SciLitLLM: How to Adapt LLMs for Scientific Literature Understanding

Sihang Li*†
sihang0520@gmail.com
University of Science and Technology of China
China

Jin Huang*†
huangjin@dp.tech
DP Technology
China

Jiaxi Zhuang†
zhuangjiaxi@dp.tech
DP Technology
China

Yaorui Shi†
shiyaorui@dp.tech
University of Science and Technology of China
China

Xiaochen Cai
caixiaochen@dp.tech
DP Technology
China

Mingjun Xu†
xumj@dp.tech
DP Technology
China

Xiang Wang‡
xiangwang1223@gmail.com
University of Science and Technology of China
China

Linfeng Zhang
zhanglf@dp.tech
DP Technology
China

Guolin Ke
kegl@dp.tech
DP Technology
China

Hengxing Cai‡
caihengxing@dp.tech
DP Technology
China

Abstract

Scientific literature understanding is crucial for extracting targeted information and garnering insights, thereby significantly advancing scientific discovery. Despite the remarkable success of Large Language Models (LLMs), they face challenges in scientific literature understanding, primarily due to (1) a lack of scientific knowledge and (2) unfamiliarity with specialized scientific tasks.

To develop an LLM specialized in scientific literature understanding, we propose a hybrid strategy that integrates continual pre-training (CPT) and supervised fine-tuning (SFT), to simultaneously infuse scientific domain knowledge and enhance instruction-following capabilities for domain-specific tasks. In this process, we identify two key challenges: (1) constructing high-quality CPT corpora, and (2) generating diverse SFT instructions. We address these challenges through a meticulous pipeline, including PDF text extraction, parsing content error correction, quality filtering, and synthetic instruction creation. Applying this strategy, we present a suite of LLMs: **SciLitLLM**, specialized in scientific literature understanding. These models demonstrate promising performance on scientific literature understanding benchmarks. Specifically, the 7B model shows an average performance improvement of 3.6% on SciAssess and 10.1% on SciRIFF compared to leading LLMs with fewer than 15B parameters. Additionally, the 72B model, trained using QLoRA, achieves state-of-the-art performance among widely adopted open-source models.

Our contributions are threefold: (1) We present an effective framework that integrates CPT and SFT to adapt LLMs to scientific literature understanding, which can also be easily adapted to other domains. (2) We propose an LLM-based synthesis method to generate diverse and high-quality scientific instructions, resulting in a new instruction set – **SciLitIns** – for supervised fine-tuning in less-represented scientific domains. (3) SciLitLLM achieves promising performance improvements on scientific literature understanding benchmarks. Our model is available in anonymous cloud drive[1].

*Equal contribution.
†Work done while these authors interned at DP Technology.
‡Corresponding.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

CCS Concepts

• **Computing Methodologies**; • **Artificial Intelligence**; • **Natural Language Processing**; • **Natural Language Generation**;

Keywords

Large Language Model, Pre-training, Supervised Fine-tuning, Scientific Literature Understanding

[1] https://osf.io/a7mtc/?view_only=cf934e65ab0443fbb9f83a0e26bf97b3

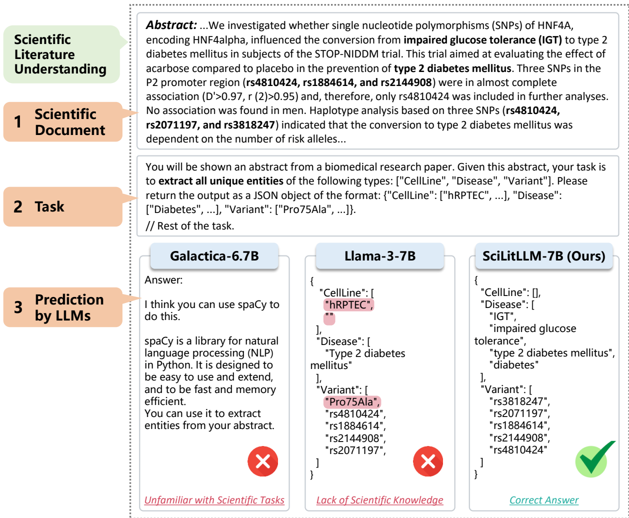

Figure 1: An example of scientific literature understanding in SciRIFF involves extracting accurate entities from a biomedicine paper. SciLitLLM-7B demonstrates sufficient scientific knowledge and instruction-following ability to accurately identify and extract these entities.

1 Introduction

Scientific literature understanding involves the systematic evaluation and interpretation of scientific texts and publications, to identify trends, extract targeted information, and garner insights [3, 64], significantly contributing to scientific discovery. Concurrently, Large Language Models (LLMs) [7, 38, 49, 56] have achieved remarkable success in natural language processing, prompting the development of domain-specific LLMs across various fields [12, 14, 54]. However, recent studies [8, 44, 50] indicate that LLMs face challenges when specializing in scientific literature understanding, particularly in context understanding and question answering. Take Figure 1 as an example, where the LLM is asked to understand the content of a biomedical research paper and then extract the targeted information. LLMs' potential might be hindered by two major barriers: (1) a lack of *scientific knowledge*, which results in errors such as the missing important entities in Llama-3-7B [4], and (2) unfamiliarity with *scientific tasks*, leading to the inability of Galactica-6.7B [47] to follow task instructions accurately.

To make LLMs specialized in science-relevant tasks, existing studies mostly adopt two strategies, as illustrated in Figure 2: (1) Fine-tuning with scientific instructions [27, 45, 50, 62]. A general-purpose LLM is fine-tuned with collected domain-specific instructions to adapt it to science-relevant tasks. However, instruction fine-tuning alone is insufficient to imbue the models with comprehensive scientific knowledge. (2) Pre-training on scientific corpora [6, 47, 60]. This approach involves training models on vast scientific corpora. While this method equips LLMs with domain knowledge, the lack of instruction-tuning confines them to solving relevant tasks. Moreover, it is hampered by substantial computational costs and data requirements [36, 57]. To address these

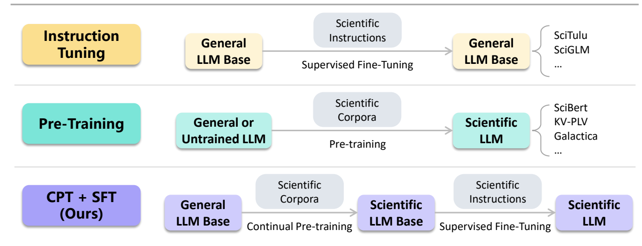

Figure 2: Comparison of strategies to adapt LLMs to scientific tasks. Previous approaches typically either fine-tune a general LLM with scientific instructions or pre-train an LLM on extensive scientific corpora. We propose a combined method of both CPT and SFT.

obstacles while balancing efficiency, we propose a hybrid strategy that incorporates continual pre-training (CPT) and supervised fine-tuning (SFT), to simultaneously infuse domain knowledge and enhance domain-specific instruction-following capabilities.

However, as illustrated in Figure 3, developing a scientific literature understanding model using this CPT and SFT pipeline presents two critical requirements:

- **High-quality CPT Corpora.** Scientific corpora, predominantly in PDF format such as textbooks and research papers, are not directly digestible for LLM training. Converting these documents to text using tools like PyPDF2[2] often introduces formatting and syntax errors, degrading corpus quality. Worse still, scientific documents often contain segments that contribute little information (*e.g.,* references), necessitating quality control to filter them out. See the first row in Figure 3 for a comparison of high- and low-quality CPT texts.
- **Diverse Scientific Instructions.** Effective instruction following for scientific literature understanding requires a large, high-quality, and diverse set of task-related instructions. However, to the best of our knowledge, there is a scarcity of well-designed instruction datasets for scientific literature understanding, and hiring human annotators to curate such a dataset from scratch is prohibitively expensive [18, 40]. See the second row in Figure 3 for an illustration of high- and low-quality instructions.

To address these challenges, we devise an effective pipeline to construct high-quality domain corpora for CPT and diverse scientific instructions for SFT, as illustrated in Figure 4:

- In the **CPT** stage for domain knowledge injection, we start with an extensive in-house corpus consisting of 73k textbooks and 625k academic papers in the scientific field, all in PDF format. Initially, we leverage PyPDF2, a widely used open-source PDF parsing tool, to extract raw texts from these documents. We then employ a moderate yet powerful model, Llama3-7B-Instruct [4], to correct the format and spelling errors introduced by PDF parsing (*cf.* Section 3.1.1). Subsequently, we train a small text quality classifier to score the corpus and filter out texts of low educational value[3] in the scientific field (*cf.* Section 3.1.2). These

[2]https://pypdf2.readthedocs.io
[3]Phi models [1, 22, 35] propose to determine the quality of a pre-training text by its educational value for a student whose goal is to learn basic domain concepts.

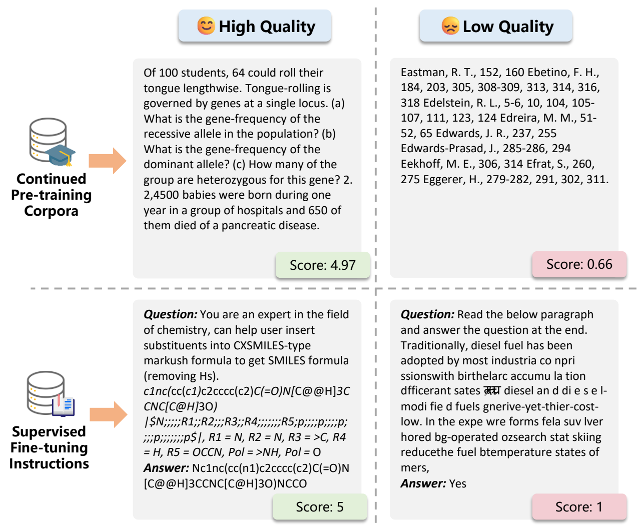

Figure 3: Examples of high and low-quality CPT text and SFT instructions. Scores are labeled by the CPT and SFT quality filter (*cf.* Section 3.1.2 & 3.2.2), ranging from 0 to 5. Higher scores indicate better quality.

two simple yet effective textual refinement and quality control measures ensure the high quality of our CPT corpus, culminating 12.7 billion tokens for CPT via the Qwen2 [56] tokenizer.
- In the **SFT** stage for domain instruction fine-tuning, to overcome the scarcity of domain-specific instructions and the high cost of human annotations, we propose a novel instruction synthesis method (*cf.* Section 3.2.1). It enables us to generate diverse instructions to better equip the model for domain-specific tasks. Moreover, we sequentially apply instruction duplication based on Levenshtein distance and an LLM-based filtering method to ensure the quality of synthetic instructions (*cf.* Section 3.2.2).

Having established such high-quality datasets, we apply the CPT-SFT integration strategy on a general-purpose LLM – Qwen2 [56] and obtain SciLitLLM in two scales: a 7B (full parameter training) and a 4-bit quantized 72B (QLoRA [15] parameter-efficient training) model. Evaluations on benchmarks of scientific literature understanding demonstrate the effectiveness of our strategy. We observe promising performance enhancements, with an average improvement of 3.6% on SciAssess [8] and 10.1% on SciRIFF [50], compared to the leading LLMs under 15B parameters. Notably, SciLitLLM-7B even outperforms Llama3 and Qwen2 with 70B parameters on SciRIFF. Additionally, SciLitLLM-72B, with QLoRA training, achieves leading results on both benchmarks, surpassing other open-source LLMs. Further ablation studies demonstrate the effectiveness of each module in our pipeline.

In summary, our contributions are threefold:

- We devise an effective and comprehensive pipeline to adapt general LLMs to a specific domain – scientific literature understanding. It combines continual pre-training (CPT) and supervised fine-tuning (SFT), to enhance scientific knowledge base and instruction-following capabilities for specialized domain tasks.
- We propose a novel domain instruction synthesis method to curate instructions for scientific literature understanding, resulting in a new dataset – **SciLitIns** .
- **SciLitLLM**, trained through the proposed pipeline, outperforms leading open-source LLMs on scientific literature understanding. It has been deployed in real-world application scenarios and has demonstrated promising performance. Our models are available anonymously[4]. Code and data will be released after administrative procedure.

2 Related Works

In this section, we provide a review of literature related to continual pre-training, supervised fine-tuning in the scientific domain, and LLMs for scientific literature understanding.

2.1 Knowledge Injection via Continual Pre-training

Pre-training a language model is usually conducted on a large corpus of textual data to learn the statistical properties of language [7, 42]. Formally, given a sequence of textual tokens $x = (x_1, x_2, ..., x_T)$, an LLM parameterized by θ performs autoregressive language modeling (ALM) by predicting the next token in the sequence given the previous tokens:

$$\mathcal{L}_{\text{ALM}}(\theta) = \sum_{t=1}^{T} \log P_\theta (x_t \mid x_{<t}) . \tag{1}$$

To further inject domain knowledge into a general LLM after pre-training, researchers engage in continual pre-training (CPT) on high-quality domain-specific corpora [30, 46], sometimes combined with general corpora. This process enhances the model's fundamental understanding abilities in specific downstream domains while mitigating catastrophic forgetting of general knowledge [31, 37, 55]. See the comprehensive study [23] for different warm-up strategies for CPT. Additionally, the CPT corpora can be augmented by transforming them into an instruction-response format [9, 10]. Furthermore, the scaling law [25] of domain-specific CPT [41] is explored to determine the optimal mix of data. However, these studies primarily focus on training dynamics and data recipes, leaving the pre-processing for scientific data, especially raw PDF files, largely unexplored. Exhibiting such steps is essential for generating high-quality domain corpora and effectively injecting domain knowledge, representing a significant challenge for practitioners.

2.2 Domain Adaptation via Supervised Fine-tuning

Supervised fine-tuning (SFT) modifies a pre-trained language model to follow specific instructions or perform designated tasks by fine-tuning it on a targeted, task-specific dataset [43, 51]. Let $\mathcal{D}_{\text{fine}} = \{(x_i, y_i)\}_{i=1}^{M}$ represent the fine-tuning dataset, where x_i is an input sequence (instruction) and y_i is the corresponding output sequence. The fine-tuning objective can be expressed as:

$$\mathcal{L}_{\text{fine-tune}}(\phi) = \sum_{i=1}^{M} \log P_\phi (y_i \mid x_i) . \tag{2}$$

[4] https://osf.io/a7mtc/?view_only=cf934e65ab0443fbb9f83a0e26bf97b3

During fine-tuning, the model adjusts its parameters ϕ to better fit the task-specific data, typically involving parameter-efficient [15, 26] or full parameter training [43, 52]. Applying SFT to a general LLM for specific domain adaptation has demonstrated effectiveness in various fields: in medicine [12], corpora of medical literature and clinical notes are used; in law [14], legal documents and case law are compiled; and in finance [54], financial reports and market data are utilized. In the scientific domain, several studies have specialized LLMs for scientific tasks, often necessitating the construction of a substantial domain-specific dataset with SFT. For example, SciGLM [61] leverages existing LLMs to generate step-by-step reasoning for unlabelled scientific instructions. ChemLLM [62], a more specified LLM in the chemistry field, collects structured chemical data from a vast selection of online databases and transforms this structured data into a question-answering format for SFT. SciRIFF [50] converts existing literature understanding datasets into natural language input-output pairs suitable for instruction-tuning using pre-defined templates. However, benchmark studies [8, 20] indicate that SFT alone may not provide adequate scientific knowledge to excel in relevant tasks. This suggests the need for a more comprehensive approach that combines domain knowledge infusion with instruction-following enhancements.

2.3 LLMs for Scientific Literature Understanding

In the scientific domain, existing strategies for developing specialized LLMs mostly fall into two categories: (1) Supervised fine-tuning with scientific instructions. This approach requires a large, high-quality, and diverse set of instructions to cultivate problem-solving abilities for scientific tasks. Representative works (*e.g.,* SciGLM [61], ChemLLM [62], and SciRIFF [50]) have been detailed in Section 2.2. (2) Pre-training with scientific corpora. This approach involves pre-training on a large corpus of scientific texts to improve performance on downstream scientific tasks. Early attempts, such as SciBert [6] and KV-PLV [60], are based on BERT [16] and pre-trained on a large corpus of scientific text for downstream scientific task enhancement. More recently, Galactica [47] is pre-trained on a vast corpus of scientific literature, including research papers, scientific articles, and other relevant scientific texts. Despite these advances, two major limitations hinder these models from excelling in scientific literature understanding: (1) lack of scientific knowledge, and (2) inability to follow task instructions. To address these challenges, we propose a combined pipeline of CPT and SFT to devise a specialized LLM for scientific literature understanding. It injects domain-specific knowledge through CPT while enhancing task-specific instruction-following abilities through SFT, leading to a more capable LLM for scientific literature understanding.

3 Method

In this section, we discuss the details of our proposed pipeline (*cf.* Figure 4): continual pre-training for scientific knowledge injection (*cf.* Section 3.1) and supervised fine-tuning for scientific tasks enhancement (*cf.* Section 3.2).

3.1 CPT for Scientific Knowledge Injection

What are high-quality pre-training corpora? Researchers [1, 22, 35] suggest that language models benefit from corpora that possess the same qualities as an exemplary textbook for human learners: clarity, self-containment, instructiveness, and balance. These characteristics ensure that the material is not only comprehensible but also informative and comprehensive, providing a solid foundation for knowledge acquisition. Over the past decades, the efforts of scientists and educators have resulted in a wealth of high-quality scientific textbooks and research papers, which serve as invaluable resources for learning and teaching. Recognizing this, we have curated a substantial collection of over 73,000 textbooks and 625,000 research papers within the scientific domain, ensuring all documents are copyright-compliant. To inject their rich scientific knowledge into a general LLM, we perform continual pre-training (CPT) on these high-quality textbooks and papers. This process equips the model with a robust scientific knowledge base, thereby paving the way for developing a specialized LLM tailored for scientific literature understanding.

However, we still face two practical obstacles when dealing with those textbooks and research papers: (1) Formatting and syntax errors. Most textbooks and research paper documents are in PDF format, which is not directly digestible by LLMs. Consequently, we need to transform them into plain text. Converting these documents using tools like PyPDF2 often introduces formatting and syntax errors, which degrade the quality of the corpus. (2) Corpus quality control. Despite their overall high quality, textbooks and research papers also contain segments with little useful information, such as references and garbled text introduced during the PDF parsing process. Given the large scale of the pre-training corpora, an effective and computation-efficient quality control measure is essential.

To tackle these obstacles, we devised the following modules of format & grammar correction and CPT quality filter:

3.1.1 Format & Grammar Correction. As illustrated in Appendix A, a parsed text from a PDF document often contains many formatting and syntax errors. To address this issue, we prompt a moderate yet powerful language model, Llama3-7B-Instruct [4], to correct these errors introduced during the PDF parsing process. Utilizing the vLLM [32] backend, Llama3-7B-Instruct can process approximately 2.52 million tokens per Nvidia A100 GPU hour. The process takes over 5,000 A100 GPU hours to handle all 73,000 textbooks and 625,000 research papers. Example texts – both before and after processing – along with the prompt template are provided in Appendix A to demonstrate the improvements made through this correction process.

3.1.2 CPT Quality Filter. During CPT, maintaining the quality of the training corpus is crucial for effective knowledge injection. Given the extremely large scale of pre-training corpora, assessing quality through human annotation is not feasible [18, 40]. Consequently, leading LLMs (*e.g.,* Phi [22], Llama [49], and Qwen [56]) employ model-based quality filters. The typical process involves using larger LLMs to score the quality of a subset of texts, which then serve as labels for training small classifiers (*e.g.,* random forest [22] and Bert [4]) to annotate the entire training corpus. Inspired by

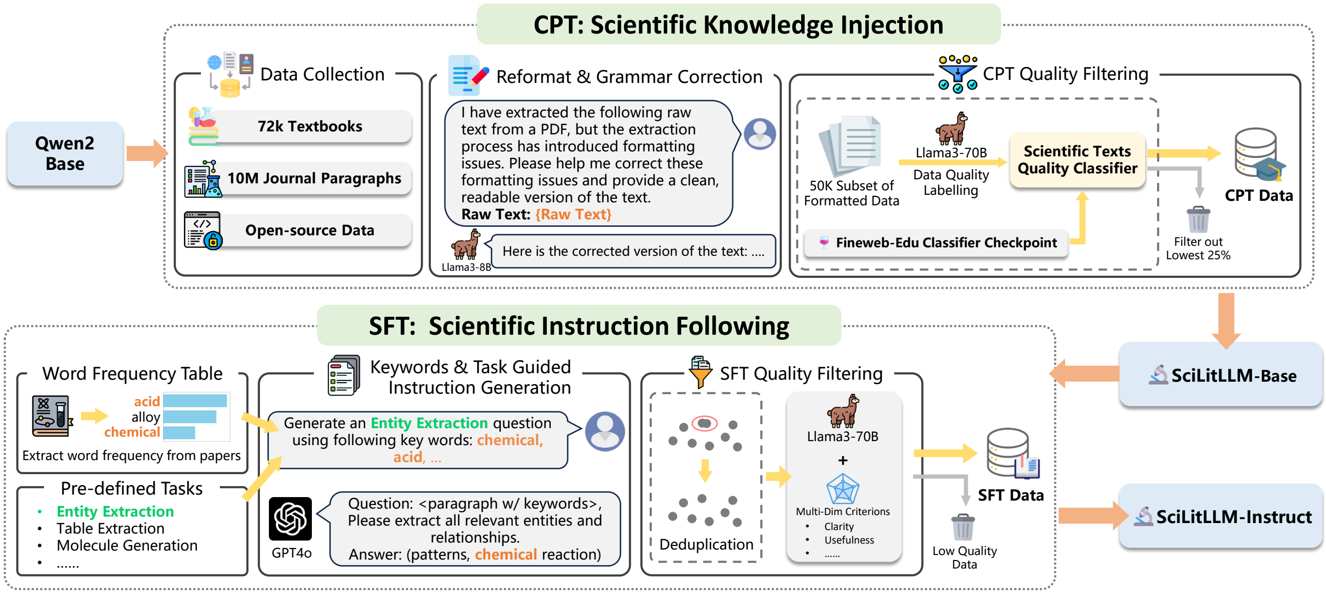

Figure 4: The pipeline of SciLitLLM consists of two key stages: continual pre-training (CPT) for scientific knowledge injection and supervised fine-tuning (SFT) for scientific instruction following. Specifically, the CPT stage involves several modules: PDF parsing, format & grammar correction (*cf.* Section 3.1.1), and quality filtering (*cf.* Section 3.1.2) modules. These modules ensure the model is equipped with high-quality scientific domain knowledge. The SFT stage includes LLM-based instruction generation (*cf.* Section 3.2.1) and instruction quality control (*cf.* Section 3.2.2) measures. These steps are designed to fine-tune the model's ability to follow scientific instructions accurately and effectively.

this approach, we design a resource-efficient method based on a lightweight text quality classifier.

Following prior studies [4, 5, 22, 56], we first annotate a random subset of 50k CPT texts using a powerful model – Llama3-70B-Instruct [4]. We adapt the quality assessment prompt from fineweb-edu-classifier [5], a widely-used quality classifier for web data, to evaluate the educational value [22] of the scientific knowledge in each sampled text, assigning scores ranging from 0 (lowest quality) to 5 (highest quality). After annotation, we perform supervised transfer learning on the fineweb-edu-classifier [5] checkpoint – a Bert-based [16] quality classifier in the web domain. This process results in a scientific text quality classifier tailored for scientific corpus assessment. See Appendix B for more details about classifier training and hyperparameters.

We then utilize this classifier to assess the quality of the entire CPT dataset (See Figure 3 for concrete samples). Each sample is evaluated and assigned with a continuous real number as the quality score. To enhance the overall quality of training data, we then exclude the lowest-scoring 25% from the dataset.

By leveraging the CPT quality classifier, we can efficiently filter out low-quality texts and ensure that only high-quality, informative content is retained. This step is crucial for enhancing the scientific knowledge base of our LLM, thereby improving its performance in scientific literature understanding.

3.1.3 CPT Training Settings. We perform CPT on Qwen2-Base [56] for one epoch, encompassing 23.7 billion tokens (*cf.* Table 1), with a sequence length of 2,048 tokens. To maintain the model's general knowledge, we also include a similar scale of general corpus tokens from Redpajama [13]. To stabilize the learning procedure, we gradually decrease the learning rate from 1×10^{-5} to 0 for SciLitLLM-7B, and from 5×10^{-6} to 0 for SciLitLLM-72B (QLoRA), with a cosine scheduler. To address overfitting, we apply a weight decay of 0.1 and gradients were clipped at a maximum value of 1.0. The CPT training took approximately 3 days on 32 Nvidia A100 GPUs for SciLitLLM-7B-Base (full parameters) and about 10 days for SciLitLLM-72B-Base (QLoRA).

Stage	Data source	Domain	#Doc/# Ins	# Tokens
CPT	In-house Textbooks	Science	73k	10B
	In-house Journals	Science	625k	2.7B
	Redpajama [13]	General	-	11B
SFT	SciLitIns	Science	110k	86M
	SciRIFF [50]	Science	70k	40M
	Infinity-Instruct[5]	General	3M	1.7B

Table 1: Data statistics of continual pre-training and supervised fine-tuning. #Doc/#Ins denotes the number of documents of CPT corpora and the number of instructions for SFT, respectively. Underlined datasets are curated by us.

3.2 SFT for Scientific Instruction Following

After performing CPT on an extensive scientific corpus to incorporate domain knowledge, we subsequently conduct SFT on domain-specific instructions to enhance the model's ability to understand scientific literature. We identify two major challenges in SFT for scientific instruction following:

- Existing instruction-tuning datasets in the scientific domain [19, 20, 34] primarily focus on fields such as physics, chemistry, and biology. Manually collecting instruction-tuning data for other less-represented vertical domains (*e.g.,* alloy, biomedicine, and material) is both time-consuming and costly [18, 40].
- Few instruction-tuning datasets adequately reflect the scenario of scientific literature understanding, which typically involves a segment of scientific literature accompanied by a question that requires deriving an answer from the text.

To address these challenges, we draw inspiration from leading models (*e.g.,* Nemotron-4 [2], Phi [22], and Qwen [56]), which leverage existing LLMs to construct synthetic instruction sets. We propose a novel instruction synthesis method to curate instructions specifically for scientific literature understanding.

3.2.1 Instruction Synthesis of Less-represented Domains. Unlike typical question-answer pairs, an instruction for a scientific literature understanding task comprises three components: (1) a segment of scientific literature, (2) a question pertaining to the context, and (3) the corresponding answer [50]. Simply prompting an LLM to generate a scientific context along with an associated question-answer pair – without variations in the instructions or parameters – often yields similar or repeated contents. This phenomenon arises because language models tend to adhere to the most probable or common paths dictated by their memory base and priors, thereby lacking the creativity to explore diverse generation [22]. Consequently, we are motivated to devise a strategy to encourage the language model to produce more creative and diverse instructions for scientific literature understanding while simultaneously ensuring the quality and coherence of the generated contexts.

We design a simple yet effective three-step pipeline to generate diverse and high-quality instructions for scientific contexts and corresponding question-answer pairs, consisting of the following:

(1) *Probability table of domain keywords.* For a target scientific domain (*e.g.,* alloy, biomedicine, and material), we collect dozens of high-impact research papers via Google Scholar[6] and count the frequency of each word appearing in these papers. Subsequently, we remove spaces and meaningless articles such as "a," "an," "the," etc. After normalization, a probability table of domain keywords, representing the word-level distribution of domain literature, is obtained.
(2) *Scientific task descriptions.* Since LLMs are expected to handle various types of scientific tasks, an instruction set with task diversity is essential. Therefore, we compile a list of task descriptions by including representative tasks from existing scientific NLP datasets [8, 20, 50], covering as many scenarios as possible that an LLM may encounter in real applications.
(3) *Instruction Generation.* Given a word probability table and the task list for a specific scientific domain, we sample 20 keywords and a task description each time. Subsequently, GPT-4o [38] is prompted to generate a scientific context containing the sampled keywords and a question-answer pair according to the provided task description.

[6]https://scholar.google.com/

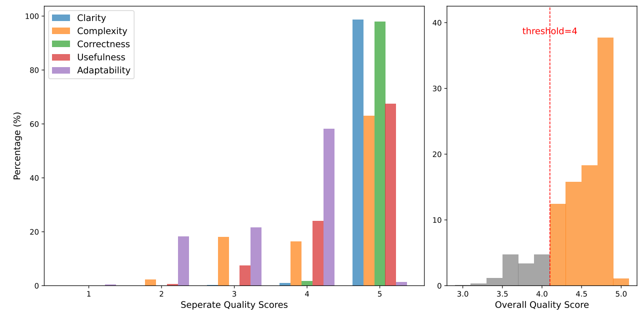

Figure 5: The quality of SciLitInsis evaluated from five aspects: clarity, complexity, correctness, usefulness, and adaptability (the higher the better). Instructions with an average score of less than 4 are filtered out.

The detailed generation process and example prompts are presented in Appendix C.1. Utilizing this pipeline, we obtain over 100k synthetic instructions for scientific literature understanding, covering less-represented scientific domains and various types of specialized tasks.

3.2.2 Instruction Quality Control. To ensure the diversity and quality of generated instructions, effective measures for quality control are essential. Specifically, we incorporate heuristic deduplication and LLM-based filtering.

(1) *Heuristic deduplication.* Despite the measures taken during the generation process to prevent high homogeneity in the instructions, the generated data points may still contain similar questions or identical answers. To eliminate such redundancy, we implement a simple yet effective deduplication process using the Levenshtein distance to calculate the similarity score between instructions. Based on this score, 5% to 10% of similar data are removed for each question type. Detailed processing steps are provided in Appendix C.2.
(2) *LLM-based filtering.* Inspired by recent efforts [11, 17, 63] to measure the quality of generated content using LLMs, we leverage Llama-3-70B-Instruct [4] to assess the quality of generated instructions for scientific literature understanding. Specifically, the quality is evaluated from five aspects: clarity, complexity, correctness, usefulness, and adaptability, assigning each instruction a score from 0 (lowest quality) to 5 (highest quality). Instructions with an average score of less than 4 are filtered out. We show the quality statistics of synthetic instructions in Figure 5. The detailed recipe for instruction quality evaluation with concrete examples is included in Appendix C.3.

Through instruction synthesis and quality control pipeline, we obtain **SciLitIns**, consisting of approximately 110,000 high-quality and diverse instructions for scientific literature understanding. With these instructions, models' problem-solving abilities in this specialized field could be enhanced.

Models	MMLU-s	CMMLU-s	Xiezhi-en-s	Xiezhi-ch-s
# Parameters < 15B				
ChatGLM-6B	46.99	46.42	58.33	63.50
Mistral-7B	59.84	36.43	64.97	53.34
Qwen1.5-7B	57.09	73.74	65.87	71.60
Qwen2-7B	66.62	86.04	71.53	74.36
Llama2-7B	39.87	28.33	42.40	36.40
Llama3-8B	61.52	42.16	66.29	63.46
Llama2-13B	49.06	32.29	58.30	42.65
Qwen1.5-14B	66.67	80.19	68.60	73.94
SciLitLLM-7B	**70.85**	**91.84**	**73.42**	**78.24**
# Parameters > 50B				
Mixtral-8x7B	67.58	44.88	69.54	65.81
Qwen2-57B-A14B	73.79	89.65	70.35	73.73
Llama2-70B	65.03	43.94	66.75	65.98
Llama3-70B	76.43	66.41	71.36	73.28
Qwen1.5-72B	74.89	86.87	71.18	74.47
Qwen2-72B	80.86	92.31	72.41	75.03
Qwen1.5-110B	77.06	90.72	73.45	73.14
SciLitLLM-72B (QLoRA)	**82.31**	**93.08**	**74.23**	**76.93**

Table 2: Performance comparison of base models. Bold indicates the highest performance for LLMs under 15B parameters or above 50B parameters. SciLitLLM achieves leading performance on all four scientific knowledge benchmarks.

3.2.3 SFT Training Settings. Our SFT training dataset consists of three parts: SciLitIns, SciRIFF [50] and Infinity-Instruct[7], as shown in Table 1. Infinity-Instruct is a collection of more than twenty open-source instructions datasets, covering various general domains. SciRIFF and SciLitIns contain specialized instructions for scientific literature understanding. We use full parameter training for SciLitLLM-7B-Base and QLoRA [15] parameter-efficient training for SciLitLLM-72B-Base. For both models, we train for one epoch on Infinity-Instruct to cultivate their general instruction-following abilities, then for five epochs on SciLitIns and SciRIFF for scientific literature understanding enhancement. The training is conducted with a sequence length of 4,096, a maximum learning rate of 5×10^{-6}, and a cosine scheduler. The SFT training takes approximately 32 hours on 32 A100 GPUs for the 7B and 100 hours for the 72B model, resulting in SciLitLLM-7B-Instruct and SciLitLLM-72B-Instruct.

4 Experiments

In this section, we perform experiments to answer the following research questions:

- **RQ1:** How does SciLitLLM perform on scientific literature understanding tasks?
- **RQ2:** Can CPT with domain-specific corpora aid in scientific knowledge injection?
- **RQ3:** Can SFT with synthetic instructions improve performance on scientific literature understanding tasks?

4.1 Experimental Setup

4.1.1 Benchmarks. To evaluate the performance of LLMs regarding scientific knowledge base and specialized task-solving abilities, our benchmarks include:

- *CPT benchmarks.* We evaluate the base models on three widely adopted benchmarks: MMLU [24], CMMLU [33], and Xiezhi [21]. Specifically, we select the STEM subsets from these benchmarks to assess their scientific knowledge, which serves as the foundation for scientific literature understanding.
- *SFT benchmarks.* We evaluate the instruct models on scientific literature understanding benchmarks: SciRIFF [50] and SciAssess [8]. Brief descriptions of them are provided in Appendix D.

4.1.2 Baselines. We test the following baselines:

- CPT baselines: We compare SciLitLLM-base against leading open-source base models: ChatGLM [59], Llama3 [4], Llama2 [49], Qwen2 [56], Qwen1.5 [48], Mistral-7B [28] and Mixtral-8x7B [29].
- SFT baselines: We benchmark leading instruction LLMs including GPT-4o [38], GPT-3.5 [7], Llama3 [4] and Qwen2 [56]. We also report the performance of SciTulu-7B [50], which is a fine-tuned Llama2-7B [49] on SciRIFF.

4.2 Performance Overview (RQ1)

4.2.1 Base model performance. The performance comparison of base models is shown in Table 2. SciLitLLM-base consistently outperforms other general base models across four scientific benchmarks. Specifically, compared with LLMs of less than 15 billion parameters, SciLitLLM-7B-Base shows an average accuracy improvement of 3.9% over Qwen2-7B. For LLMs with more than 50 billion parameters, SciLitLLM-72B-Base, with QLoRA training, outperforms all other LLMs (without quantization) as large as 110 billion parameters. The results demonstrate the effectiveness of CPT on high-quality scientific corpora, paving the way to a specialized LLM for scientific literature understanding.

4.2.2 Instruct model performance. As shown in Table 3, SciLitLLM-7B-Instruct achieves the highest performance in 4 out of 5 domains on SciAssess, outperforming the second-best model by 3.6%. Notably, on SciRIFF, it surpasses baseline models by a substantial margin of 10.1%. Additionally, SciLitLLM-72B, trained using QLoRA, shows a 1.7% and 0.9% performance improvement over Qwen2-72B on SciAssess and SciRIFF, respectively.

Detailed model performance on SciAssess is presented in Table 7, where SciLitLLM-7B and SciLitLLM-72B both lead in 12 and 13 out of 29 sub-tasks. Specifically, SciLitLLM-7B excels in tasks such as table extraction and molecule generation, likely benefiting from the comprehensive task coverage in our synthetic instruction dataset, SciLitIns. On SciRIFF, SciLitLLM-7B/SciLitLLM-72B ranks first in 8/6 out of 11 evaluations[8].

4.3 Ablation Study (RQ2 & RQ3)

We conducted ablation experiments on three key components in our pipeline: the CPT stage, the SFT data recipe, and the instruction quality filter, to demonstrate their effectiveness. It is important to note that all ablation experiments were performed on SciLitLLM-7B due to budget constraints.

4.3.1 Scientific knowledge injection via CPT (RQ2). We investigate the contribution of the CPT stage for SciLitLLM. We compare the three variants: (1) *Qwen2-7B-Instruct:* official instruct-model checkpoint; (2) *Qwen2-7B-base + SFT:* applying our SFT stage directly

[7] https://huggingface.co/datasets/BAAI/Infinity-Instruct

[8] In SciRIFF, the Qasper and SciFact tasks have two different evaluation methods and thus two results.

Dataset	Domain/ Task	# Parameter ~7B					# Parameter ~70B			API	
		SciTulu-7B	Mistral-7B	Llama3-8B	Qwen2-7B	SciLitLLM-7B	Llama3-70B	Qwen2-72B	SciLitLLM-72B (QLoRA)	GPT3.5	GPT4o
SciAssess	FundSci	32.3	48.3	58.5	70.3	**74.8**	70.9	77.1	**78.4**	62.2	76.7
	AlloyMat	23.9	28.0	32.9	32.8	**35.6**	44.9	42.7	**49.1**	32.0	52.1
	Biomed	67.8	76.0	77.4	**80.8**	79.6	79.6	**81.0**	79.6	78.0	82.3
	DrugDisc	25.4	30.2	32.0	31.7	**33.2**	41.5	35.5	**41.8**	31.0	43.4
	OrgMat	16.7	20.6	24.5	28.3	**38.9**	41.5	**52.7**	48.6	24.4	62.7
	Mean	33.2	40.6	45.0	48.8	**52.4**	55.7	57.8	**59.5**	45.5	63.4
SciRIFF	BioASQ	37.5	43.9	44.7	40.7	**51.0**	46.3	43.6	**50.7**	47.3	46.7
	BioR	55.7	48.2	45.3	44.3	**74.0**	59.9	59.1	**63.0**	53.9	61.0
	DiscMT	61.5	44.6	58.7	59.9	**77.4**	71.9	73.5	**73.5**	67.9	78.3
	EI	11.6	17.1	14.7	14.4	**22.3**	22.0	**24.1**	23.1	19.2	24.7
	MC	34.6	47.0	49.5	51.6	**68.0**	59.9	57.3	**70.5**	47.8	58.7
	MuP	72.1	93.4	90.7	**96.6**	76.8	96.4	**97.8**	77.9	76.8	86.9
	Qasper	54.2/38.6	**58.6**/39.4	58.2/41.9	56.8/34.5	58.4/**56.9**	25.0/19.4	**63.3**/47.2	60.5/**54.1**	54.7/39.8	67.8/50.5
	SciERC	35.6	30.2	19.9	27.5	**39.9**	35.2	34.1	**46.9**	28.6	42.2
	SciFact	66.0/49.2	**68.5**/51.3	64.6/51.7	65.2/44.3	68.5/**59.7**	**85.1**/**67.3**	82.3/65.9	75.2/60.6	69.7/53.3	84.3/68.7
	Mean	47.0	49.3	49.1	48.7	**59.4**	53.5	58.9	**59.7**	50.8	60.9

Table 3: Model performances on scientific literature understanding benchmarks: SciAssess and SciRIFF. SciLitLLM-7B and SciLitLLM-72B achieve leading performance compared with models of similar scales. The best-performing models in the 7B and 70B scales are highlighted in bold. Results for SciTulu-7B, GPT-3.5, and GPT-4o on SciRIFF are taken from its original papers, while all other results are generated by our experiments.

Model	SciAssess	SciRIFF
Qwen2-7B-Instruct	48.8	48.7
Qwen2-7B-Base + SFT	48.1	51.6
Qwen2-7B-Base + CPT + SFT	**52.4**	**59.4**

Table 4: Ablation study of the CPT stage. The results demonstrate the effectiveness of the CPT stage in improving performance on scientific literature understanding.

SFT Dataset	SciAssess	SciRIFF
Infinity-Instruct	44.5	44.7
Infinity-Instruct + SciRIFF	42.2	53.9
Infinity-Instruct + SciRIFF + SciLitIns	**52.4**	**59.4**

Table 5: Ablation study of SFT data recipes.

to Qwen2-7B-base without CPT; (2) *Qwen2-7B-base + CPT + SFT:* SciLitLLM-7B-Instruct.

As shown in Table 4, applying SFT alone to the Qwen2-7B-Base model does not lead to clear performance gains on SciAssess (-0.7%) and yields only a modest improvement on SciRIFF (+2.9%). In contrast, incorporating both CPT and SFT results in substantial performance enhancements: a 3.6% improvement on SciAssess and a 10.7% gain on SciRIFF. These results demonstrate that the CPT is crucial for effectively injecting scientific knowledge and significantly enhancing LLM performance on scientific literature understanding tasks.

4.3.2 Influence of SFT Data Recipes (RQ3). We explore the influence of each ingredient in SFT data recipes. We incrementally add three datasets to the SFT training set: Infinity-Instruct, SciRIFF, and SciLitIns. As shown in Table 5, using only the Infinity-Instruct results in the lowest performance on both SciAssess (44.5%) and SciRIFF (44.7%). This indicates that fine-tuning LLMs on general instructions alone is insufficient for scientific literature understanding, likely because Infinity-Instruct lacks specialized contents.

Dataset	SciAssess	SciRIFF
SciLitIns w/o filtering	51.1	56.2
SciLitIns w/ filtering	**52.4**	**59.4**

Table 6: Ablation study of SFT instruction quality filtering.

Adding SciRIFF to Infinity-Instruct improves performance on SciRIFF significantly but decreases performance on SciAssess. This discrepancy may be due to the disjoint coverage of scientific domains between SciRIFF and SciAssess. Finally, including SciLitIns along with Infinity-Instruct and SciRIFF boosts performance on both benchmarks, with SciAssess at 52.4% and SciRIFF at 59.4%. This demonstrates that including SciLitIns that covers less-represented scientific domains and tasks is beneficial for enhancing model performance in scientific literature understanding.

4.3.3 Influence of Instruction Quality Filter. We conduct an ablation study to assess the impact of quality filter for synthetic instructions by varying whether the dataset SciLitIns was filtered. As discussed in Section 3.2.2, this filter removes low-quality instructions evaluated from five key aspects. Table 6 shows that applying the filter significantly improved the performance of SciLitLLM-7B on SciAssess (+1.3%) and SciRIFF (+3.2%). This demonstrates that SFT quality filtering process effectively selects high-value educational instructions, thereby boosting the performance of SciLitLLMon scientific literature understanding.

5 Limitations

Despite the promising results achieved by SciLitLLM, there are several limitations that should be acknowledged:

- *Insufficient data volume.* Compared with existing pre-training datasets [4, 47, 56], the amount of data used for CPT is not satisfying. Future work should consider incorporating a larger scientific corpus, potentially including scientific blogs or purely synthetic data, to enhance the model's scientific knowledge base and overall performance.

- *Lack of reasoning enhancement.* The current pipeline does not explore advanced reasoning techniques such as Chain-of-Thought [53] or Tree-of-Thought [58] in the data construction or model inference stages. Investigating these methods could potentially improve the model's inference capabilities and overall performance.
- *Lack of preference alignment.* Due to a limited financial budget, the model lacks Reinforcement Learning from Human Feedback (RLHF) [39]. RLHF has shown significant improvements in aligning models with human preferences and ensuring more reliable outputs. Implementing RLHF in future iterations could further enhance the model's reliability.

Addressing these limitations in future research will be crucial developing a more robust and capable LLM specialized in scientific literature understanding.

6 Conclusion and Future Works

In this paper, we introduce **SciLitLLM**, a specialized model for scientific literature understanding. It is initialized with a general base model – Qwen2 [56], and trained through a sequential pipeline of continual pre-training (CPT) and supervised fine-tuning (SFT). For effective scientific knowledge injection during CPT, we propose model-based format and grammar correction method, along with text quality filtering measures. To ensure high-quality and diverse instructions during SFT, we devise instruction synthesis and quality control approaches. Our experiments on widely-used benchmarks demonstrate the effectiveness of this pipeline in adapting a general model to the field of scientific literature understanding. Specifically, SciLitLLM-7B achieves a 3.6% improvement on the SciAssess [8] and a 10.1% improvement on the SciRIFF [50] compared to leading models with fewer than 10 billion parameters. SciLitLLM-72B, trained with QLoRA, also surpasses baseline open-source LLMs. We note that this pipeline could be easily adapted to other specialized domains, particularly those lacking adequate open-source corpora and high-quality instruction sets.

Our future work will focus on expanding the diversity and quality of the training data, as well as exploring more efficient methods for domain-specific knowledge injection and high-quality instruction generation. Moreover, we plan to expand our pipeline to include the RLHF [39] stage for better human preference alignment and enhanced safety.

References

[1] Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat S. Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. *CoRR* abs/2404.14219 (2024).

[2] Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H. Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan M. Cohen, Sirshak Das, Ayush Dattagupta, Olivier Delalleau, Leon Derczynski, Yi Dong, Daniel Egert, Ellie Evans, Aleksander Ficek, Denys Fridman, Shaona Ghosh, Boris Ginsburg, Igor Gitman, Tomasz Grzegorzek, Robert Hero, Jining Huang, Vibhu Jawa, Joseph Jennings, Aastha Jhunjhunwala, John Kamalu, Sadaf Khan, Oleksii Kuchaiev, Patrick LeGresley, Hui Li, Jiwei Liu, Zihan Liu, Eileen Long, Ameya Sunil Mahabaleshwarkar, Somshubra Majumdar, James Maki, Miguel Martinez, Maer Rodrigues de Melo, Ivan Moshkov, Deepak Narayanan, Sean Narenthiran, Jesus Navarro, Phong Nguyen, Osvald Nitski, Vahid Noroozi, Guruprasad Nutheti, Christopher Parisien, Jupinder Parmar, Mostofa Patwary, Krzysztof Pawelec, Wei Ping, Shrimai Prabhumoye, Rajarshi Roy, Trisha Saar, Vasanth Rao Naik Sabavat, Sanjeev Satheesh, Jane Polak Scowcroft, Jason Sewall, Pavel Shamis, Gerald Shen, Mohammad Shoeybi, Dave Sizer, Misha Smelyanskiy, Felipe Soares, Makesh Narsimhan Sreedhar, Dan Su, Sandeep Subramanian, Shengyang Sun, Shubham Toshniwal, Hao Wang, Zhilin Wang, Jiaxuan You, Jiaqi Zeng, Jimmy Zhang, Jing Zhang, Vivienne Zhang, Yian Zhang, and Chen Zhu. 2024. Nemotron-4 340B Technical Report. *CoRR* abs/2406.11704 (2024). https://doi.org/10.48550/arXiv.2406.11704

[3] Microsoft Research AI4Science and Microsoft Azure Quantum. 2023. The Impact of Large Language Models on Scientific Discovery: a Preliminary Study using GPT-4. *CoRR* abs/2311.07361 (2023).

[4] AI@Meta. 2024. Llama 3 Model Card. (2024). https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

[5] Lozhkov Anton, Ben Allal Loubna, von Werra Leandro, and Wolf Thomas. 2024. *FineWeb-Edu*. https://doi.org/10.57967/hf/2497

[6] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *EMNLP/IJCNLP (1)*. Association for Computational Linguistics, 3613–3618.

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

[8] Hengxing Cai, Xiaochen Cai, Junhan Chang, Sihang Li, Lin Yao, Changxin Wang, Zhifeng Gao, Hongshuai Wang, Yongge Li, Mujie Lin, Shuwen Yang, Jiankun Wang, Yuqi Yin, Yaqi Li, Linfeng Zhang, and Guolin Ke. 2024. SciAssess: Benchmarking LLM Proficiency in Scientific Literature Analysis. *CoRR* (2024). https://doi.org/10.48550/arXiv.2403.01976

[9] Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. 2024. Instruction Pre-Training: Language Models are Supervised Multitask Learners. *arXiv preprint arXiv:2406.14491* (2024).

[10] Daixuan Cheng, Shaohan Huang, and Furu Wei. 2023. Adapting Large Language Models via Reading Comprehension. *CoRR* abs/2309.09530 (2023).

[11] David Cheng-Han Chiang and Hung-yi Lee. 2023. Can Large Language Models Be an Alternative to Human Evaluations?. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, 15607–15631. https://doi.org/10.18653/v1/2023.acl-long.870

[12] Jan Clusmann, Fiona R Kolbinger, Hannah Sophie Muti, Zunamys I Carrero, Jan-Niklas Eckardt, Narmin Ghaffari Laleh, Chiara Maria Lavinia Löffler, Sophie-Caroline Schwarzkopf, Michaela Unger, Gregory P Veldhuizen, et al. 2023. The future landscape of large language models in medicine. *Communications medicine* 3, 1 (2023), 141.

[13] Together Computer. 2023. *RedPajama: an Open Dataset for Training Large Language Models.* https://github.com/togethercomputer/RedPajama-Data

[14] Jiaxi Cui, Munan Ning, Zongjian Li, Bohua Chen, Yang Yan, Hao Li, Bin Ling, Yonghong Tian, and Li Yuan. 2024. Chatlaw: A Multi-Agent Collaborative Legal Assistant with Knowledge Graph Enhanced Mixture-of-Experts Large Language Model. arXiv:2306.16092 [cs.CL] https://arxiv.org/abs/2306.16092

[15] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. In *NeurIPS*.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*. Association for Computational Linguistics, 4171–4186.

[17] Ronen Eldan and Yuanzhi Li. 2023. TinyStories: How Small Can Language Models Be and Still Speak Coherent English? *CoRR* abs/2305.07759 (2023). https://doi.org/10.48550/arXiv.2305.07759

[18] Alexander Erdmann, David Joseph Wrisley, Benjamin Allen, Christopher Brown, Sophie Cohen-Bodénès, Micha Elsner, Yukun Feng, Brian Joseph, Béatrice Joyeux-Prunel, and Marie-Catherine de Marneffe. 2019. Practical, Efficient, and Customizable Active Learning for Named Entity Recognition in the Digital Humanities. In

NAACL-HLT (1). Association for Computational Linguistics, 2223–2234.

[19] Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. 2023. Mol-Instructions: A Large-Scale Biomolecular Instruction Dataset for Large Language Models. *CoRR* abs/2306.08018 (2023). https://doi.org/10.48550/arXiv.2306.08018

[20] Kehua Feng, Keyan Ding, Weijie Wang, Xiang Zhuang, Zeyuan Wang, Ming Qin, Yu Zhao, Jianhua Yao, Qiang Zhang, and Huajun Chen. 2024. SciKnowEval: Evaluating Multi-level Scientific Knowledge of Large Language Models. *CoRR* (2024). https://doi.org/10.48550/arXiv.2406.09098

[21] Zhouhong Gu, Xiaoxuan Zhu, Haoning Ye, Lin Zhang, Jianchen Wang, Yixin Zhu, Sihang Jiang, Zhuozhi Xiong, Zihan Li, Weijie Wu, Qianyu He, Rui Xu, Wenhao Huang, Jingping Liu, Zili Wang, Shusen Wang, Weiguo Zheng, Hongwei Feng, and Yanghua Xiao. 2024. Xiezhi: An Ever-Updating Benchmark for Holistic Domain Knowledge Evaluation. In *AAAI*. AAAI Press, 18099–18107. https://doi.org/10.1609/aaai.v38i16.29767

[22] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks Are All You Need. *CoRR* abs/2306.11644 (2023).

[23] Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats L. Richter, Quentin Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. 2023. Continual Pre-Training of Large Language Models: How to (re)warm your model? *CoRR* abs/2308.04014 (2023).

[24] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. In *ICLR*. https://openreview.net/forum?id=d7KBjmI3GmQ

[25] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training Compute-Optimal Large Language Models. *CoRR* abs/2203.15556 (2022).

[26] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *CoRR* abs/2106.09685 (2021). https://arxiv.org/abs/2106.09685

[27] Quzhe Huang, Mingxu Tao, Zhenwei An, Chen Zhang, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. 2023. Lawyer LLaMA Technical Report. *CoRR* abs/2305.15062 (2023). https://doi.org/10.48550/arXiv.2305.15062

[28] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *CoRR* abs/2310.06825 (2023). https://doi.org/10.48550/arXiv.2310.06825

[29] 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, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of Experts. *CoRR* abs/2401.04088 (2024). https://doi.org/10.48550/arXiv.2401.04088

[30] Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew O. Arnold, and Xiang Ren. 2022. Lifelong Pretraining: Continually Adapting Language Models to Emerging Corpora. In *NAACL-HLT*. Association for Computational Linguistics, 4764–4780.

[31] Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual Pre-training of Language Models. In *ICLR*. OpenReview.net.

[32] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *SOSP*. ACM, 611–626.

[33] Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023. CMMLU: Measuring massive multitask language understanding in Chinese. *CoRR* abs/2306.09212 (2023). https://doi.org/10.48550/arXiv.2306.09212

[34] Sihang Li, Zhiyuan Liu, Yanchen Luo, Xiang Wang, Xiangnan He, Kenji Kawaguchi, Tat-Seng Chua, and Qi Tian. 2024. Towards 3D Molecule-Text Interpretation in Language Models. *CoRR* abs/2401.13923 (2024).

[35] Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks Are All You Need II: phi-1.5 technical report. *CoRR* abs/2309.05463 (2023).

[36] Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun Li, Hejie Cui, Xuchao Zhang, Tianjiao Zhao, Amit Panalkar, Wei Cheng, Haoyu Wang, Yanchi Liu, Zhengzhang Chen, Haifeng Chen, Chris White, Quanquan Gu, Carl Yang, and Liang Zhao. 2023. Beyond One-Model-Fits-All: A Survey of Domain Specialization for Large Language Models. *CoRR* abs/2305.18703 (2023). https://doi.org/10.48550/ARXIV.2305.18703 arXiv:2305.18703

[37] Sanket Vaibhav Mehta, Darshan Patil, Sarath Chandar, and Emma Strubell. 2023. An Empirical Investigation of the Role of Pre-training in Lifelong Learning. *J. Mach. Learn. Res.* 24 (2023), 214:1–214:50.

[38] OpenAI. 2023. GPT-4 Technical Report. *CoRR* abs/2303.08774 (2023). https://doi.org/10.48550/arXiv.2303.08774

[39] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*. http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html

[40] Hang Qiu, Krishna Chintalapudi, and Ramesh Govindan. 2023. MCAL: Minimum Cost Human-Machine Active Labeling. In *ICLR*. OpenReview.net.

[41] Haoran Que, Jiaheng Liu, Ge Zhang, Chenchen Zhang, Xingwei Qu, Yinghao Ma, Feiyu Duan, Zhiqi Bai, Jiakai Wang, Yuanxing Zhang, et al. 2024. D-CPT Law: Domain-specific Continual Pre-Training Scaling Law for Large Language Models. *arXiv preprint arXiv:2406.01375* (2024).

[42] 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 (2019), 9.

[43] 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. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67.

[44] Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2023. SciRepEval: A Multi-Format Benchmark for Scientific Document Representations. In *EMNLP*. 5548–5566. https://doi.org/10.18653/v1/2023.emnlp-main.338

[45] Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Kumar Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Schärli, Aakanksha Chowdhery, Philip Andrew Mansfield, Blaise Agüera y Arcas, Dale R. Webster, Gregory S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle K. Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. 2022. Large Language Models Encode Clinical Knowledge. *CoRR* abs/2212.13138 (2022). https://doi.org/10.48550/arXiv.2212.13138

[46] Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding. In *AAAI*. AAAI Press, 8968–8975.

[47] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A Large Language Model for Science. *CoRR* abs/2211.09085 (2022). https://doi.org/10.48550/arXiv.2211.09085

[48] Qwen Team. 2024. Introducing Qwen1.5. https://qwenlm.github.io/blog/qwen1.5/

[49] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR* abs/2302.13971 (2023).

[50] David Wadden, Kejian Shi, Jacob Morrison, Aakanksha Naik, Shruti Singh, Nitzan Barzilay, Kyle Lo, Tom Hope, Luca Soldaini, Shannon Zejiang Shen, Doug Downey, Hannaneh Hajishirzi, and Arman Cohan. 2024. SciRIFF: A Resource to Enhance Language Model Instruction-Following over Scientific Literature. *CoRR* abs/2406.07835 (2024). https://doi.org/10.48550/arXiv.2406.07835

[51] Jason Wei, Maarten Bosma, Vincent Y 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. *arXiv preprint arXiv:2109.01652* (2021).

[52] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned Language Models are Zero-Shot Learners. In *ICLR*. OpenReview.net.

[53] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *NeurIPS*.

[54] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David S. Rosenberg, and Gideon Mann. 2023. BloombergGPT: A Large Language Model for Finance. *CoRR* abs/2303.17564 (2023). https://doi.org/10.48550/ARXIV.2303.17564 arXiv:2303.17564

[55] Tongtong Wu, Massimo Caccia, Zhuang Li, Yuan-Fang Li, Guilin Qi, and Gholamreza Haffari. 2022. Pretrained Language Model in Continual Learning: A Comparative Study. In *ICLR*. OpenReview.net.

[56] 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. https://arxiv.org/abs/2407.10671

[57] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. FinGPT: Open-Source Financial Large Language Models. *CoRR* abs/2306.06031 (2023). https://doi.org/10.48550/arXiv.2306.06031

[58] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In *NeurIPS*.

[59] 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, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. *CoRR* abs/2406.12793 (2024). https://doi.org/10.48550/arXiv.2406.12793

[60] Zheni Zeng, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2022. A Deep-learning System Bridging Molecule Structure and Biomedical Text with Comprehension Comparable to Human Professionals. *Nature communications* 13, 862 (2022).

[61] Dan Zhang, Ziniu Hu, Sining Zhoubian, Zhengxiao Du, Kaiyu Yang, Zihan Wang, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024. SciGLM: Training Scientific Language Models with Self-Reflective Instruction Annotation and Tuning. *CoRR* abs/2401.07950 (2024). https://doi.org/10.48550/arXiv.2401.07950

[62] Di Zhang, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyu Yue, Dongzhan Zhou, Shufei Zhang, Mao Su, Hansen Zhong, Yuqiang Li, and Wanli Ouyang. 2024. ChemLLM: A Chemical Large Language Model. *CoRR* abs/2402.06852 (2024). https://doi.org/10.48550/arXiv.2402.06852

[63] Xingjian Zhang, Yutong Xie, Jin Huang, Jinge Ma, Zhaoying Pan, Qijia Liu, Ziyang Xiong, Tolga Ergen, Dongsub Shim, Honglak Lee, and Qiaozhu Mei. 2024. MASSW: A New Dataset and Benchmark Tasks for AI-Assisted Scientific Workflows. *CoRR* abs/2406.06357 (2024). https://doi.org/10.48550/arXiv.2406.06357

[64] Yizhen Zheng, Huan Yee Koh, Jiaxin Ju, Anh T. N. Nguyen, Lauren T. May, Geoffrey I. Webb, and Shirui Pan. 2023. Large Language Models for Scientific Synthesis, Inference and Explanation. *CoRR* abs/2310.07984 (2023).

A Format & Grammar Correction Examples

Raw text parsed by PyPDF2

Highly p e n e t r a t i n g radiation, such as γ-rays or fast electorns, deposits ener gy
throughout the solid t a r g e t material. Gas production occurs w i t h i n the solid phase and
must d i f f u s e to the surface to be observed. The a p p a r e n t yield of H
2 can depend on the radiolysis
pro c e d u r e or the particle size if some of the gas remains in the solid. Experiments have
shown
that the apparent y i e l d of H 2 can vary by a f a c t o r of 3 in the r a d i o l y s i s of polyethylene
spheres
of 7 to 2100 cm2/g (about 9 to 0.03 mm) [12]. The e f f e c t s of gas trapping and diffusion are
not understood in the c o n t e x t of waste storage. Extremely h i g h dose rates in the p r o c e s
s i n g of certain materials may lead to bubble formation, which could a l t e r product quality.
The y i e l d of H
2 in the r a d i o l y s i s of polymers w i t h γ-rays is well known for several
types of p o l y m e r s [2].

Format & Grammar Correction Prompt

I have extracted the following raw text from a PDF, but the extraction process has introduced many formatting issues such as unnecessary line breaks, extra spaces, and other artifacts that disrupt the text flow. Could you please help me correct these formatting issues and provide a clean, readable version of the text? Respond with the Corrected Version only.
Raw Text:
{RawText}
Start your response with "Here is the corrected version of the text:".

Text after Format & Grammar Correction

Highly penetrating radiation, such as γ-rays or fast electrons, deposits energy throughout the solid target material. Gas production occurs within the solid phase and must diffuse to the surface to be observed. The apparent yield of H2 can depend on the radiolysis procedure or the particle size if some of the gas remains in the solid. Experiments have shown that the apparent yield of H2 can vary by a factor of 3 in the radiolysis of polyethylene spheres of 7 to 2100 cm2/g (about 9 to 0.03 mm) [12]. The effects of gas trapping and diffusion are not understood in the context of waste storage. Extremely high dose rates in the processing of certain materials may lead to bubble formation, which could alter product quality.
The yield of H2 in the radiolysis of polymers with γ-rays is well known for several types of polymers [2].

B CPT Quality Filter

We randomly select 50k samples from our 700k CPT data. These selected samples are then scored using the Llama3-70B model. The prompt utilized for this scoring process is as follows:

Prompt for CPT Data Quality Labelling

Below is an extract from a textbook. Evaluate whether the text has a high educational value and could be useful in an educational setting for teaching from primary school to grade school levels using the additive 5-point scoring system described below. Points are accumulated based on the satisfaction of each criterion:

- Add 1 point if the extract provides some basic information relevant to educational topics, even if it includes some irrelevant or non-academic content like advertisements and promotional material.
- Add another point if the extract addresses certain elements pertinent to education but does not align closely with educational standards. It might mix educational content with non-educational material, offering a superficial overview of potentially useful topics, or presenting information in a disorganized manner and incoherent writing style.
- Award a third point if the extract is appropriate for educational use and introduces key concepts relevant to school curricula. It is coherent though it may not be comprehensive or could include some extraneous information. It may resemble an introductory section of a textbook or a basic tutorial that is suitable for learning but has notable limitations like treating concepts that are too complex for grade school students.
- Grant a fourth point if the extract is highly relevant and beneficial for educational purposes for a level not higher than grade school, exhibiting a clear and consistent writing style. It could be similar to a chapter from a textbook or a tutorial, offering substantial educational content, including exercises and solutions, with minimal irrelevant information, and the concepts aren't too advanced for grade school students. The content is coherent, focused, and valuable for structured learning.
- Bestow a fifth point if the extract is outstanding in its educational value, and perfectly suited for teaching either at primary school or grade school. It follows detailed reasoning, the writing style is easy to follow, and offers profound and thorough insights into the subject matter, devoid of any non-educational or complex content.

After examining the extract:
- Briefly justify your total score, up to 100 words.
- Conclude with the score using the format: "Educational score: <total points>"

We train a Scientific Texts Quality Classifier on these labeled data samples. The classifier is a 109M BERT [16] classifier, fine-tuned from the checkpoint of fineweb-edu-classifier [5]. The model is trained for 20 epochs with a learning rate of 0.001 and a batch size of 1024. Ninety percent of the 50K samples are used as the training set, and the rest 10% are used as the validation set. The training process costs approximately 50 minutes on 4 A100 GPUs. We select the checkpoint from the epoch that yields the highest validation micro F1 score as our final checkpoint.

During Inference, we set batch size to 2048, and beam number to 1. The inference process costs 90 minutes on 4 A100 GPUs. We utilize the generated to filter out 25% data with the lowest quality. The distribution of the scores is demonstrated in Figure 6. The filtered-out 25% data are marked **gray**, while the remaining 75% CPT data are marked **orange**.

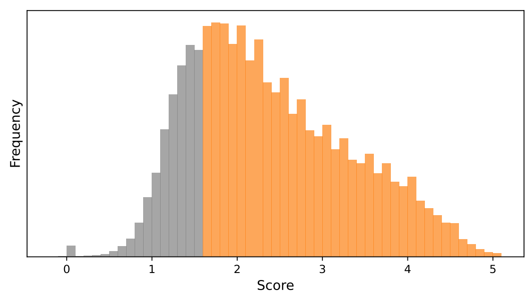

Figure 6: Score distribution of the CPT Data

C SFT Details

C.1 Instruction Generation Pipeline

In SciLitIns, we focus on generating instructions for three less-represented domains (materials science, medicine, and drug discovery) and five question types:

- **Table Extraction:** Table Extraction tasks evaluate a model's proficiency in extracting, summarizing, and structuring data from an article into a table format.
- **Entity Extraction:** Entity Extraction tasks are designed to evaluate a model's ability to extract specific information, such as entities or relationships, from the text.
- **Molecule Translation:** Molecule Translation tasks evaluate a model's ability to translate molecules between different SMILES formats.
- **Molecule Extraction:** Molecule Extraction tasks ask a model to extract an appropriate molecule from a scientific paragraph that contains multiple molecules.
- **Multiple Choice and True-or-False:** Multiple Choice and True-or-False questions assess a model's ability to select the correct answer from a set of options, testing its knowledge and reasoning on both simple and complex scenarios.

For each of the three scientific domains, we collect a set of high-impact research papers and construct a word frequency table. To generate a question in a given domain, we sample 20 keywords from the corresponding word table and insert them into the prompt for that question. To ensure fair representations of less frequent keywords, we use random sampling with a temperature setting of 3. We release our code, prompt templates, and word frequency tables. Below is an example of generating a table extraction question:

Prompt for Generating a Table Extraction Question

I need synthetic training data for training a machine learning model that extracts tables from text correctly. The data should be formatted in JSON, with each entry containing "text" and "answer" attributes. You should generate a paragraph that includes the keywords:
{{keywords}}.
The "text" part must contain enough information for the table to be extracted! In "text" part, You must you include a table description in latex format.
Special notice for the table content:
You should generate a table that has complicated numbers and characters, include non-standard characters, and have a variety of values. Make sure the value you generated do not follow simple patterns, for example, never include deplicate values or values with constant interval in columns.
Your answer should contain as much details as possible. You should only generate one JSON.
The value for the two attributes should be two string. Use {{ and }} to warp your output. Pay attention to the escape characters in the latex format. Remember to put a comma at the end of the first string. Never use a json block to wrap your output. Here is the format for your output:
{
"text": "Your paragraph here, remember to include a table in latex format",
"answer": "Your answer table here"
}
Now start your answer:

C.2 Instruction Deduplication

The generated synthetic data may contain similar questions or identical answers. To eliminate redundancy, we implement a fuzzy deduplication process using the Levenshtein distance to calculate the similarity score between question-answer pairs. Specifically, for two pairs (q_1, a_1) and (q_2, a_2), their textual similarity is defined as $(1 - \text{lev}(q_1, q_2))(1 - \text{lev}(a_1, a_2))$, where $\text{lev}(\cdot, \cdot)$ denotes the Levenshtein distance. Due to significant differences between texts from different question types, we compute similarity matrices separately for each type. We then use a disjoint-set data structure to merge highly similar data points. We use this process to remove approximately 5% to 10% of duplicated data for each question type.

C.3 Quality Assessment of Generated SFT Instructions

In section 3.2.2, we sample 10k instruction pairs from SciLitIns and evaluate them by Llama-3-70B using the below prompt.

SFT Evaluation Prompt

You are a helpful and precise assistant for checking the quality of instruction-tuning data for large language models. Your task is to evaluate the given instruction using the criterions described below.
- Clarity: The sample should be clear, specific, and unambiguous, providing a well-defined task for the model to perform.
- Complexity: The sample should be advanced complexity that necessitate a high level of comprehension and cognitive processing, challenging the language model significantly.
- Correctness: The sample is impeccably written, with flawless grammar, syntax, and structure, demonstrating exceptional clarity and professionalism.
- Usefulness: The sample should be highly useful, and contribute to expanding the model's knowledge base.
- Adaptability: The sample could be adapted to different contexts or use cases, showing some flexibility.
After examining the instruction-response pair:
- Briefly justify your scores with a paragraph in the field "Explanation", up to 500 words.
- For each point of criterion above, assign a score from 1 to 5.
- You should only provide the rest of your answer in a structured format as shown below, and make sure your response can be directly parsed by computer programs.
Below is a template for your response:
Explanation: <string, your explanations to the scores>
====================
{
"Clarity": <int, complexity_score>,
"Complexity": <int, complexity_score>,
"Correctness": <int, quality_score>,
"Usefulness": <int, usefulness_score>,
"Adaptability": <int, adaptability_score>,
"Total": <int, total_score>
}

Below is an example of SciLitIns, which will be sent to Llama-3-70B for evaluation.

An Example in SciLitIns

Context
In recent studies regarding material science, the crmatrix has shown significant importance in understanding fracture behaviors. Alternatively, many researchers have proposed a series of experiments to analyze these phenomena. Even though the process is delicate, the enthusiasm for discovering new findings is understandable. The following Table 1 presents a schematic illustration of the enthalpy changes observed during the different fracture phases. These studies are indispensable for advancing our knowledge of the material's resilience.
In addition, in certain cases, pronounced errors in measurement can occur, but the references provided can guide through the correct methodology. Prime among the tools used is the differential scanning calorimetry which helps measure these enthalpy changes accurately.

```
\begin{table}[h!]\n\centering\n\begin{tabular}
    {|c|c|c|c|}\n\hline\nMaterial & Initial Enthalpy
    (kJ/mol) & Final Enthalpy (kJ/mol) & Fracture Phase
    \\\n\hline\nMaterial A & 25.4 & 47.8 & Brittle
    \\\n\hline\nMaterial B & 22.3 & 45.0 & Ductile
    \\\n\hline\nMaterial C & 28.9 & 50.2 & Semi-brittle
    \\\n\hline\n\end{tabular}\n\caption{Enthalpy changes
    during fracture
    phases}\n\label{table:enthalpy}\n\end{table}
```

The process of accurately measuring these values is crucial for the material selection in various engineering applications, making it an indispensable research area.

Instruction
Extract the table in the paragraph.

Response
Material,Initial Enthalpy (kJ/mol),Final Enthalpy (kJ/mol),Fracture Phase
Material A,25.4,47.8,Brittle
Material B,22.3,45.0,Ductile
Material C,28.9,50.2,Semi-brittle

Below is the evaluation by Llama-3-70B on the example from SciLitIns. Each criterion is scored out of 5 points, making the total possible points 25. The total score is then averaged, resulting in an overall score out of 5.

Evaluation by Llama-3-70B

Explanation:
The instruction is clear and specific, asking the model to extract a table from a given paragraph. The paragraph itself is well-written, with proper grammar and syntax, making it easy to understand. The table is also well-structured and provides valuable information about enthalpy changes during fracture phases. The task requires a good understanding of the text and the ability to identify the table. However, the complexity of the task is relatively low, as it only involves extracting a table, and the context is not particularly nuanced or ambiguous. The task is useful for advancing knowledge in material science, and the table could be adapted to different contexts or use cases.
====================
"Clarity": 5,
"Complexity": 2,
"Correctness": 5,
"Usefulness": 4,
"Adaptability": 4,
"Total": 20

D Benchmark Details

To the best of our knowledge, there are two commonly-adopted datasets for scientific literature understanding:

- SciRIFF [50] evaluates essential scientific literature understanding capabilities, including information extraction, summarization, question answering, claim verification, and classification. Data points in SciRIFF are notable for their long input contexts and complicated structured outputs. The Qasper and SciFact tasks have two different evaluation methods and thus two results. We note that SciRIFF contains a separate training set used in the SFT stage in our study.
- SciAssess [8] features an end-to-end benchmark of understanding PDF content. It includes 29 tasks from five scientific domains: fundamental science, alloy materials, biomedicine, drug discovery, and organic materials. SciAssess is used exclusively for testing in our evaluation.

Overall, SciRIFF provides basic benchmarks for comprehending short scientific segments and various instructions, while SciAssess presents more challenging tasks involving longer contexts from raw PDFs.

E Detailed Performance on SciAssess

The detailed results on each tasks in SciAssess are shown in Table 7.

Domain	Task	SciTulu-7B	Mistral-7B	Llama3-8B	Qwen2-7B	SciLitLLM-7B	Llama3-70B	Qwen2-72B	SciLitLLM-72B	GPT3.5	GPT4o
Fundamental Science	Average	32.3	48.3	58.5	70.3	**74.8**	70.9	77.1	**78.4**	62.2	76.7
	MMLU	35.5	52.1	59.4	64.0	**65.3**	76.2	**78.7**	78.5	64.3	84.2
	CMMLU	27.5	31.8	46.1	79.2	**87.8**	65.5	86.6	**89.6**	44.9	78.7
	Xz-Ch	34.6	51.7	64.7	71.7	**75.4**	73.4	74.0	**75.5**	73.2	73.4
	Xz-En	31.5	57.6	63.6	66.2	**70.7**	68.4	69.1	**69.9**	66.5	70.3
Alloy Materials	Average	23.9	28.0	32.9	32.8	**35.6**	44.9	42.7	**49.1**	32.0	52.1
	AlloyQA	6.7	33.3	26.7	**53.3**	**53.3**	46.7	53.3	**66.7**	53.3	46.7
	CompEx	9.0	9.0	8.1	**10.1**	7.2	**34.7**	19.3	19.5	24.8	50.5
	TempEx	**34.3**	28.5	32.9	28.5	30.9	55.6	**58.0**	57.9	30.9	60.9
	SampDiff	27.4	14.3	**29.1**	6.3	18.1	26.6	7.6	**32.8**	12.7	34.6
	TreatSeq	42.2	54.9	67.6	65.7	**68.6**	60.8	**75.5**	68.6	38.2	67.6
Biomedicine	Average	67.8	76.0	77.4	**80.8**	79.6	79.6	**81.0**	79.6	78.0	82.3
	BioQA	33.3	37.4	**43.4**	41.4	38.4	50.5	**55.6**	54.5	29.3	59.6
	ChemER	68.3	**93.2**	84.3	92.1	90.5	86.1	90.9	**91.4**	93.3	90.3
	DisER	80.8	82.2	80.9	87.2	**88.4**	79.8	**81.7**	80.9	90.5	81.1
	CompDis	67.7	70.3	**74.6**	73.8	74.5	**78.2**	76.3	74.0	71.6	72.6
	GeneFunc	70.9	79.9	88.8	**92.9**	89.6	**87.1**	85.5	81.3	88.8	96.4
	GeneReg	85.9	92.8	92.1	**97.1**	95.9	**95.9**	**95.9**	95.7	94.2	93.7
Drug Discovery	Average	25.4	30.2	32.0	31.7	**33.2**	41.5	35.5	**41.8**	31.0	43.4
	AffEx	1.1	4.6	**5.7**	4.3	3.0	3.5	**6.1**	5.4	8.1	31.4
	DrugQA	40.0	**53.3**	33.3	20.0	46.7	**40.0**	33.3	**40.0**	33.3	53.3
	TagMol	7.3	0.0	10.2	1.5	**20.1**	23.0	13.0	**27.1**	0.6	7.8
	MarkMol	28.4	15.9	18.0	**34.4**	32.8	**53.3**	31.7	52.4	48.8	63.8
	MolDoc	44.0	46.0	50.0	**56.0**	**56.0**	48.0	50.0	**58.0**	44.0	54.0
	ReactQA	25.3	25.3	**32.6**	28.4	26.3	**50.5**	36.8	32.6	34.7	37.9
	ResTarg	31.9	66.2	74.3	**77.0**	47.7	72.3	**77.3**	**77.3**	47.4	55.7
Organic Materials	Average	16.7	20.6	24.5	28.3	**38.9**	41.5	**52.7**	48.6	24.4	62.7
	ElecQA	26.0	20.0	**41.0**	30.0	28.0	33.0	**49.0**	33.0	26.0	68.0
	OLEDEx	1.8	**8.0**	6.5	7.2	6.8	16.1	**17.4**	11.0	13.5	27.9
	PolyQA	6.7	6.7	13.3	20.0	**93.3**	**80.0**	73.3	**80.0**	0.0	80.0
	PolyCompQA	23.9	25.7	35.8	32.1	**49.5**	53.2	73.4	**74.9**	33.0	82.6
	PolyPropEx	4.9	20.2	22.4	32.2	**43.4**	35.9	**54.4**	**54.4**	39.5	75.3
	SolEx	26.2	31.8	34.0	**35.7**	32.9	36.2	**42.3**	38.5	35.8	45.9
	ReactMechQA	27.3	31.8	18.2	**40.9**	18.2	36.4	**59.1**	48.2	22.7	59.1
Overall	**Average**	33.2	40.6	45.0	48.8	**52.4**	55.7	57.8	**59.5**	45.5	63.4

Table 7: Detailed model performance on SciAssess tasks. SciLitLLM-7B shows significant improvement in the Fundamental Science and Organic Materials domains while maintaining comparable performance in other domains. Overall, SciLitLLM-7B achieves approximately 3.6% improvement over the second-best LLM.