and efficacy of MDCure with open- and closed-source generator LLMs.
- The value of MDCureRM toward providing reward signals for synthetic MD data policy optimization.

Models and Training Details. We apply MDCure to LLMs across different widely used model families and sizes: FlanT5, Qwen2, and LLAMA3.1. We choose these models as they are capable open-source instruction-following models with varying architectures and parameter scales. We utilize the instruction-tuned variants as our base models to showcase the complementary effect of MDCure instructions versus typical IT data. To assess MDCure at scale, we create MDCure instruction datasets of size 12000, 36000, and 72000 examples[5] and use them to fine-tune FlanT5-Base and -Large (250M & 750M), Qwen2-Instruct (1.5B & 7B), and LLAMA3.1-Instruct (8B & 70B). Hereafter, we refer to resulting models that have undergone MDCure post-training as "MDCure'd" models. To assess the efficacy of MDCureRM, we additionally repeat these experiments using unfiltered and GPT-3.5-filtered instruction sets.[6] Further data and training setup details are in App.s B and C.1.

Baselines. Beyond their base model counterparts, our MDCure'd models are compared against state-of-the-art pre-trained models for multi-document processing, namely PRIMERA (Xiao et al., 2022) and QAMDEN (Caciularu et al., 2023), which are comparable in scale to FlanT5-Base and -Large, as well as several strong open-source long-context LLMs, namely LongAlign-7B (Bai et al., 2024) and ProLong-8B-64k-Instruct (Gao et al., 2024b), which are comparable to Qwen2-7B-Instruct and LLAMA3.1-8B-Instruct, and Jamba-1.5B-Mini (Labs, 2024) which has 54B parameters. We also compare MDCure'd models to SOTA general LLMs GPT-4o and Gemini 1.5 Pro (Team et al., 2024). Additionally, we inspect the effect of our instruction data versus baseline MD and long-context *pre-training* data at the 7B- and 8B-scales by fine-tuning Qwen2-7B-Instruct with PRIMERA and QAMDEN data and ProLong-8B-64k-Instruct with MDCure data (72k). Further details regarding baselines are in App. D.2.

Benchmarks. We evaluate all baselines and MDCure'd models over a range of long- and short-form tasks through use of two challenging MD and long-context benchmarks: SEAM (Lior et al., 2024) and ZeroScrolls (Shaham et al., 2023). Within SEAM, we specifically consider the MultiNews, OpenAsp, MuSiQue, ECB+, and SciCo datasets to ensure coverage of a spread of domains and MD tasks. We additionally profile 0-shot prompting performance over a suite of four widely used MD benchmarks: WikiHop (Welbl et al., 2018) and HotpotQA-distractor (Yang et al., 2018) for MHQA, Multi-XScience (Lu et al., 2020) for MDS, and QMDSCNN (Pasunuru et al., 2021) for QMDS. While these datasets were built for use in fine-tuning evaluation, we draw upon the prompting approach proposed by Shaham et al. (2023) to formulate effective 0-shot prompts for each model. App.s D.1 and D.3 provide additional details.

Metrics. For SEAM and ZeroScrolls we employ the metrics and corresponding implementations defined by each for their component datasets. For WikiHop and HotpotQA-distractor, we use the standard F1 score. For Multi-XScience and QMDSCNN, we utilize LLM scores issued by GPT-3.5-Turbo and Gemini 1.5 Flash, respectively, to assess summarization coherence, relevance, fluency, and consistency. This follows recent adoption of LLM-as-a-Judge as a proxy for human-level rating (Liu et al., 2024b; Ye et al., 2024), given the limitations of traditional metrics like ROUGE for long-form generation quality assessment. (Sai et al., 2019; Gao et al., 2024a). Further details regarding the evaluator LLMs and the prompts are in App. D.4.

5 Results

5.1 Main Results

We report experimental results in Table 1, with full results on ZeroScrolls included in App. F. For fair comparison, we report all MDCure results based on training with 72K samples, which led to best

[5] Dataset sizes were determined as a function of API costs and based on the sizes of representative IT datasets used in related works (Bai et al., 2024; Köksal et al., 2023).

[6] We utilize GPT-3.5-Turbo for comparison as it is a more cost-effective alternative to commonly used scorer GPT4. Cheaper models such as GPT-4o-mini had not yet been released at the time of our starting experimentation.

		Model / IT Data Setting	HQA	WH	MXS	QC	SEAM						ZS Score	Avg
							ECB+	MN	MSQ	OA	SC	Avg		
FLANT5	Base	PRIMERA	0.4	0.5	70.7	24.2	7.1	7.7	0.5	3.9	10.4	**5.9**	4.2	8.8
		QAMDEN	1.8	1.9	63.6	27.1	0.0	0.4	0.0	1.0	3.1	0.9	3.8	7.2
		None	4.4	45.1	38.7	48.0	0.0	5.8	0.2	2.6	0.1	1.7	13.1	14.5
		Unfiltered	31.7	47.5	89.8	52.2	0.0	6.7	0.1	2.7	0.2	1.9	21.0	23.2
		GPT-Filtered	44.1	46.0	92.4	54.2	0.0	6.0	0.2	2.7	0.0	1.8	21.3	24.1
		MDCureRM	**47.3**	**48.3**	**93.8**	**57.3**	0.0	7.5	0.3	2.7	0.0	2.1	**22.6**	**25.4**
	Large	None	24.4	54.6	70.9	61.8	1.1	7.9	0.1	3.1	0.3	2.5	23.2	24.0
		Unfiltered	46.9	62.9	91.1	64.7	0.7	8.0	0.0	3.8	0.1	2.5	24.2	27.3
		GPT-Filtered	46.3	65.1	91.8	64.0	0.5	8.3	0.0	3.9	0.1	2.6	24.2	27.5
		MDCureRM	**49.6**	**66.1**	**93.1**	**66.0**	1.2	9.1	0.1	4.2	0.0	**2.9**	**25.3**	**28.5**
QWEN2-INS	1.5B	None	21.5	17.8	93.3	73.3	9.4	10.6	0.5	5.0	13.0	7.7	24.0	25.5
		Unfiltered	32.9	30.5	94.2	79.3	15.7	12.0	0.4	5.0	16.8	10.0	23.9	27.7
		GPT-Filtered	33.3	32.9	94.2	81.3	16.1	12.0	0.5	5.1	15.9	9.9	24.2	28.1
		MDCureRM	**37.7**	**34.8**	**94.4**	**82.9**	16.4	12.0	0.6	5.9	18.1	**10.6**	**25.6**	**29.4**
	7B	LongAlign-7B	10.4	14.3	92.2	83.3	11.5	16.5	0.0	4.1	16.8	9.8	23.2	25.3
		None	30.5	39.6	95.6	79.3	5.0	11.9	0.5	6.4	13.1	7.4	23.9	27.4
		Unfiltered	40.5	43.3	94.7	84.2	8.0	15.3	0.5	6.6	11.9	8.5	26.3	29.9
		GPT-Filtered	42.0	44.0	94.7	85.3	8.7	15.3	0.9	6.6	12.1	8.7	28.7	31.4
		MDCureRM	**44.7**	**46.0**	**95.1**	**87.3**	13.8	15.4	0.6	6.7	14.9	**10.3**	**29.8**	**32.7**
LLAMA3.1-INS	8B	ProLong-8B	43.6	34.3	85.8	54.7	9.5	17.4	0.4	10.7	18.0	11.2	32.2	32.1
		None	35.5	27.1	95.1	65.3	10.5	15.0	0.6	7.6	17.4	10.2	18.7	24.3
		Unfiltered	37.6	34.4	84.7	90.4	15.3	16.3	0.5	7.8	18.9	11.8	28.6	31.1
		GPT-Filtered	38.0	42.3	95.3	87.8	8.7	16.0	0.6	7.8	18.3	10.3	29.6	32.1
		MDCureRM	**44.7**	**43.7**	**95.3**	**93.8**	16.3	16.4	0.6	7.9	18.5	**11.9**	**30.9**	**34.0**
	70B	Jamba 1.5 Mini	47.5	41.8	94.2	87.1	20.1	14.2	0.0	5.3	20.4	12.0	34.1	35.3
		None	53.9	38.1	95.1	88.2	25.1	21.9	0.6	6.1	11.5	13.0	36.4	37.1
		Unfiltered	55.9	40.6	88.7	70.4	3.6	21.9	0.6	6.3	11.2	8.7	34.9	34.1
		GPT-Filtered	57.4	41.2	88.2	74.9	4.9	22.0	0.7	6.4	10.7	8.9	37.5	35.9
		MDCureRM	**58.4**	**45.5**	**95.1**	**88.7**	25.2	22.1	0.7	6.7	12.0	**13.3**	**37.7**	**38.5**
		GPT-4o	57.5	50.0	**100.0**	**94.4**	8.9	18.6	0.3	14.2	28.9	14.1	**39.8**	**40.6**
		Gemini 1.5 Pro	**66.6**	**55.6**	93.8	79.8	21.4	17.8	0.6	15.1	30.1	**17.0**	32.0	36.9

Table 1: Evaluation of MDCure'd models versus baselines on 6 benchmarks in the zero-shot prompting setting. The rightmost "Avg" column reports the average of individual dataset scores. Dataset abbreviations are described in App. D.1.1. Full results on ZeroScrolls are provided in App. F. Rows specified by "MDCureRM" refer to our full MDCure pipeline applied to the corresponding base model and size. Size-comparable baselines are highlighted in blue and results of our final method is highlighted in yellow. Bold numbers show best performance in each group.

performance overall across all models. Our key findings are summarized as follows.

MDCure instructions are effective across model families and sizes. MDCure exhibits the best performance across all base models, demonstrating a clear and consistent advantage relative to non-MDCure'd base models over all benchmarks. In particular, we see improvements of up to 975%, 96%, 143%, 434%, 39%, and 73% for HotpotQA, WikiHop, Multi-XScience, QMDSCNN, SEAM, and ZeroScrolls, respectively. MDCure'd LLAMA3.1-70B-Instruct achieves the best results across benchmarks, followed by MDCure'd Qwen2-7B-Instruct. As our models are trained from -instruct versions of base LLMs, it is clear that MDCure instruction data imparts complementary abilities versus typical IT data.

The relative improvement of MDCure'd models versus corresponding base models varies across model families, and improvements tend to shrink with model size. For FlanT5, MDCure provides 75.1% average improvement at the 250M-parameter scale (FlanT5-Base) versus 18.9% at the 750M-parameter scale (FlanT5-Large). For LLAMA3.1, MDCure provides 40.2% average improvement at the 8B scale versus 3.8% at the 70B scale. For Qwen2, we observe alternatively that MDCure gains shift from 15.4% on average at the 1.5B scale to 19.4% at the 7B scale. More generally, these results suggest the importance of instruction quality may weaken with base model scale and capability, similar to findings by Ivison et al. (2023).

Filtering with MDCureRM is essential to improve MD capabilities of LLMs. Comparing to

Model	HQA	WH	MXS	QC	SEAM	ZSS	Avg
ProLong (PL)	43.6	34.3	85.8	54.7	11.2	33.6	32.1
PL+MDCure	**45.8**	**45.1**	**87.3**	**75.6**	**12.5**	**34.7**	**34.9**

Table 2: MDCure performance when applied to an already strong long-context LLM, ProLong-8B.

Data Source	HQA	WH	MXS	QC	SEAM	ZSS	Avg
Qwen2-7B-Ins	30.5	39.6	95.6	79.3	7.4	23.9	27.4
+MDCure	**44.7**	**46.0**	**95.1**	**87.3**	**10.3**	**29.8**	**32.7**
+PRIMERA	44.6	45.4	88.0	48.2	8.1	27.9	28.7
+QAMDen	42.6	45.8	87.1	52.7	8.8	27.2	28.6

Table 3: Efficacy of MDCure data versus synthetic MD pre-training data, applied to Qwen2-7B-Instruct.

models trained with unfiltered MDCure instruction samples, MDCure'd models trained with the highest-scoring MDCureRM-filtered instructions are superior across all benchmarks and base models. Furthermore, MDCureRM outperforms GPT-3.5-Turbo as a judge of MD data quality and efficacy, evidenced by MDCure'd models superiority over those trained with GPT-filtered instructions. As use of naive, unfiltered instructions yields relatively little improvement to MD capabilities especially at larger base model scales, this suggests curation of high-quality synthetic MD data is a nontrivial goal which MDCure is able to efficaciously accomplish.

MDCure improves performance on both MD and single-document long-context tasks, enabling cross-task and cross-domain generalization to unseen tasks and domains. Across MD benchmarks, MDCure enables models to achieve improved performance in all settings, with strong generalization to out-of-distribution (OOD) tasks such as MD coreference resolution, MD classification, text reordering, and aggregated sentiment classification and to OOD domains such as science, literature, and media. Beyond MD tasks, MDCure generally boosts single-document long-context performance, particularly on tasks in ZeroScrolls, which emphasize long-context abilities. These trends are consistent across downstream task domains, suggesting MDCure data boosts MD performance in a domain-general way.

MDCure consistently outperforms pre-trained long-context baselines. Our MDCure'd models achieve results that exceed the performance of size-comparable baselines[7] trained on much longer inputs (64-256K tokens) for all benchmarks by a substantial margin, confirming the efficacy of our design. For a controlled setup, we apply MDCure to strong open-source long-context LLM Prolong-8B (Gao et al., 2024a) to study if our data provides additional advantages to long-context continual pre-training and fine-tuning. Table 2 demonstrates the results. Note that Prolong-8B has undergone extensive training on ∼81B tokens to improve long-context performance. Fine-tuning ProLong-8B with much less MDCure data (∼162M tokens) yields further improvement, verifying the added long-context value of our data.

# IT Samples	HQA	WH	MXS	QC	SEAM	ZSS
FlanT5-Base						
None	4.4	45.1	38.7	48.0	1.7	13.1
12K RM-Filtered	43.8	**48.4**	92.4	54.0	1.9	22.0
36K RM-Filtered	45.7	46.1	92.7	56.7	1.9	22.2
72K RM-Filtered	**47.3**	48.3	**93.8**	**57.3**	**2.1**	**22.6**
72K Unfiltered	31.7	47.5	89.8	52.2	1.9	21.0
Qwen2-7B-Instruct						
None	30.5	39.6	95.6	79.3	7.4	23.9
12K RM-Filtered	44.1	45.4	94.7	82.7	8.9	25.9
36K RM-Filtered	44.4	45.8	94.7	84.7	8.6	28.1
72K RM-Filtered	**44.7**	**46.0**	**95.1**	**87.3**	**10.3**	**29.8**
72K Unfiltered	40.5	43.3	94.7	82.2	8.5	26.3

Table 4: Performance at different training data scales.

Our MDCure pipeline improves over existing MD synthetic pre-training data. We also compare our MDCure pipeline with existing MD-focused synthetic pre-training datasets inluding PRIMERA and QAMDen. As indicated in Table 3, MDCure data offers performance improvements across benchmarks at the 7B-scale. The weak performance of the original PRIMERA and QAMDen models (Table 1) can be explained as they are intended to serve as strong base pre-trained models for finetuning as opposed to zero-shot prompting settings. Overall, our results emphasize the utility of our framework as a powerful approach for improving MD abilities of LLMs via post-training.

5.2 Influence of Data Scale

To understand the importance of data quantity, we compare the impact of fine-tuning with MDCure instruction datasets of various sizes. We utilize FlanT5-Base and Qwen2-7B-Instruct as base models to demonstrate the effect of data size at different model scales. Results are compared versus non-MDCure'd baselines and shown in Table 4. We observe consistent, albeit modest, improvements to downstream MD performance as we scale

[7] We consider Jamba 1.5 Mini (54B) comparable with LLAMA3.1-70B-Instruct, as their parameter scales are within the same ballpark, falling within the same order of magnitude.

	HQA	WH	MXS	QC	SEAM	ZSS
Zero-Shot Prompting Evaluation						
LLAMA3.1-8B-Instruct	35.5	27.1	95.1	65.3	10.2	18.7
+MDCure	**41.0**	40.2	95.6	87.6	12.0	31.7
+MDCure+LongContext	38.9	**40.5**	**95.8**	**88.2**	**12.1**	**32.5**
5-Shot Prompting Evaluation						
LLAMA3.1-8B-Instruct	57.5	44.2	88.2	65.1	15.1	—
+MDCure	63.9	47.0	89.8	**73.8**	17.3	—
+MDCure +Few-Shot	**65.0**	**49.2**	**90.7**	**73.8**	**17.9**	—

Table 5: Efficacy of MDCure for few-shot and long-context MD instruction data generation.

up the instruction data size. Notably, use of 12K MDCure-filtered samples consistently surpasses performance with use of 72K *unfiltered* samples for both base models, consistent with other work finding use of only a few thousand high-quality samples is sufficient for alignment (Zhou et al., 2023). This evidences MDCureRM's ability to promote data efficiency and reduce costs involved in generating and using an effective MD IT dataset.

5.3 Generalization to Few-Shot & Extended-Context Training

We investigate the scalability of MDCure instructions to the few-shot prompting and extended-context settings. The few-shot setting is of interest as it provides an inexpensive alternative for adapting models to new tasks with limited compute (Dong et al., 2023). Likewise, real-world MD tasks often occur over "extreme" numbers of documents such as in extreme MDS (Lu et al., 2020). We extend our MDCure data to these settings by packing few-shot exemplars or distractor articles into instruction inputs up to 32K tokens in length.[8] Experiments utilize LLAMA3.1-8B-Instruct as the base model, with training setup details described in App. C.1. Results are displayed in Table 5, where we evaluate the few-shot setting via 5-shot prompting, omitting ZeroScrolls which targets 0-shot evaluation. We find that use of MDCure with extended contexts further improves performance for most benchmarks. Combined with prior results showing MDCure generalizes beyond training context length, this suggests MDCure is effective and applicable in both resource-rich and resource-constrained settings, enhancing its practicality.

Generator LLM	HQA	WH	MXS	QC	SEAM	ZSS	Avg
Qwen2-1.5B-Instruct							
GPT-3.5-Turbo	**37.0**	33.2	**94.4**	**83.3**	10.0	24.4	42.3
LLAMA3.1-70B	36.7	**33.9**	94.2	82.4	**10.3**	**24.5**	**42.8**
Qwen2-7B-Instruct							
GPT-3.5-Turbo	44.1	45.4	94.7	82.7	**8.9**	25.9	43.0
LLAMA3.1-70B	**46.9**	**45.6**	**94.9**	**82.7**	8.2	**28.7**	**44.0**
LLAMA3.1-8B-Instruct							
GPT-3.5-Turbo	**40.9**	**41.6**	95.8	87.6	11.9	30.7	50.3
LLAMA3.1-70B	40.0	37.5	**97.3**	**88.0**	**12.2**	**34.3**	**52.9**

Table 6: MDCure results with different generator LLMs.

5.4 Efficacy with Open-Source Generation

We demonstrate the distinct cost advantage of MDCure over existing synthetic data pipelines which rely on GPT-based generations and target only non-MD contexts by studying the impact of using an open-source generator model in place of GPT-3.5-Turbo. Following the same procedure as in previous settings, we alternately use LLAMA3.1-70B-Instruct generation with MDCureRM filtering to create an MDCure instruction dataset of size 12K examples, since §5.2 finds this quantity to be sufficient for strong improvements to MD task performance. Fine-tuning results on Qwen2-1.5B-Instruct, Qwen2-7B-Instruct, and LLAMA3.1-8B-Instruct are reported in Table 6. Across model sizes and families, use of open-source generations in fact leads to better performance versus use of GPT generations. These emphasize the flexibility of MDCure and validate the unique cost-saving nature of our synthetic data framework.

5.5 Integration with Policy Optimization

Observing MDCureRM's strength as a judge of MD instruction quality, we explore the value of using MDCureRM with reinforcement learning methods such as PPO to iteratively improve synthetic MD data generations. We utilize LLAMA3.1-8B-Instruct as the initial policy model[9] and use MDCureRM to provide reward signals whilst training with a PPO objective. Details of our PPO training setup can be found in App. C.3. As in §5.4, we use the final policy model to create a size-12K MDCure instruction dataset and report fine-tuning results on Qwen2-1.5B-Instruct, Qwen2-7B-Instruct, and LLAMA3.1-8B-Instruct in Table 7. Across benchmarks, use of generations from PPO-trained LLAMA3.1-8B-Instruct enables all three models to achieve close or superior performance versus use of

[8] Details of extended-context data construction is in App. B.11. Context length is selected based on GPU memory limits.

[9] We select this model as it is relatively small while remaining effective at instruction-following.

Generator LLM	HQA	WH	MXS	QC	SEAM	ZSS	Avg
Qwen2-1.5B-Instruct							
GPT-3.5-Turbo	37.0	33.2	**94.4**	**83.3**	10.0	24.4	42.3
Llama3.1-8B+PPO	**38.5**	**34.0**	94.3	81.9	**11.3**	**24.8**	**43.5**
Qwen2-7B-Instruct							
GPT-3.5-Turbo	44.1	45.4	94.7	82.7	8.9	25.9	43.0
Llama3.1-8B+PPO	**46.7**	**47.8**	**95.1**	**82.5**	**9.1**	**26.9**	**45.3**
LLAMA3.1-8B-Instruct							
GPT-3.5-Turbo	40.9	**41.6**	**95.8**	87.6	**11.9**	30.7	50.3
Llama3.1-8B+PPO	**41.5**	40.6	94.9	**93.5**	11.8	**32.1**	**53.1**

Table 7: MDCure performance when MDCureRM is used to train a custom MD instruction generator model.

GPT generations. Notably, MDCure enables models far smaller than SOTA LLMs to become effective MD instruction generators, further enhancing the cost-effectiveness of our approach.

5.6 Impact on Advanced Model Capabilities

We assess the impact of MDCure on general LLM capabilities including language understanding, advanced reasoning, mathematics, and coding by evaluating our models at the 7B and 8B parameter scales on GSM8K (Cobbe et al., 2021b), MBPP (Austin et al., 2021), and MMLU (Hendrycks et al., 2021a,b) via the OLMES framework (Gu et al., 2024). Results are shown in Table 8. We observe that MDCure'd models achieve comparable average performance across the three benchmarks versus non-MDCure'd base models, demonstrating that MDCure data enhances MD modeling abilities of LLMs while preserving other advanced capabilities as well.

6 Conclusion

In this work, we presented MDCure, a framework for scalably and inexpensively generating high-quality MD post-training data to improve any base LLM's ability to leverage long-range dependencies for MD tasks. MDCure addresses the challenge of data scarcity in MD task domains without depending on proprietary LLMs and is robust across model families and sizes. We further introduced MDCureRM, a novel RM-as-a-Judge LLM to filter MD instruction data more cheaply and effectively than GPT-3.5-Turbo and enable moderately-sized LLMs to excel over SOTA proprietary counterparts as generators of high-quality MD instruction data. Our extensive experiments on various LLMs in many task and content domains show MDCure adds value beyond typical IT and substantially boosts MD performance while overcoming limitations of pre-training-based approaches. Future work could explore extending MDCure to generate instructions over additional topic domains.

Model	GSM8K	MBPP	MMLU	Avg
Qwen2-7B-Instruct	0.781	0.634	**0.706**	0.707
+MDCure	**0.801**	**0.668**	0.694	**0.721**
LLAMA3.1-8B-Instruct	0.800	0.669	**0.717**	0.729
+MDCure	**0.806**	**0.720**	0.713	**0.746**

Table 8: General performance of MDCure'd models.

7 Limitations

As abstractive tasks are the focus of many downstream MD applications, MDCure focuses on instruction generation for abstractive settings; future work can consider incorporation of extractive instruction templates to expand MDCure's utility. Additionally, the design and application of our approach is limited to texts in English; future work can also consider exploring the utility of MDCure for non-English tasks. As observed with existing state-of-the-art generative language models, post-trained models may still suffer from factuality errors during generation. Although the focus of this work is not on factuality, we adopt measures to improve factual consistency by eliminating non-factual or unfaithful instruction generations in the Filtering stage; faithfulness aspects of our pipeline could be the subject of other future investigation. Lastly, potential improvements may arise from generating instructions and answers separately as opposed to together (Zhao et al., 2024a), from incorporation of short-context data during instruction-tuning (Bai et al., 2024; Gao et al., 2024b), and with inclusion of document sets in other text genres. Such design choices are left as possibilities for future research given the cost-effectiveness, efficacy, and domain-general nature of our approach.

8 Ethics Statement

As with any use of LLMs, there is a need to ensure the safety of our post-trained models' responses through red-teaming or other safety measures. Use of our factuality criterion in assessing instruction data reduces the chances for MDCure to lead to unfactual examples due to hallucination during instruction generation. However, it is generally difficult to guarantee factual consistency during generative language modeling without further fine-tuning. Beyond this, we anticipate our work will improve

LLMs' efficacy in processing long documents and large document collections, which can accelerate value generation in domains such as business analytics and scientific literature mining.

Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-2139841. We are grateful for the compute support provided by Google. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or Google.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Devanshu Agrawal, Shang Gao, and Martin Gajek. 2024. Can't remember details in long documents? you need some r&r. *Preprint*, arXiv:2403.05004.

Joshua Ainslie, Tao Lei, Michiel de Jong, Santiago Ontanon, Siddhartha Brahma, Yury Zemlyanskiy, David Uthus, Mandy Guo, James Lee-Thorp, Yi Tay, Yun-Hsuan Sung, and Sumit Sanghai. 2023. CoLT5: Faster long-range transformers with conditional computation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5100, Singapore. Association for Computational Linguistics.

Chenxin An, Fei Huang, Jun Zhang, Shansan Gong, Xipeng Qiu, Chang Zhou, and Lingpeng Kong. 2024a. Training-free long-context scaling of large language models. *Preprint*, arXiv:2402.17463.

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, and Jian-Guang Lou. 2024b. Make your llm fully utilize the context. *Preprint*, arXiv:2404.16811.

Stefanos Angelidis, Reinald Kim Amplayo, Yoshihiko Suhara, Xiaolan Wang, and Mirella Lapata. 2021. Extractive opinion summarization in quantized transformer spaces. *Transactions of the Association for Computational Linguistics*, 9:277–293.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large language models. *Preprint*, arXiv:2108.07732.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional ai: Harmlessness from ai feedback. *Preprint*, arXiv:2212.08073.

Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. 2024. Longalign: A recipe for long context alignment of large language models. *ArXiv*, abs/2401.18058.

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

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324.

Avi Caciularu, Arman Cohan, Iz Beltagy, Matthew Peters, Arie Cattan, and Ido Dagan. 2021. CDLM: Cross-document language modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2648–2662, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Avi Caciularu, Ido Dagan, Jacob Goldberger, and Arman Cohan. 2022. Long context question answering via supervised contrastive learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2872–2879, Seattle, United States. Association for Computational Linguistics.

Avi Caciularu, Matthew Peters, Jacob Goldberger, Ido Dagan, and Arman Cohan. 2023. Peek across: Improving multi-document modeling via cross-document question-answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1970–1989, Toronto, Canada. Association for Computational Linguistics.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. 2021a. Realistic evaluation principles for cross-document coreference resolution. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 143–151, Online. Association for Computational Linguistics.

Arie Cattan, Sophie Johnson, Daniel S Weld, Ido Dagan, Iz Beltagy, Doug Downey, and Tom Hope. 2021b. Scico: Hierarchical cross-document coreference for scientific concepts. In *3rd Conference on Automated Knowledge Base Construction*.

Yung-Chieh Chan, George Pu, Apaar Shanker, Parth Suresh, Penn Jenks, John Heyer, and Sam Denton. 2024. Balancing cost and effectiveness of synthetic data generation strategies for llms. *Preprint*, arXiv:2409.19759.

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. 2023. Walking down the memory maze: Beyond context limit through interactive reading. *Preprint*, arXiv:2310.05029.

Jiuhai Chen, Rifaa Qadri, Yuxin Wen, Neel Jain, John Kirchenbauer, Tianyi Zhou, and Tom Goldstein. 2024a. Genqa: Generating millions of instructions from a handful of prompts. *Preprint*, arXiv:2406.10323.

Longze Chen, Ziqiang Liu, Wanwei He, Yinhe Zheng, Hao Sun, Yunshui Li, Run Luo, and Min Yang. 2024b. Long context is not long at all: A prospector of long-dependency data for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8222–8234, Bangkok, Thailand. Association for Computational Linguistics.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2022. SummScreen: A dataset for abstractive screenplay summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8602–8615, Dublin, Ireland. Association for Computational Linguistics.

Zhi Chen, Qiguang Chen, Libo Qin, Qipeng Guo, Haijun Lv, Yicheng Zou, Wanxiang Che, Hang Yan, Kai Chen, and Dahua Lin. 2024c. What are the essential factors in crafting effective long context multi-hop instruction datasets? insights and best practices. *Preprint*, arXiv:2409.01893.

Eric Chu and Peter Liu. 2019. MeanSum: A neural model for unsupervised multi-document abstractive summarization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1223–1232. PMLR.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021a. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021b. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey for in-context learning. *ArXiv*, abs/2301.00234.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Alon Eirew, Arie Cattan, and Ido Dagan. 2021. WEC: Deriving a large-scale cross-document event coreference dataset from Wikipedia. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2498–2510, Online. Association for Computational Linguistics.

Ori Ernst, Avi Caciularu, Ori Shapira, Ramakanth Pasunuru, Mohit Bansal, Jacob Goldberger, and Ido Dagan. 2022. Proposition-level clustering for multi-document summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1765–1779, Seattle, United States. Association for Computational Linguistics.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Preprint*, arXiv:2007.12626.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. 2024a. Enabling large language models to generate text with citations.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. 2024b. How to train long-context language models (effectively). *Preprint*, arXiv:2410.02660.

Suyu Ge, Xihui Lin, Yunan Zhang, Jiawei Han, and Hao Peng. 2024. A little goes a long way: Efficient long context training and inference with partial contexts. *Preprint*, arXiv:2410.01485.

Alireza Ghadimi and Hamid Beigy. 2023. Sgcsumm: An extractive multi-document summarization method based on pre-trained language model, submodularity, and graph convolutional neural networks. *Expert Systems with Applications*, 215:119308.

John Giorgi, Luca Soldaini, Bo Wang, Gary Bader, Kyle Lo, Lucy Wang, and Arman Cohan. 2023. Open domain multi-document summarization: A comprehensive study of model brittleness under retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8177–8199, Singapore. Association for Computational Linguistics.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Xiaotao Gu, Yuning Mao, Jiawei Han, Jialu Liu, Hongkun Yu, You Wu, Cong Yu, Daniel Finnie, Jiaqi Zhai, and Nicholas Zukoski. 2020. Generating Representative Headlines for News Stories. In *Proc. of the the Web Conf. 2020*.

Yuling Gu, Oyvind Tafjord, Bailey Kuehl, Dany Haddad, Jesse Dodge, and Hannaneh Hajishirzi. 2024. Olmes: A standard for language model evaluations. *Preprint*, arXiv:2406.08446.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021a. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021b. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.

Peyman Hosseini, Ignacio Castro, Iacopo Ghinassi, and Matthew Purver. 2024. Efficient solutions for an intriguing failure of llms: Long context window does not mean llms can analyze long sequences flawlessly. *Preprint*, arXiv:2408.01866.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A Smith, Iz Beltagy, et al. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *arXiv preprint arXiv:2311.10702*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Wojciech Kryscinski, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2022. BOOKSUM: A collection of datasets for long-form narrative summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6536–6558, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Litton J Kurisinkel and Nancy F chen. 2023. Controllable multi-document summarization: Coverage & coherence intuitive policy with large language model based rewards. *Preprint*, arXiv:2310.03473.

Litton J Kurisinkel and Nancy F. Chen. 2023. Llm based multi-document summarization exploiting main-event biased monotone submodular content extraction. *Preprint*, arXiv:2310.03414.

Wai-Chung Kwan, Xingshan Zeng, Yufei Wang, Yusen Sun, Liangyou Li, Yuxin Jiang, Lifeng Shang, Qun Liu, and Kam-Fai Wong. 2024. M4LE: A multi-ability multi-range multi-task multi-domain long-context evaluation benchmark for large language models. In *Proceedings of the 62nd Annual Meeting* *of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15568–15592, Bangkok, Thailand. Association for Computational Linguistics.

Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. 2023. Longform: Effective instruction tuning with reverse instructions. *Preprint*, arXiv:2304.08460.

AI21 Labs. 2024. [link].

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. 2024. Tulu 3: Pushing frontiers in open language model post-training. *Preprint*, arXiv:2411.15124.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2024. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. *Preprint*, arXiv:2309.00267.

Bing Li, Peng Yang, Zhongjian Hu, Yuankang Sun, and Meng Yi. 2023a. Graph-enhanced multi-answer summarization under question-driven guidance. *J. Supercomput.*, 79(18):20417–20444.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2023b. Loogle: Can long-context language models understand long contexts? *arXiv preprint arXiv:2311.04939*.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason E Weston, and Mike Lewis. 2024. Self-alignment with instruction backtranslation. In *The Twelfth International Conference on Learning Representations*.

Gili Lior, Avi Caciularu, Arie Cattan, Shahar Levy, Ori Shapira, and Gabriel Stanovsky. 2024. Seam: A stochastic benchmark for multi-document tasks. *Preprint*, arXiv:2406.16086.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2023a. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *Preprint*, arXiv:2312.15685.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Yixin Liu, Alexander Fabbri, Jiawen Chen, Yilun Zhao, Simeng Han, Shafiq Joty, Pengfei Liu, Dragomir Radev, Chien-Sheng Wu, and Arman Cohan. 2024a. Benchmarking generation and evaluation capabilities of large language models for instruction controllable summarization. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4481–4501, Mexico City, Mexico. Association for Computational Linguistics.

Yixin Liu, Kejian Shi, Alexander R. Fabbri, Yilun Zhao, Peifeng Wang, Chien-Sheng Wu, Shafiq Joty, and Arman Cohan. 2024b. Reife: Re-evaluating instruction-following evaluation. *Preprint*, arXiv:2410.07069.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

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.

Alisia Lupidi, Carlos Gemmell, Nicola Cancedda, Jane Dwivedi-Yu, Jason Weston, Jakob Foerster, Roberta Raileanu, and Maria Lomeli. 2024. Source2synth: Synthetic data generation and curation grounded in real data sources. *Preprint*, arXiv:2409.08239.

Yuning Mao, Yanru Qu, Yiqing Xie, Xiang Ren, and Jiawei Han. 2020. Multi-document summarization with maximal marginal relevance-guided reinforcement learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1737–1751, Online. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. 2022. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358, Seattle, United States. Association for Computational Linguistics.

Ramakanth Pasunuru, Asli Celikyilmaz, Michel Galley, Chenyan Xiong, Yizhe Zhang, Mohit Bansal, and Jianfeng Gao. 2021. Data augmentation for abstractive query-focused multi-document summarization. In *AAAI Conference on Artificial Intelligence*.

Joseph Peper, Wenzhao Qiu, and Lu Wang. 2024. PELMS: Pre-training for effective low-shot multi-document summarization. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7652–7674, Mexico City, Mexico. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Noveen Sachdeva, Benjamin Coleman, Wang-Cheng Kang, Jianmo Ni, Lichan Hong, Ed H. Chi, James Caverlee, Julian McAuley, and Derek Zhiyuan Cheng. 2024. How to train data-efficient llms. *Preprint*, arXiv:2402.09668.

Ananya B. Sai, Mithun Das Gupta, Mitesh M. Khapra, and Mukundhan Srinivasan. 2019. Re-evaluating adem: a deeper look at scoring dialogue responses. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. 2023. ZeroSCROLLS: A zero-shot benchmark for long text understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7977–7989, Singapore. Association for Computational Linguistics.

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. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12007–12021, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Victoria Lin, Noah A Smith, Luke Zettlemoyer, Scott Yih, and Mike Lewis. 2023. In-context pretraining: Language modeling beyond document boundaries. *arXiv preprint arXiv:2310.10638*.

Zhengyan Shi, Adam Yang, Bin Wu, Laurence Aitchison, Emine Yilmaz, and Aldo Lipani. 2024. Instruction tuning with loss over instructions.

Chih-Wei Song, Yu-Kai Lee, and Yin-Te Tsai. 2024. A new pipeline for generating instruction dataset via rag and self fine-tuning. *Preprint*, arXiv:2408.05911.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Preprint*, arXiv:2206.04615.

Zecheng Tang, Keyan Zhou, Juntao Li, Baibei Ji, Jianye Hou, and Min Zhang. 2024. L-citeeval: Do long-context models truly leverage context for responding? *Preprint*, arXiv:2410.02115.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S. Morcos. 2024. D4: improving llm pretraining via document de-duplication and diversification. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Alex Wang, Richard Yuanzhe Pang, Angelica Chen, Jason Phang, and Samuel R. Bowman. 2022. SQuALITY: Building a long-document summarization dataset the hard way. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1139–1156, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chonghua Wang, Haodong Duan, Songyang Zhang, Dahua Lin, and Kai Chen. 2024a. Ada-leval: Evaluating long-context llms with length-adaptable benchmarks. *Preprint*, arXiv:2404.06480.

Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. 2024b. Arithmetic control of llms for diverse user preferences: Directional preference alignment with multi-objective rewards. In *ACL*.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024c. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *EMNLP*.

Minzheng Wang, Longze Chen, Cheng Fu, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, et al. 2024d. Leave no document behind: Benchmarking long-context llms with extended multi-doc qa. *arXiv preprint arXiv:2406.17419*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Ruben Wolhandler, Arie Cattan, Ori Ernst, and Ido Dagan. 2022. How “multi” is multi-document summarization? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5761–5769, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wenhao Wu, Yizhong Wang, Yao Fu, Xiang Yue, Dawei Zhu, and Sujian Li. 2024. Long context alignment with short instructions and synthesized positions. *Preprint*, arXiv:2405.03939.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-grained human feedback gives better rewards for language model training. *arXiv preprint arXiv:2306.01693*.

Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2022. PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5245–5263, Dublin, Ireland. Association for Computational Linguistics.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. 2024. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. *Preprint*, arXiv:2312.11456.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Retrieval meets long context large language models. In *The Twelfth International Conference on Learning Representations*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland. Association for Computational Linguistics.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. 2024. Flask: Fine-grained language model evaluation based on alignment skill sets. *Preprint*, arXiv:2307.10928.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izasak, Moshe Wasserblat, and Danqi Chen. 2024. Helmet: How to evaluate long-context language models effectively and thoroughly. *Preprint*, arXiv:2410.02694.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.

Xiang Yue, Tuney Zheng, Ge Zhang, and Wenhu Chen. 2024. Mammoth2: Scaling instructions from the web. *Preprint*, arXiv:2405.03548.

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.

Chenyang Zhao, Xueying Jia, Vijay Viswanathan, Graham Neubig, and Tongshuang Wu. 2024a. Self-guide: Better task-specific instruction following via self-synthetic finetuning. In *The Conference on Language Modeling (COLM)*.

Jun Zhao, Can Zu, Hao Xu, Yi Lu, Wei He, Yiwen Ding, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024b. Longagent: Scaling language models to 128k context through multi-agent collaboration. *Preprint*, arXiv:2402.11550.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. Association for Computational Linguistics.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*.

He Zhu, Junyou Su, Tianle Lun, Yicheng Tao, Wenjia Zhang, Zipei Fan, and Guanhua Chen. 2024. Fanno: Augmenting high-quality instruction data with open-sourced llms only. *Preprint*, arXiv:2408.01323.

Ingo Ziegler, Abdullatif Köksal, Desmond Elliott, and Hinrich Schütze. 2024. Craft your dataset: Task-specific synthetic dataset generation through corpus retrieval and augmentation. *Preprint*, arXiv:2409.02098.

A Additional Ablation Results

A.1 Effect of Generation Approach

The efficacy of MDCure is dependent on the quality of initial candidate instruction generations. To this end, we conduct an ablation study to investigate the impact of several different prompting approaches on downstream instruction-tuned model performance. We consider only the zero-shot prompting setting given the high cost of prompting GPT models with few-shot examples that each contain multiple long documents.

In our preliminary investigation, we find that prompt templates emphasizing the MD nature of generated instructions tend to yield higher-scoring results (e.g., suggesting penalties if generated instructions concern only a single document). Thereafter, we explore the utility of providing general vs. highly detailed specifications for the style of the generated instruction-answer pairs. We ultimately converge on two classes of prompt templates which we refer to as either *General* or *Style-Specific*. Details are given in in App. B.2.

	HQA	WH	MXS	QC	SEAM	ZSS
	FlanT5-Base					
Baseline (No IT)	4.4	45.1	38.7	48.0	1.7	13.1
General	42.3	40.5	90.9	52.0	1.1	20.6
Style-Specific	37.9	48.1	91.8	48.9	1.3	20.0
Both	**43.8**	**48.4**	**92.4**	**54.0**	**1.9**	**22.0**
	Qwen2-7B-Instruct					
Baseline (No IT)	30.5	39.6	95.6	79.3	7.4	23.9
General	40.8	44.9	93.8	81.3	7.8	24.2
Style-Specific	40.2	**45.8**	94.2	80.7	7.8	23.7
Both	**44.1**	45.4	**94.7**	**82.7**	**8.9**	**25.9**

Table 9: Efficacy of different prompt approaches.

We examine the impact of General vs. Style-Specific prompt templates by fine-tuning FlanT5-Base and Qwen2-7B-Instruct using MDCureRM-filtered datasets created using templates from one or both categories. As in §5.4, we limit the number of samples to 12K given the findings in §5.2. When both templates are used, we employ a 1:3 ratio of General:Style-Specific instructions, found to be most effective in initial experiments.

Results are shown in Table 9. We observe that use of General or Style-Specific templates alone to produce instructions does not always lead to downstream MD performance improvements (e.g., FlanT5-Base IT'ed with General-based instructions does not surpass base model performance on WikiHop). In comparison, combining the two gives the best performance gains in a consistent fashion across benchmarks.

A.2 Ablations on Scoring Criteria

To justify our choice of scoring criteria in MDCureRM, we conduct an ablation study to show the advantage of combining all six criteria, rather than using a subset of them. We also evidence the benefit of applying greater weight to MD-specific criteria when computing the overall score for each candidate instruction, as opposed to computing an evenly weighted sum (details in App. B.9). Ablation experiments were run using FlanT5-Base and Qwen2-7B-Instruct. Table 10 shows the results. Across all benchmarks, filtering instruction data using a weighted average of six scoring criteria with emphasis on multi-document factors yields the best results. Further, removing any one criterion or utilizing even weighting across criteria leads to worsened IT'ed performance.

	HQA	WH	MXS	QC	SEAM	ZSS
			FlanT5-Base			
all criteria, MD emphasis	**43.8**	**48.4**	**92.4**	**54.0**	**1.9**	**22.0**
all criteria, evenly weighted	43.4	47.1	92.4	52.2	1.8	21.2
without Relevance	37.6	46.1	91.6	50.2	1.6	20.4
without Coh. & Fac.	40.1	46.0	92.0	52.2	1.5	20.4
without Creativity	38.4	44.2	92.2	51.8	1.3	20.8
without Context Int.	14.1	45.4	85.3	45.8	1.5	17.0
without Inter-Doc. Rel.	40.9	45.8	91.1	53.6	1.5	21.5
without Complexity	40.2	45.5	91.3	54.2	1.4	20.0
			Qwen2-7B-Instruct			
all criteria, MD emphasis	**44.1**	**45.4**	94.7	**82.7**	**8.9**	**25.9**
all criteria, evenly weighted	41.3	43.1	88.2	68.9	8.5	25.4
without Relevance	37.5	40.7	94.4	82.0	8.1	25.0
without Coh. & Fac.	41.9	43.8	**94.9**	80.7	8.4	23.2
without Creativity	43.6	44.3	94.0	**82.7**	**8.9**	25.4
without Context Int.	37.1	42.4	94.2	80.4	8.5	25.0
without Inter-Doc. Rel.	39.0	42.4	94.2	81.1	8.5	25.7
without Complexity	37.1	41.8	94.4	81.1	8.5	25.7

Table 10: Scoring criteria ablation study results.

B Dataset Construction

B.1 Source Texts

As noted in §3.1, our experiments base data generation on the NewSHead corpus (Gu et al., 2020). NewSHead consists of 369940 clusters of 3-5 related English news articles curated for the task of news story headline generation. We utilize the already-preprocessed version of the corpus supplied by Caciularu et al. (2023) at `https://storage.googleapis.com/primer_summ/processed_pre-training_data.tar.gz`. This follows the pre-processing procedure described in Xiao et al. (2022).

B.2 Data Generation Prompts

During the Generation phase of MDCure, we employ two types of prompt templates to encourage the generator model to produce a diverse set of instruction data. We distinguish these templates as either General or Style-Specific.

General templates are inspired by Köksal et al. (2023) and broadly ask the generator model to produce an instruction-answer pair adhering to a certain answer length (e.g., 1-2 words, 5 or more sentences). Emphasis is placed on creation of instructions that cannot be answered if any one context document is removed, and we employ various phrasings of this specification. We utilize a total of 14 general prompts. General templates A-D (Figure 4) are inspired by recent work that demonstrates the efficacy of grounding synthetic generations in pre-selected answer content from a given source. Modeling after (Köksal et al., 2023), for these 4 prompt templates, we consider all possible combinations of two related documents in NewSHead and split the contents of every possible document pair into texts 1-3 sentences in length. Each segment is embedded via HuggingFace Sentence Transformers (Reimers and Gurevych, 2019) and aligned with the others via cosine similarity. Distinct segments are then paired according to highest alignment for according use in the prompt templates. To ensure segments are not too similar, we set a maximum similarity threshold of 0.70 per pair. General templates E-N (Figures 5 and 6) consider 2 or more context documents per generation.

On the other hand, Style-Specific templates (Figure 7) are adapted from Zhu et al. (2024) and employ a single phrasing while incorporating varying combinations of constraints regarding instruction complexity (e.g., analysis, recall), type (e.g., paraphrasing, inference), style (e.g., imperative, interrogative), and output length (e.g., 3-4 words, 3 sentences), with pre-defined options for each specification category. The aim is to enforce constraints on diversity of the generated candidate data. As defined below, we consider 4 characteristics, 4 types, and 3 styles of instruction, in combination with 8 possible answer lengths (Figure 8). Each possible resulting prompt template considers 2 or more context documents per generation.

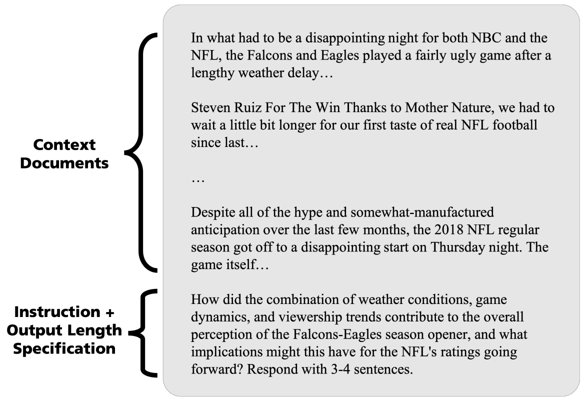
To ensure a balanced composition for the pool of candidate instructions, we sample uniformly from all possible prompt templates during generation for each cluster of related documents.

B.3 Data Generation Hyperparameters

To elicit high-quality candidate instructions in a cost-effective manner for our experiments, we utilize GPT-3.5-Turbo as the generator model in MDCure with all inference hyperparameters set to their default values. The cost is approximately $25.00 per 35,000 candidate instruction generations.

B.4 Instruction Format

We format the MD instruction inputs to place the generated instruction after the source documents, but alternate formats may be equally effective. Once all instruction-answer pairs are generated, since the expected output length of downstream MD tasks has a high variance (e.g., keyword based QA vs. long-form summarization), we append a brief direction indicating the expected length of the answer (i.e., expected number of words or sentences).[10] Options for this direction differ depending on whether an instruction was created using a General or Style-Specific template (Figure 9). As shown in Fig. 3, the direction is appended to the end of the finalized instruction input, separated by a space.



In what had to be a disappointing night for both NBC and the NFL, the Falcons and Eagles played a fairly ugly game after a lengthy weather delay…

Steven Ruiz For The Win Thanks to Mother Nature, we had to wait a little bit longer for our first taste of real NFL football since last…

…

Despite all of the hype and somewhat-manufactured anticipation over the last few months, the 2018 NFL regular season got off to a disappointing start on Thursday night. The game itself…

How did the combination of weather conditions, game dynamics, and viewership trends contribute to the overall perception of the Falcons-Eagles season opener, and what implications might this have for the NFL's ratings going forward? Respond with 3-4 sentences.

Figure 3: Illustrative example of input format.

B.5 Sample Instruction Data

We provide several qualitative examples demonstrating the diverse, complex, and open-ended nature of our MDCure data, formatted as prompt-response pairs (Table 11). Note that while each instruction input is formatted as discussed in App. B.4, in Table 11 we present only the generated prompt for each input and omit the context documents for easier visibility.

B.6 Diversity Analysis

To provide further insight regarding the syntactical and semantic diversity of our MDCure data, we follow the methodology used in Wang et al. (2023); Köksal et al. (2023) to analyze our size-12K MDCure dataset versus 12K samples from the training sets of HotpotQA-distractor and WikiHop, which are human-annotated MD benchmark datasets curated specifically for fine-tuning evaluation, along with 12K samples from the Tulu V2 SFT mixture (Rafailov et al., 2024), which is a mixture of human-annotated and synthetic data curated for general instruction tuning.

As in Wang et al. (2023), for each dataset, we parse the instruction in each training sample using Berkeley Neural Parser (Kitaev and Klein, 2018; Kitaev et al., 2019) and extract either the contained auxiliary and its dependent verb (e.g., “do hear”) or the contained verb and its direct object (e.g., “summarize events”). Additionally, as in Zhu et al. (2024), we analyze the relative complexity of instruction-response pairs using the Deita-complexity-scorer model (Liu et al., 2023a). We report statistics regarding unique verb-noun combinations and average complexity of each dataset in Table 12.

Instruction Prompt	Answer
What is the standout feature of Samsung’s Galaxy S10 fingerprint sensor identified by the majority of the articles? Answer with a single word or brief phrase.	Magical unlocking transition.
Describe the mechanism by which synthetic sandalwood stimulates hair growth. Give a response that is 3-4 words.	Activates receptor OR2AT4.
Describe the implications of NASCAR’s new reduced horsepower package being used at the Auto Club 400 while considering the historical context of the race distance changes and past driver performances. Respond with 1-2 sentences.	The implementation of the reduced horsepower package at the Auto Club 400 marks a significant shift in NASCAR’s strategy for intermediate tracks, aiming to enhance competitive balance; historically, the race has been shortened from 500 miles to 400 miles since 2011, impacting driver strategies and outcomes, as seen in past performances of drivers like Kevin Harvick and Kyle Busch.
Elaborate on the emotional and artistic significance of Adam Lambert's song "Feel Something" as it reflects his personal journey and professional struggles, highlighting how this piece serves as a precursor to his upcoming album. Craft your response in 6 sentences.	Adam Lambert’s song "Feel Something" signifies a pivotal moment in his life, as it encapsulates his journey of overcoming personal struggles and creative disillusionment. Written during a phase of introspection, the track addresses Lambert’s emotional experiences of feeling detached amidst the chaos of his career, illustrating his realization that the pursuit of love and connection could not be separated from the quest for personal happiness. The song represents both a cathartic release and an assertive step towards reclaiming his artistic vision, allowing Lambert to finally embrace the importance of self-care and vulnerability. This piece serves as the emotional cornerstone of his upcoming album, which aims to chronicle his commitment to taking ownership of his happiness and strength through music. The collaborative efforts with writers Benedict Cork and Josh Cumbee during its creation reflect a renewed artistic authenticity that Lambert seeks to reclaim. Thus, "Feel Something" not only marks a return to his musical roots but also signals a larger narrative of resilience, artistry, and self-discovery that will unfold in his forthcoming work.

Table 11: Representative instances of high-quality instruction-answer pairs, encompassing various MD task formulations and answer lengths, produced using our MDCure framework.

Versus human-annotated SFT and MD datasets, MDCure data covers a comparably wide range of tasks encompassing various instruction types, as indicated by its greater spread of unique verb+noun pairs, diverse and complex top 5 verbs, and highest-rated average complexity. These comparisons demonstrate the ability for MDCure prompts to

[10] Our preliminary experiments demonstrated non-negligible performance gains on downstream MD tasks with the attachment of such length direction.

General Template A

Snippets: ‘{segment1}’, ‘{segment2}’
Context Paragraphs: ‘{doc1}’, ‘{doc2}’
Based on the given snippets and context paragraphs, construct an instruction-answer pair such that (1) the answer is based on the two snippets and (2) the instruction is a plausible prompt or question to which the answer would be the expected response. Make sure both snippets are required to answer the instruction. You will be penalized if the instruction concerns only one snippet. Format your response as:
Instruction: <prompt or question>
Answer: <answer>

General Template B

Snippets: ‘{segment1}’, ‘{segment2}’
Based on the given snippets, construct an instruction-answer pair such that (1) the answer is yes and (2) the instruction is a plausible prompt or question to which yes would be the expected response. Make sure the answer does not conflict with the information in the snippets. You will be penalized if the instruction-answer pair is unfactual. Do NOT mention the snippets in the instruction. Format your response as:
Instruction: <prompt or question>
Answer: <yes>

General Template C

Snippets: ‘{segment1}’, ‘{segment2}’
Based on the given snippets, construct an instruction-answer pair such that (1) the answer is no and (2) the instruction is a plausible prompt or question to which no would be the expected response. Make sure the answer does not conflict with the information in the snippets. You will be penalized if the instruction-answer pair is unfactual. Do NOT mention the snippets in the instruction. Format your response as:
Instruction: <prompt or question>
Answer: <no>

General Template D

Snippets: ‘{segment1}’, ‘{segment2}’
Context Paragraphs: ‘{doc1}’, ‘{doc2}’
Based on the given snippets and context paragraphs, construct an instruction-answer pair such that (1) the answer is a brief phrase and NOT a sentence and (2) the instruction is a plausible prompt or question to which the answer is the expected response. Make sure both snippets are required to answer the instruction. You will be penalized if the instruction concerns only one snippet. Make sure the answer is a brief phrase less than 7 words in length, with NO periods. You will be penalized if the answer is longer than 7 words or if the answer is a sentence. Format your response as:
Instruction: <prompt or question>
Answer: <answer>

Figure 4: General Prompt Templates A-D

General Template E

Context Paragraphs: ‘{doc1}’, ‘{doc2}’
Based on the two given context paragraphs, construct an instruction-answer pair such that (1) the answer is summary of the two paragraphs and (2) the instruction is a plausible prompt or question to which the answer is the expected response. Make sure both paragraphs are required to answer the instruction. You will be penalized if the instruction concerns only one paragraph. Make sure the answer does not conflict with the information in the paragraphs. You will be penalized if the instruction-answer pair is unfactual. Make sure the answer is at least 5 sentences in length. Do not mention the context paragraphs in the instruction. Format your response as:
Instruction: <prompt or question>
Answer: <answer>

General Template F

Context Paragraphs: ‘{doc1}’, ‘{doc2}’
Based on the two given context paragraphs, construct an instruction-answer pair such that (1) the answer is summary of the two paragraphs and (2) the instruction is a plausible prompt or question to which the answer is the expected response. Make sure both paragraphs are required to answer the instruction. You will be penalized if the instruction concerns only one paragraph. Make sure the answer does not conflict with the information in the paragraphs. You will be penalized if the instruction-answer pair is unfactual. Make sure the answer is less than 5 sentences in length. Do not mention the context paragraphs in the instruction. Format your response as:
Instruction: <prompt or question>
Answer: <answer>

General Template G

{context_docs}
A question or command that can ONLY be answered by utilizing ALL of the above documents and that CANNOT be answered if any one document is removed is:
Question: <respond here>
Answer: <respond here briefly>

General Template H

{context_docs}
What is a question or command that can ONLY be answered by utilizing ALL of the above documents and that CANNOT be answered if any one document is removed?
Question: <respond here>
Answer: <respond here briefly>

General Template I

Articles:
{context_docs}
What is an exam question that can ONLY be answered by utilizing ALL of the above documents and that CANNOT be answered if any one document is removed?
Exam Question: <respond here>
Answer: <respond here briefly>

Figure 5: General Prompt Templates E-I

General Template J

{context_docs}
What is a question or command that can ONLY be answered by utilizing ALL of the above documents and that CANNOT be answered if any one document is removed?
Question: <respond here>
Answer: <respond here, feel free to use a single word or phrase>

General Template K

{context_docs}
A question or command that can ONLY be answered by utilizing ALL of the above documents and that CANNOT be answered if any one document is removed is:
Question: <respond here>
Answer: <respond here>

General Template L

{context_docs}
What is a question or command that can ONLY be answered by utilizing ALL of the above documents and that CANNOT be answered if any one document is removed?
Question: <respond here>
Answer: <respond here, using ONLY a single word or phrase>

General Template M

Articles:
{context_docs}
Contrasting Question: <respond here>
Answer: <respond here briefly>

General Template N

Articles:
{context_docs}
What is an exam question that can ONLY be answered by utilizing ALL of the above documents and that CANNOT be answered if any one document is removed?
Exam Question: <respond here>
Answer Choices: <respond here>
Answer: <answer letter only>

Figure 6: General Prompt Templates J-N

Style-Specific Template

You're proficient in crafting complex questions. Generate only one question and one answer that adheres to the provided #Articles#. Make sure the question and answer are factually consistent with the #Articles#. The question should meet the following criteria:
0. The person answering the question cannot see the #Articles#, so the question must not contain phrases like 'Given the information provided', 'Based on the provided information', or similar expressions that imply direct citations or references from #Articles#.
1. The question must REQUIRE synthesis of information in at least 2 of the provided documents in order to answer correctly. The more documents are involved the better. Ideally all documents are required to answer the question, such that losing any one of them will lead person answering the question to provide an incorrect response. You will lose your job if this criterion is not satisfied.
2. {complexity}
3. {type}
4. {style}.
5. It requires {answer_length} to answer correctly.

The answer must be {answer_length} in length.

Articles:
{context_docs}

Question: <respond here>
Answer: <respond here>

Figure 7: Style-Specific Prompt Template.

Options for {complexity} Specification

- "It should be complex and require multiple-step reasoning across the documents to solve."
- "It demands critical thinking skills to analyze, evaluate, and synthesize multiple pieces of information from the different documents."
- "It demands integrating knowledge from multiple documents to address its multifaceted nature."
- "It should be simple and require only a few words to answer, yet utilize supporting evidence from at least 2 documents."

Options for {type} Specification

- "It is a Natural language inference question: Assessing if evidence supports a conclusion."
- "It is a Paraphrasing question: Rewording a statement while retaining its meaning."
- "It is a Summarization question: Condensing key information from a larger text."
- "It is an Informational question: Locating a specific piece of information in the given evidence."

Options for {style} Specification

- "It should be in the style of a command or imperative. For example, ‘Write a paragraph about...’ or ‘Describe the...’"
- "It should be in the style of a question or interrogative. For example, ‘What is the..?’ or ‘How do you...?’"
- "It should be in the style of a short phrase that serves as a query. For example, ‘today’s forecast.’ or ‘Donna’s car accident.’"

Options for {answer_length} Specification

- "1-2 words"
- "3-4 words"
- "a phrase of at least 5-6 words"
- "1-2 sentences"
- "3-4 sentences"
- "6 sentences"
- "8 sentences"
- "10 sentences"

Figure 8: Detailed options for Style-Specific prompt template.

Length Direction Options (General)

- "Answer briefly in 1-2 sentences."
- "Answer ‘yes’ or ‘no’"
- "Answer ‘yes’ or ‘no’"
- "Answer with a single word or brief phrase."
- "Answer with at least 5 sentences."
- "Answer with at most 5 sentences."

Length Direction Options (Style-Specific)

- "Answer with {answer_length}."
- "Answer using {answer_length}."
- "Respond with {answer_length}."
- "Respond using {answer_length}."
- "Formulate your answer in {answer_length}."
- "Reply with a {answer_length} answer."
- "Craft your response in {answer_length}."
- "Give a response that is {answer_length}."
- "Answer in around {answer_length}."

Figure 9: Options for direction regarding expected output length for instructions generated via General and Style-Specific templates.

Dataset	MDCure	HotpotQA	WikiHop	Tulu V2
Unique Verbs	210	302	248	197
Unique Noun+Verb/ Aux-Verb Pairs	1269	1392	552	422
Avg Complexity	2.65	2.07	1.60	2.10
Top 5 Verbs	describe, analyze, provide, summarize, reflect	direct, have, take, play, write	take, compose, use, get, make	answer, solve, ask, modify, explain

Table 12: Data statistics according to instruction composition.

yield generation of diverse MD instructions concerning tasks beyond those seen in typical SFT data, supporting the complementary nature of our instruction data and confirming the design of MDCure to reflect a broad array of task types and content.

B.7 GPT Scoring Prompt

We use the prompt shown in Figure 10 to elicit effective scores from GPT-3.5-Turbo for the candidate generations in a zero-shot fashion. This prompt is adapted from that used in the self-curation step of Li et al. (2024); other prompt variations (e.g., additive scoring as in (Yuan et al., 2024), rubric-free scoring as in (Li et al., 2023b)) yielded relatively worse MD performance in models instruction-tuned with the resulting filtered samples and were therefore discarded from consideration.

B.8 MDCureRM Training Data Prompt

As described in §3.2, we use the prompt shown in Figure 11 to generate fine-grained reward training data for MDCureRM.

B.9 MDCure Data Scoring

Given a candidate instruction-answer pair, MDCureRM generates a vector of six floating point values in the range $[0, 1]$. To re-scale each value to the 1-5 range, we multiply each criterion score by 4 and add 1. To obtain the overall score for the instruction sample, we take a weighted sum of the resulting values, using a weight of $\frac{1}{9}$ for the general quality criteria (Relevance, Coherence & Factuality, Creativity) and a weight of $\frac{2}{9}$ for the multi-document specific criteria (Context Integration, Inter-Document Relationships, Complexity). In §A.2, we compare this weighting assignment with the evenly weighted alternative, which utilizes a weight of $\frac{1}{6}$ for each criterion in the case where all 6 criteria are used, and a weight of $\frac{1}{5}$ for each criterion in the case where one criterion is ablated at a time.

B.10 MDCure Dataset Statistics

We summarize statistics for the unfiltered, GPT-3.5-Turbo-filtered, and MDCureRM-Filtered instruction data as shown in Table 13. Quantitatively, we observe that the MDCureRM-Filtered data tends to have a slightly higher number of documents per sample compared to the GPT-filtered data. Meanwhile, the instruction lengths in the MDCureRM-Filtered and unfiltered pools are quite similar, both having longer and more detailed instructions on average, whereas the GPT-filtered pool features shorter instruction lengths. Complexity in terms of context length and instruction lengths is apparent across all data settings, but the MDCureRM-Filtered data handles the most complex tasks while GPT-filtered data leans towards handling smaller, more straightforward tasks.

Data Setting	# Instr.	Avg. # Context Docs	Avg. Instr. Length
Unfiltered	221,835	3.4	1,754
GPT-Filtered	70,598	3.2	1,377
RM-Filtered	125,994	3.4	1,719

Table 13: Data statistics according to instruction composition.

B.11 Long-Context & Few-Shot Data Construction

For long-context and few-shot post-training experiments (§5.3), we construct instruction inputs by incorporating distractor articles or few-shot examples, respectively. Distractor articles are identified by embedding all NewSHead documents using the Sentence Transformers model `all-distilroberta-v1`,[11] and taking the most cosine-similar documents to the documents in the source cluster. We choose similar instead of random documents to increase task difficulty and encourage identification of relevant documents in a long context, as recent work finds using packing related texts during training can assist models to read and reason across text boundaries (Shi et al., 2023). For few-shot training, exemplars are chosen randomly, with each exemplar appearing once across the duration of training.

[11] We choose this embedding model from https://www.sbert.net/docs/sentence_transformer/pretrained_models.html due to its strong performance, long context length, and compatibility with cosine similarity.

Instruction:
{generation_prompt}

Response:
{generated_instruction_answer_pair}

Above are an Instruction from a user and a candidate Response from an AI Assistant. The goal of this AI Assistant is to generate a Response that effectively addresses the user's Instruction and that, in order to answer, requires the ability to reason over multiple documents. The Response should be a targeted question, instruction, prompt, or task that requires the use of information from different positions in the provided texts.

Evaluate whether the Response is a good example of how an AI Assistant should respond to the user's Instruction. Assign a score to the Response using the following 5-point scale:

1: It means the Response is incomplete, vague, off-topic, or not exactly what the user asked for in the Instruction. Perhaps it provides an incomplete prompt. Or it can be answered without looking at source documents provided in the Instruction. Or some content seems missing, the opening sentence repeats user's question, or it contains is irrelevant to the source documents or snippets provided in the Instruction.
2: It means the Response addresses some of the asks from the user but does not directly address the user's Instruction. For example, the Response only leverages one of several source documents or snippets provided in the Instruction. Or the Response can be answered using only ONE source document or snippet, and thus does not effectively assess and require use of multi-document reasoning capabilities.
3: It means the Response is fair and addresses all the basic asks from the user. It is complete and self contained and is relevant to most of the documents or snippets provided, but not all. It may be somewhat helpful toward assessing an agent's multi-document reasoning capability but still has room for improvement.
4: It means the Response is good quality. Specifically, the Response can only be answered by performing reasoning across most of the documents or snippets provided in the Instruction. The provided documents or snippets include all the information required to answer the Response, i.e. no information beyond that provided in the Instruction is needed. The Response has minor room for improvement, e.g. more concise and focused.
5: It means the Response is perfect, i.e. it can only be answered with strong ability to extract and synthesize information across the documents or snippets provided in the Instruction. The Response utilizes ALL documents or snippets provided in the instruction. The provided documents or snippets include all the information required to answer the Response. It is well-written and effective toward the AI Assistant's goal and has no irrelevant content.

Assess the Response and assign a rating score using this scale. Respond with "Score: <rating>".

Figure 10: Prompt used for GPT-3.5-Turbo to evaluate the quality of candidate MD instruction-answer pairs.

Instruction Quality Rating Task
Rate the quality of the generated instruction based on the provided documents, using a scale of 1-5. Only provide numerical scores, without any rationale or explanation.

Relevance: Does the instruction align well with the content of the documents? Does it make sense given the provided information?
Coherence & Factuality: Is the instruction-answer pair coherent, logical, and factually accurate? Does the answer appropriately address the instruction and is it well-supported by the documents?
Creativity: How diverse or creative is the instruction in terms of question type (e.g., factual, inferential) and format (e.g., multiple-choice, open-ended)?
Context Integration: How well does the instruction leverage and synthesize information from multiple documents to form a comprehensive response?
Inter-Document Relationships: Does the instruction encourage understanding relationships (e.g., comparisons, contrasts, discrepancies) between different documents?
Complexity: Does the instruction appropriately challenge the answerer to think critically and synthesize information from multiple sources?

Input:
Context: {context_docs}
{instruction_sample}

Output (provide only numerical scores, no rationale):
Relevance: [score]
Coherence & Factuality: [score]
Creativity: [score]
Context Integration: [score]
Inter-Document Relationships: [score]
Complexity: [score]

Figure 11: Prompt to elicit fine-grained score training data for MDCureRM.

C Training Details

C.1 Instruction Tuning

All FlanT5 models are trained using 8 NVIDIA A6000 GPUs or 8 A100 40GB GPUs. For LLAMA3.1 and Qwen2 fine-tuning experiments, we employ the widely-used Unsloth repository with HuggingFace to perform 4-bit quantization and Low-Rank Adaptation (LoRA). We choose to use LoRA over full fine-tuning due to compute limitations and since preliminary experiments yielded similar performance in both settings. LoRA adapters are used for every linear layer for all query-, key-, value-, output-, gate-, up-, and down-projection matrices with a rank of 16, alpha parameter of 32, and zero adapter dropout. As Unsloth only accommodates single-GPU training, all LLAMA3.1 and Qwen2 models are trained on a single A6000 or A100 GPU. All fine-tuning experiments utilized gradient checkpointing aside from those involving FlanT5-Base.

As the FlanT5 models could be trained with a maximum length of 4096 tokens without GPU memory overflow, we set the maximum input length of the training data to 4096 for all fine-tuning experiments beyond those in §5.3 to ensure comparability. Any samples exceeding this length were truncated from the right, by shortening component context documents to equal lengths and preserving the entirety of the subsequent instruction. For fine-tuning experiments with long-context and few-shot MDCure data (§5.3), we set the maximum input length to 32000 tokens.

All fine-tuning experiments utilized a global batch size of 64, AdamW optimization with $\beta_1 = 0.9$ and $\beta_2 = 0.98$, a linear learning rate schedule, warmup ratio of 0.15, dropout of 0.05, weight decay of 0.01, and the best of either 1e-4 or 5e-5 for the maximum learning rate. Fine-tuning is performed for 1100, 1650, and 2250 steps for the 12K-, 36K-, and 72K-training sample settings. For the long-context and few-shot training settings, we use a total of 36K samples, with the first 90% of training steps utilizing 4096 maximum input length and the remaining 10% utilizing 32000 maximum input length, in a similar fashion to Dubey et al. (2024). In terms of training data composition, unless otherwise stated, all models employed a 1:3 ratio of General:Style-Specific prompt template generations. Using the hardware specified above, all fine-tuning experiments require 24-120 GPU-hours.

In all settings, we determine the best checkpoint for each model by comparing checkpoint performances in the zero-shot prompting setting on 500 test samples from MultiNews, Multi-XScience, HotpotQA-distractor, WikiHop, and QMDSCNN; we use LLM evaluation to assess MDS performance and F1 and EM scores to assess MHQA performance. Evaluation details are provided in App. D.4. For language modeling settings, we apply the loss to the instruction input as well during training, per recommendation by Shi et al. (2024).

C.2 MDCureRM Training

To train MDCure RM, we utilize 8 A100 40GB GPUs. Training occurs for 4 epochs using a training set of size approximately 17K samples and validation and test sets of size approximately 1.5K. We set the global batch size to 64 and use AdamW 8-bit optimization, a linear learning rate schedule, warmup ratio of 0.1. We perform a small hyperparameter search over learning rates of 7e-6, 1e-5, 5e-5, 1e-4, 7e-4, and 1e-3.

Beyond use of a single regression head to construct MDCureRM as in Wang et al. (2024c), we investigate the impact of several variations in the architecture and design of MDCureRM. These variations include the addition of 2 or 3 linear layers to the backbone LLM as opposed to a single regression head, as well as use of an MLP head with attention pooling. Due to compute limitations we were unable to investigate the impact of unfreezing the backbone LLAMA3 model during MDCureRM training. Across these variations, we determined the best scoring model based on test loss, finding use of a single regression head and learning rate 1e-3 most effective. Small-scale experiments on FlanT5-Base demonstrated this improves over the MLP variant.

C.3 PPO Training

For PPO training of LLAMA3.1-8B-Instruct (§5.5), we utilized 4 A100 80GB GPUs. Training occurred over 72K samples for a single epoch, with a global batch size of 64, learning rate of 1e-5, and LORA as described in §C.1. Additional hyperparameter search was performed over learning rates of 1e-4 and 1e-3. All other hyperparameter settings remained consistent with default values.

D Evaluation Details

D.1 Benchmark Descriptions

We provide details on the datasets used to benchmark our instruction-tuned models versus baselines. All benchmarks are in English.

HotpotQA (Yang et al., 2018) is a multi-hop QA dataset consisting of approximately 113,000 questions and paragraphs sourced from Wikipedia, where for each sample there are eight paragraphs that serve as distractors and two paragraphs containing the information necessary to answer the given question. The objective is to identify the correct answer and evidence spans in the text.

WikiHop (Welbl et al., 2018) is a multi-hop QA dataset consisting of approximately 43,000 samples constructed from entities and relations from WikiData and supporting documents from WikiReading. Each sample provides a question, 2-79 potential answers, and 3-63 supporting contexts. The aim is to arrive at the correct answer based on the context and question.

Multi-XScience (Lu et al., 2020) is a MDS dataset sourced from Arxiv and Microsoft scientific graphs. The aim is to produce the related works section for a query publication given the abstract of the query paper and several abstracts of other referenced works. Multi-XScience is considered relatively more difficult versus datasets such as Multi-News since it is less susceptible to positional and extractive biases.

QMDSCNN (Pasunuru et al., 2021) is a query-focused MDS dataset of approximately 287K samples created by extending the CNN/DailyMail single-document summarization dataset to the multi-document setting. Given paragraph-scale chunks of various news articles, the aim is to return relevant context portions given queried article titles.

SEAM (Lior et al., 2024) is a recent multi-document benchmark that addresses the sensitivity of LMs to prompt variations through repeated evaluations with controlled input-output formats on a diverse set of multi-document datasets. Tasks include multi-document summarization, multi-hop question-answering, and cross-document coreference resolution.

ZeroSCROLLS (Shaham et al., 2023) is a collection of datasets for long-context natural language tasks across domains of summarization, QA, sentiment classification, and information reordering. Successful performance on the benchmark requires synthesizing information across long texts.

D.1.1 Dataset Abbreviations

We provide a list of dataset name abbreviations in Table 14.

Dataset Name	Abbreviation
HotpotQA-distractor	HQA
WikiHop	WH
Multi-XScience	MXS
QMDSCNN	QC
SEAM	
MultiNews	MN
OpenAsp	OA
MuSiQue	MSQ
SciCo	SC
ECB+	ECB+
ZeroScrolls	
GovReport	GvRp
SummScreenFD	SSFD
QMSum	QMsm
NarrativeQA	NrQA
Qasper	Qspr
QuALITY	QLTY
SQuALITY	SQLT
MuSiQue	MuSQ
SpaceDigest	SpDg
BookSumSort	BkSS
ZeroScrolls (ZS) Score	ZSS

Table 14: Dataset name abbreviations used for results tables in the main text.

D.2 Baselines

PRIMERA (Xiao et al., 2022) is a 447M-parameter model initialized from LED-large (Beltagy et al., 2020) and trained using an entity salience-based pre-training objective over the NewSHead dataset. It previously stood as the state-of-the-art in multi-document summarization.

QAMDen (Caciularu et al., 2023) is a 447-parameter model initialized in PRIMERA sparse attention style over LED-large and trained using a QA-based pre-training objective over the NewSHead dataset. It surpasses PRIMERA in performance on MDS, QMDS, and MDQA tasks.

LongAlign (Bai et al., 2024) is a recipe for long-context alignment that enables improved long-context performance of fine-tuned models through a combination of data and packing strategies during training. It provides a series of instruction-tuned long-context models up to 13B parameters in size adapted from the ChatGLM3 (GLM et al., 2024) and LLAMA2 (Touvron et al., 2023) model families. We use LongAlign-7B for comparability to Qwen2-7B-Instruct and LLAMA3.1-8B-Instruct, which we use for MDCure experiments.

ProLong-8B-64k-Instruct (Gao et al., 2024b) is continued-trained and supervised-fine-tuned from the LLAMA-3-8B model, with a maximum context window of 64K tokens. It achieves state-of-the-art long-context performance versus similarly sized models and outperforms LLAMA3.1-8B-Instruct on many long-context tasks.

Jamba 1.5 Mini (Labs, 2024) is a speed-optimized model with 52B total parameters and 12B active parameters that is specifically trained for improved long-context handling and quality. It utilizes a hybrid transformer architecture.

D.3 Zero-Shot Evaluation Prompts

To perform zero-shot prompting evaluations on Multi-XScience, QMDSCNN, HotpotQA, and WikiHop, we devise effective zero-shot prompts for each dataset in a similar fashion to Shaham et al. (2023) and utilize the same input format as Shaham et al. (2023) as shown in Figure 12. For each model family (FlanT5, LLAMA3.1, Qwen2), we choose the best one of 5 tailored prompt variations based on base model performance to utilize for evaluations of all subsequent instruction-tuned models. We display the final prompts utilized for each model family in Figure 13. During inference, we use greedy decoding and limit the model generation length to at most the 90th percentile of the maximum target output length. The input context is truncated to length 8192 for every model, and for all results reported in the main text, we utilize 1500 test samples per dataset as opposed to using the entire test set for each dataset to accommodate cost limitations imposed by LLM evaluations for MDS.

D.4 LLM Evaluation Details

As noted in §4, to evaluate MDS performance, we turn to LLM evaluation to overcome limitations of n-gram-based methods such as ROUGE in long-context settings (Shaham et al., 2023). We employ the open-source LLM evaluation framework G-Eval (Liu et al., 2023b), which is one of the most highly human-correlated evaluation frameworks to date when paired with GPT-4. For a given document and summary, G-Eval determines the quality of the summary by issuing four fine-grained scores for Relevance, Coherence, Consistency, and Fluency. The overall LLM score for a given summarization response is obtained by taking the average of the four scores and normalizing the result to the 0-100 range. We adapt G-Eval to the multi-document setting using the prompts shown in Figures 14, 15, 16, and 17.

As GPT4 is expensive, we investigate use of several other commercial LLMs with the adapted G-Eval framework for evaluation on Multi-XScience and QMDSCNN, with the goal of ensuring strong alignment of resulting ratings with human evaluation. To validate our choice of LLMs to evaluate MDS performance, we employ two expert annotators similarly to the procedure used in Fabbri et al. (2021). We observe strong alignment between annotators as indicated by the inter-annotator agreements presented in Table 15, measured via the Krippendorff's alpha coefficient. Given the demanding nature of the annotation task and cost of human expert annotations, since the agreement is high additional annotations were deemed not necessary.

	Multi-XScience	QMDSCNN
Krippendorff α	0.91	0.94

Table 15: Inter-annotator agreements for human scores on summarization performance across six base models.

	Multi-XScience	QMDSCNN
Pearson's r	0.54	0.58

Table 16: Correlations between LLM and average human scores of summarization performance across six base models. GPT-3.5-Turbo is used for evaluation of Multi-XScience performance and Gemini-1.5-Flash is used for evaluation of QMDSCNN performance.

Based on computed correlations with scores from the annotators on 500 samples per dataset, we find GPT-3.5-Turbo and Gemini-1.5-Flash to be most human-aligned for evaluation of model performances on Multi-XScience and QMDSCNN, respectively, and therefore use these for all reported evaluations. The correlations are displayed in Table 16. These values are comparable with the human-LLM score correlations observed in leading LLM evaluation works such as G-Eval (Liu et al., 2023b), from which we adapted our LLM evaluation prompts.

E Meta-Evaluation on MDCureRM

To validate the reliability of using MDCureRM as an evaluator for MD instruction data, we conducted a meta-evaluation study to compare MDCureRM with humans. Here we provide additional details regarding the prompt and setup used in the meta-evaluation study. We utilized two annotators (graduate students in NLP working directly with LLMs) to obtain fine-grained scores from 1-5 according to the six criteria defined in MDCure on 100 randomly sampled MDCure-generated candidate instructions. Each candidate was generated using a different

Zero-Shot Input Format for HotpotQA

{prompt}

Question:
{question}

Supporting Documents:
{supporting_documents}

Answer:

Zero-Shot Input Format for WikiHop

{prompt}

Documents:
{supporting_documents}

Question:
{question}

Answer Candidates:
{answer_choices}

Answer:

Zero-Shot Input Format for Multi-XScience

{prompt}

Documents:
{source_and_reference_abstracts}

Related Work Section:

Zero-Shot Input Format for QMDSCNN

{prompt}

Query:
{question}

Articles:
{articles}

Summary:

Figure 12: Task input formats used for zero-shot prompting evaluation on HotpotQA, WikiHop, Multi-XScience, and QMDSCNN, adapted from Shaham et al. (2023).

Final Task Prompts for FlanT5 Models

- HotpotQA: "You are given a question and multiple supporting documents separated by ‘|||||’. Answer the question as concisely as you can, using a single phrase if possible."
- WikiHop: "You are given multiple supporting documents separated by ‘|||||’, a question, and list of answer candidates. Answer the question by selecting one of the provided answer candidates."
- Multi-XScience: "You are given several scientific documents, separated by ‘|||||’. Write the related-work section of a paper based on the given abstracts and articles. Answer in a single paragraph."
- QMDSCNN: "You are given a query and an article. Answer the query as concisely as you can by summarizing the relevant information from the article. Respond in a single paragraph."

Final Task Prompts for LLAMA3.1 Models

- HotpotQA: "You are given a question and several supporting documents. Answer the question as concisely as you can, using a single word or phrase."
- WikiHop: "You are given multiple supporting documents separated by ‘|||||’, a question, and list of answer candidates. Answer the question by selecting one of the provided answer candidates."
- Multi-XScience: "You are given several scientific abstracts. Write a related works section based on the abstracts. Answer in a short paragraph. Be very brief."
- QMDSCNN: "You are given a query and several articles. Answer the query as concisely as you can by supplying the relevant information from the articles. Respond briefly, in less than one paragraph."

Final Task Prompts for Qwen2 Models

- HotpotQA: "You are given a question and multiple supporting documents separated by ‘|||||’. Answer the question as concisely as you can, using a single word or phrase."
- WikiHop: "You are given multiple supporting documents separated by ‘|||||’, a question, and list of answer candidates. Answer the question by selecting one of the provided answer candidates."
- Multi-XScience: "You are given several scientific abstracts, separated by ‘|||||’. Write the related works section of a paper based on the given abstracts. Answer in a single paragraph."
- QMDSCNN: "You are given a query and several articles. Answer the query as concisely as you can by supplying the relevant information from the articles. Respond briefly, in less than one paragraph."

Figure 13: Task prompts yielding the best performance for base models in the FlanT5, LLAMA3.1, and Qwen2 model families, selected from among 5 tailored variations.

G-Eval Prompt Template for LLM Evaluation of MDS Relevance

You will be given one summary written for a set of related documents.

Your task is to rate the summary on one metric.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Relevance (1-5) - selection of important content from the source. The summary should include only important information from the source documents. Annotators were instructed to penalize summaries which contained redundancies and excess information.

Evaluation Steps:

1. Read the summary and the source documents carefully.
2. Compare the summary to the source documents and identify the main points of the articles.
3. Assess how well the summary covers the main points of the source documents, and how much irrelevant or redundant information it contains.
4. Assign a relevance score from 1 to 5.

Example:

Source Text:

{documents}

Summary:

{summary}

Evaluation Form (scores ONLY):

- {Relevance}

Figure 14: G-Eval prompt for NLG evaluation of the relevance of a given summarization response.

G-Eval Prompt Template for LLM Evaluation of MDS Coherence

You will be given one summary written for a set of related documents.

Your task is to rate the summary on one metric.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Coherence (1-5) - the collective quality of all sentences. We align this dimension with the DUC quality question of structure and coherence whereby "the summary should be well-structured and well-organized. The summary should not just be a heap of related information, but should build from sentence to a coherent body of information about a topic.

Evaluation Steps:

1. Read the source documents carefully and identify the main topic and key points.
2. Read the summary and compare it to the source documents. Check if the summary covers the main topic and key points of the source documents, and if it presents them in a clear and logical order.
3. Assign a score for coherence on a scale of 1 to 5, where 1 is the lowest and 5 is the highest based on the Evaluation Criteria.

Example:

Source Text:

{documents}

Summary:

{summary}

Evaluation Form (scores ONLY):

- {Coherence}

Figure 15: G-Eval prompt for NLG evaluation of the coherence of a given summarization response.

G-Eval Prompt Template for LLM Evaluation of MDS Consistency

You will be given one summary written for a set of related documents.

Your task is to rate the summary on one metric.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Consistency (1-5) - the factual alignment between the summary and the summarized sources. A factually consistent summary contains only statements that are entailed by the source documents. Annotators were also asked to penalize summaries that contained hallucinated facts.

Evaluation Steps:

1. Read the source documents carefully and identify the main facts and details they present.
2. Read the summary and compare it to the source documents. Check if the summary contains any factual errors that are not supported by the source documents.
3. Assign a score for consistency based on the Evaluation Criteria.

Example:

Source Text:

{documents}

Summary:

{summary}

Evaluation Form (scores ONLY):

- {Consistency}

Figure 16: G-Eval prompt for NLG evaluation of the consistency of a given summarization response.

G-Eval Prompt Template for LLM Evaluation of MDS Fluency

You will be given one summary written for a set of related documents.

Your task is to rate the summary on one metric.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

Fluency (1-3): the quality of the summary in terms of grammar, spelling, punctuation, word choice, and sentence structure.

- 1: Poor. The summary has many errors that make it hard to understand or sound unnatural.
- 2: Fair. The summary has some errors that affect the clarity or smoothness of the text, but the main points are still comprehensible.
- 3: Good. The summary has few or no errors and is easy to read and follow.

Evaluation Steps:

1. Read the summary.
2. Assign a score for fluency based on the Evaluation Criteria.

Example:

Source Text:

{documents}

Summary:

{summary}

Evaluation Form (scores ONLY):

- {Fluency}

Figure 17: G-Eval prompt for NLG evaluation of the fluency of a given summarization response.

prompt template over distinct news article clusters from NewSHead. Annotators were informed of the purpose, aims, and intended use of the study and annotations, and informed consent was collected prior to their performing the task. No compensation was provided given the small-scale nature of the task. The meta-evaluation task was formulated using the direction shown in Figure 18, and ratings were collected via a Google form.

	Human	Human & GPT	Human & MDCureRM
Krippendorff α	0.81	0.18	0.33

Table 17: IAA for human annotations, versus IAA and correlation for human and LLM annotations.

The results of the meta-evaluation study are shown in Table 17. We compute the inter-annotator agreement (IAA) among human annotators by calculating the Krippendorff's alpha coefficient for each criterion and then taking the average over all 6 criteria. We also compute the IAA when GPT-3.5-Turbo and MDCureRM are each separately considered as an additional annotator to demonstrate the value of MDCureRM as a cost-effective, more human-aligned alternative to proprietary model scoring. Overall, we find that while GPT-3.5-Turbo and MDCureRM have weak agreement with human annotators, MDCureRM yields stronger alignment with humans versus GPT. This is consistent with demonstrated MD performance gains when using MDCureRM-filtered instruction data as opposed to GPT-filtered data for IT. We note that since MDCureRM is trained using data with target scores generated by GPT-4o, its alignment with human scoring could perhaps be improved by using human annotations of training data in future work.

F Full Experimental Results

We display full experimental results, previously abbreviated in Table 1, in Table 18.

Meta-Evaluation Annotation Task Directions

Task Description
In this task, you will evaluate the quality of instruction-answer pairs generated by a large language model (LLM). Each pair is based on a set of related news articles, with the instruction potentially being a question, command, or descriptive phrase.

Your goal is to assess how helpful each instruction-answer pair would be for instruction-tuning an LLM, specifically to improve its multi-document capabilities.

To this end, you will score each instruction-answer pair across six key criteria, which focus on both the pair's overall quality (criteria 1-3) and the pair's effectiveness in handling multi-document content (criteria 4-6). While evaluating, prioritize the instruction over the answer, except where the criterion explicitly calls for focus on the answer (e.g., criterion #2). But loosely keep in mind the quality of the answer, especially in cases where the answer's accuracy or depth affects the overall value of the pair.

To correctly complete the task, please follow these steps:

- Keep this document open on the side, such that this document and the Google Form for responses are both visible at once.
- Read the context documents, to be aware of the information contained in each article. Note that some samples have many documents or very long documents. It is not required to scrutinize the articles in meticulous detail; rather, the annotator should obtain an understanding of the content so as to make accurate judgments regarding the instruction-answer pair.
- Read the proposed instruction and answer.
- Rate the instruction-answer pair on a scale from 1 (worst) to 5 (best) according to the following criteria: relevance, coherence & factuality, creativity, context integration, inter-document relationships, and complexity.

Criteria Definitions

- **Relevance**: How well does the instruction align with the context provided by the documents? This involves assessing the pertinence of the instruction to the content and the degree to which it makes sense given the information in the documents.
- **Coherence & Factuality**: Do the instruction and answer form a coherent, logical, and factually accurate pair? This involves checking if the answer directly addresses the instruction, if the level of detail in the answer is appropriate for the instruction, and if the reasoning behind the answer is sound and well-supported by the documents.
- **Creativity**: How creative or diverse is the instruction in terms of the type of question or task it involves? This involves assessing the question type (e.g., factual, inferential, analytical, summarization, etc.) and answer format (e.g., multiple-choice, open-ended, etc.).
- **Context Integration**: How well does the instruction leverage the context provided by multiple documents? This assesses whether the instruction involves synthesizing information from various documents to form a comprehensive response.
- **Inter-Document Relationships**: How well does the instruction encourage understanding relationships between different documents? This assesses whether the instruction requires comparing, contrasting, drawing connections, or identifying discrepancies between the documents.
- **Complexity**: Does the instruction appropriately and effectively challenge the answerer's ability to understand and leverage information across multiple documents? This evaluates how well the instruction involves higher-order thinking skills like analysis, synthesis, and evaluation.

Figure 18: Instructions for human annotation task.

		Model / IT Data Setting	HQA	WH	MXS	QC	SEAM						ZeroScrolls (ZS)											Avg
							ECB+	MN	MSQ	OA	SC	Avg	GvRp	SSFD	QMsm	NrQA	Qspr	QLTY	SQLT	MuSQ	SpDg	BkSS	ZSS	
FlanT5	Base	PRIMERA	0.4	0.5	70.7	24.2	7.1	7.7	0.5	3.9	10.4	5.9	6.9	4.6	3.5	0.6	1.5	13.4	9.7	0.2	1.9	0.0	4.2	8.8
		QAMDen	1.8	1.9	63.6	27.1	0.0	0.4	0.0	1.0	3.1	0.9	6.0	6.2	5.7	2.4	6.1	3.4	7.1	0.9	0.1	0.0	3.8	7.2
		None	4.4	45.1	38.7	48.0	0.0	5.8	0.2	2.6	0.1	1.7	5.2	5.4	9.9	16.7	14.2	48.2	4.8	26.9	0.0	0.0	13.1	14.5
		Unfiltered	31.7	47.5	89.8	52.2	0.0	6.7	0.1	2.7	0.2	1.9	5.3	8.6	12.4	15.5	28.7	46.6	6.2	30.0	54.3	2.4	21.0	23.2
		GPT-Filtered	44.1	46.0	92.4	54.2	0.0	6.0	0.2	2.7	0.0	1.8	6.1	9.2	13.4	16.4	25.8	49.6	6.8	27.8	54.4	3.6	21.3	24.1
		MDCureRM	**47.3**	**48.3**	**93.8**	**57.3**	0.0	7.5	0.3	2.7	0.0	**2.1**	6.6	9.2	14.5	16.1	34.6	48.2	8.1	31.2	54.4	3.4	**22.6**	**25.4**
	Large	None	24.4	54.6	70.9	61.8	1.1	7.9	0.1	3.1	0.3	2.5	19.5	13.5	15.5	9.4	24.4	31.0	21.1	11.8	44.4	41.1	23.2	24.0
		Unfiltered	46.9	62.9	91.1	64.7	0.7	8.0	0.0	3.8	0.1	2.5	6.6	10.4	12.1	19.7	37.6	61.8	9.3	35.1	48.7	0.1	24.2	27.3
		GPT-Filtered	46.3	65.1	91.8	64.0	0.5	8.3	0.0	3.9	0.1	2.6	7.2	9.9	11.9	19.8	39.1	63.4	8.3	33.8	48.8	0.0	24.2	27.5
		MDCureRM	**49.6**	**66.1**	**93.1**	**66.0**	1.2	9.1	0.1	4.2	0.0	**2.9**	7.5	10.1	12.8	20.0	43.2	64.0	10.7	35.4	48.6	0.1	**25.3**	**28.5**
Qwen2-Ins	1.5B	None	21.5	17.8	93.3	73.3	9.4	10.6	0.5	5.0	13.0	7.7	17.8	11.4	15.0	13.3	33.8	48.2	16.7	14.5	58.7	10.5	24.0	25.5
		Unfiltered	32.9	30.5	94.2	79.3	15.7	12.0	0.4	5.0	16.8	10.0	14.1	9.8	15.3	15.7	40.3	48.0	15.7	16.1	53.8	10.1	23.9	27.7
		GPT-Filtered	33.3	32.9	94.2	81.3	16.1	12.0	0.5	5.1	15.9	9.9	17.9	11.3	14.5	13.8	35.9	49.2	16.5	16.0	56.9	10.1	24.2	28.1
		MDCureRM	**37.7**	**34.8**	**94.4**	**82.9**	16.4	12.0	0.6	5.9	18.1	**10.6**	19.8	12.0	14.9	13.1	38.0	55.0	16.4	19.7	54.9	12.5	**25.6**	**29.4**
	7B	LongAlign-7B	10.4	14.3	92.2	83.3	11.5	16.5	0.0	4.1	16.8	9.8	19.5	13.5	15.5	9.4	24.4	31.0	21.1	11.8	44.4	41.1	23.2	25.3
		None	30.5	39.6	95.6	79.3	5.0	11.9	0.5	6.4	13.1	7.4	21.1	12.2	15.0	13.2	33.8	72.8	19.0	29.4	16.3	6.5	23.9	27.4
		Unfiltered	40.5	43.3	94.7	84.2	8.0	15.3	0.5	6.6	11.9	8.5	14.4	11.9	16.5	17.5	42.7	70.2	17.4	29.6	35.1	7.4	26.3	29.9
		GPT-Filtered	42.0	44.0	94.7	85.3	8.7	15.3	0.9	6.6	12.1	8.7	11.5	12.0	16.8	18.1	43.1	67.4	15.9	28.4	54.1	19.6	28.7	31.4
		MDCureRM	**44.7**	**46.0**	**95.1**	**87.3**	13.8	15.4	0.6	6.7	14.9	**10.3**	25.8	13.2	15.3	14.4	45.8	74.8	19.7	30.8	26.0	31.7	**29.8**	**32.7**
Llama3.1-Ins	8B	ProLong-8B	43.6	34.3	85.8	54.7	9.5	17.4	0.4	10.7	18.0	11.2	23.8	16.0	17.5	23.5	39.9	70.6	22.6	19.2	54.6	48.5	33.6	32.1
		None	35.5	27.1	95.1	65.3	10.5	15.0	0.6	7.6	17.4	10.2	23.7	5.8	5.2	10.5	4.6	75.6	13.3	0.7	47.2	0.3	18.7	24.3
		Unfiltered	37.6	34.4	84.7	90.4	15.3	16.3	0.5	7.8	18.9	11.8	20.2	14.1	17.4	21.9	52.3	50.0	19.4	25.3	32.6	32.1	28.6	31.1
		GPT-Filtered	38.0	42.3	95.3	87.8	8.7	16.0	0.6	7.8	18.3	10.3	20.3	14.2	17.5	22.2	53.6	59.8	19.5	26.2	36.1	26.4	29.6	32.1
		MDCureRM	**44.7**	**43.7**	**95.3**	**93.8**	16.3	16.4	0.6	7.9	18.5	**11.9**	19.9	14.5	17.6	22.6	52.2	68.2	19.2	27.5	34.3	33.4	**30.9**	**34.0**
	70B	Jamba 1.5 Mini	47.5	41.8	94.2	87.1	20.1	14.2	0.0	5.3	20.4	12.0	21.1	15.1	17.3	22.2	47.3	84.4	20.0	28.1	59.1	26.0	34.1	35.3
		None	53.9	38.1	95.1	88.2	25.1	21.9	0.6	6.1	11.5	13.0	21.9	14.9	18.1	25.5	45.5	66.7	23.4	42.5	61.4	43.8	36.4	37.1
		Unfiltered	55.9	40.6	88.7	70.4	3.6	21.9	0.6	6.3	11.2	8.7	20.5	14.8	17	32.4	49.7	77.4	18	33.1	62.8	23.4	34.9	34.1
		GPT-Filtered	57.4	41.2	88.2	74.9	4.9	22.0	0.7	6.4	10.7	8.9	21.5	15.3	18.3	30.1	50	83.8	19.3	32.7	65.2	38.7	37.5	35.9
		MDCureRM	**58.4**	**45.5**	**95.1**	**88.7**	25.2	22.1	0.7	6.7	12.0	**13.3**	21.5	15.4	18.3	29.8	50.7	83.4	19.4	33.6	66.0	38.7	**37.7**	**38.5**
		GPT-4o	57.5	50.0	**100.0**	**94.4**	8.9	18.6	0.3	14.2	28.9	14.1	25.7	14.8	15.9	35.3	58.8	88.6	20.7	49.2	83.5	5.6	**39.8**	**40.6**
		Gemini 1.5 Pro	**66.6**	**55.6**	93.8	79.8	21.4	17.8	0.6	15.1	30.1	**17.0**	22.1	14.3	14.3	53.9	36.1	85.7	18.1	36.1	39.4	0.0	32.0	36.9

Table 18: Full evaluation of MDCure'd models versus baselines on 6 benchmarks in the zero-shot prompting setting. The rightmost "Avg" column reports the average of individual dataset scores. Dataset abbreviations are described in App. D.1.1. Rows specified by "MDCureRM" refer to our full MDCure pipeline applied to the corresponding base model and size. Size-comparable baselines are highlighted in blue and results of our final method is highlighted in yellow. Bold numbers show best performance in each group.