Instruction Pre-Training:
Language Models are Supervised Multitask Learners

Daixuan Cheng[†*] **Yuxian Gu**[‡] **Shaohan Huang**[†✉] **Junyu Bi**[†] **Minlie Huang**[‡✉] **Furu Wei**[†]

[†] Microsoft Research

[‡] The CoAI Group, DCST, Institute for Artificial Intelligence, State Key Lab of Intelligent Technology and Systems, Beijing National Research Center for Information Science and Technology, Tsinghua University

daixuancheng6@gmail.com guyx21@mails.tsinghua.edu.cn

{shaohanh,fuwei}@microsoft.com aihuang@tsinghua.edu.cn

https://huggingface.co/instruction-pretrain

Abstract

Unsupervised multitask pre-training has been the critical method behind the recent success of language models (LMs). However, supervised multitask learning still holds significant promise, as scaling it in the post-training stage trends towards better generalization. In this paper, we explore *supervised multitask pre-training* by proposing *Instruction Pre-Training*, a framework that scalably augments massive raw corpora with instruction-response pairs to pre-train LMs. The instruction-response pairs are generated by an efficient instruction synthesizer built on open-source models. In our experiments, we synthesize 200M instruction-response pairs covering 40+ task categories to verify the effectiveness of *Instruction Pre-Training*. In pre-training from scratch, *Instruction Pre-Training* not only consistently enhances pre-trained base models but also benefits more from further instruction tuning. In continual pre-training, *Instruction Pre-Training* enables Llama3-8B to be comparable to or even outperform Llama3-70B. Our model, code, and data are available at https://github.com/microsoft/LMOps.

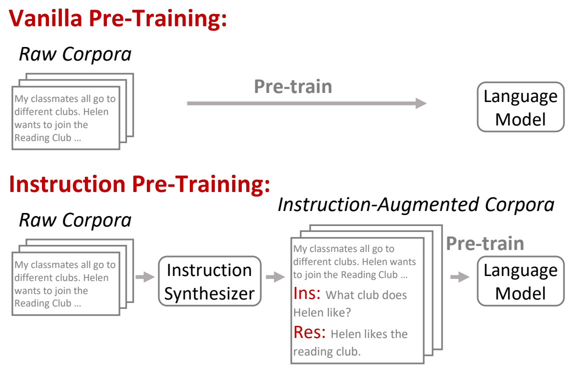

Figure 1: **Comparison between Instruction Pre-Training and Vanilla Pre-Training.** Instead of directly pre-training on raw corpora, *Instruction Pre-Training* augments the corpora with instruction-response pairs generated by an instruction synthesizer, then pre-trains LMs on the augmented corpora. "Ins" and "Res" represent instruction and response, respectively.

1 Introduction

On the path towards general artificial intelligence, multitask learning (Caruana, 1997) emerges as a promising approach. However, scaling supervised multitask learning to the necessary degree is very challenging. This motivates GPT-2 (Radford et al., 2019) to explore unsupervised multitask learning: pre-training on raw corpora through causal language modeling, which facilitates scaling up training data. Over time, unsupervised multitask learning has evolved into the standard approach for pre-training language models (LMs) (Brown et al., 2020; Chowdhery et al., 2023), which is referred to as *Vanilla Pre-Training* in this paper.

Despite the success of unsupervised approaches, supervised multitask learning still holds significant promise. Instruction tuning (Wei et al., 2021), which fine-tunes pre-trained models using diverse tasks framed through natural language instructions, significantly enhances task generalization (Sanh et al., 2021; Chung et al., 2024), re-emphasizing the value of supervised multitask learning.

In this paper, we introduce *Instruction Pre-Training* to explore supervised multitask learning for pre-training. As shown in Figure 1, instead of directly pre-training on raw corpora, *Instruction Pre-Training* augments each raw text with a set of instruction-response pairs[1] generated by an instruction synthesizer, and then pre-trains LMs using the augmented corpora. These pairs are synthesized based on the content of massive raw corpora, ensuring high knowledge coverage and correctness. Therefore, we can scale up task synthesis with great diversity and quality (Li et al., 2023a).

[*] Contribution during an internship at the CoAI Group, Tsinghua University. [✉] Corresponding Author.

[1]We use "task" and "instruction-response pair" interchangeably, with the instruction as task input and the response as task output.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 2529–2550
November 12-16, 2024 ©2024 Association for Computational Linguistics

To develop the instruction synthesizer, we convert a wide range of existing datasets into our required format: each example consists of a set of instruction-response pairs and a piece of raw text that these pairs condition on. Using this data collection, we fine-tune a language model to generate instruction-response pairs based the corresponding raw text. The high diversity of the tuning data enables the synthesizer to generalize to unseen data, facilitating the synthesis of instruction-response pairs for raw pre-training corpora. Unlike existing works (Li et al., 2023b; Yehudai et al., 2024) using large or closed-source models (OpenAI, 2023; Yehudai et al., 2024) to generate synthetic data, we build our instruction synthesizer based on open-source models (typically with 7B parameters), which is much more cost-effective. This efficiency allows us to further scale up task synthesis: augmenting the raw corpora with 200M instruction-response pairs across more than 40 task categories.

We conduct experiments in both general pre-training from scratch and domain-adaptive continual pre-training. In pre-training from scratch, our 500M model pre-trained on 100B tokens reaches performance of the 1B model pre-trained on 300B tokens. Moreover, models that have undergone *Instruction Pre-Training* gain significantly more from further instruction tuning. In continual pre-training, *Instruction Pre-Training* consistently improves performance of Llama3-8B[2] on two domains: finance and biomedicine, enabling it to be comparable to or even surpass Llama3-70B.

In summary, our contributions include:

- We propose *Instruction Pre-Training* to explore supervised multitask pre-training, and verify its effectiveness through extensive experiments.
- We develop an instruction synthesizer capable of scalably generating diverse instruction-response pairs based on various raw corpora.
- We comprehensively analyze the instruction synthesizer and the synthetic data to reveal the key factors towards the success of our method.

2 Instruction Pre-Training

Instead of directly pre-training on raw corpora, *Instruction Pre-Training* augments each text from the raw corpora with a set of instruction-response pairs generated by an instruction synthesizer, where the instruction serves as the task input and the response serves as the task output, then pre-trains LMs on the augmented corpora.

2.1 Instruction Synthesizer

To facilitate the scaling of supervised task learning, we develop an instruction synthesizer to generate instruction-response pairs based on raw corpora. Studies suggest that raw corpora contain numerous intrinsic tasks (Gu et al., 2022b; Chen et al., 2024), which enables efficient scaling of task synthesis (Cheng et al., 2023; Li et al., 2023a; Yue et al., 2024) along with the upscale of raw corpora.

Our instruction synthesizer is developed through multitask fine-tuning on a language model. As illustrated in Figure 2, during tuning, the instruction synthesizer is given a piece of raw text and tuned to create a set of instruction-response pairs. The tuning data are curated to be highly diverse, enabling the instruction synthesizer to generalize to unseen data (Wei et al., 2021). Therefore, during inference, we can directly employ the instruction synthesizer to create instruction-response pairs based on the raw pre-training corpora. Furthermore, we incorporate specific designs to synthesize both one-shot and few-shot examples for subsequent pre-training.

Data Collection We sample from and reformat a diverse range of context-based task completion datasets, which require models to perform tasks based on a given context, to meet our fine-tuning requirements. Each data sample's context serves as the raw text, and the downstream tasks serve as the instruction-response pairs. The contexts span various domains such as encyclopedias, social media, and academic tests (Rogers et al., 2023), and the tasks encompass a wide range such as commonsense reasoning and sentiment analysis. Further details are in Appendix A.

Tuning We tune the instruction synthesizer using few-shot examples. As depicted in Figure 3, a one-shot example consists of a piece of raw text followed by its instruction-response pairs. Each sequence fed into the synthesizer concatenates multiple such examples, all sampled from the same dataset. This ensures that the concatenation of multiple examples within one sequence constitutes a few-shot example, maintaining consistency in patterns (i.e., task format or category) among different sets of instruction-response pairs. Fine-tuning on these examples enables the instruction synthesizer to generate instruction-response pairs with similar patterns to those in the given examples (Min et al.,

[2]https://llama.meta.com/llama3/

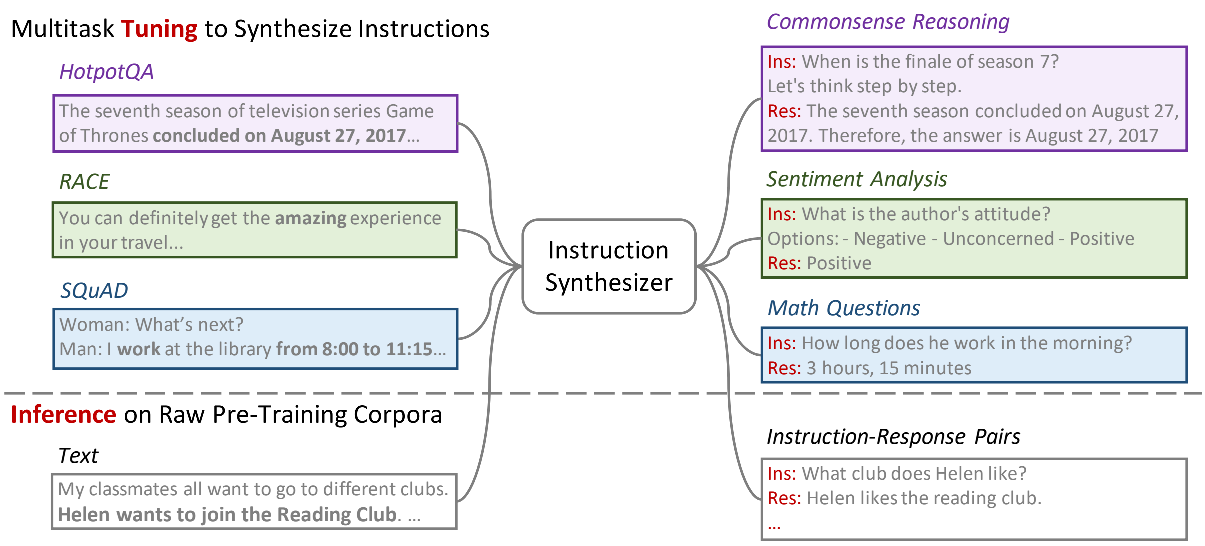

Figure 2: **Tuning and inference framework of instruction synthesizer.** During tuning, the instruction synthesizer learns to generate instruction-response pairs for a given raw text. The tuning data are curated to be highly diverse, enabling the synthesizer to generalize to unseen data. During inference, we use this tuned instruction synthesizer to generate instruction-response pairs for raw texts from pre-training corpora.

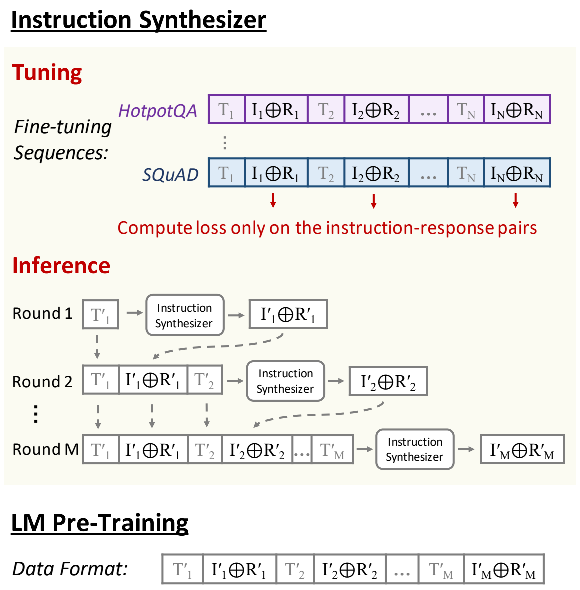

Figure 3: **For instruction synthesizer**, a one-shot example consists of a raw text (T_N) and a set of instruction-response pairs ($I_N \bigoplus R_N$); data denoted without $'$ are for tuning the instruction synthesizer, and data with $'$ are for synthesizer inference and LM pre-training. During instruction synthesizer tuning, each sequence fed into the synthesizer concatenates multiple one-shot examples sampled from the same dataset. During inference, multi-round inference is conducted to synthesize instruction-response pairs with patterns similar to those of previous rounds. **For LM pre-training**, a few-shot example concatenates raw texts and synthesized pairs from multiple rounds.

2022). Additionally, we calculate the tuning loss only on the instruction-response pairs to guide the model to focus on these pairs.

Inference We conduct multi-round inference to create few-shot examples. As depicted in Figure 3, in each round, we prepend the texts and instruction-response pairs from previous rounds to the current text. This allows the instruction synthesizer to generate new instruction-response pairs based on the previous ones.

2.2 LM Pre-Training

After collecting the synthesized instruction-response pairs, we employ templates from Longpre et al. (2023) to diversify instruction formats, and templates from Cheng et al. (2023) to concatenate each raw text with its instruction-response pairs. As shown in Figure 3, by concatenating the texts and instruction-pairs from M rounds, we create an M-shot example for subsequent pre-training.

Except for the pre-training data, *Instruction Pre-Training* keeps all other pre-training settings the same as *Vanilla Pre-Training*: training with the next-token prediction objective (Radford et al., 2018) and computing loss on all tokens. We conduct both general pre-training from scratch and domain-adaptive continued pre-training to verify the effectiveness in different pre-training scenarios.

General Pre-Training From Scratch Considering the large amount of data required for general pre-training from scratch, we only convert part of the raw corpora into instruction-augmented corpora, leaving the rest unchanged. Besides, we mix the corpora with the data for fine-tuning the instruction synthesizer to enhance task diversity.

Domain-Adaptive Continual Pre-Training For domain-adaptive continual pre-training, the data requirement is much smaller. Therefore, we convert all raw corpora into instruction-augmented corpora. Following Cheng et al. (2023), we mix the corpora with the general instructions to benefit from improved prompting ability. Since the general instructions collection contains the fine-tuning data for the instruction synthesizer, we do not include these fine-tuning data.

3 Experiment Settings

3.1 Instruction Synthesizer

Our synthesizer is fine-tuned from Mistral-7B-v0.1 (Jiang et al., 2023), an open-source model with 7B parameters. This model is much more cost-effective than large-scale (Almazrouei et al., 2023; Jiang et al., 2024a; Bai et al., 2023) or closed-source (OpenAI, 2023) models typically used for generating synthetic data (Li et al., 2023b; Yehudai et al., 2024; Yue et al., 2024). During inference, about 5 instruction-response pairs are created per raw text, where each pair contains about 52 tokens. Further tuning and inference details are in Appendix B.

3.2 General Pre-training From Scratch

Pre-Training Corpora We randomly sample a subset of RefinedWeb (Penedo et al., 2023) dataset for raw pre-training corpora, consisting of 200M pieces of text containing about 100B tokens.

To create instruction-augmented corpora, we conduct two rounds of instruction synthesis, converting 1/5 of the raw corpora (40M raw texts) into instruction-augmented texts. The first round converts 20M raw texts, and the second round uses the raw texts and instruction-response pairs from the first round to convert another 20M raw texts. The resulted corpora contain 200M synthesized pairs amounting to about 10B tokens. An example of a 2-shot instruction-augmented text is shown in Table 16 in Appendix.

We then mix the fine-tuning data for instruction synthesizer. Since the fine-tuning data amount (0.2B tokens) is too small compared to that of the raw corpora, we increase its sample ratio so that it repeats 4 times throughout pre-training.

Training and Evaluation We adopt the architecture and tokenizer of Mistral (Jiang et al., 2023) to implement models of two different parameters: 500M and 1.3B.

Our pre-training settings largely follow Brown et al. (2020). To enhance training efficiency, we implement the memory-efficient attention of *xformers* (Lefaudeux et al., 2022). Detailed hyperparameters are listed in Table 14 in Appendix. The lm-evaluation-harness framework (Gao et al., 2023) is used for model evaluation, detailed evaluation settings are in Appendix C.

We also conduct instruction tuning on the pre-trained model with 500M parameters using the data from Longpre et al. (2023). The instruction-tuned models are evaluated on MMLU (Hendrycks et al., 2020) benchmark.

3.3 Domain-Adaptive Continual Pre-Training

Pre-Training Corpora We use raw corpora from two domains: PubMed Abstracts (Gao et al., 2020) for biomedicine and financial news (Yang et al., 2023) for finance.

We conduct 3-round inference to covert all the domain-specific corpora. Each round processes 1/3 of the raw corpora, inheriting the raw texts and instruction-response pairs from previous rounds. Examples of the instruction-augmented texts are in Table 17 and 18 in Appendix.

We then mix the instruction-augmented corpora with general instructions (Zhou et al., 2024; Xu et al., 2023; Lian et al., 2023), using the same mixing ratio as Cheng et al. (2023).

Training and Evaluation We continue to pre-train Llama3-8B on each domain respectively, detailed settings are in Table 14 in Appendix. We follow the prompting settings in Cheng et al. (2023) to evaluate models on the domain-specific tasks. Detailed evaluation settings are in Appendix C.

4 Results

4.1 General Pre-Training From Scratch

Pre-Trained Base Models Table 1 presents the general performance of the models after pre-training. To ensure a fair comparison with *Vanilla Pre-Training*, which uses only raw corpora, we include a baseline (Mix PT) that mixes the raw corpora with the fine-tuning data for our instruction synthesizer. Compared to *Vanilla Pre-Training* (Vanilla PT), incorporating the fine-tuning data in Mix PT improves model performance on

	ARC-e	ARC-c	BoolQ	SIQA	WinoGrande	PIQA	OBQA	HellaSwag	MMLU
500M									
Vanilla PT	50.3	26.4	57.5	44.6	53.8	**71.1**	29.8	47.2	25.4
Mix PT	52.8	26.7	46.8	46.6	52.7	70.1	30.0	47.0	**26.7**
Instruct PT	**54.8**	**27.4**	**62.0**	**47.2**	**54.8**	69.9	**30.8**	**47.3**	25.3
1.3B									
Vanilla PT	58.5	28.8	60.3	47.9	54.9	73.0	**33.6**	**54.9**	25.7
Instruct PT	**60.5**	**30.9**	**62.2**	**49.2**	**55.9**	**73.6**	33.4	54.3	**27.3**

Table 1: **General performance of the pre-trained base models** via *Vanilla Pre-Training* (Vanilla PT), mixing raw corpora with fine-tuning data for the instruction synthesizer (Mix PT), and *Instruction Pre-Training* (Instruct PT) in general pre-training from scratch. All the pre-training methods use the same number of tokens for model training.

	# Param.	# Token	Average
GPT-2	774M	-	45.7
Pythia	1B	300B	47.1
BLOOM	1.1B	341B	45.1
Instruct PT	500M	100B	46.6
OPT	1.3B	300B	49.3
GPT-2	1.5B	-	48.6
BLOOM	3B	341B	50.1
Instruct PT	1.3B	100B	49.7

Table 2: **Comparison between our pre-trained base models and others** on general benchmarks. Detailed results are in Table 15.

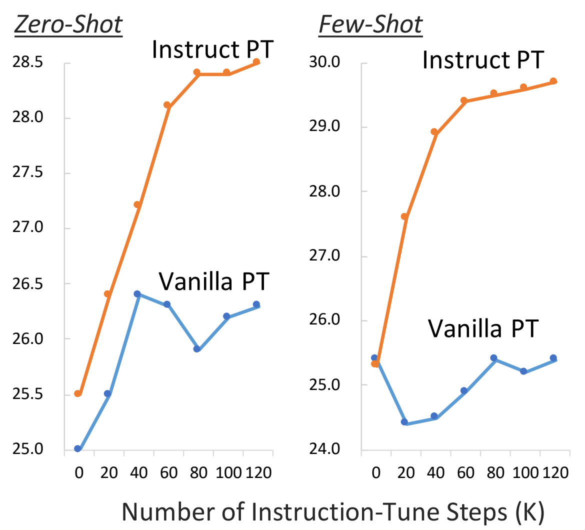

Figure 4: **MMLU performance during instruction tuning** of models pre-trained via *Vanilla Pre-Training* (Vanilla PT) and *Instruction Pre-Training* (Instruct PT).

several benchmarks. By further transforming the raw corpora into instruction-augmented corpora, *Instruction Pre-Training* (Instruct PT) achieves even better performance. Note that none of the evaluated datasets are included in our fine-tuning data for the instruction synthesizer. Nevertheless, the model pre-trained on the data generated by the instruction synthesizer shows improved performance on these unseen datasets, demonstrating the effectiveness of our method in enhancing model generalization.

In Table 2, we compare our pre-trained models with other open-source models. Using 100B tokens, our 500M model reaches the performance of Pythia-1B (Biderman et al., 2023) trained with 300B tokens and our 1.3 B model reaches the performance of BLOOM-3B (Workshop et al., 2022) trained with 341B tokens. This shows consistent data efficiency of *Instruction Pre-Training* across different model scales.

Instruction-Tuned Models Figure 4 shows the zero/few-shot performance on MMLU during instruction tuning from the pre-trained models. The model pre-trained via *Instruction Pre-Training* quickly outperforms the model pre-trained via *Vanilla Pre-Training*, and we observe a stable increasing trend of our model throughout the instruction tuning process. We infer that the closer alignment of training tasks during the instruction pre-training and instruction tuning stages facilitates a smoother transition between pre-training and fine-tuning. This alignment enables the model to learn more rapidly on downstream tasks. Therefore, *Instruction Pre-Training* offers a promising solution to significantly reduce the number of further fine-tuning steps (Longpre et al., 2023; Jiang et al., 2024c).

4.2 Domain-Adaptive Continual Pre-Training

Main Results As shown in Table 3, *Instruction Pre-Training* consistently outperforms *Vanilla Pre-Training* on almost all domain-specific

BioMed.	PubMedQA	ChemProt	RCT	MQP	UMSLE	AVERAGE
Llama3-70B	54.3	51.8	82.2	84.8	46.7	63.9
Llama3-8B	59.8	27.6	**73.6**	66.2	**40.6**	53.6
Vanilla PT-8B	65.1	42.4	72.4	76.4	35.5	58.4
Instruct PT-8B	**68.7**	**47.2**	73.4	**79.3**	38.0	**61.3**

Finance	ConvFinQA	Headline	FiQA SA	FPB	NER	AVERAGE
Llama3-70B	59.1	86.3	81.0	68.5	64.4	71.9
Llama3-8B	49.9	81.1	**83.3**	63.5	**72.8**	70.1
Vanilla PT-8B	62.9	84.7	82.2	65.4	64.9	72.0
Instruct PT-8B	**74.6**	**87.1**	82.4	**65.7**	63.6	**74.7**

Table 3: **Domain-specific task performance** of Llama3-8B without continued pre-training, after continued pre-training via *Vanilla Pre-Training* (Vanilla PT), and after continued pre-training via *Instruction Pre-Training* (Instruct PT). Both Vanilla PT and Instruct PT mix domain-specific corpora with general instructions to boost prompting ability, and use the same number of tokens for model training. The performance of Llama3-70B is displayed for reference.

	w/o Corpora	Rule-based	1-shot	Ours
Med.	58.6	58.8	58.5	**61.3**
Fin.	73.3	73.1	73.1	**74.7**

Table 4: **Ablations on training data.** *w/o Corpora* removes domain-specific pre-training corpora. *Rule-based* replaces instruction-augmented corpora with those created by the rule-based methods in Cheng et al. (2023). *1-shot* replaces instruction-augmented corpora with those created through single-turn synthesis. We report the average task scores within each domain.

tasks. Continual pre-training with *Instruction Pre-Training* significantly enhances the domain-specific performance of Llama3-8B, achieving parity with or even surpassing Llama3-70B. On the finance NER benchmark, where *Instruction Pre-Training* underperforms *Vanilla Pre-Training*, we observe considerable variance, where even Llama3-70B underperforms Llama3-8B, suggesting that this benchmark may not be reliable.

Ablations Table 4 presents ablation results for our pre-training data, which consist of a mixture of domain-specific instruction-augmented corpora and general instructions.

- *w/o Corpora*: Removing the domain-specific instruction-augmented corpora eliminates the source of domain-specific knowledge, leading to reduced domain-specific performance.
- *Rule-based*: Constructing instruction-augmented corpora using rule-based methods results in limited diversity, thereby constraining performance.
- *1-shot*: Limiting synthesis to 1-turn instead of multi-turn synthesis results in instruction-augmented corpora containing only 1-shot examples, leading to decreased prompting performance (Longpre et al., 2023).

5 Analysis

We conduct a detailed analysis of the instruction synthesizer and the instruction-augmented corpora to understand their impact on LM pre-training.

5.1 Instruction Synthesizer

Our goal in multitask fine-tuning is to develop a general synthesizer capable of generating instruction-response pairs for any raw text. Therefore, we evaluate its performance on both seen datasets (listed in Appendix A) and unseen datasets. The unseen datasets include SocialIQA (Sap et al., 2019), TextbookQA (Kembhavi et al., 2017), WikiWhy (Ho et al., 2022), and FEVER (Thorne et al., 2018), each representing a specific instruction format. Each example in these datasets comprises a context (raw text) and a set of context-based tasks (instruction-response pairs).

Response Accuracy Given a raw text and a task instruction, the instruction synthesizer generates a response. We compute the F1 similarity between the generated response and the gold response to evaluate response accuracy. Our instruction synthesizer is fine-tuned from the base Mistral-7B model. For comparison, we also present the results of the base model. As shown in Table 5, our fine-tuned synthesizer significantly outperforms

	Accuracy		Quality			
	Seen	Unseen	Seen		Unseen	
			Zero	Few	Zero	Few
Base	30.6	29.2	16.5	21.8	12.1	19.6
Ours	**70.0**	**55.2**	**49.4**	**49.9**	**25.3**	**30.8**

Table 5: **Response accuracy and instruction-response pair quality** of our instruction synthesizer (Ours) and Mistral-7B (Base). "Zero" indicates the zero-shot setting where no examples are presented before the testing raw text, and "Few" prepends 3-shot examples to the testing raw text.

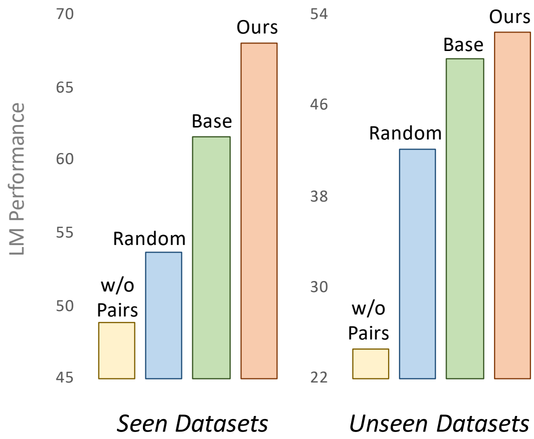

Figure 5: **Helpfulness on LM generalization** measured by LM performance with or without synthesized instruction-response pairs in the prompt.

the base model on both seen and unseen datasets, demonstrating the effectiveness of our fine-tuning.

Instruction-Response Pair Quality Given a raw text, the instruction synthesizer generates a set of instruction-response pairs. We compute the F1 similarity between the generated pairs and the gold pairs to evaluate their quality. The evaluation is conducted in both zero-shot and few-shot settings: 1) Zero-shot: the input to the instruction synthesizer contains only the raw text. 2) Few-shot: following Wang et al. (2023); Yehudai et al. (2024), a few examples from the same dataset as the gold instruction-response pairs, each consisting of a raw text and corresponding instruction-response pairs, are prepended to the testing raw text.

As shown in Table 5, compared to the base model, our fine-tuned synthesizer significantly outperforms the baseline across all four dimensions: zero-shot, few-shot, seen, and unseen datasets. In unseen datasets, the few-shot setting substantially outperforms the zero-shot setting, indicating that our synthesizer effectively leverages the pattern of the few-shot examples to create instruction-response pairs for the testing text.

Helpfulness on LM Generalization We conduct experiments using an LM (base Mistral-7B in our analysis) to assess the impact of synthesized instruction-response pairs on helping LMs generalize to unseen tasks. Given a prompt concatenating a testing raw text, synthesized pairs, and a testing instruction, the LM generates a response. We then compare the LM's performance on the testing task with and without the synthesized pairs in the prompt to evaluate their effectiveness.

We evaluate instruction-response pairs generated using different methods: 1) Random: randomly sampled instruction-response pairs of a different context. 2) Base: pairs synthesized based on the testing raw text by the base Mistral-7B model prompted with a few examples. 3) Ours: pairs synthesized based on the testing raw text by our instruction synthesizer using the same few-shot examples as Base.

As shown Figure 5, "w/o Pairs" denotes the setting where synthesized pairs are excluded from the prompt. On both seen and unseen datasets, ours consistently enhances the LM's performance on the testing task, surpassing all baselines. This demonstrates the effectiveness of our synthesized tasks in improving the LM's ability to perform a wide range of tasks.

5.2 Instruction-Augmented Corpora

We analyze the instruction-augmented pre-training corpora in terms of context relevance, response accuracy and task diversity. We sample 500 instruction-augmented texts from the augmented corpora and use GPT-4 (OpenAI, 2023) to evaluate the synthesized instruction-response pairs. Specifically, GPT-4 is prompted to assess whether the synthesized instruction is relevant to the context of the raw text (context relevance) and whether the response is accurate based on the instruction and context (response accuracy). Additionally, to evaluate task diversity, we prompt GPT-4 to categorize each instruction-response pair using a predefined list of task categories from Wang et al. (2022).

As shown in Table 6, our instruction synthesizer generates instruction-response pairs spanning 49 different task categories, with over 85% relevance to the context and 70% response accuracy. We further group the task categories into 9 general task scenarios. Figure 6 shows the percentages of

	Accuracy	Relevance	# Category
General	77.5	92.9	49
BioMed.	86.2	99.4	26
Finance	69.8	85.8	41

Table 6: **Response accuracy, context relevance, and number of task categories** of the instruction-augmented corpora.

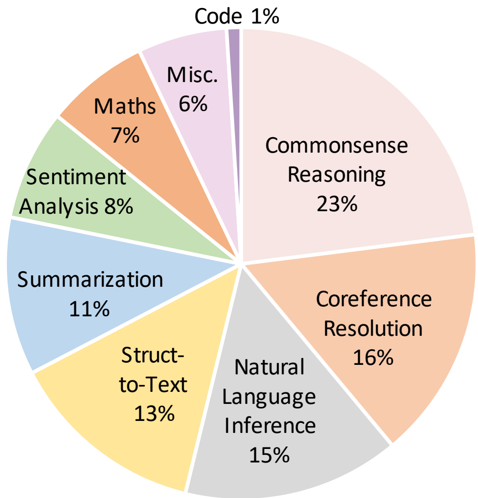

Figure 6: **Distribution of task scenarios of synthesized instruction-response pairs** in the instruction-augmented corpora.

each task scenario in the instruction augmented corpora for general pre-training. Our synthesized tasks cover all general task scenarios, demonstrating the effectiveness of our instruction synthesizer in generating a highly diverse tasks. We conduct further analysis in Appendix E for human evaluation, Appendix D for data contamination, and Appendix F for domain distribution and diversity.

6 Related Work

Synthetic Instruction Generation There have been many works studying synthetic instruction generation, but they mainly focus on post-training (Xu et al., 2023; Li et al., 2023a), while we focus on pre-training. This makes these works complementary to ours. Moreover, our experiments demonstrate that instruction pre-trained models gain more from instruction post-training, highlighting the complementary nature.

Regardless of the training stage, our method differs from related works in several ways. Firstly, we focus on learning from the raw corpora rather than distilling knowledge from strong models (Xu et al., 2023; Mukherjee et al., 2023; Li et al., 2024). Secondly, ours can be task-agnostic, in contrast to the more task-specific approaches (Wang et al., 2023; Honovich et al., 2023; Yehudai et al., 2024) relying on a few gold examples. Additionally, we outperforms rule-based methods (Cheng et al., 2023; Gu et al., 2022b) by increasing instruction diversity. Moreover, the iterative techniques used in Li et al. (2023a); Lee et al. (2024); Yue et al. (2024) could potentially complement our method, areas we plan to explore in future research.

Data Curation for LM Pre-Training Data curation for LM pre-training typically involves collection, cleaning, and organization. Most pre-training data are collected from the Internet to ensure diversity (Raffel et al., 2020; Penedo et al., 2023; Wenzek et al., 2020; Gao et al., 2020). Although diverse, web-scraped data often contain low-quality and duplicate content. Therefore, data cleaning techniques are applied to these corpora, including language identification (Joulin et al., 2016), perplexity-based (Wenzek et al., 2020), classifier-based (Brown et al., 2020), and rule-based (Raffel et al., 2020; Rae et al., 2021) filtering. Data organization aims at performing more fine-grained programming of the data, including data selection (Albalak et al., 2024; Xie et al., 2024) and constructing training instances related to downstream usage (Gu et al., 2022a, 2023; Shi et al., 2023; Jiang et al., 2024b; Maini et al., 2024). Our work explores an orthogonal direction: augmenting raw corpora with large-scale supervised signals.

7 Conclusion

This paper proposes *Instruction Pre-Training* to explore supervised multitask learning for pre-training. Instead of directly pre-training on raw corpora, *Instruction Pre-Training* augments the corpora with instruction-response pairs generated by an instruction synthesizer. Our instruction synthesizer, fine-tuned from a highly diverse data collection, is capable of generating diverse instruction-response pairs from various corpora. In pre-training from scratch, *Instruction Pre-Training* not only outperforms *Vanilla Pre-Training* on the pre-trained base models but also benefits more from further instruction tuning. In continual pre-training, *Instruction Pre-Training* substantially enhances the performance of Llama3-8B in two different domains. Looking ahead, we hope our work can inspire further exploration into this promising area of supervised multitask pre-training, effectively enhancing the general abilities of LMs.

Limitations

While synthetic data offer numerous benefits, it is crucial to acknowledge the potential limitations. Our work, along with other works utilizing synthetic data (Liu et al., 2024), is inevitably limited by the possibility of introducing hallucinations. As shown in our analysis in Section 5, the accuracy of our instruction-augmented corpora is approximately 70%, which may potentially mislead the pre-trained model. Future work could explore post-verification techniques such as those proposed by Li et al. (2023a); Lee et al. (2024); Yue et al. (2024); Yehudai et al. (2024) to filter out low-quality data or develop methods to enhance the reliability of the instruction synthesizer.

Furthermore, works like Touvron et al. (2023); Jiang et al. (2023) have achieved impressive performance by pre-training on trillions of tokens, whereas our pre-training is currently limited to the scale of billions of tokens. Future research should investigate scaling laws for synthetic data and determine the optimal balance between quantity and quality of synthetic samples (Liu et al., 2024).

Ethics Statement

Except for the pre-training corpora in the finance domain, all datasets and language models used in this work are publicly available.

Acknowledgements

This work was supported by the National Science Foundation for Distinguished Young Scholars (with No. 62125604) and the NSFC key project (with No. 61936010).

References

Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2022. Topiocqa: Open-domain conversational question answering with topic switching. *Transactions of the Association for Computational Linguistics*, 10:468–483.

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021. Explanations for commonsenseqa: New dataset and models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3050–3065.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. 2024. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Julio Cesar Salinas Alvarado, Karin Verspoor, and Timothy Baldwin. 2015. Domain adaption of named entity recognition to support credit risk assessment. In *Proceedings of the Australasian Language Technology Association Workshop 2015*, pages 84–90.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jon Ander Campos, Arantxa Otegi, Aitor Soroa, Jan Milan Deriu, Mark Cieliebak, and Eneko Agirre. 2020. Doqa-accessing domain-specific faqs via conversational qa. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7302–7314.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28:41–75.

Mayee Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. 2024. Skill-it! a data-driven skills framework for understanding and training language models. *Advances in Neural Information Processing Systems*, 36.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022. Convfinqa: Exploring the chain of numerical reasoning in conversational finance question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6279–6292.

Daixuan Cheng, Shaohan Huang, and Furu Wei. 2023. Adapting large language models via reading comprehension. In *The Twelfth International Conference on Learning Representations*.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

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.

Franck Dernoncourt and Ji Young Lee. 2017. Pubmed 200k rct: a dataset for sequential sentence classification in medical abstracts. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 308–313.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Pre-training to learn in context. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4849–4870.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022a. Ppt: Pre-trained prompt tuning for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8410–8423.

Yuxian Gu, Pei Ke, Xiaoyan Zhu, and Minlie Huang. 2022b. Learning instructions with unlabeled data for zero-shot cross-task generalization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1617–1634.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Matthew Ho, Aditya Sharma, Justin Chang, Michael Saxon, Sharon Levy, Yujie Lu, and William Yang Wang. 2022. Wikiwhy: Answering and explaining cause-and-effect questions. In *The Eleventh International Conference on Learning Representations*.

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.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401.

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*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024a. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Ting Jiang, Shaohan Huang, Shengyue Luo, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, et al. 2024b. Improving domain adaptation through extended-text reading comprehension. *arXiv preprint arXiv:2401.07284*.

Zhengbao Jiang, Zhiqing Sun, Weijia Shi, Pedro Rodriguez, Chunting Zhou, Graham Neubig, Xi Victoria Lin, Wen-tau Yih, and Srinivasan Iyer. 2024c. Instruction-tuned language models are better knowledge learners. *arXiv preprint arXiv:2402.12847*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*, pages 4999–5007.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8082–8090.

Tomáš Kočiskỳ, 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.

Jens Kringelum, Sonny Kim Kjaerulff, Søren Brunak, Ole Lund, Tudor I Oprea, and Olivier Taboureau. 2016. Chemprot-3.0: a global chemical biology diseases mapping. *Database*, 2016:bav123.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794.

Matthew Lamm, Jennimaria Palomaki, Chris Alberti, Daniel Andor, Eunsol Choi, Livio Baldini Soares, and Michael Collins. 2021. Qed: A framework and dataset for explanations in question answering. *Transactions of the Association for computational Linguistics*, 9:790–806.

Nicholas Lee, Thanakul Wattanawong, Sehoon Kim, Karttikeya Mangalam, Sheng Shen, Gopala Anumanchipali, Michael W Mahoney, Kurt Keutzer, and Amir Gholami. 2024. Llm2llm: Boosting llms with novel iterative data enhancement. *arXiv preprint arXiv:2403.15042*.

Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, Daniel Haziza, Luca Wehrstedt, Jeremy Reizenstein, and Grigory Sizov. 2022. xformers: A modular and hackable transformer modelling library. `https://github.com/facebookresearch/xformers`.

Haoran Li, Qingxiu Dong, Zhengyang Tang, Chaojun Wang, Xingxing Zhang, Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, et al. 2024. Synthetic data (almost) from scratch: Generalized instruction tuning for language models. *arXiv preprint arXiv:2402.13064*.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason E Weston, and Mike Lewis. 2023a. Self-alignment with instruction backtranslation. In *The Twelfth International Conference on Learning Representations*.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023b. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.

Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. Openorca: An open dataset of gpt augmented flan reasoning traces. `https://https://huggingface.co/Open-Orca/OpenOrca`.

Yichan Liang, Jianheng Li, and Jian Yin. 2019. A new multi-choice reading comprehension dataset for curriculum learning. In *Asian Conference on Machine Learning*, pages 742–757. PMLR.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 58–62.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2021. Logiqa: a challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3622–3628.

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, et al. 2024. Best practices and lessons learned on synthetic data for language models. *arXiv preprint arXiv:2404.07503*.

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. In *International Conference on Machine Learning*, pages 22631–22648. PMLR.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Www'18 open challenge: financial opinion mining and question answering. In *Companion proceedings of the the web conference 2018*, pages 1941–1942.

Pratyush Maini, Skyler Seto, He Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. 2024. Rephrasing the web: A recipe for compute and data-efficient language modeling. *arXiv preprint arXiv:2401.16380*.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796.

Clara H McCreery, Namit Katariya, Anitha Kannan, Manish Chablani, and Xavier Amatriain. 2020. Effective transfer learning for identifying similar questions: matching user questions to covid-19 faqs. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3458–3465.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. Metaicl: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809.

Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjamshidi. 2021. Spartqa: A textual question answering benchmark for spatial reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4582–4598.

Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. 2020. Covid-qa: A question answering dataset for covid-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 193–203.

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2023. Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. *ACM Computing Surveys*, 55(10):1–45.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020. Getting closer to ai complete question answering: A set of prerequisite real tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8722–8731.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, et al. 2021. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473.

Weijia Shi, Sewon Min, Maria Lomeli, Chunting Zhou, Margaret Li, Xi Victoria Lin, Noah A Smith, Luke Zettlemoyer, Wen-tau Yih, and Mike Lewis. 2023. In-context pretraining: Language modeling beyond document boundaries. In *The Twelfth International Conference on Learning Representations*.

Ankur Sinha and Tanmay Khandait. 2021. Impact of news on the commodity market: Dataset and results. In *Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Volume 2*, pages 589–601. Springer.

Amir Soleimani, Christof Monz, and Marcel Worring. 2021. Nlquad: A non-factoid long question answering data set. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1245–1255.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. Dream: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019. Quartz: An open-domain dataset of qualitative relationship questions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5941–5946.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819.

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

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200.

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.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. 2022. Supernaturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M

Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Édouard Grave. 2020. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2024. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36.

Wenhan Xiong, Jiawei Wu, Hong Wang, Vivek Kulkarni, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Tweetqa: A social media focused question answering dataset. *arXiv preprint arXiv:1907.06292*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2023. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Wu, Zheng Zhang, Toby Li, Nora Bradford, Branda Sun, et al. 2022. Fantastic questions and where to find them: Fairytaleqa–an authentic dataset for narrative comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 447–460.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. Fingpt: Open-source financial large language models. *FinLLM at IJCAI*.

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.

Asaf Yehudai, Boaz Carmeli, Yosi Mass, Ofir Arviv, Nathaniel Mills, Assaf Toledo, Eyal Shnarch, and Leshem Choshen. 2024. Genie: Achieving human parity in content-grounded datasets generation. *arXiv preprint arXiv:2401.14367*.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2019. Reclor: A reading comprehension dataset requiring logical reasoning. In *International Conference on Learning Representations*.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

A Data Collection for Fine-Tuning Instruction Synthesizer

Figure 7 displays our dataset collection for fine-tuning the instruction synthesizer. For each context in the datasets, we gather all the downstream tasks corresponding to the context, and regard the context as the raw text and the downstream tasks as the instruction-response pairs. For each dataset, we sample a maximum of 10K examples with the highest number of instruction-response pairs, to enhance task diversity while avoiding dataset predominance. Instruction-response pairs covers all the formats defined in (Longpre et al., 2023), including free-form completion, multiple-choice, free-form completion with chain-of-thought (CoT; Wei et al., 2022) and multiple-choice with CoT.

B Tuning and Inference Settings for Instruction Synthesizer

Data Format We fill each data example into a specifically designed template to explicitly separate different parts. This facilitates the direct extraction of instruction-response pairs after inference. We use the template `<CON> {text} </CON>` to wrap the raw text. As shown in Table 7, we design different templates for different formats of instructions, and `\n\n` is used to connect instruction-response pairs and link them with the raw text. Additionally, we use `<s>` before the beginning of each example and `</s>` after the end of each example. An N-shot example is made by directly concatenating N examples in a sequence. A case of a formatted 2-shot data example for fine-tuning is displayed in Table 13.

Tuning To constitute a few-shot example for fine-tuning, we concatenate as many formatted examples as possible from the same dataset to match the maximum sequence length. The tuning hyperparameters are in Table 8.

Inference During each round of inference, we concatenate the formatted examples from previous rounds with the formatted raw text of the current round as the input for the instruction synthesizer. Subsequently, the instruction synthesizer generates a sequence of instruction-response pairs. The maximum sequence length for inference corresponds to that of the target LM intended for pre-training. We use the vLLM (Kwon et al., 2023) framework for acceleration. It takes about 1 day to synthesize instruction-response pairs for 1B tokens of raw corpora on a single A100-80GB GPU.

Instruction Synthesizer Template

Free-form Completion

```
<QUE> {instruction} <ANS> {response} </END>
```

Multiple Choice

```
<QUE> {instruction}
Options:
- {option_1}
- {option_2} <ANS> {response} </END>
```

Free-form Completion with CoT

```
<QUE> {instruction}
Let's think step by step. <ANS> {CoT}
Therefore, the answer is {response} </END>
```

Multiple Choice with CoT

```
<QUE> {instruction}
Options:
- {option_1}
- {option_2}
Let's think step by step. <ANS> {CoT}
Therefore, the answer is {response} </END>
```

Table 7: **Templates for different formats of instruction-response pairs** for tuning and inference of the instruction synthesizer.

Hyper-Parameter	Assignment
Base model	Mistral-7B-v0.1
Computing infrastructure	4 A100-80GB GPUs
Run-time	2 days
Epochs	5
Batch size	16384 tokens
Max sequence length	4096
Max learning rate	5e-6
Optimizer	Adam
Adam beta weights	0.9, 0.95
Learning rate scheduler	cosine
Weight decay	0.1
Warm-up steps	1000
Gradient clipping	1.0
Dropout ratio	0.1

Table 8: **Hyper-parameters of fine-tuning the instruction synthesizer.**

C LM Evaluation

General Models We evaluate 0-shot performance on tasks originally formatted as language modeling, including WinoGrande (Sakaguchi et al., 2021), PIQA (Bisk et al., 2020)

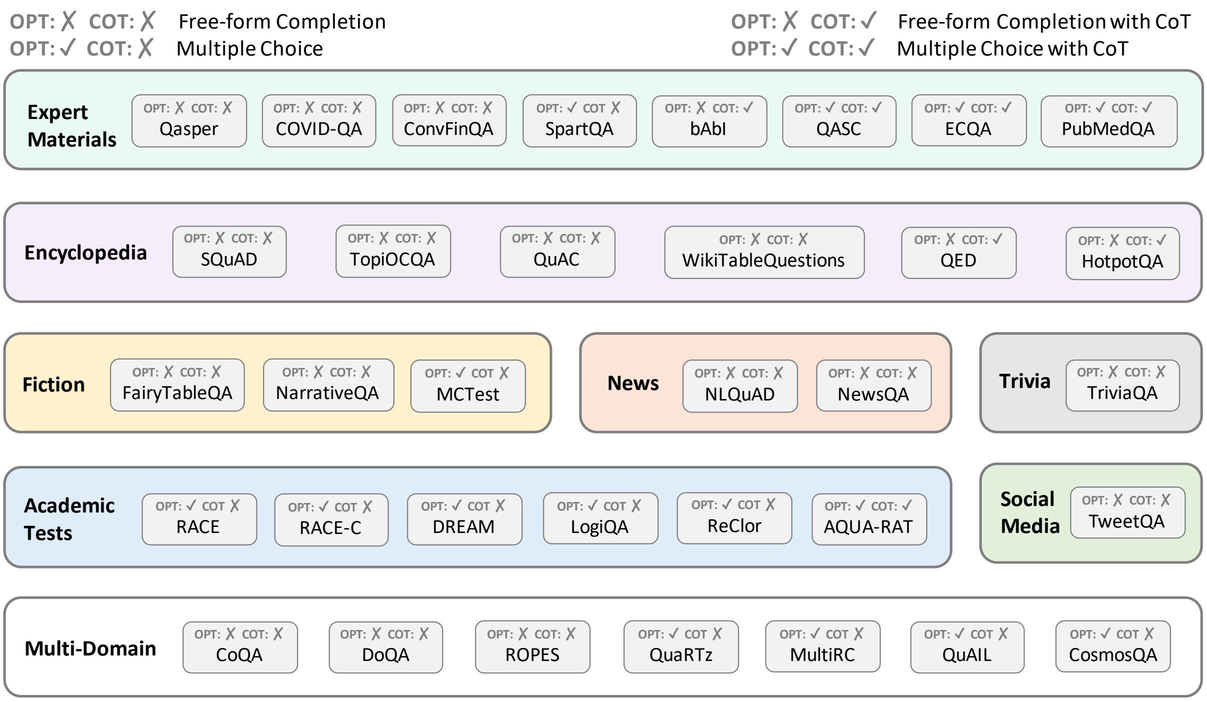

Figure 7: **Datasets for fine-tuning the instruction synthesizer**, including Dasigi et al. (2021); Möller et al. (2020); Chen et al. (2022); Mirzaee et al. (2021); Weston et al. (2015); Khot et al. (2020); Aggarwal et al. (2021); Jin et al. (2019) in the expert materials domain, Rajpurkar et al. (2016); Adlakha et al. (2022); Choi et al. (2018); Pasupat and Liang (2015); Lamm et al. (2021); Yang et al. (2018) in the encyclopedia domain, Xu et al. (2022); Kočiskỳ et al. (2018); Richardson et al. (2013) in the fiction domain, Soleimani et al. (2021); Trischler et al. (2017) in the news domain, Joshi et al. (2017) in the trivia domain, Lai et al. (2017); Liang et al. (2019); Sun et al. (2019); Liu et al. (2021); Yu et al. (2019); Ling et al. (2017) in the academic tests domain, Xiong et al. (2019) in the social media domain, and Reddy et al. (2019); Campos et al. (2020); Lin et al. (2019); Tafjord et al. (2019); Khashabi et al. (2018); Rogers et al. (2020); Huang et al. (2019) in the multi-domains sources domain.

and HellaSwag (Zellers et al., 2019), and 5-shot performance on tasks that are rather challenging and formatted as question-answering, including ARC (Clark et al., 2018), BoolQ (Clark et al., 2019), SIQA (Sap et al., 2019), OBQA (Mihaylov et al., 2018), and MMLU (Hendrycks et al., 2020). Using the lm-evaluation-harness framework, we report the acc-norm score to follow Brown et al. (2020).

Domain-Specific Models We follow the prompting settings of AdaptLLM (Cheng et al., 2023): for biomedicine domain, we evaluate zero-shot performance on PubMedQA (Jin et al., 2019) and USMLE (Jin et al., 2021), few-shot performance on ChemProt (Kringelum et al., 2016), MQP (McCreery et al., 2020) and RCT (Dernoncourt and Lee, 2017); for finance domain, we evaluate zero-shot performance on ConvFinQA (Chen et al., 2022) and few-shot performance on FPB (Malo et al., 2014), FiQA SA (Maia et al., 2018), Headline (Sinha and Khandait, 2021), and NER (Alvarado et al., 2015).

D Data Contamination Analysis

We measure cross-contamination between the evaluation datasets and the training data using the sub-string match method described in OpenAI (2023): an evaluated example is considered contaminated if a sub-string of it appears in the training data. Table 9 shows:

- *Total Eval Examples*: The number of all evaluated examples in each dataset.
- *Contam in Raw Corpora*: The number of contaminated examples in the raw corpora.
- *Contam in Ins-Aug Corpora*: The number of contaminated examples in the instruction-augmented corpora, which includes the raw corpora and the synthesized instruction-response pairs.
- *Contam in Synthesized Pairs*: The number of contaminated examples introduced by synthesized pairs, calculated by subtracting the number of contaminated examples in the raw corpora from those in the instruction-augmented corpora.

The results indicate the synthesized pairs introduce minimal contamination to the training data.

	ARC-e/c	BoolQ	SIQA	WG	PIQA	OBQA	HS	MMLU
Total Eval Examples	2376/1172	3270	1954	1267	1838	500	10042	14042
Contam in Raw Corpora	5/3	144	0	0	3	0	4	20
Contam in Ins-Aug Corpora	5/4	144	0	0	3	0	4	22
Contam in Synthesized Pairs	0/1	0	0	0	0	0	0	2

Table 9: **Data contamination analysis** of raw corpora, instruction-augmented corpora and the synthesized instruction-response pairs. "WG" and "HS" represent WinoGrande and HellaSwag, respectively.

	Accuracy	Relevance	# Category
General	75.5	87.5	51
BioMed.	81.0	97.0	21
Finance	73.5	88.0	39

Table 10: **Human evaluation of response accuracy, context relevance, and number of task categories** on the instruction-augmented corpora.

Coverage	Coverage (multi-domain)	Overlap
86.8	77.8	84.9

Table 11: **Domain coverage and overlap** between the raw text and the synthesized instruction-response pairs.

E Human Evaluation on Instruction-Augmented Corpora

We conduct human evaluation to analyze the instruction-augmented corpora from the following aspects:

- *Response Accuracy*: A binary score indicating whether the response is accurate based on the instruction and context, where 1 means accurate and 0 means inaccurate. We report the average score of all responses.
- *Context Relevance*: A binary score indicating whether the instruction-response pair is relevant to the context of the raw text, where 1 means relevant and 0 means irrelevant. We report the average score of all instruction-response pairs.
- *# Task Category*: The evaluator categorizes each instruction-response pair using a predefined list of task categories from Wang et al. (2022). We report the number of different categories of all the instruction-response pairs to show diversity.

From the results in Table 10, the synthesized instruction-response pairs span 51 different task categories, with over 85% relevance to the context and 70% response accuracy.

F Domain Distribution Analysis

We analyze domain distribution to evaluate the effectiveness of our instruction synthesizer in generating instruction-response pairs closely aligned with the domain of the given raw context.

Domain Coverage and Overlap For each instruction-augmented text, we calculate the following scores and report the average on all the instruction-augmented texts:

- *Domain Coverage*: The ratio of text domains included in the instruction domains to all text domains.
- *Domain Coverage (multi-domain only)*: We specifically compute domain coverage for the cases where a raw text contains multiple domains.
- *Domain Overlap*: The overlap of raw text domains and instruction domains divided by the union of raw text and instruction domains.

As shown in Table 11, the synthesized instruction-response pairs cover most of the domains in the raw text, with a high domain overlap with the raw text. For the texts containing more than one domain, our instruction synthesizer generates, on average, 5 instruction-response pairs per raw text, with each pair potentially covering a different domain. According to the *domain coverage (multi-domain only)*, when a single raw text includes multiple domains, our instruction synthesizer can effectively generate instruction-response pairs that cover most of the text domains.

Domain Distribution We analyze domain distributions of the following sources:

- Fine-tuning data for the instruction synthesizer.
- Raw pre-training corpora (Penedo et al., 2023).
- Synthesized instruction-response pairs based on the raw pre-training corpora.

As shown in Table 12, despite the domain distributions of fine-tuning data and raw corpora being very different, the synthesized pairs closely follow the domain distribution of the raw corpora.

	Encyclo	Fiction	Academic	Trivia	News	Expert	Social	Code
Fine-tune Data	22.2	11.1	22.2	3.7	7.4	29.6	3.7	0.0
Raw Corprora	5.8	9.6	3.5	0.0	20.3	42.8	14.7	3.3
Synthesized Pairs	5.8	11.8	3.3	0.1	18.8	46.0	11.1	3.1

Table 12: **Domain distribution** of fine-tuning data for the instruction synthesizer, raw corpora and synthesized instruction-response pairs. “Encyclo", “Academic", “Expert" and “Social" represent Encyclopedia, Academic Tests, Expert Materials and social media domains, respectively.

<s> <CON> Our school life is very interesting! My friends and I study hard at school. And we are good at our lessons. We are very happy. We have lots of time for our hobbies. My classmates all want to go to different clubs. Helen wants to join the Reading Club. She loves reading books. The Reading Club meets every Wednesday at three thirty. Lily enjoys dancing. She wants to join the Dancing Club. It meets on Mondays at four thirty. There's also an Art Club. It meets on Fridays at four o'clock. Nick doesn't want to join the Art Club. He doesn't like drawing. He thinks it is too difficult for him . Nick likes playing computer games. He wants to join the Computer Club. It meets every Thursday at three forty-five. Mike loves sports. He wants to join the football team. They play football every Monday at three thirty. I want to join the Music Club. I like listening to music with my friends. The Music Club meets on Tuesday at three fifteen. </CON>

<QUE> What club does Helen like? <ANS> Helen likes the reading club. </END>

<QUE> How many friends does the story teller describe? <ANS> I have four friends. </END>

<QUE> Are you and your friends smart? <ANS> unknown </END> </s><s> <CON> Billy and Sara are brother and sister. They went to the beach with their family last July for a week, and had the best time ever! On Monday, Billy and Sara wanted to build a giant sandcastle. They invited their new friends Jack and Jane to help build the sandcastle. Jack and Jane had a house on the beach, so they were really good when it came to building sandcastles. They hoped that they could make the sandcastle taller than themselves, but they soon found they needed more help. They asked their cousin Joey to help them build the biggest sandcastle in the world! Joey wasn't the friendliest cousin in the world, but to Billy and Sara's surprise, Joey was happy to help build the sandcastle. Billy, Sara, Jake, Jane and Joey had spent the whole day building the sandcastle, and finally, right before dinner time, they completed it. The sandcastle was huge! It had a river around the castle, and even a bridge to cross the river. It even had a flag at the top, and a wall that went around the castle too! They were so happy!

The rest of the week at the beach was a lot of fun for Billy and Sara. On Tuesday, they went for ice cream. Sara's ice cream fell and dripped all the way down to her tummy, but Billy gave her some of his. On Wednesday, they watched the fireworks at night. On Thursday, they went swimming all day long, moving like worms in the water. On Friday, they had to go back home. They were sad, so they started counting down the days until next year at the beach! </CON>

<QUE> how do billy and Sara know each other? <ANS> Billy and Sara are brother and sister. </END>

<QUE> Did they do something yesterday? <ANS> no. </END>

<QUE> When did they do something? <ANS> last July </END>

<QUE> What did they do? <ANS> They went to the beach </END> </s>

Table 13: **An example of a sequence for fine-tuning the instruction synthesizer.** This sequence contains two examples, both from the CoQA dataset (Reddy et al., 2019), constituting a 2-shot example.

Hyper-Parameter	Pre-Train from Scratch		Continual Pre-Train
Parameters	500M	1.3B	8B
Hidden size	1536	2048	4096
Intermediate size	4320	8192	14336
Max Position Embeddings	2048	2048	8192
Num attention heads	24	32	32
Num hidden layers	16	20	32
Num key value heads	24	8	8
Rope theta	10000	10000	500000
Vocab Size	32000	32000	128256
Tokenizer	Mistral	Mistral	Llama3
Computing infrastructure	8 A100-80GB GPUs	8 A100-80GB GPUs	4 A100-80GB GPUs
Run-time	5 days	10 days	1 day
Train steps	200K	100K	4K
Batch size	0.5M tokens	1M tokens	0.25M tokens
Max Sequence Length	2048	2048	4096
Max Learning Rate	3e-4	2e-4	1e-5
Optimizer	Adam	Adam	Adam
Adam beta weights	0.9, 0.95	0.9, 0.95	0.9, 0.95
Learning rate scheduler	cosine	cosine	cosine
Weight decay	0.1	0.1	0.1
Warm-up steps	2000	2000	1000
Gradient clipping	1	1	1
Dropout ratio	0.1	0.1	0.1

Table 14: **Hyper-parameters of pre-training from scratch and continual pre-training.**

	# Param.	# Token	ARC-e/c	BoolQ	SIQA	WG	PIQA	OBQA	HS	MMLU
Instruct PT	500M	100B	54.8/27.4	62.0	47.2	54.8	69.9	30.8	47.3	25.3
GPT-2	774M	-	53.8/24.9	62.1	45.5	54.5	69.3	30.6	45.3	25.5
Pythia	1B	300B	59.0/28.8	61.6	46.3	52.6	69.3	32.6	47.2	26.1
BLOOM	1.1B	341B	52.3/28.3	61.5	45.9	52.7	67.2	28.6	43.0	26.6
Instruct PT	1.3B	100B	60.5/30.9	62.2	49.2	55.9	73.6	33.4	54.3	27.3
OPT	1.3B	300B	60.1/31.1	62.4	48.4	58.2	71.0	34.0	53.8	25.1
GPT-2	1.5B	-	60.2/29.6	63.5	47.3	56.2	70.5	33.2	50.8	26.3
BLOOM	3B	341B	63.1/35.3	62.2	48.8	57.4	70.5	33.0	54.6	25.9

Table 15: **Comparison between our pre-trained models and other open-source models** (Radford et al., 2019; Biderman et al., 2023; Workshop et al., 2022; Zhang et al., 2022) on general benchmarks. "WG" and "HS" represent WinoGrande and HellaSwag, respectively.

Not a writer, a writer wannabe, editor, lit maj, or pretend literary critic. Just an avid reader/listener. My ratings are opinion only.
I love all genres of books. However, when I listen to audio books as I clean, garden, drive they are better with a lot of heat!
"Laborious"
This might have been a bit more tolerable if narrator was better. I am happy to say that I did finish the book but it just seemed to go and on. Like other listeners the book itself reminded me of a bad TV show. Not horrible but of all the books I have listened to this is just bearly average.

Problem: Pick your answer from:
a). They didn't like the genre.;
b). They did n't have enough time to read it.;
c). They did n't like the author.;
d). They did n't like the narrator.;
Q: What may be the reason for them not finishing the book?

Answer: d).

Customer Web Interaction: Fundamentals and Decision Tree From Virtual Communities
Authors
Enrico Senger, Sandra Gronover, and Gerold Riempp, University of St. Gallen
Abstract
In order to utilise the new possibilities of Internet technology efficiently, many companies invest considerable sums in the development of communication channels to customers. In this context, the often-quoted objective of cost saving per interaction appears to be questionable, since new communication media have not been able to fully substitute the existing systems. Costs are therefore more likely to rise than drop. The following article discusses potentials, criteria, conditions and consequences related to the use of computer-mediated environments for customer interaction. The objective is to derive recommendations for action in respect of a context-dependent support, especially by means of web collaboration and self-service-options.
Download Customer Web Interaction: Fundamentals and Decision Tree

Problem: Pick your answer from:
a). It can be edited.;
b). It can be read offline.;
c). It can be read online.;
d). It can be used offline.;
Q: What may happen after the download?

Answer: c).

Table 16: **A case of a 2-shot example in the general instruction-augmented corpora.**

Read this article and answer questions
Correlation between increased airway responsiveness and severity of pulmonary edema.
To determine whether the severity of the pulmonary edema in sheep models of cardiogenic and non-cardiogenic pulmonary edema correlate with concomitant alterations in airway responsiveness using three separate measures of pulmonary edema: post-mortem wet-to-dry lung weight ratio (W/D), chest radiograph (CXR) scores, and small airway wall area. Cardiogenic pulmonary edema was induced by increasing left atrial pressure (increase PLA) and non-cardiogenic pulmonary edema was induced by intravenous administration of Perilla ketone (PK). **(...)**

Does increased airway responsiveness correlate with pulmonary edema severity in sheep?
–
Let's think first: Increased airway responsiveness correlates with severity of pulmonary edema in sheep... So the answer is [Yes]

Read this article and answer questions
Immobilization and bioactivity evaluation of FGF-1 and FGF-2 on powdered silicon-doped hydroxyapatite and their scaffolds for bone tissue engineering.
Fibroblast growth factors (FGFs) are polypeptides that control the proliferation and differentiation of various cell types including osteoblasts. FGFs are also strong inducers of angiogenesis, necessary to obtain oxygen and nutrients during tissue repair. **(...)**

Do immobilization of fibroblast growth factors 1 and 2 on silicon-doped hydroxyapatite scaffolds for bone repair?
–
Let's think first: This study demonstrated the efficient immobilization of FGF-1 and FGF-2 on Si-HA and Si-HA scaffolds, retaining their biological activity on osteoblasts. Thus, these FGF/scaffolds may be useful in bone tissue engineering applications... So the answer is [Yes]

Read this article and answer questions
Phytotoxicity, cytotoxicity and genotoxicity evaluation of organic and inorganic pollutants rich tannery wastewater from a Common Effluent Treatment Plant (CETP) in Unnao district, India using Vigna radiata and Allium cepa.
The leather industry is a major source of environmental pollution in India. The wastewater generated by leather industries contains very high pollution parameters due to the presence of a complex mixture of organic and inorganic pollutants even after the treatment at a Common Effluent Treatment Plant (CETP) and disturbs the ecological flora and fauna. The nature, characteristics and toxicity of CETP treated wastewater is yet to be fully elucidated. Thus, this study aims to characterize and evaluate the toxicity of CETP treated tannery wastewater collected from the Unnao district of Uttar Pradesh, India. In addition to measuring the physico-chemical parameters, the residual organic pollutants was identified by GC-MS analysis and phytotoxicity, cytotoxicity and genotoxicity of the treated wastewater was evaluated using Vigna radiata L. and Allium cepa L. **(...)**

Is common effluent treatment plant wastewater safe for the environment?
–
Let's think first: The present study revealed the presence of high levels of various pollutants in CETP treated tannery wastewater. Moreover, the toxicity assessment showed the phytotoxic and genotoxic nature of the wastewater which suggests that this wastewater cannot be directly discharged into the environment without any further treatment... So the answer is [No]

Table 17: **A case of a 3-shot example in the instruction-augmented corpora for biomedicine domain.** Certain portions are omitted for brevity and are represented as **(...)**

Answer questions based on this article:
Once the MOASS is truly over would anyone like an AMA with DFV AND RC? I would love to learn what went on through their minds and the events all the way from 2019 to post-MOASS.
They must be dying to talk about all the things that went on (but couldn't because of all the potential controversy and lawsuits that can be had) and apes would love to get the official explanation on the cryptic, and some not so cryptic tweets from DFV and RC. Edit: it may be obvious but it's just an opinion of mine on to see what they may have to say. If it does somehow gain enough traction, we would respectfully ask them if they're interested. If not, no AMA. Simple as that. I've been thinking what we should do is once the squeeze is over let it die down a bit and then we should start a gmecon or something similar. I wanted to right a post about it but my karma is too low so if someone else wants to put it out there and see what people think that would be great. Personally I'm in this stock for life and would love an annual event where we could all meet up and have in person Q and A's with RC, DFV and others, even someone like Jordan Belfort to hype up the apes after we take our tendies. also would be good to see all gamestops ideas for the future. Just a thought hope there's some way we could make this happen.

question below:
What might happen after the MOASS?
answer below:
People will want an AMA with DFV and RC

question below:
What might happen if they did an AMA with DFV and RC?
answer below:
They would ask questions about the cryptic tweets

Answer questions based on this article:
Pixar's 'Lightyear' snares $51 million in domestic opening
Pixar's "Lightyear" rocketed to a $51 million domestic opening, the best performance of an animated feature since the pandemic began. Internationally, the Disney film tallied $34.6 million in ticket sales, bringing its global haul to $85.6 million. The animated film's performance, while strong for a pandemic release, fell short of expectations. Box office analysts had foreseen "Lightyear" bringing in between $70 million and $85 million domestically. Expectations were high because the last two films in the Toy Story franchise both opened to more than $100 million in ticket sales, according to data from Comscore. "Toy Story 4" in 2019 topped $120 million in its domestic debut and "Toy Story 3" generated more than $110 million during its opening 2010. "'Lightyear' had a great deal of potential on paper, but a number of factors resulted in this very rare box office misfire for a Pixar release," said Shawn Robbins, chief media analyst at BoxOffice.com. It's unclear if tough box office competition with Universal's "Jurassic World: Dominion," which generated $58.6 million over the weekend, and Paramount and Skydance's "Top Gun: Maverick," which secured another $44 million, was the reason for "Lightyear's" smaller-than-expected opening or if consumers were confused about the film release. After all, there has not been a theatrical release of a Pixar film since 2020's "Onward." **(...)**

question below:
What is the main point of the article?
answer below:
Lightyear fell short of expectations

question below: What is the author's opinion of why the movie had a smaller than expected opening?
answer below:
It had tough box office competition

Table 18: **A case of a 2-shot example in the instruction-augmented corpora for finance domain.** Certain portions are omitted for brevity and are represented as **(...)**