SciRIFF: A Resource to Enhance Language Model Instruction-Following over Scientific Literature

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**[♠♣]
Arman Cohan[♡♣]

[♣]Allen Institute for AI [♡]Yale University
[♠]University of Washington [◇]IIT Gandhinagar [♦]Hebrew University
[♥]Massachusetts Institute of Technology [♧]Northwestern University

`kejian.shi@yale.edu`

Abstract

We present SciRIFF (**Sci**entific **R**esource for **I**nstruction-**F**ollowing and **F**inetuning), a dataset of 137K instruction-following demonstrations for 54 tasks covering five essential scientific literature understanding capabilities: information extraction, summarization, question answering, claim verification, and classification. SciRIFF demonstrations are notable for their long input contexts, detailed task specifications, and complex structured outputs. While instruction-following resources are available in specific domains such as clinical medicine and chemistry, SciRIFF is the first dataset focused on extracting and synthesizing information from research literature across a wide range of scientific fields. To demonstrate the utility of SciRIFF, we develop a sample-efficient strategy to adapt a general instruction-following model for science by performing additional finetuning on a mix of general-domain and SciRIFF demonstrations. In evaluations on nine held-out scientific tasks, our model—called SciTülu—improves over a strong LLM baseline by 28.1% and 6.5% at the 7B and 70B scales respectively, while maintaining general instruction-following performance within 2% of the baseline. We are optimistic that SciRIFF will facilitate the development and evaluation of LLMs to help researchers navigate the ever-growing body of scientific literature. We release our dataset, model checkpoints, and data processing and evaluation code to enable further research.

`https://github.com/allenai/SciRIFF`
🤗 `https://huggingface.co/datasets/allenai/SciRIFF`

1 Introduction

Large language models (LLMs) have the potential to advance scientific progress by helping researchers navigate and draw insights from the scientific literature. To accomplish these tasks, LLMs must be able to reliably follow a range of *instructions*—e.g. to extract information, summarize content, or answer questions—when given research articles as input. These instructions will often feature long input contexts, such as an entire research article. In addition, the model's responses may need to be *structured* according to a specific format or schema that supports aggregation for literature

---

[*]Equal contribution.

Foundation Models for Science Workshop,38th Conference on Neural Information Processing Systems (NeurIPS 2024).

review [Marshall and Wallace, 2019], or is consumable by software components like augmented reading interfaces [Lo et al., 2023, Palani et al., 2023]. While bespoke models are available for specific scientific literature understanding tasks, models that can flexibly follow instructions are preferable both for their ease of use (offering a unified input / output interface) and for their ability to generalize to novel applications and settings.

The general instruction-following capabilities of LLMs have advanced rapidly in recent years, largely due to the availability of general-purpose instruction datasets [Zhang et al., 2023a]. In addition, some instruction-following resources are available for specific scientific and medical tasks, such as describing the properties of a molecule [Fang et al., 2024, Yu et al., 2024] or answering medical exam questions [Toma et al., 2023, Han et al., 2023] (see §5 for a review). However, there is a scarcity of resources aimed at enabling flexible scientific literature understanding capabilities across a range of domains.

In this work, we present SciRIFF (**Sci**entific **R**esource for **I**nstruction-**F**ollowing and **F**inetuning), a dataset to enable progress on instruction-following over scientific literature. SciRIFF includes 137K demonstrations for 54 tasks spanning five scientific literature understanding task categories: information extraction, summarization, question answering, claim verification, and classification. SciRIFF covers five scientific domains, ranging from artificial intelligence to clinical medicine (Figure 2). The tasks in SciRIFF are derived from existing scientific literature understanding datasets with human-annotated inputs and outputs, and are converted to a common instruction-following format via templates written by the paper authors (Figure 1). Many of the tasks feature long input contexts and require structured model responses.

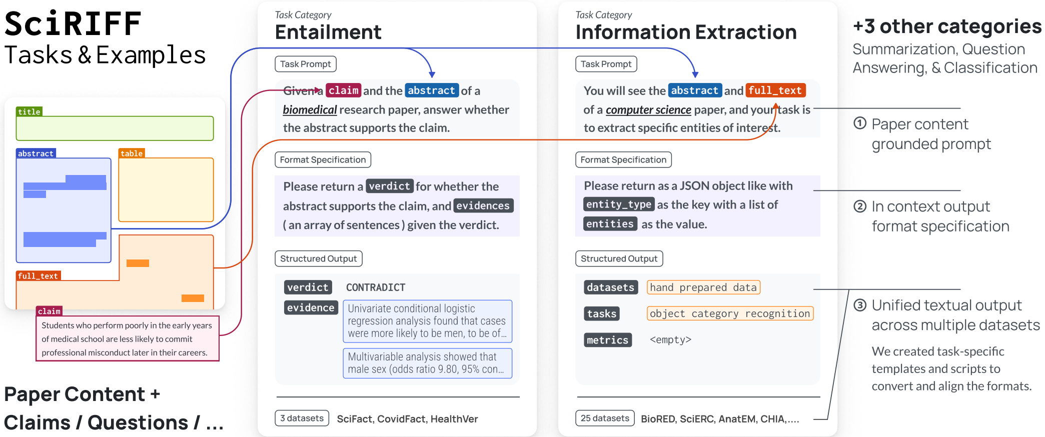
We hold out nine representative tasks from SciRIFF for use as an evaluation benchmark, which we call SciRIFF-Eval (§3.1). We then perform supervised finetuning experiments to identify a sample-efficient strategy to adapt Tülu V2 [Ivison et al., 2023]—a strong open instruction-following model—for scientific literature use cases. We find that, by starting from the original Tülu V2 model and performing additional finetuning on a downsampled mix of SciRIFF and data from the Tülu V2 Mix, we are able to match the performance achieved by training on all instances, while using less than 20% of the available data. Using this sample-efficient training strategy, we improve performance on SciRIFF-Eval by 28.1% over a directly comparable baseline at 7B scale, and by 6.5% at 70B scale. At the same time, we achieve performance within 2% of the baseline model on a general instruction-following benchmark (§4.1). We publicly release our 7B and 70B models, which we call SciTülu.

In summary, our contributions are as follows:

- We introduce SciRIFF, a dataset with 137K instruction-following demonstrations covering 54 literature understanding tasks spanning five scientific domains. Many tasks in SciRIFF feature long input contexts and require structured model responses.
- We employ a sample-efficient approach to adapt a family of general instruction-following models to scientific literature use cases. The resulting SciTülu models achieve substantial performance gains on held-out scientific tasks, without sacrificing general capabilities.
- We release the SciRIFF dataset, SciTülu model checkpoints, and code to recreate the dataset and perform evaluations on nine held-out tasks from SciRIFF.

2 SciRIFF

SciRIFF is a comprehensive instruction-tuning resource focused on real-world scientific literature understanding, consisting of 137k high-quality instructions derived from 54 datasets, which span five task categories and five subject domains. The primary design objective of SciRIFF is to enhance and evaluate instruction-following capabilities of LLMs in this specialized domain. Our focus is on document-grounded scientific literature understanding tasks, rather than tasks that evaluate general reasoning or mathematical problem-solving abilities without reference to scientific literature (e.g., MMLU [Hendrycks et al., 2021a]). In addition to coverage of a wide range of tasks, the instructions in SciRIFF often are grounded in long input contexts (i.e., scientific papers), and they support *structured* outputs according to a specific schema useful for tasks in literature understanding (such as relation extraction, fact checking with rationale selection, question answering with attribution, etc). The instances in SciRIFF are sourced from existing high-quality scientific datasets and converted into



Students who perform poorly in the early years of medical school are less likely to commit professional misconduct later in their careers.

Given a `claim` and the `abstract` of a *biomedical* research paper, answer whether the abstract supports the claim.

Please return a `verdict` for whether the abstract supports the claim, and `evidences` (an array of sentences) given the verdict.

Univariate conditional logistic regression analysis found that cases were more likely to be men, to be of...

Multivariable analysis showed that male sex (odds ratio 9.80, 95% con...

You will see the `abstract` and `full_text` of a *computer science* paper, and your task is to extract specific entities of interest.

Please return as a JSON object like with `entity_type` as the key with a list of `entities` as the value.

We created task-specific templates and scripts to convert and align the formats.

Figure 1: Example SciRIFF tasks. Given an input context from a research paper, the `text prompt` instructs an LLM to perform an operation on the input—e.g. determine whether the `abstract` entails a scientific `claim`, extract information over the `full_text`, answer a question, etc. The model's `output` must conform to a task-specific, user-specified `structure`. SciRIFF unifies 54 scientific literature understanding tasks under a common input / output format, enabling the development of LLMs that can flexibly generalize to novel scientific use cases.

instructions using human expert-written instruction templates. Below we discuss each of the steps involved in the creation of SciRIFF.

2.1 Dataset construction

We create SciRIFF by repurposing existing scientific literature understanding datasets for instruction-following—similar to resources like Flan [Longpre et al., 2023] and Super-NaturalInstructions [Wang et al., 2022]—rather than the alternative recent trend of generating synthetic data using an LLM (e.g., Köksal et al. [2023], Li et al. [2023]). We make this choice for two reasons. First, we believe it is sensible to exhaust available human-annotated resources, which we can be confident are correctly-annotated, before turning to potentially noisy synthetic data. Second, given the lack of existing evaluation benchmarks for scientific literature understanding, it may be difficult to assess the utility of synthetic data generation approaches. For this reason, we hold out nine SciRIFF tasks as an evaluation benchmark (§3.1), which we hope will provide valuable signal for future synthetic data generation efforts.

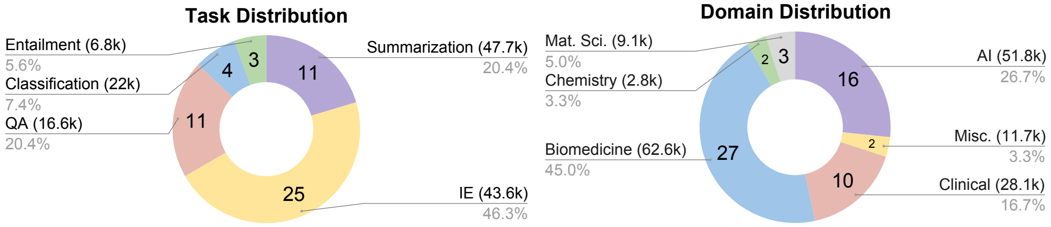
Dataset selection criteria In forming SciRIFF, we focus on scientific literature understanding tasks in which the model is given a portion of scientific text as input, and is instructed to produce output derived directly from the text. This task family includes summarization, reading comprehension, information extraction and other tasks, and is the most relevant setting for real-world use cases (e.g., meta-analysis of literature, clinical decision-making, augmented reading). We *exclude* datasets that require retrieval from document collections (e.g., open-domain QA) and datasets that assess reasoning and mathematical problem-solving skills without necessarily relying on scientific literature, such as ScienceQA [Lu et al., 2022], SciBench [Wang et al., 2023a], and MATH [Hendrycks et al., 2021b]. Additionally we only keep datasets that are publicly available and have a permissive license. Finally we include datasets that are well-documented and actively maintained. See Appendix A.1 for the complete task list.

Instruction templates We convert our 54 datasets into natural language input-output pairs suitable for instruction tuning using dataset-specific instruction templates[2] created by the paper authors, who are NLP experts.[3]

---

[2] We used a single template per dataset. Future work could explore the utility of multiple templates.

[3] We conducted initial experiments using an LLM to create templates, but found that the resulting instructions were often vague and did not clearly specify the desired output format.



(a) Task distribution

(b) Domain distribution

Figure 2: Distribution of task categories and domains in SciRIFF. The numbers in the pie charts indicate the number of datasets present in every task category/domain, while the numbers in brackets indicate the total number of instances per task category/domain.

We use `json` as the common output format for all structured tasks, which facilitates consistent evaluation and matches with larger industry trends to request JSON model outputs. Instruction templates are written in Jinja [Pallets, 2024]. Guidelines and best practices for prompt-writing are detailed in our GitHub repository. Each prompt was double-checked by an additional author for clarity and correctness.

2.2 Instruction Mix Statistics

Figures 2a and 2b present an overview of the SciRIFF training set distribution over task categories and domains respectively. Given the significant presence of information extraction tasks, it is unsurprising that a large percentage of datasets in SciRIFF (34 datasets; 63%) require structured outputs.

We construct three instruction mixes from this dataset collection, with maximum context lengths (input + output tokens) of 4,096, 8,192 and 16,382 per instance (longer instances are truncated where possible and discarded otherwise; see Appendix A.3). Due to model and hardware limitations, we conduct experiments in this work using the SciRIFF-4096 mixture, and make the longer mixtures available to enable future research. In what follows, we refer to SciRIFF-4096 simply as SciRIFF.

Input prompts in SciRIFF average over 1,200 tokens in length (Figure 5). As shown in Appendix A.2, this is over three times longer than the average prompt length for a range of representative general-purpose and domain-specific instruction datasets; for instance, Flan prompts average 350 tokens.

3 Experimental setup

We conduct finetuning experiments to determine the best way to use SciRIFF to improve LLM performance on scientific instruction-following.

3.1 Evaluation

Science evaluation We hold out 9 tasks from SciRIFF for evaluation, covering a representative range of task categories and scientific domains. The inputs, outputs, and evaluation metrics for each task are detailed in Table 1. Additional details along with full input / output examples for all evaluation tasks are included in Appendix D.

General instruction-following evaluation We re-use the instruction-following evaluations from Ivison et al. [2023]. These test a broad range of general abilities including world knowledge (MMLU), reasoning (GSM8k, Big Bench Hard), coding (CodexEval), open-ended generation (AlpacaEval), and truthfulness (TruthfulQA)[4].

---

[4] We consider the toxicity and multilinguality evaluations from Ivison et al. [2023] as out-of-scope for this work and do not evaluate on them; hence our reported averages are not directly comparable Ivison et al. [2023].

Name	Input	Output	Metrics
BioASQ List QA QA	Question over a set of paper excerpts	List of answer entities	Exact match F1
BioRED IE (NER)	Biomedical research abstract	Mentions of 6 biomedical entity types	Exact match F1
DiSCoMaT IE (Table extraction)	LaTex excerpt containing a table	Values of table entries	BLEU score vs. reference values
Evidence Inference (EI) IE (Rel. extraction)	Clinical trial report abstract	`(intervention, comparator, outcome, effect, evidence)` tuples	String overlap approximate match F1
MultiCite (MC) Classification	Citation context	List of expressed citation intents (7 categories)	Exact match F1
MuP Summarization	Machine learning paper full text	Short summary useful for a peer review	Similarity to reference summary, evaluated by LLM judge
Qasper QA	Question over an NLP paper full text	Answer to question / Paragraphs providing attribution	Similarity to reference answer, evaluated by LLM judge / Token F1 vs. reference paragraphs
SciERC IE (Rel. extraction)	Computer science abstract	Mentions of 6 computer science entity types	Exact match F1
SciFact Entailment	Claim to verify against a biomedical research abstract	Fact-checking verdict / Evidentiary sentences	Label F1 / Token F1 vs. reference evidence

Table 1: Evaluation tasks included in SciRIFF-Eval. "/" separators indicate two separate subtasks. We use GPT-3.5 Turbo as our LLM judge and evaluate similarity on a 1-5 scale; see Appendix D. "Rel. extraction" stands for "Relation extraction".

3.2 Finetuning approach

We experiment with two training data sources and two starting model checkpoints. In §4.2, we present ablation results examining the effects of both components on model performance.

Data sources We finetune on combinations of two datasets. (1) **SciRIFF**. We train on a fixed number of instances per task n_{sci} by sampling from each task at random; we use $n_{sci} = 1000$ for our final models based on the results of ablation experiments (§4.2). For tasks with fewer than n_{sci} instances, we include all examples. (2) **Tülu V2 Mix** [Ivison et al., 2023]. Tülu V2 Mix[5] is a high-quality general instruction-following dataset. It includes demonstrations from a range of sources, both human-written (e.g. Flan [Wei et al., 2022]) and distilled from proprietary LLMs (e.g. ShareGPT[6], Open Assistant[7]).

Starting checkpoints We conduct finetuning experiments starting from two model checkpoints: (1) **Llama 2**. We train on all available Tülu V2 Mix demonstrations, combined with n_{sci} instances per SciRIFF task. (2) **Tülu V2**. We start from a Tülu V2 checkpoint already trained on the full Tülu V2 Mix, and perform additional finetuning on n_{sci} instances per SciRIFF task,

Checkpoint	SciRIFF	Tülu-V2	Total
Llama 2 base	35,357	318,686	354,043
Tülu V2	35,357	35,357	70,714

Table 2: SciRIFF and Tülu V2 Mix[8] instances used for finetuning, with $n_{sci} = 1000$.

---

[5] We generally use "Tülu V2" to refer to the model and "Tülu V2 Mix" to refer to the dataset. In cases where it is clear from context, we may refer to the dataset simply as "Tülu V2".

[6] `https://sharegpt.com/`

[7] `https://github.com/LAION-AI/Open-Assistant`

[8] The original Tülu V2 Mix contains 326,154 examples, including 7.5K scientific literature understanding demonstrations. We remove these for our experiments.

Model	BioASQ	BioR	DiscMT	EI	MC	MuP	Qasper	SciERC	SciFact	Sci.	Gen.
GPT-3.5T	**47.3**	53.9	67.9	19.2	47.8	76.8	54.7 / 39.8	28.6	69.7 / 53.3	50.8	76.2
GPT-4	46.7	**61.0**	**78.3**	**24.7**	**58.7**	**86.9**	**67.8** / **50.5**	**42.2**	**84.3** / **68.7**	**60.9**	**88.8**
Llama 2 7B chat	34.2	0.0	4.8	7.4	37.8	72.0	15.7 / 8.5	0.3	27.7 / 6.2	19.5	36.4
Llama 3 8B instruct	43.3	40.3	37.3	13.5	**37.9**	**84.6**	41.1 / 25.9	25.4	42.3 / 40.1	39.2	**72.9**
TÜLU V2 7B	**44.5**	15.1	47.8	**15.1**	33.6	71.6	43.2 / 26.3	21.2	49.5 / 35.4	36.7	47.8
SCITÜLU 7B (ours)	37.5	**55.7**	**61.5**	11.6	34.6	72.1	**54.2** / **38.6**	**35.6**	**66.0** / **49.2**	**47.0**	47.5
TÜLU V2 70B	38.3	50.6	68.2	16.9	48.5	**64.9**	**49.1** / 20.7	32.5	**76.4** / **57.2**	47.6	**69.8**
SCITÜLU 70B (ours)	**42.7**	**69.3**	**72.6**	**17.5**	**62.8**	61.1	43.0 / **27.3**	**35.9**	70.7 / 55.6	**50.7**	68.3

Table 3: Performance of SCITÜLU. TÜLU V2 serves as a directly comparable baseline. Popular open (Llama) and closed (GPT) instruction-following models are also shown for reference. We report performance on each task in SCIRIFF-EVAL, along with the average performance on our general-purpose evaluations. The last two columns are averages. Complete results for general-purpose metrics can be found in Appendix C. Columns with a "/" indicate two evaluation metrics as described in §3.1. By training on SCIRIFF, SCITÜLU improves over TÜLU V2 by 28.1% at 7B, and 6.5% at 70B.

together with a matching number of instances randomly sampled from TÜLU V2 MIX. Including TÜLU V2 MIX instances prevents the model from "forgetting" its general instruction-following capabilities; see §4.2. Table 2 shows the number of training examples used to finetune each starting checkpoint.

SCITÜLU For our final 7B and 70B models, we use TÜLU V2 as our starting checkpoint and set n_{sci} = 1000. This approach leads to performance comparable to finetuning Llama base, while requiring 80% less data (Table 2). We call our final models SCITÜLU.

4 Results

We present the main results for SCITÜLU in §4.1. In §4.2, we perform ablations examining the effects of the different finetuning data sources and model checkpoints.

4.1 Main Results

Results for our final model, SCITÜLU, are shown in Table 3. TÜLU V2 serves as a directly comparable baseline. We also include the performance of widely used open (Llama 2 7B Chat and Llama 8B Instruct) and closed (GPT-3.5 and GPT-4[9]) LLMs.

SCIRIFF greatly improves scientific performance at 7B, with moderate improvement at 70B. SCITÜLU achieves a 28.1% average improvement on SCIRIFF-EVAL over TÜLU V2, while achieving nearly identical performance on the general evaluations. Llama 3 exhibits the strongest performance on the general evaluations but underperforms SCITÜLU on SCIRIFF-EVAL by 20%. Llama 2 has the lowest performance on both evaluations.[10] At the 70B scale, training on SCIRIFF provides a 6.5% average improvement on SCIRIFF-EVAL while causing a slight 2.1% decrease on general tasks. The smaller SCIRIFF-EVAL performance gain may be due to the stronger reasoning capabilities of 70B models, which can more readily perform complex tasks zero-shot.

SCITÜLU 7B matches TÜLU V2 70B on science. Given that many scientific literature understanding workflows (e.g. literature review) require running an LLM over large research corpora, the ability to run inference using a 7B model in place of a 70B could provide substantial efficiency gains and cost savings to practitioners.

---

[9] Predictions were made using `gpt-3.5-turbo-1106` and `gpt-4-turbo-0125-preview`.

[10] During manual inspection, we found that Llama 2 often struggles to generate properly-structured responses, leading to very low scores on some tasks. Future work could leverage tools like jsonformer to alleviate this issue.

Starting Checkpoint	**Finetuning data**	**Sci Avg.**	**Gen. Avg.**	**Sci Avg.**	**Gen. Avg.**
		Model size=7B		Model size=70B	
Llama 2 base	Tülu	36.7	47.8	47.5	69.8
	SciRIFF	**48.0**	23.9	**51.1**	44.2
	SciRIFF+Tülu	46.0	**48.9**	50.8	**70.4**
Tülu V2	SciRIFF	**47.0**	33.3	48.8	56.9
	SciRIFF+Tülu	**47.0**	**47.5**	**50.7**	**68.3**

Table 4: Ablations examining the effect of starting checkpoint and training data on scientific and general instruction-following. For **Data**, SciRIFF and Tülu each train on one of the two data sources described in §3.2, while SciRIFF + Tülu trains on both. All settings use n_{sci} = 1000. See Appendix C for individual task metrics.

Gains at 7B are largest on tasks requiring extraction and attribution. SciTülu 7B outperforms Tülu V2 most substantially on information extraction tasks (BioRED, DiSCoMaT, and SciERC)[11] and tasks requiring attribution (Qasper, SciFact), achieving a 51% average improvement. On the other hand, it performs worse on summarization (MuP) and question answering (BioASQ). This may occur because summarization and QA are well-covered by existing instruction-following resources, while tasks like information extraction and attribution have received less attention. It may also be an instance of *negative task transfer* [Jang et al., 2023, Asai et al., 2022], where training on many tasks at once can damage performance on a subset of target tasks. Based on these observations, we encourage practitioners using SciRIFF to conduct ablations to determine which training tasks are most beneficial for their particular use case.

GPT-4 achieves the strongest performance, but has room to improve. GPT-3.5 Turbo performs roughly on par with SciTülu on SciRIFF-Eval. Interestingly, GPT-4 performance is still relatively low on many tasks; this is due to a combination of task difficulty and evaluation challenges, which we discuss in §6.

4.2 Ablations

We conduct ablations to characterize the effects of (1) the inclusion of each data source, (2) the choice of starting checkpoint, and (3) the number of instances per SciRIFF task.

Table 4 reports average SciRIFF-Eval and general metrics for our two starting checkpoints using three data configurations: (1) Tülu trains on the Tülu V2 mix with no science instructions, (2) SciRIFF trains only on science instructions, and (3) SciRIFF + Tülu represents our final training mix described in §3.2.

Training on SciRIFF+Tülu gives the strongest overall performance Models trained only on SciRIFF perform well on science evaluations but struggle at general instruction-following. The Llama-2 checkpoint trained only on Tülu performs well on general tasks but is weaker on science. Training on the combined mix achieves the best overall performance on our general metrics and achieves comparable science performance to training only on

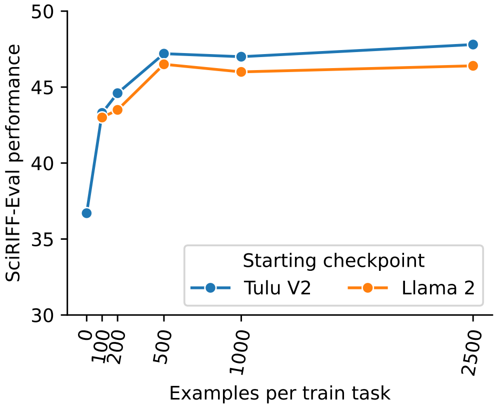


Figure 3: Performance on SciRIFF-Eval as a function of n_{sci}, the number of science instances per task. Performance gains largely saturate by n_{sci} = 1000. Experiments are done on 7B models.

---

[11] SciTülu does not improve on Evidence Inference, which is an IE task. This likely occurs because the task is very difficult (Appendix D) and evaluation is challenging (§6), making current evaluations unreliable. Improving modeling and evaluation for complex IE tasks represents important future work.

SciRIFF, with slight increases in some settings (e.g. 70B Tülu V2) and slight decreases in others (e.g. 7B Llama 2 base).

Starting from Tülu V2 performs comparably to Llama 2 base, while using 20% of the compute When trained on SciRIFF+Tülu data, models trained starting from Tülu V2 vs Llama 2 base are comparable. Tülu V2 is slightly better on science at 7B and nearly identical at 70B, while it underperforms Llama 2 by less than 3% on general evals. Given that finetuning Tülu V2 requires only 20% of the data (§3.2), we recommend that practitioners save on compute by instruction-tuning a strong instruction-following model on a mix of SciRIFF and general-domain instructions. This aligns with findings from prior works, e.g. Dong et al. [2024], Shi et al. [2023].

1,000 instances per science task is sufficient for peak performance Figure 3 shows that performance on SciRIFF-Eval increases sharply as n_{sci} rises from 100 to 500 and levels off subsequently. We set $n_{sci} = 1000$ based on these findings.

5 Related Work

Strategies for creation of instruction-following resources. Instruction tuning, or finetuning LLMs to improve their instruction-following ability, has emerged as a crucial technique for enhancing generalizability and controllability of LLMs [Wei et al., 2022, Sanh et al., 2022, Mishra et al., 2022, Ivison et al., 2023]. Several strategies have been explored for the creation of instruction-following resources, such as repurposing existing datasets using human-written instruction templates [Wei et al., 2022, Chung et al., 2024, Sanh et al., 2022, Mishra et al., 2022], crowdsourcing instructions [Databricks [2023], Zhou et al. [2023], ShareGPT[12]] and using LLMs to generate synthetic instructions data. As LLM capabilities grow stronger, synthetic instruction generation approaches, often including humans in the loop as correctors, have shown promising results. Broadly, these approaches use LLMs to either generate new dataset/task instances alongside instructions [Wang et al., 2023c, Xu et al., 2024, Nayak et al., 2024, Lou et al., 2024], or to "back-translate" existing datasets into instructions [Yin et al., 2023, Köksal et al., 2023, Li et al., 2023]. In this work, we create instructions using human-written templates (§2.1).

Instruction-following resources for scientific literature. While numerous open-domain instruction-following collections exist, resources for enhancing and evaluating LLMs' instruction-following capabilities on scientific literature are limited. Such resources are critical for developing models that can assist researchers and accelerate scientific discovery [Taylor et al., 2022, Xie et al., 2023]. Recent work has taken steps in this direction with the development of instruction-following datasets for specific domains such as mathematics [Yue et al., 2024a,b, Shao et al., 2024, Luo et al., 2023, Tang et al., 2024, Toshniwal et al., 2024], medicine [Parmar et al., 2022, Wu et al., 2024, Rohanian et al., 2023], chemistry [Yu et al., 2024, Zhang et al., 2024b], molecular biology [Fang et al., 2024, Tran et al., 2023], materials science [Song et al., 2023], and college-level foundational science [Zhang et al., 2024a]. Besides domain limitations, these resources primarily focus on improving LLMs' abilities to solve college-level science problems or reasoning tasks (see also, MMLU [Hendrycks et al., 2021a], SciEval [Sun et al., 2023], TheoremQA [Chen et al., 2023], SciBench [Wang et al., 2023a], and GPQA [Rein et al., 2023]). In contrast, SciRIFF both covers a broader set of scientific domains and focuses on document-grounded scientific literature understanding tasks that can power real-world scientific use cases. Another distinguishing factor of our work is our inclusion of tasks that require structured outputs, following a uniform JSON output format, besides text-to-text tasks. Some instruction-tuning resources have explored structured output formats [Zhang et al., 2023b, Wang et al., 2023b, Jiao et al., 2023, Gao et al., 2023], but not with a focus on scientific literature. Finally, most datasets in SciRIFF require long input contexts, leading to longer instruction contexts than prior work (see Appendix Table 5 for a comparison).

Other scientific literature benchmarks. In addition to instruction-following resources, prior works have also developed benchmarks to improve and assess scientific literature understanding. Notable efforts in the biomedical domain include BLUE [Peng et al., 2019], BLURB [Gu et al., 2021], InBoXBART [Parmar et al., 2022], and BigBio [Fries et al., 2022]; SciRIFF covers a broader set of domains than these resources. Other efforts such as SciRepEval [Singh et al., 2023], Galactica [Taylor et al., 2022], and AcademicGPT [Wei et al., 2023] cover domains beyond biomedicine, but are not suitably formatted for training or evaluating instruction-following models. SciASSESS [Cai

---

[12] `https://sharegpt.com/`

et al., 2024] evaluates LLMs' proficiency in scientific literature analysis, focusing on tasks like memorization and reasoning. Li et al. [2024] introduces a hybrid strategy that combines continual pretraining and supervised finetuning to specialize LLMs for scientific literature understanding. In contrast to these, SciRIFF provides both a benchmark and training resource for advancing LLMs in flexible instruction-following across scientific literature tasks.

6 Conclusion and Future Work

In this work, we introduced SciRIFF, a resource to facilitate progress on LLM instruction-following over scientific literature. We demonstrated that training on SciRIFF leads to improved performance on held-out scientific tasks, with especially large improvements at the 7B scale on tasks requiring structured extraction or attribution.

Our work points toward a number of future research directions. As observed in §4.1, GPT-4 performance on SciRIFF-Eval is fairly low. This is partly due to the difficulty of the tasks, but also due the challenges associated with evaluating structured LLM responses in cases where the predicted surface form does not match the reference, but the underlying meaning is the same [Wadhwa et al., 2023]. Utilizing LLMs to perform more flexible evaluations [Kim et al., 2024] represents a promising direction. Future work could also explore whether diversifying tasks by using multiple templates—potentially with different input and output formats—could enable models to learn from a larger number of demonstrations and to generalize more readily to unseen tasks. Synthetic data generation techniques may also be beneficial.

As presented in §2.2 the distribution of domains in SciRIFF concentrates heavily on AI and Bio / Clinical medicine. This reflects the availability of existing resources for scientific NLP. Looking forward, we hope that researchers will release new instruction-following resources covering a wider range of scientific domains, and we encourage the community to contribute novel tasks and datasets to SciRIFF as they become available.

In conclusion, we are optimistic that the SciRIFF data and evaluations, as well as the SciTülu models, will serve as valuable resources to build systems which can boost the productivity and creativity of scientific researchers.

References

A. Asai, M. Salehi, M. E. Peters, and H. Hajishirzi. Attempt: Parameter-efficient multi-task tuning via attentional mixtures of soft prompts. In *Conference on Empirical Methods in Natural Language Processing*, 2022. URL `https://api.semanticscholar.org/CorpusID:254125751`.

H. Cai, X. Cai, J. Chang, S. Li, L. Yao, C. Wang, Z. Gao, H. Wang, Y. Li, M. Lin, S. Yang, J. Wang, M. Xu, J. Huang, F. Xi, J. Zhuang, Y. Yin, Y. Li, C. Chen, Z. Cheng, Z. Zhao, L. Zhang, and G. Ke. Sciassess: Benchmarking llm proficiency in scientific literature analysis. *arXiv preprint arXiv: 2403.01976*, 2024.

W. Chen, M. Yin, M. Ku, P. Lu, Y. Wan, X. Ma, J. Xu, X. Wang, and T. Xia. TheoremQA: A theorem-driven question answering dataset. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7889–7901, Singapore, Dec. 2023. Association for Computational Linguistics.

H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

Databricks. Free dolly: Introducing the world's first truly open instruction-tuned llm. Blog post, 2023. URL `https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm`.

G. Dong, H. Yuan, K. Lu, C. Li, M. Xue, D. Liu, W. Wang, Z. Yuan, C. Zhou, and J. Zhou. How abilities in large language models are affected by supervised fine-tuning data composition, 2024.

Y. Fang, X. Liang, N. Zhang, K. Liu, R. Huang, Z. Chen, X. Fan, and H. Chen. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. In *International Conference on Learning Representations* (ICLR), 2024.

J. Fries, L. Weber, N. Seelam, G. Altay, D. Datta, S. Garda, S. Kang, R. Su, W. Kusa, S. Cahyawijaya, F. Barth, S. Ott, M. Samwald, S. Bach, S. Biderman, M. Sänger, B. Wang, A. Callahan, D. León Periñán, T. Gigant, P. Haller, J. Chim, J. Posada, J. Giorgi, K. R. Sivaraman, M. Pàmies, M. Nezhurina, R. Martin, M. Cullan, M. Freidank, N. Dahlberg, S. Mishra, S. Bose, N. Broad, Y. Labrak, S. Deshmukh, S. Kiblawi, A. Singh, M. C. Vu, T. Neeraj, J. Golde, A. Villanova del Moral, and B. Beilharz. Bigbio: A framework for data-centric biomedical natural language processing. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 25792–25806. Curran Associates, Inc., 2022.

C. Gao, W. Zhang, G. Chen, and W. Lam. Jsontuning: Towards generalizable, robust, and controllable instruction tuning. *arXiv preprint arXiv:2310.02953*, 2023.

Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23, Oct. 2021. ISSN 2637-8051.

T. Han, L. C. Adams, J.-M. Papaioannou, P. Grundmann, T. Oberhauser, A. Löser, D. Truhn, and K. K. Bressem. Medalpaca - an open-source collection of medical conversational ai models and training data. *ArXiv*, abs/2304.08247, 2023.

D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a.

D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021b.

H. Ivison, Y. Wang, V. Pyatkin, N. Lambert, M. Peters, P. Dasigi, J. Jang, D. Wadden, N. A. Smith, I. Beltagy, and H. Hajishirzi. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *arXiv preprint arXiv: 2311.10702*, 2023.

J. Jang, S. Kim, S. Ye, D. Kim, L. Logeswaran, M. Lee, K. Lee, and M. Seo. Exploring the benefits of training expert language models over instruction tuning. In *ICLR*, 2023.

Y. Jiao, M. Zhong, S. Li, R. Zhao, S. Ouyang, H. Ji, and J. Han. Instruct and extract: Instruction tuning for on-demand information extraction. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10030–10051, Singapore, Dec. 2023. Association for Computational Linguistics.

S. Kim, J. Shin, Y. Cho, J. Jang, S. Longpre, H. Lee, S. Yun, S. Shin, S. Kim, J. Thorne, and M. Seo. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*, 2024.

A. Köksal, T. Schick, A. Korhonen, and H. Schütze. Longform: Effective instruction tuning with reverse instructions. *arXiv preprint arXiv: 2304.08460*, 2023.

S. Li, J. Huang, J. Zhuang, Y. Shi, X. Cai, M. Xu, X. Wang, L. Zhang, G. Ke, and H. Cai. Scilitllm: How to adapt llms for scientific literature understanding. *arXiv preprint arXiv: 2408.15545*, 2024.

X. Li, P. Yu, C. Zhou, T. Schick, O. Levy, L. Zettlemoyer, J. Weston, and M. Lewis. Self-alignment with instruction backtranslation. *arXiv preprint arXiv: 2308.06259*, 2023.

K. Lo, J. C. Chang, A. Head, J. Bragg, A. X. Zhang, C. Trier, C. Anastasiades, T. August, R. Authur, D. Bragg, E. Bransom, I. Cachola, S. Candra, Y. Chandrasekhar, Y.-S. Chen, E. Y.-Y. Cheng, Y. Chou, D. Downey, R. Evans, R. Fok, F. Hu, R. Huff, D. Kang, T. S. Kim, R. M. Kinney, A. Kittur, H. B. Kang, E. Klevak, B. Kuehl, M. Langan, M. Latzke, J. Lochner, K. MacMillan, E. S. Marsh, T. C. Murray, A. Naik, N.-U. Nguyen, S. Palani, S. Park, C. Paulic, N. Rachatasumrit, S. R. Rao, P. Sayre, Z. Shen, P. Siangliulue, L. Soldaini, H. Tran, M. van Zuylen, L. L. Wang, C. Wilhelm, C. M. Wu, J. Yang, A. Zamarron, M. A. Hearst, and D. S. Weld. The semantic reader project: Augmenting scholarly documents through ai-powered interactive reading interfaces. *ArXiv*, abs/2303.14334, 2023.

S. Longpre, L. Hou, T. Vu, A. Webson, H. W. Chung, Y. Tay, D. Zhou, Q. V. Le, B. Zoph, J. Wei, and A. Roberts. The flan collection: Designing data and methods for effective instruction tuning. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR, 23-29 Jul 2023.

R. Lou, K. Zhang, J. Xie, Y. Sun, J. Ahn, H. Xu, Y. su, and W. Yin. MUFFIN: Curating multi-faceted instructions for improving instruction following. In *The Twelfth International Conference on Learning Representations*, 2024.

P. Lu, S. Mishra, T. Xia, L. Qiu, K.-W. Chang, S.-C. Zhu, O. Tafjord, P. Clark, and A. Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

H. Luo, Q. Sun, C. Xu, P. Zhao, J. Lou, C. Tao, X. Geng, Q. Lin, S. Chen, and D. Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv: 2308.09583*, 2023.

I. J. Marshall and B. C. Wallace. Toward systematic review automation: a practical guide to using machine learning tools in research synthesis. *Systematic Reviews*, 8, 2019.

S. Mishra, D. Khashabi, C. Baral, and H. Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland, May 2022. Association for Computational Linguistics.

N. V. Nayak, Y. Nan, A. Trost, and S. H. Bach. Learning to generate instruction tuning datasets for zero-shot task adaptation. *arXiv preprint arXiv: 2402.18334*, 2024.

S. Palani, A. Naik, D. Downey, A. X. Zhang, J. Bragg, and J. C. Chang. Relatedly: Scaffolding literature reviews with existing related work sections. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023.

Pallets. Jinja. `https://github.com/pallets/jinja/`, 2024. GitHub repository.

M. Parmar, S. Mishra, M. Purohit, M. Luo, M. Mohammad, and C. Baral. In-BoXBART: Get instructions into biomedical multi-task learning. In M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 112–128, Seattle, United States, July 2022. Association for Computational Linguistics.

Y. Peng, S. Yan, and Z. Lu. Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, 2019.

D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv: 2311.12022*, 2023.

O. Rohanian, M. Nouriborji, and D. A. Clifton. Exploring the effectiveness of instruction tuning in biomedical language processing. *arXiv preprint arXiv: 2401.00579*, 2023.

V. Sanh, A. Webson, C. Raffel, S. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, A. Raja, M. Dey, M. S. Bari, C. Xu, U. Thakker, S. S. Sharma, E. Szczechla, T. Kim, G. Chhablani, N. Nayak, D. Datta, J. Chang, M. T.-J. Jiang, H. Wang, M. Manica, S. Shen, Z. X. Yong, H. Pandey, R. Bawden, T. Wang, T. Neeraj, J. Rozen, A. Sharma, A. Santilli, T. Fevry, J. A. Fries, R. Teehan, T. L. Scao, S. Biderman, L. Gao, T. Wolf, and A. M. Rush. Multitask Prompted Training Enables Zero-Shot Task Generalization. In *International Conference on Learning Representations* (ICLR), 2022.

Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. Zhang, Y. Li, Y. Wu, and D. Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024.

C. Shi, Y. Su, C. Yang, Y. Yang, and D. Cai. Specialist or generalist? instruction tuning for specific NLP tasks. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15336–15348, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.947.

A. Singh, M. D'Arcy, A. Cohan, D. Downey, and S. Feldman. SciRepEval: A multi-format benchmark for scientific document representations. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5548–5566, Singapore, Dec. 2023. Association for Computational Linguistics.

Y. Song, S. Miret, H. Zhang, and B. Liu. HoneyBee: Progressive instruction finetuning of large language models for materials science. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5724–5739, Singapore, Dec. 2023. Association for Computational Linguistics.

L. Sun, Y. Han, Z. Zhao, D. Ma, Z.-W. Shen, B. Chen, L. Chen, and K. Yu. Scieval: A multi-level large language model evaluation benchmark for scientific research. *AAAI Conference on Artificial Intelligence*, 2023.

Z. Tang, X. Zhang, B. Wang, and F. Wei. Mathscale: Scaling instruction tuning for mathematical reasoning. *arXiv preprint arXiv: 2403.02884*, 2024.

R. Taylor, M. Kardas, G. Cucurull, T. Scialom, A. Hartshorn, E. Saravia, A. Poulton, V. Kerkez, and R. Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv: 2211.09085*, 2022.

A. Toma, P. R. Lawler, J. Ba, R. G. Krishnan, B. Rubin, and B. Wang. Clinical camel: An open-source expert-level medical language model with dialogue-based knowledge encoding. *ArXiv*, abs/2305.12031, 2023.

S. Toshniwal, I. Moshkov, S. Narenthiran, D. Gitman, F. Jia, and I. Gitman. Openmathinstruct-1: A 1.8 million math instruction tuning dataset. *arXiv preprint arXiv: Arxiv-2402.10176*, 2024.

H. Tran, Z. Yang, Z. Yao, and H. Yu. Bioinstruct: Instruction tuning of large language models for biomedical natural language processing. *JAMIA Journal of the American Medical Informatics Association*, 2023.

S. Wadhwa, S. Amir, and B. C. Wallace. Revisiting relation extraction in the era of large language models. *Proceedings of the conference. Association for Computational Linguistics. Meeting*, 2023: 15566–15589, 2023.

X. Wang, Z. Hu, P. Lu, Y. Zhu, J. Zhang, S. Subramaniam, A. Loomba, S. Zhang, Y. Sun, and W. Wang. SCIBENCH: Evaluating college-level scientific problem-solving abilities of large language models. In *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*, 2023a.

X. Wang, W. Zhou, C. Zu, H. Xia, T. Chen, Y. Zhang, R. Zheng, J. Ye, Q. Zhang, T. Gui, J. Kang, J. Yang, S. Li, and C. Du. Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv: 2304.08085*, 2023b.

Y. Wang, S. Mishra, P. Alipoormolabashi, Y. Kordi, A. Mirzaei, A. Naik, A. Ashok, A. S. Dhanasekaran, A. Arunkumar, D. Stap, E. Pathak, G. Karamanolakis, H. Lai, I. Purohit, I. Mondal, J. Anderson, K. Kuznia, K. Doshi, K. K. Pal, M. Patel, M. Moradshahi, M. Parmar, M. Purohit, N. Varshney, P. R. Kaza, P. Verma, R. S. Puri, R. Karia, S. Doshi, S. K. Sampat, S. Mishra, S. Reddy A, S. Patro, T. Dixit, and X. Shen. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics.

Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada, July 2023c. Association for Computational Linguistics.

J. Wei, M. Bosma, V. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le. Finetuned language models are zero-shot learners. *International Conference on Learning Representations*, 2022.

S. Wei, X. Xu, X. Qi, X. Yin, J. Xia, J. Ren, P. Tang, Y. Zhong, Y. Chen, X. Ren, Y. Liang, L. Huang, K. Xie, W. Gui, W. Tan, S. Sun, Y. Hu, Q. Liu, N. Li, C. Dai, L. Wang, X. Liu, L. Zhang, and Y. Xie. Academicgpt: Empowering academic research, 2023.

C. Wu, W. Lin, X. Zhang, Y. Zhang, W. Xie, and Y. Wang. PMC-LLaMA: toward building open-source language models for medicine. *Journal of the American Medical Informatics Association*, page ocae045, 04 2024. ISSN 1527-974X.

T. Xie, Y. Wan, W. Huang, Z. Yin, Y. Liu, S. Wang, Q. Linghu, C. Kit, C. Grazian, W. Zhang, I. Razzak, and B. Hoex. Darwin series: Domain specific large language models for natural science. *arXiv preprint arXiv: 2308.13565*, 2023.

C. Xu, Q. Sun, K. Zheng, X. Geng, P. Zhao, J. Feng, C. Tao, Q. Lin, and D. Jiang. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*, 2024.

D. Yin, X. Liu, F. Yin, M. Zhong, H. Bansal, J. Han, and K.-W. Chang. Dynosaur: A dynamic growth paradigm for instruction-tuning data curation. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4031–4047, Singapore, Dec. 2023. Association for Computational Linguistics.

B. Yu, F. N. Baker, Z. Chen, X. Ning, and H. Sun. Llasmol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset. *arXiv preprint arXiv: 2402.09391*, 2024.

X. Yue, X. Qu, G. Zhang, Y. Fu, W. Huang, H. Sun, Y. Su, and W. Chen. MAmmoTH: Building math generalist models through hybrid instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024a.

X. Yue, T. Zheng, G. Zhang, and W. Chen. Mammoth2: Scaling instructions from the web. *arXiv preprint arXiv: 2405.03548*, 2024b.

D. Zhang, Z. Hu, S. Zhoubian, Z. Du, K. Yang, Z. Wang, Y. Yue, Y. Dong, and J. Tang. Sciglm: Training scientific language models with self-reflective instruction annotation and tuning. *arXiv preprint arXiv: 2401.07950*, 2024a.

D. Zhang, W. Liu, Q. Tan, J. Chen, H. Yan, Y. Yan, J. Li, W. Huang, X. Yue, W. Ouyang, D. Zhou, S. Zhang, M. Su, H.-S. Zhong, and Y. Li. Chemllm: A chemical large language model. *arXiv preprint arXiv: 2402.06852*, 2024b.

S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu, and G. Wang. Instruction tuning for large language models: A survey. *arXiv preprint arXiv: 2308.10792*, 2023a.

T. Zhang, X. Yue, Y. Li, and H. Sun. Tablellama: Towards open large generalist models for tables. *arXiv preprint arXiv: 2311.09206*, 2023b.

X. Zhang, C. Tian, X. Yang, L. Chen, Z. Li, and L. R. Petzold. Alpacare:instruction-tuned large language models for medical application. *arXiv preprint arXiv: 2310.14558*, 2023c.

C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu, S. Zhang, G. Ghosh, M. Lewis, L. Zettlemoyer, and O. Levy. Lima: Less is more for alignment. In *Advances in Neural Information Processing Systems* (NeurIPS), 2023.

A Additional information on SciRIFF

A.1 SciRIFF task list

The full list of SciRIFF tasks is visualized in Figure 4. Detailed information on all tasks—including citations, URLs to source websites, and licensing information where available—is provided in our dataset card, available at `https://huggingface.co/datasets/allenai/SciRIFF`. Where convenient, we use datasets as preprocessed by the BigBio resource (`https://huggingface.co/bigbio`); details are in the dataset card.

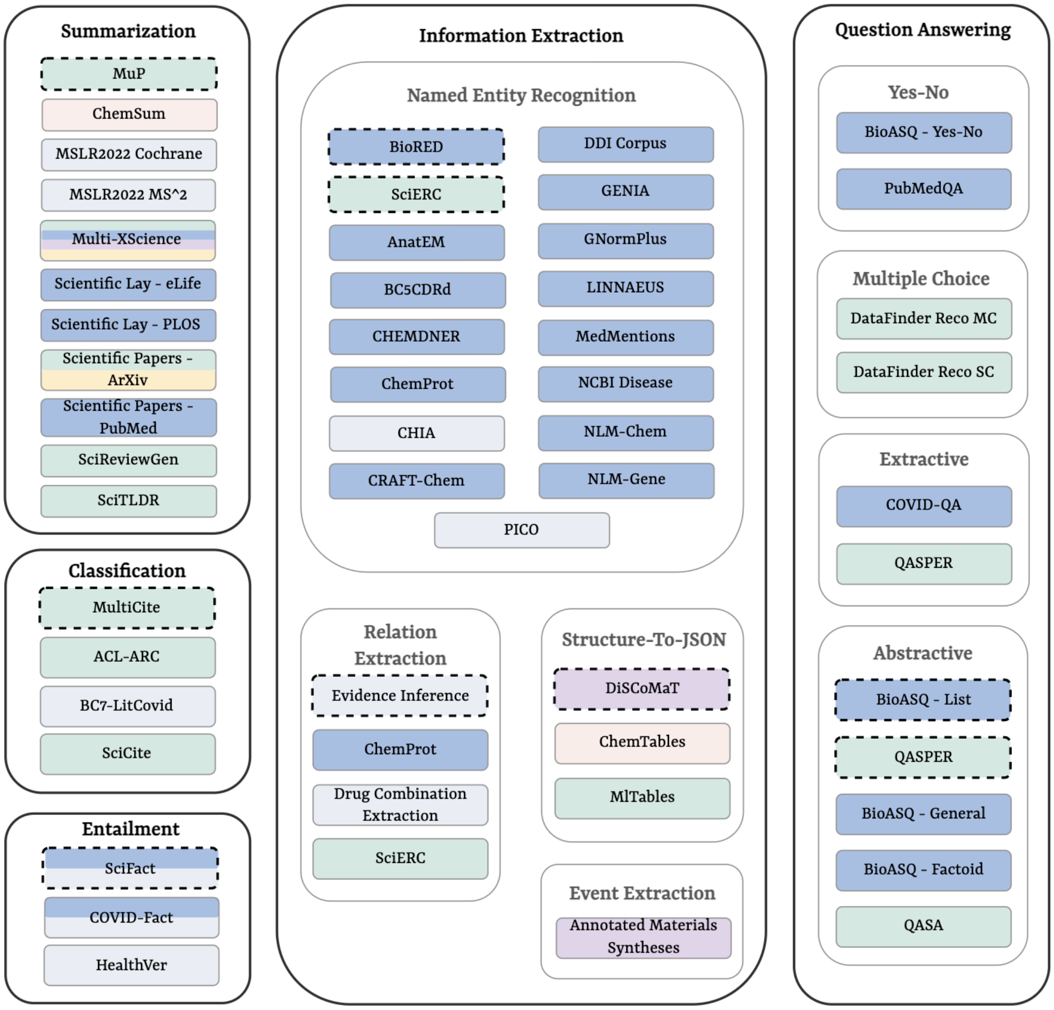


Figure 4: Overview of SciRIFF dataset. Dashed black lines indicate that a task is included in SciRIFF-Eval and held out during model training. Scientific domains are colored as follows: ■Biomedicine;■AI;■Clinical Medicine;■Chemistry;■Materials Science;■Miscellaneous.

A.2 Task length distribution

Figure 5 shows the distribution of input and output lengths for demonstrations in SciRIFF.

Table 5 compares SciRIFF with selected instruction-following datasets, including canonical collections commonly used for general fine-tuning and selected recent datasets specialized in scientific domains. Our dataset features longer input contexts than existing resources.

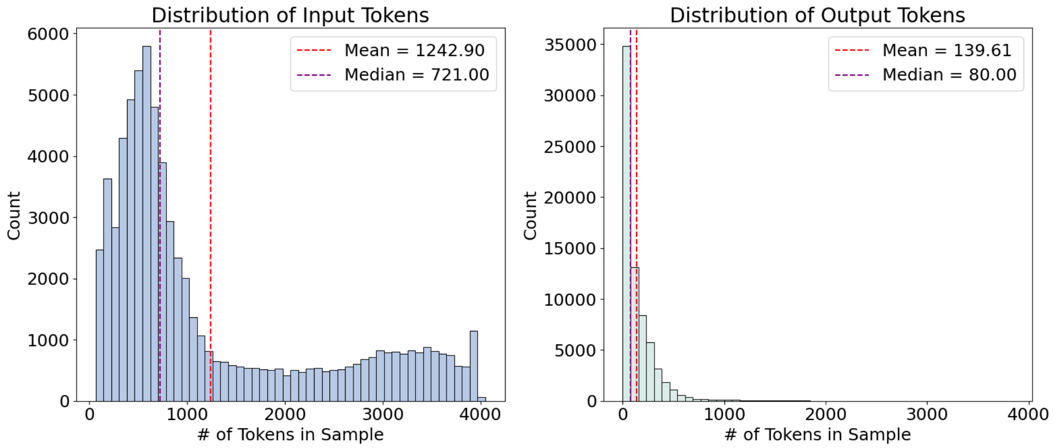


Figure 5: Distribution of input (left) and output (right) token lengths over SciRIFF training instances.

Name	# Instances	Domain	Avg. Length
General Domain			
Flan V2 [Chung et al., 2024]	15M	General	355.6 / 31.2
SuperNI [Wang et al., 2022]	97K	General	291.1 / 38.7
Tülu V2 Mix [Ivison et al., 2023]	326K	General	353.3 / 696.9
Scientific Domain			
BoX [Parmar et al., 2022]	141K	Biomed	X*
SciInstruct [Zhang et al., 2024a]	254K	Math, PH, Chem, FP	88.4 / 265.6
Mol-Instructions [Fang et al., 2024]	2.04M	Biomolecular	126.3 / 112.9
MathInstruct [Yue et al., 2024a]	262K	Math	82.5 / 174.0
MedInstruct-52K [Zhang et al., 2023c]	52K	Medical	148.2 / 96.9
LlaSMol [Yu et al., 2024]	3.29M	Chem	81.9 / 53.0
SciRIFF (Our work)	137K	AI, Biomed, Clinical, Chem, MatSci	**1242.9 / 139.6**

Table 5: Comparison with selected instruction-following datasets. We use the following abbreviations: PH – Physics; FP – Formal Proof; MatSci – Materials Science. We report average token counts for `input/output` using Llama 2 tokenizer using up to 200k subsamples from each dataset. *BoX dataset is not readily available.

A.3 Truncation strategy

In §2.2, we mention that when an instance exceeds the maximum context length for a given version of SciRIFF, we truncate where possible and discard otherwise. In particular, we truncate for tasks (like question answering) where the task output can be localized to particular passages in the input document by randomly removing irrelevant passages until the document fits in the desired context. For tasks like summarization, where the task output cannot easily be localized, we simply discard examples that are longer than the context window.

B Training Details

For instruction-tuning, our training hyperparameters were as follows:

- Precision: BFloat16
- Epochs: 2
- Weight decay: 0
- Warmup ratio: 0.03
- Learning rate: 2e-5 (1e-5 for 70B)
- Max. seq. length: 4,096
- Effective batch size: 128

All of our models were trained on v3-128 TPUs on the Google TPU Research Cloud.

C Full modeling results

Table 4 in §4.2 reports average science and general evaluation metrics for our ablations. Here, we include individual task metrics. Science metrics are shown in Table 6. General evaluation metrics are shown in Table 7.

Checkpoint	Data	BioASQ	BioRED	DiSCoMaT	Ev. Inf.	MultiCite	MUP	QASPER	SciERC	SciFact
					7B					
Llama 2	Tülu	**44.5**	15.1	47.8	**15.1**	33.6	71.6	43.2 / 26.3	21.2	49.5 / 35.4
Llama 2	SciRIFF	36.1	**61.0**	62.0	4.2	**54.4**	70.9	49.8 / **45.2**	35.6	60.9 / 47.6
Llama 2	SciRIFF+Tülu	27.0	50.1	60.6	11.8	44.1	70.9	50.0 / 44.4	34.5	65.2 / 47.4
Tülu V2	SciRIFF	39.1	55.6	**64.5**	0.0	52.2	67.8	48.6 / 38.2	**35.7**	**67.3** / 47.5
Tülu V2	SciRIFF+Tülu	37.5	55.7	61.5	11.6	34.6	**72.1**	**54.2** / 38.6	35.6	66.0 / **49.2**
					70B					
Llama 2	Tülu	38.3	50.6	68.2	16.9	48.5	64.9	**49.1** / 20.7	32.5	76.4 / 57.2
Llama 2	SciRIFF	39.2	68.1	71.6	14.2	**64.0**	62.5	43.9 / 27.1	**43.8**	73.3 / 54.5
Llama 2	SciRIFF+Tülu	36.5	59.7	**72.8**	**18.6**	63.7	64.9	45.1 / 26.6	37.0	77.5 / 56.9
Tülu V2	SciRIFF	12.5	66.8	69.1	14.4	56.8	**70.7**	41.1 / 25.0	41.0	**81.4** / **58.0**
Tülu V2	SciRIFF+Tülu	**42.7**	**69.3**	72.6	17.5	62.8	61.1	43.0 / **27.3**	35.9	70.7 / 55.6

Table 6: Model ablation results on science tasks.

		BBH	**Codex-eval**	**GSM8K**	**MMLU**	**TruthfulQA**	**AlpacaEval**	**Average**
Checkpoint	**Data**	**3-shot CoT**	**Pass@10**	**8-shot CoT**	**0-shot**	**Info * True**	**% Win vs Davinci-003**	**-**
				7B				
Llama 2	Tülu	**45.6**	**37.7**	29.5	**50.2**	50.4	73.1	47.8
Llama 2	SciRIFF	38.4	18.7	12.0	39.5	27.8	6.7	23.9
Llama 2	SciRIFF+Tülu	43.8	36.6	**34.0**	49.7	**54.3**	**74.7**	**48.9**
Tülu V2	SciRIFF	42.5	29.0	23.0	45.8	41.0	18.8	33.3
Tülu V2	SciRIFF+Tülu	42.7	35.9	29.5	49.6	52.6	**74.7**	47.5
				70B				
Llama 2	Tülu	66.8	**67.6**	66.5	**67.7**	**63.8**	**86.6**	69.8
Llama 2	SciRIFF	54.4	44.5	53.5	65.3	36.1	11.2	44.2
Llama 2	SciRIFF+Tülu	**68.1**	67.3	**73.5**	66.8	60.7	86.1	**70.4**
Tülu V2	SciRIFF	66.2	57.2	67.0	66.3	49.9	34.5	56.9
Tülu V2	SciRIFF+Tülu	67.0	60.3	67.5	66.8	62.2	85.8	68.3

Table 7: Model ablation results on general tasks.

D Evaluation details

The following pages show full input / output examples for all SciRIFF-Eval tasks, along with details on metric calculations. This information is also available on the project GitHub page. For tasks using an LLM judge, we found in preliminary experiments that the results of GPT-3.5 were similar to other proprietary LLMs like GPT-4 and Claude-2; we used GPT-3.5 in the interest of cost and efficiency.

Evaluation tasks

This doc has a list of all evaluation tasks, including input / output examples and evaluation metrics.

Table of contents

- BioASQ: question answering
- BioRED: named entity recognition
- Discomat: table extraction
- Evidence inference: evidence tuple extraction
- Multicite: citation intent classification
- MUP: summarization
- Qasper: paper question answering
- SciERC: named entity recognition
- SciFact: claim verification

BioASQ

- Task input: A collection of biomedical research excerpts and a question answerable from the excerpts.
- Task output: A list of answers to the question.
- Metrics: Compare predicted vs. reference answers using exact-match F1.

Input

```
Below are a collection of excerpts from biomedical research articles. Excerpts are separated by
newlines. Your task is to answer a question based these excerpts. Your response should be
formatted
as a `json` array.

For instance, given excerpts from articles studying breast cancer, and the question "what are some
common genes associated with breast cancer?", an answer might be formatted like: ["BRCA1",
"BRCA2",
"TP53", ...]. Only include answers that are mentioned in the provided exerpts. The array should
always have at least one answer; do not respond with an empty array []. Do not include any text in
your response other than the answer array.

Context: sensitization, behavioral changes, and low body mass index (BMI). One possible cellular
target that may mediate some of these findings is the hypocretin/orexin neurons. This neuronal
system plays a role in regulating wakefulness/sleep cycles, pain perception, and appetite. Food
intake, in contrast, receives circadian modulation through hormones such as leptin, ghrelin,
insulin
and orexin. A low level of hypocretin-1/orexin-A in the cerebrospinal fluid is sufficient to
diagnose narcolepsy type 1, being a highly specific and sensitive biomarker, and the irreversible
loss of hypocretin neurons is responsible for the main symptoms of the disease: Orexins, or
hypocretins, are excitatory neuropeptides involved in the regulation of feeding behavior and the
sleep and wakefulness states.

[Lines omitted for space]
```

```
Orexin A (OXA) and orexin B (OXB) are recently discovered neuropeptides that appear to play a role
in various distinct functions such as arousal and the sleep-wake cycle as well as on appetite and
regulation of feeding and energy homeostasis. Orexins were first described as neuropeptides
expressed by a sp Orexin/hypocretin neurons located in the lateral hypothalamus play a critical
role
in the maintenance of arousal and contribute to the regulation of multiple homeostatic and
behavioral processes.

Question: What processes do orexin/hypocretin neurons regulate?
```

Output

```
[
  "sleep",
  "appetite",
  "wakefullness",
  "pain",
  "reward",
  "energy homeostasis",
  "goal-directed behaviors",
  "Arousal",
  "addiction"
]
```

BioRed

- Task input: Abstract of a biomedical research article.
- Task output: All entities in the article of the following types:
  - cell line
  - chemical
  - disease
  - gene
  - gene variant
  - species
- Metrics: Compare predicted vs. reference entities using exact-match F1.

Input

```
You will be shown an abstract from a biomedical research paper. Given this abstract, your task is
to
extract all unique entities of the following types: ["Chemical", "Variant", "Gene", "CellLine",
"Disease", "Species"].

Please return the output as a JSON object of the format: {"CellLine": ["hRPTEC", ...], "Chemical":
["Glucose", ...], "Disease": ["Diabetes", ...], "Gene": ["HNF-6", ...], "Species": ["Patients",
...], "Variant": ["Pro75Ala", ...]}. The keys should be entity types and values should be lists of
extracted entities belonging to the corresponding type. If you cannot find entities belonging to a
specific type, the value should be [].
```

```
Only output the JSON object and do not include any additional text.

Abstract:

Fatal carbamazepine induced fulminant eosinophilic (hypersensitivity) myocarditis: emphasis on
anatomical and histological characteristics, mechanisms and genetics of drug hypersensitivity and
differential diagnosis. The most severe adverse reactions to carbamazepine have been observed in
the
haemopoietic system, the liver and the cardiovascular system. A frequently fatal, although
exceptionally rare side effect of carbamazepine is necrotizing eosinophilic (hypersensitivity)
myocarditis. We report a case of hypersensitivity myocarditis secondary to administration of
carbamazepine. Acute hypersensitivity myocarditis was not suspected clinically, and the diagnosis
was made post-mortem. Histology revealed diffuse infiltration of the myocardium by eosinophils and
lymphocytes with myocyte damage. Clinically, death was due to cardiogenic shock. To best of our
knowledge this is the second case of fatal carbamazepine induced myocarditis reported in English
literature.
```

Output

```
{
  "CellLine": [],
  "Chemical": ["carbamazepine"],
  "Disease": [
    "hypersensitivity",
    "death",
    "myocarditis",
    "cardiogenic shock",
    "drug hypersensitivity"
  ],
  "Gene": [],
  "Species": [],
  "Variant": []
}
```

Discomat

- Task input: A passage from a research paper including a table.
- Task output: The table, with each cell as a `json` line.
- Metrics: BLEU score between predicted and gold reference. Manual inspection showed that BLEU was pretty reliable for this task.

Input

```
| Sample no. | Ph, volume percent of crystals | Activation energy (kJ/mol) of the scale factor for
normalised frequency | ln(t 0, s) of the scale factor | G  unrelaxed shear modulus (GPa) |
| Glas 0     | 0                             | 137+-18
| -50.15                        | 24.3                             |
| Glas 1     | 17                            | 129+-13
| -47.68                         | 23.8                             |
| Glas 3     | 22                            | 126+-16
| -45.72                         | 24.7                             |
| Glas 5     | 27                            | 117+-10
| -42.25                         | 25.0                             |
Caption: Activation energies of shear stress relaxation and unrelaxed shear modulus of disilicate
lithium glasses

You are provided with a table from a material science paper. Here are JSON templates for two types
of numeric cells: "Other" and "Glass_Compound_Amount": {"value": "xx", "type": "Other"} {"value":
"xx", "type": "Glass_Compound_Amount", "constituent": "xx", "unit": "xx", "material": "xx"}

Please describe all numeric cells in the above table following the JSON templates (proceeding by
row
in a left-right, top-down direction). For each cell, output one JSON description per line. For any
unanswerable attributes in the templates, set their value to the placeholder "xx".

Cell Description:
```

Output

```
{"value": "0", "type": "Other"}
{"value": "137", "type": "Other"}
{"value": "24.3", "type": "Other"}
{"value": "17", "type": "Other"}
{"value": "129", "type": "Other"}
{"value": "23.8", "type": "Other"}
{"value": "22", "type": "Other"}
{"value": "126", "type": "Other"}
{"value": "24.7", "type": "Other"}
{"value": "27", "type": "Other"}
{"value": "117", "type": "Other"}
{"value": "25.0", "type": "Other"}
```

Evidence Inference

- Task input: Abstract of a clinical trial report.
- Task output: List of all ICO `(intervention / comparator / outcome)` tuples, together with the effect of the intervention on the outcome and the textual evidence of this effect.
- Metrics: "Fuzzy" F1. Given a prediction and a reference tuple, compute the token overlap for each tuple item. If token overlaps for all fields exceed 0.3, the predicted tuple is judged as a match to the reference.

Input

```
You will be shown the abstract of a medical clinical trial report. Your task is to extract all the
findings from this report into a JSON array. Each finding should contain the following five
elements:

- Intervention: The medical intervention being tested. This should be a text span copied from the
```

```
  input passage.
- Comparator: The baseline against which the intervention is being evaluated. This should be a
text
  span copied from the input passage. If no comparator is reported, set to `null`.
- Outcome: The medical outcome whose effect is being measured. This should be a text span copied
  from the input passage.
- Effect: The effect of the intervention on the outcome, relative to the comparator. The effect
  should be one of the following three values: ("significantly increased", "significantly
  decreased", "no significant difference").
- Evidence: The evidence for the effect. This should be a text span copied from the input passage.

Please format your output as a JSON array. Each entry in the output should be an array containing
the 5 elements listed above, in the following order: [<intervention>, <comparator>, <outcome>,
<effect>, <evidence>].

For example, an output with two findings might read: [["aspirin", "placebo", "headache severity",
"significantly decreased", "Mean headache severity was significantly decreased in the aspirin
group
compared to the placebo group (p < 0.05)."], ["aspirin", "placebo", "weight loss", "no significant
difference", "We did not observe any difference in weight loss between the group given aspirin
relative to the control group"]]

There are 3 finding(s) in the abstract below. Please extract them. Output only the JSON array with
these 3 findings. Do not include any additional text.

Abstract: ABSTRACT.OBJECTIVES: To compare the efficacy and safety of SB4 (an etanercept
biosimilar)
with reference product etanercept (ETN) in patients with moderate to severe rheumatoid arthritis
(RA) despite methotrexate (MTX) therapy.

ABSTRACT.METHODS: This is a phase III, randomised, double-blind, parallel-group, multicentre study
with a 24-week primary endpoint. Patients with moderate to severe RA despite MTX treatment were
randomised to receive weekly dose of 50 mg of subcutaneous SB4 or ETN. The primary endpoint was
the
American College of Rheumatology 20% (ACR20) response at week 24. Other efficacy endpoints as well
as safety, immunogenicity and pharmacokinetic parameters were also measured.

ABSTRACT.RESULTS: 596 patients were randomised to either SB4 (N=299) or ETN (N=297). The ACR20
response rate at week 24 in the per-protocol set was 78.1% for SB4 and 80.3% for ETN. The 95% CI
of
the adjusted treatment difference was -9.41% to 4.98%, which is completely contained within the
predefined equivalence margin of -15% to 15%, indicating therapeutic equivalence between SB4 and
ETN. Other efficacy endpoints and pharmacokinetic endpoints were comparable. The incidence of
treatment-emergent adverse events was comparable (55.2% vs 58.2%), and the incidence of antidrug
antibody development up to week 24 was lower in SB4 compared with ETN (0.7% vs 13.1%).

ABSTRACT.CONCLUSIONS: SB4 was shown to be equivalent with ETN in terms of efficacy at week 24. SB4
was well tolerated with a lower immunogenicity profile. The safety profile of SB4 was comparable
with that of ETN.

ABSTRACT.TRIAL REGISTRATION NUMBERS: NCT01895309, EudraCT 2012-005026-30.

Findings:
```

Output

```
[
  [
    "SB4 (an etanercept biosimilar)",
    "etanercept (ETN)",
    "therapeutic equivalence – Patients with moderate to severe rheumatoid arthritis (RA) despite
methotrexate (MTX) treatment",
    "no significant difference",
    "The 95% CI of the adjusted treatment difference was \u22129.41% to 4.98%, which is completely
contained within the predefined equivalence margin of \u221215% to 15%, indicating therapeutic
equivalence between SB4 and ETN."
  ],
  [
    "SB4 (an etanercept biosimilar)",
    "etanercept (ETN)",
    "incidence of antidrug antibody development up to week 24 – Patients with moderate to severe
rheumatoid arthritis (RA) despite methotrexate (MTX) treatment",
    "significantly decreased",
    "the incidence of antidrug antibody development up to week 24 was lower in SB4 compared with
ETN (0.7% vs 13.1%)."
  ],
  [
    "SB4 (an etanercept biosimilar)",
    "etanercept (ETN)",
    "incidence of treatment-emergent adverse events – Patients with moderate to severe rheumatoid
arthritis (RA) despite methotrexate (MTX) treatment",
    "no significant difference",
    "The incidence of treatment-emergent adverse events was comparable (55.2% vs 58.2%)"
  ]
]
```

Multicite

- Task Input: A citation sentence from a research paper.
- Task output: A list of intents for the citation sentence.
- Metrics: Compare predicted vs. reference intents using exact-match F1.

Input

```
Your task is to classify the citation intent within the following provided text from a
computational
linguistics research paper. The cited work is demarcated by "<cite>" and "</cite>". Determine the
purpose of the cited work by selecting from the listed categories:

- Background: The cited paper underpins the subject matter.
- Motivation: The cited paper inspires or provides a rationale for the current research.
- Uses: The current work utilizes concepts or tools from the cited paper.
- Extends: The current work advances ideas or methods from the cited paper.
- Similarities: The current work identifies commonalities with the cited paper.
- Differences: The current work delineates its distinction from the cited paper.
- FutureWork: The cited paper is acknowledged as groundwork for prospective research.

Indicate the intents by listing them in a `json` array, e.g. ["Background", "Uses"]. More than one
intent may be applicable. Do not include any extraneous text in your response.

Context with Citation: In addition to that, we implemented semi-supervised classification by
training in the positive samples of the <cite>[9]</cite> dataset and training in only the lexicon
as
negative samples.
```

Output

```
["Similarities", "Uses"]
```

MUP

- Task input: Full text of a machine learning paper.
- Task output: Short paper summary that a reviewer might write as part of a paper review.
- Metrics: Use GPT-3.5 to judge similarity of generated summary to human reference on 1-5 scale. Based on manual inspection, this was higher-quality than automated metrics like ROUGE.

Input

```
You will be presented with the title and body text of a computer science research paper. Please
write a summary of the work that would be informative for a peer reviewer assessing its quality.
Your summary should be 3 sentences long. In your response, include only the summary and no
additional text.

Paper title: Reinforcement Learning with Efficient Active Feature Acquisition

Paper body: 1 INTRODUCTION . Recently , machine learning models for automated sequential decision
making have shown remarkable success across many application areas , such as visual recognition (
Mathe et al. , 2016 ; Das et al. , 2017 ) , robotics control ( Finn et al. , 2016 ; Zhang et al. ,
2018 ) , medical diagnosis ( Ling et al. , 2017 ; Peng et al. , 2018 ) and computer games ( Mnih
et
al. , 2015 ; Silver et al. , 2016 ) . One fundamental reason that drives the success of such
models
and enables them to outperform classical algorithms is the availability of large amounts of
training
data . Typically such training data is either fully observed or the features stem from an
action-independent observation model ( which clearly can depend on the state of the system ) .
However , the fundamental assumption that the same features are always readily available during
deployment could not hold in many real-world applications . For instance , consider a medical
support system for monitoring and treating patients during their stay at hospital which was
trained
on rich historical medical data . To provide the best possible treatment , the system might need
to
```

```
perform several measurements of the patient over time , while some of them could be costly or even
pose a health risk . Therefore , during deployment , it is more ideal that the system could
function
with minimal features while during training more features might have been available . In such
cases
, we are interested in decision making models that actively take the measurement process , i.e. ,
feature acquisition , into account and only acquire the information relevant for making a decision
.
In this paper , we consider the challenging problem of learning effective policies when the cost
of
information acquisition can not be neglected . To be successful , we need to learn policies which
acquires the information required for solving a task in the cheapest way possible . [Truncated for
space].

3-sentence paper summary:
```

Output

```
In this paper the authors propose an approach for simultaneously learning how to explore more
efficiently in POMDPs via targeted feature acquisition, and learning a reward-maximizing control
policy, balancing the cost of feature acquisition with the expected reward. Learning is done via a
VAE framework which combines a belief inference model and an observation decoder, with a key
innovation being that inference is done as a sequential process. Results comparing this approach
to other variational inference approaches show the proposed framework reaches better performance
with lower cost (particularly, number of acquired features).
```

Qasper

- Task input: The full text of an NLP research paper, and a question answerable from the paper body (but not the abstract).
- Task output: An answer to the question, accompanied by the extracts from the paper body supplying the answer.
- Metrics: We compute metrics for both the answer and the evidence.
  - Answer: GPT-3.5 judge of similarity of model answer to human reference (1-5 scale).
  - Evidence: Token F1 overlap with gold evidence.

Input

```
You will be shown sections from a scientific research paper, together with a question about the
paper. Paragraphs in the paper are separated by newlines. Your task is to answer the question
based
on the contents of the paper.

Paper:
----------------------------------------
Named Entity Disambiguation for Noisy Text

We address the task of Named Entity Disambiguation (NED) for noisy text. We present WikilinksNED,
a
```

```
large-scale NED dataset of text fragments from the web, which is significantly noisier and more
challenging than existing news-based datasets. To capture the limited and noisy local context
surrounding each mention, we design a neural model and train it with a novel method for sampling
informative negative examples. We also describe a new way of initializing word and entity
embeddings
that significantly improves performance. Our model significantly outperforms existing
state-of-the-art methods on WikilinksNED while achieving comparable performance on a smaller
newswire dataset.

The WikilinksNED Dataset:             Entity Mentions in the Web We introduce WikilinksNED, a
large-scale NED dataset based on text fragments from the web. Our dataset is derived from the
Wikilinks corpus BIBREF14 , which was constructed by crawling the web and collecting hyperlinks
(mentions) linking to Wikipedia concepts (entities) and their surrounding text (context).
Wikilinks
contains 40 million mentions covering 3 million entities, collected from over 10 million web
pages.
Wikilinks can be seen as a large-scale, naturally-occurring, crowd-sourced dataset where thousands
of human annotators provide ground truths for mentions of interest. This means that the dataset
contains various kinds of noise, especially due to incoherent contexts. The contextual noise
presents an interesting test-case that supplements existing datasets that are sourced from mostly
coherent and well-formed text.

[Truncated for space]
----------------------------------------

Question: How was a quality control performed so that the text is noisy but the annotations are
accurate?

To answer the question, format your response as a `json` object with two fields:

"answer": A string providing a succinct answer to the question, in your own words. "evidence": An
array of strings. Each entry should be a full paragraph from the paper. Together, the evidence
should serve as a justification for the answer.

For instance, for the question "What baselines did the authors compare against?", a sample
response
might be:

{ "answer": "BERT and RoBERTa." "evidence": ["We compare our approach against two baselines. In
  Table 1, we compare against BERT. In Table 2, we compare against RoBERTa. Our findings indicate
  that our approach improves over both baeslines..."] }

The "answer" field should be roughly 190 characters in length.

Do not include any text in your response other than the json. If the question is unanswerable
given
the provided excerpts, respond with the single word "null".

To repeat, the question is: How was a quality control performed so that the text is noisy but the
annotations are accurate?

Answer JSON object:
```

Output

```
{
  "answer": "Profile pictures from the Twitter users' profiles.",
  "evidence": [
    "The recent advancements in deep neural networks, specifically for image analysis task, can
lead to determining demographic features such as age and gender BIBREF13 . We show that by
determining and integrating heterogeneous set of features from different modalities \u2013
aesthetic features from posted images (colorfulness, hue variance, sharpness, brightness,
blurriness, naturalness), choice of profile picture (for gender, age, and facial expression), the
screen name, the language features from both textual content and profile's description (n-gram,
emotion, sentiment), and finally sociability from ego-network, and user engagement \u2013 we can
reliably detect likely depressed individuals in a data set of 8,770 human-annotated Twitter
users."
  ]
}
```

SciERC

- Task input: An abstract of an NLP paper.
- Task output: A list of all entities mentioned in the paper of the following types:
    - Material
    - Method
    - Metric
    - Task
    - Generic
    - Other scientific term
- Metrics: Exact-match F1.

Input

```
You will be shown an abstract from a computer science research paper. Given this abstract, your
task
is to extract all unique entities with the following types:

- "Task": Applications, problems to solve, systems to construct. Examples include "information
  extraction", "machine reading system", "image segmentation".
- "Method": : Methods, models, systems to use, or tools, components of a system, frameworks.
  Examples include "language model", "CORENLP", "POS parser".
- "Metric": Metrics, measures, or entities that can express quality of a system / method. Examples
  include "F1", "BLEU", "Precision", "time complexity".
- "Material": Data, datasets, resources, Corpus, Knowledge base. Examples include "image data",
  "speech data", "stereo images", "CoNLL", "Wikipedia".
- "OtherScientificTerm": Phrases that are a scientific terms but do not fall into any of the above
  classes. Examples include "physical or geometric constraints", "qualitative prior knowledge",
  "tree kernel", "noise".
- "Generic": General terms or pronouns that may refer to a entity but are not themselves
  informative, often used as connection words. Examples include "model", "approach", "them".
```

```
Please return the output as a JSON object of the format: {"type1" : ["example_entity", ...],
"type2"
: ["example_entity", ...]}. The keys should be entity types and values should be lists of
extracted
entities belonging to the corresponding type. Entity types with no matching entities should be
assigned an empty array [].

For instance, the output might look like: {"Task": ["speech recognition", ...], "Method":
["Conditional random field"], "Material": [], ...}.

Only output the JSON object and do not include any additional text.

Abstract:

We present a syntax-based constraint for word alignment, known as the cohesion constraint. It
requires disjoint English phrases to be mapped to non-overlapping intervals in the French
sentence.
We evaluate the utility of this constraint in two different algorithms. The results show that it
can
provide a significant improvement in alignment quality.
```

Output

```
{
  "Generic": ["algorithms"],
  "Material": ["English phrases", "French sentence"],
  "Method": [],
  "Metric": ["alignment quality"],
  "OtherScientificTerm": ["cohesion constraint", "syntax-based constraint"],
  "Task": ["word alignment"]
}
```

SciFact

- Task input: An abstract from a biomedical research article, and a scientific claim.
- Task output:
    - A fact-checking verdict indicating whether the abstract supports or refutes the claim, or has no relevant information.
    - The evidence -- i.e. sentences from the abstract justifying the verdict.
- Metrics: We compute metrics for both the answer and the evidence.
    - Verdict: Label F1.
    - Evidence: Token F1 overlap with gold evidence.

Input

```
You will be shown a scientific claim, and the abstract of a biomedical research paper. Each
sentence
from the abstract will be on a separate line. Your task is to return a JSON object with two
```

```
fields:

- "verdict": The fact-checking verdict. If the information in the abstract supports the claim,
write
  "SUPPORT". If the abstract contradicts the claim, write "CONTRADICT". If the abstract does not
  provide enough information to arrive at a verdict, write "NEI" (for "not enough information").
- "evidence": An array of sentences providing evidence for the verdict. Please copy all relevant
  sentences verbatim from the abstract. If the verdict was "NEI", then return an empty array.

For instance, if the model were given the claim "smoking causes cancer", the output might be {
"verdict": "SUPPORT", "evidence": ["The results of our meta-analysis provide overwhelming support
  that cigarette smoking is a risk cause for lung cancer."] }

Your response should not include any text other than the json.

Claim: Therapeutics receiving accelerated approval encounter a lower frequency of post-marketing
safety events

Abstract: Importance Postmarket safety events of novel pharmaceuticals and biologics occur when
new
safety risks are identified after initial regulatory approval of these therapeutics. These safety
events can change how novel therapeutics are used in clinical practice and inform patient and
clinician decision making. Objectives To characterize the frequency of postmarket safety events
among novel therapeutics approved by the US Food and Drug Administration (FDA), and to examine
whether any novel therapeutic characteristics known at the time of FDA approval were associated
with
increased risk. [Truncated for space] Biologics, psychiatric therapeutics, and accelerated and
near–regulatory deadline approval were statistically significantly associated with higher rates of
events, highlighting the need for continuous monitoring of the safety of novel therapeutics
throughout their life cycle.
```

Output

```
{
  "verdict": "CONTRADICT",
  "evidence": [
    "In multivariable analysis, postmarket safety events were statistically significantly more
frequent among biologics (incidence rate ratio [IRR] = 1.93; 95% CI, 1.06–3.52; P = .03),
therapeutics indicated for the treatment of psychiatric disease (IRR = 3.78; 95% CI, 1.77–8.06; P
< .001), those receiving accelerated approval (IRR = 2.20; 95% CI, 1.15–4.21; P = .02), and those
with near\u2013regulatory deadline approval (IRR = 1.90; 95% CI, 1.19–3.05; P = .008); events were
statistically significantly less frequent among those with regulatory review times less than 200
days (IRR = 0.46; 95% CI, 0.24–0.87; P = .02)."
  ]
}
```