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

David Wadden[*†1] **Kejian Shi**[*†2] **Jacob Morrison**[†1,3] **Alan Li**[†2] **Aakanksha Naik**[†1]
Shruti Singh[2] **Nitzan Barzilay**[4,1] **Kyle Lo**[1,3] **Tom Hope**[4,1] **Luca Soldaini**[1]
Shannon Zejiang Shen[5] **Doug Downey**[1,6] **Hannaneh Hajishirzi**[1,3] **Arman Cohan**[2,1]

[1]Allen Institute for AI [2]Yale University
[3]University of Washington [4]Hebrew University [5]MIT [6]Northwestern University

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

Abstract

We present SciRIFF (**Sci**entific **R**esource for **I**nstruction-**F**ollowing and **F**inetuning), a dataset of 137K instruction-following instances for training and evaluation, covering 54 tasks. These tasks span five core scientific literature understanding capabilities: information extraction, summarization, question answering, claim verification, and classification. SciRIFF is unique in being entirely expert-written, high-quality instruction-following dataset for extracting and synthesizing information from research literature across diverse scientific fields. It features complex instructions with long input contexts, detailed task descriptions, and structured outputs. To demonstrate its utility, we finetune a series of large language models (LLMs) using a mix of general-domain and SciRIFF instructions. On nine out-of-distribution held-out tasks (referred to as SciRIFF-Eval), LLMs finetuned on SciRIFF achieve 70.6% average improvement over baselines trained only on general-domain instructions. SciRIFF facilitates the development and evaluation of LLMs to help researchers navigate the rapidly growing body of scientific literature.

1 Introduction

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 are often grounded in entire scientific articles, featuring longer inputs than other typical instruction-following resources in the science domain. In addition, the model's responses may need to be *structured* according to a specific format or schema that supports aggregation for literature review (Marshall and Wallace, 2019), or is consumable by software components like augmented reading interfaces (Lo et al., 2023; Palani et al., 2023). For example, when analyzing clinical trials, responses should follow a PICO framework (Population, Intervention, Comparison, Outcome), or when examining methodology papers, follow a standardized format capturing study design, sample size, statistical methods, and key findings, or when performing question answering or fact checking, accompany appropriate evidence for attribution and verification. Such outputs can be represented as `json` to ensure structured, consistent formatting that enhance both human readability and seamless machine processing (e.g., for claim verification and the input claim `“Coffee consumption reduces diabetes risk”`, the response could be `{ “verdict”: “support”, “evidence”: [“Study A shows 23% risk reduction”, “Meta-analysis B confirms protective effect”], “confidence”: “moderate”}`).

While bespoke models are available for specific scientific literature understanding tasks, models that can flexibly follow instructions in domain-specific settings of science 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 within that domain.

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.,

[*]Equal contribution. Full author contributions here. Correspondence to: {kejian.shi,arman.cohan}@yale.edu
[†]Denotes core contributors.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 6072–6109
November 4-9, 2025 ©2025 Association for Computational Linguistics

2024; Yu et al., 2024) or answering medical exam questions (Toma et al., 2023; Han et al., 2023) (see §5 for a review). However, few resources are available for supporting instruction-following for 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 comprehensive dataset to enable progress on instruction-following over scientific literature. SciRIFF includes 137K demonstrations for 54 tasks A.1 spanning five broad scientific literature understanding task categories: information extraction, summarization, question answering, claim verification, and classification.

SciRIFF covers five scientific domains, ranging from AI to clinical medicine (Figure 2).

Unlike synthetic or LLM-distilled instruction-following data (e.g., Lambert et al., 2024a), we prioritize human-annotated data to better capture nuanced domain expertise, complex structures, and reasoning required for scientific tasks. Additionally, existing datasets undergo individualized, manually written processes for data conversion to diverse instructions and undergo expert verification, ensuring accuracy and reliability (§2.1).

Our resource is a unique and specialized instruction-following meta-dataset. As illustrated in Figure 1 and with sample prompt templates provided in Appendix F, it is characterized by: (1) grounding every instance in scientific articles or texts, (2) requiring structured and complex responses, such as answers paired with attributions (i.e., tracing the source of the answer), and (3) featuring longer input contexts compared to most existing resources in the science domain (see Figure 5 and Table 7 in Appendix A).

All instruction templates are created by experts (authors of the paper) to ensure quality. Our experiments (§4) show that simple templates—similar to those used in prior work such as FlanV2 (Chung et al., 2024) or generated by an LLM (GPT-4o)—do not capture the complexity of our tasks. As a result, models finetuned on these instructions perform substantially worse than those using our expert-crafted instructions.

We also present a new benchmark dataset SciRIFF-Eval (§3.1) for evaluating instruction-following capabilities of LLMs in the science domain. Specifically, we hold out nine datasets from SciRIFF as an unseen evaluation benchmark which covers a representative range of skills and tasks. To demonstrate the utility of SciRIFF in improving scientific literature instruction following, we perform supervised finetuning experiments on several LLMs ranging different sizes.[1] When finetuned on a mix of SciRIFF and general open-source instruction-following data (i.e., Tülu v2 (Ivison et al., 2023a)), our models show consistent improvements on SciRIFF-Eval compared to training on general-domain instructions alone. Our evaluation tasks test true out-of-distribution generalization with formats and templates entirely excluded from training.

In summary, our contributions are as follows:

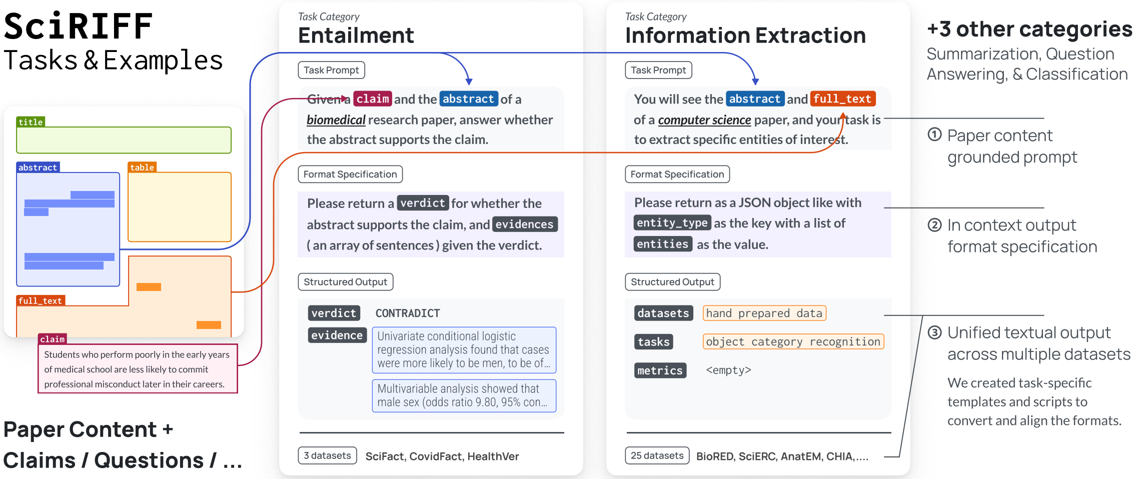
- We introduce SciRIFF, a high-quality and comprehensive resource for instruction-following in the science domain, containing 137K instances covering a wide range of tasks.
- We present SciRIFF-Eval, a diverse evaluation suite in scientific literature understanding (4.1K selected instances from unseen tasks).
- We release a range of LLMs finetuned on SciRIFF, achieving substantial improvements in scientific literature instruction-following, and conduct experiments showing insights on training strategy and instruction data scaling.
- We release SciRIFF dataset, evaluation suite SciRIFF-Eval, model checkpoints, and code to enable the community to reproduce our results and contribute to task sourcing for broader coverage.

2 SciRIFF

SciRIFF is a comprehensive instruction-following resource for real-world scientific literature understanding, with 137K instructions for training and in-domain validation. In addition, the test set SciRIFF-Eval includes 4.1K instances from held-out tasks. Our resource spans five task categories and subjects, (Figure 1), with particular emphasis on attribution and evidence in scientific tasks. Many tasks require models not only to provide answers but also to support them with evidence from the source paper to ensure verifiable outputs.

Our focus is on *document-grounded* scientific literature understanding tasks, rather than tasks that evaluate scientific knowledge recall (Feng et al., 2024), or general mathematical, problem-solving abilities without reference to scientific literature (e.g., SciInstruct (Zhang et al., 2024a), MMLU (Hendrycks et al., 2021a)). In addition

[1] Other types of post-training such as preference optimization are outside our scope.



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.

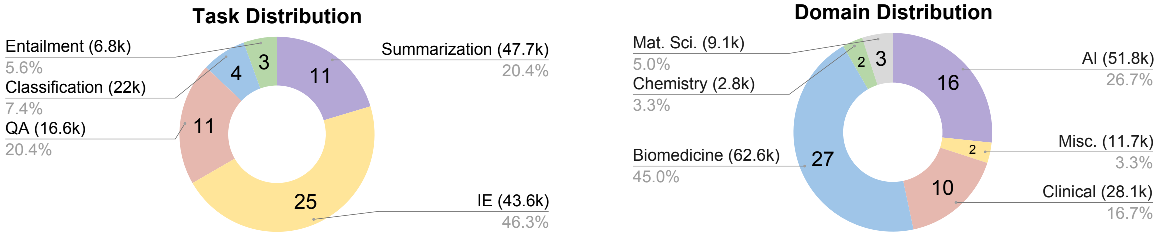


Figure 2: SciRIFF: pie charts show dataset counts and brackets indicate instance totals for task categories/domains.

to a wide coverage, the instructions in SciRIFF are grounded in long inputs (i.e., paper sections) and support *structured* outputs useful for tasks in literature understanding (such as relation extraction, fact checking with rationale selection, and QA with attribution). SciRIFF is sourced from existing high-quality scientific datasets and converted into instructions using expert-written and verified instruction templates. Out of 54 tasks, 50 involve templates paired with manually crafted Python scripts that serve to extract ground-truth answers, postprocess (e.g., removing duplicate entity mentions; converting span-level representations to instruction-following formats), or normalize the source datasets.[2]

2.1 Dataset construction

We construct SciRIFF through a rigorous pipeline that transforms existing scientific literature datasets into high-quality instruction-following instances, which involves template engineering, output schema design, and quality control steps that go well beyond simple dataset reformatting. See § A.1 for full task list.

We perform all template writing and annotation with domain experts. Domain experts are the paper authors with extensive experience in NLP. We chose this approach rather than using synthetic data (e.g., (Köksal et al., 2023; Li et al., 2023)). We believe it is sensible to exhaust available human-annotated resources for this emerging area before turning to potentially noisy synthetic data generation (see Appendix F for sample templates, which show the complexity of the tasks.) Further, in §4 we show that using templates from prior work, or using LLMs to generate templates results in significant decline in performance. In addition, we need

[2] The remaining four tasks are naturally formatted for instruction-following and ready for Jinja templating which don't require any special treatment.

high-quality evaluation data, which we construct by holding out nine SciRIFF tasks as an evaluation benchmark (§3.1).

We adopt `json` as the output format for structured tasks (34 of 54 tasks).[3] `json` is a convenient format for tasks requiring attribution, such as information extraction and question answering, where outputs must explicitly pair answers with supporting evidence in a human-and-machine-readable format. Our training set spans multiple scientific domains (Figure 2). We create instruction mixes of varying context lengths.[4] We refer readers to Appendix A.3 for details and statistics.

Dataset selection criteria 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 a response derived directly from the text. The task families include summarization, reading comprehension, entailment, classification, and information extraction, which are relevant for real-world use cases (e.g., meta-analysis of literature, clinical decision-making, augmented reading). We provide detailed information and citations of all source datasets in Appendix A. We *exclude* datasets that require retrieval from document collections (e.g., open-domain QA), since it is unclear how to build instruction-response pairs from them. We also exclude datasets that assess general reasoning and mathematical problem-solving skills without grounding on scientific literature, such as ScienceQA (Lu et al., 2022), SciBench (Wang et al., 2023b), and MATH (Hendrycks et al., 2021b) since such resources already exist. Additionally, we only keep datasets that are publicly available, have a permissive license, and are well-documented and actively maintained. See Appendix A.1 for the complete task list.

Quality Verification. Each template was verified by an additional author for clarity and correctness. We will include guidelines and best practices for prompt-writing in the release and aim to promote community contributions to expand SciRIFF through our open-sourced data collection process.

3 Experiment setup

We conduct supervised finetuning experiments to evaluate the effectiveness of SciRIFF in improving LLM performance on scientific instruction-following tasks across various model families and sizes. Our experiments explore different data configurations and their impact on scientific instruction-following as measured through SciRIFF-Eval described in §3.1.

3.1 Evaluation

We selected a set of nine tasks from SciRIFF for evaluation, designed to cover a diverse range of task categories and scientific domains. SciRIFF tests true out-of-distribution generalization with instructions entirely excluded from training. The inputs, outputs, and evaluation metrics for each task are summarized in Table 1. Additional details of evaluation tasks are included in Appendix D.

3.2 Scientific Instruction Finetuning

Our goal is to adapt pretrained LLMs to the scientific literature domain. We conduct full finetuning experiments using a range of models and data configurations to assess the effectiveness of SciRIFF. In §4.3, we present an additional analysis examining the potential of using SciRIFF for continual finetuning of instruction-tuned models, exploring a compute-efficient strategy for adaptation.

Data sources We finetune using two primary datasets: (1) **SciRIFF**, [5] and (2) **Tülu V2 Mix** (Ivison et al., 2023b), an open-source high-quality general-domain instruction-following dataset that includes demonstrations from various sources, both human-written (e.g., Flan (Wei et al., 2022)) and distilled from proprietary LLMs (e.g., ShareGPT[6], Open Assistant[7]). The original Tülu V2 Mix contains 326,154 examples, including 7.5K scientific literature understanding demonstrations which overlap (i.e. contaminated) with our evaluation set SciRIFF-Eval. We remove those 7.5K examples for clean experiments and to avoid contamination with SciRIFF-Eval. For all experiments, we consistently use this filtered version and refer to this as Tülu V2 Mix to maintain controlled finetuning and unbiased evaluations.

Base models We use following base LLMs as starting points: Llama 3.1-8B (Touvron et al., 2023b), Llama 3.2-3B (Dubey et al., 2024), and

[3] Paper authors transform raw dataset annotations into standardized json schemas before templating.

[4] We conduct our experiments using SciRIFF-4096 (hereafter SciRIFF) due to computational constraints.

[5] In our study, we use 70.5K instances for training.

[6] https://sharegpt.com/

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

Name	Type	Input	Output	Metrics
BioASQ List QA	QA	Question, paper excerpts	Answer entities	Exact match F1
BioRED	IE(NER)	Biomedical abstract	6 entity types	Exact match F1
DiSCoMaT	IE(Table)	LaTex table excerpt	Table entries	BLEU score
Evidence Inference (EI)	IE(Rel)	Clinical trial abstract	PICO	String overlap F1
MultiCite (MC)	Classification	Citation context	Citation intents	Exact match F1
MuP	Summarization	ML paper full text	Peer review summary	LLM judge similarity
Qasper	QA	NLP paper question	Answer / Attribution	LLM judge similarity / Token F1
SciERC	IE(Rel)	CS abstract	6 entity types	Exact match F1
SciFact	Entailment	Claim, abstract	Verdict / Evidence	Label F1 / Token F1

Table 1: Evaluation tasks included in SciRIFF-Eval. "/" separators indicate two separate subtasks. We use `GPT-4o` as our LLM judge and evaluate similarity on a 1-5 scale; see Appendix D for details.

Qwen 2.5-1.5B (Yang et al., 2024).[8] While our primary focus is on improving base models, we also experiment with models that have undergone proprietary instruction tuning and preference optimization ("–instruct" versions) (Ouyang et al., 2022). Although direct comparisons with "–instruct" models are complicated by unknown training details, we show that SciRIFF can provide additional value even in these cases. We note, however, that our main results and analyses focus on the controlled experiments with base models where we can fully account for all training conditions.

Finetuning data configurations For each model, we explore three data configurations: (1) **Tülu V2 Mix** only, to establish a **baseline** for general instruction-following; (2) **SciRIFF** only, to assess the impact of scientific instruction data in isolation; and (3) **SciRIFF+Tülu**, combining the general and scientific instruction data.

4 Results

This section discusses our key results and findings.

4.1 Main Results

We report our main experimental results in Table 2. For fair comparison, all models are finetuned on the same data mixes. We show that training on SciRIFF instances results in the best average performance in each model family. Six frontier models, such as GPT-5, Gemini-2.5-Pro (Gemini-2.5, 2025) and Kimi-K2 (Kimi, 2025), serve as strong baselines. Additionally, we evaluate selected domain-expert models for comprehensiveness, including SciLitLLM 7B (Li et al., 2024), BioMedical-Llama3 8B (Bolton et al., 2024), BioMistral 7B (Labrak et al., 2024), CodeLlama 7B (Rozière et al., 2023), and a weak baseline Llama 2 7B (Touvron et al., 2023a).

Furthermore, to demonstrate the necessity of expert-written templates for our tasks, we conduct an ablation study comparing our templates against alternatives in §4.2, with details in Appendix B.

Our key findings are below:

SciRIFF enhances scientific literature understanding Table 2 shows that finetuning on SciRIFF consistently enhances the overall performance on SciRIFF-Eval. Compared to the corresponding base models finetuned on Tülu, SciRIFF-trained models achieve, on average, 70.6% performance gain. Furthermore, without exception, SciRIFF also adds values when finetuning on "–instruct" models (44.6% on average). Across all model groups, the "–instruct" variants trained exclusively on SciRIFF achieve the highest average scores within their respective groups. Finally, while the new frontier models are very strong, with GPT-5 achieving the top baseline score of 61.1, out of the twelve models trained with the inclusion of SciRIFF instances, eight outperform GPT-5 on SciRIFF-Eval, with Qwen 2.5 1.5B showing the most significant improvement (from 29.1 to 57.2 in average score with SciRIFF alone). Results indicate that our specialized SciRIFF can substantially enhance scientific literature understanding and extraction capabilities beyond what general or proprietary instruction data can provide.

Task-specific impacts and room for improvement SciRIFF training achieves large gains on the three IE tasks (BioRED, DiSCoMaT, and SciERC). Relative to their Tülu-only counterparts, SciRIFF-finetuned base models improve IE task performance by, on average, 200.4%. And SciRIFF training improves performance on QA and Entailment as well. In contrast, performance on the

[8] We do not train larger models due to compute constraint. However, as shown in §4.1 improvements are consistent across sizes/families.

Model	**Data**	**BioASQ**	**BioR**	**DiscMT**	**EI**	**MC**	**MuP**	**Qasper**	**SciERC**	**SciFact**	**Avg.**
GPT-5	-	47.3	66.6	72.0	25.7	**67.9**	**94.2**	62.1 / **55.9**	44.0	73.1 / **74.0**	**61.1**
Gemini-2.5-Pro	-	45.1	65.1	71.2	23.7	63.9	92.8	**66.6** / 52.0	42.3	77.1 / 72.5	59.8
DeepSeek-V3.1	-	48.3	**68.0**	75.5	25.1	51.2	90.5	66.1 / 49.7	**44.1**	82.4 / 61.8	59.2
Kimi-K2	-	49.3	66.9	75.0	5.9	58.8	92.5	61.0 / 47.9	42.7	**87.6** / 62.0	60.0
GPT-4o	-	48.3	63.6	71.3	25.9	62.0	88.3	54.0 / 55.0	40.3	85.5 / 70.4	60.4
GPT-4o-mini	-	**49.6**	53.7	**75.6**	**27.7**	54.8	88.8	61.7 / 46.7	33.1	82.7 / 63.6	58.0
SciLitLLM 7B	-	**51.2**	**76.6**	**71.0**	**23.5**	**70.7**	67.5	**50.7** / **53.9**	49.8	**83.4** / **67.2**	**60.3**
BioMedical-Llama3 8B	-	41.1	45.7	62.9	8.4	28.6	79.8	19.0 / 11.1	**58.0**	43.1 / 38.7	41.1
BioMistral 7B	-	38.3	0.7	4.7	7.7	23.7	70.3	14.1 / 12.5	0.0	7.1 / 18.6	19.1
CodeLlama 7B	-	38.6	22.7	45.0	11.0	38.9	**80.3**	46.3 / 31.4	14.8	55.8 / 35.1	38.1
Llama 2 7B	-	34.2	0.0	4.8	7.4	37.8	72	15.7 / 8.5	0.3	27.7 / 6.2	19.5
Qwen 2.5 1.5B-`Instruct`	-	38.9	19.7	35.5	10.5	36.9	**80.8**	38.8 / 39.4	20.8	55.0 / 31.5	37.1
	SciRIFF	48.1	79.7	**80.6**	20.9	**70.9**	67.3	42.8 / **54.3**	**52.0**	**80.9** / 68.9	**60.6**
	SciRIFF +Tülu	**49.3**	**80.1**	79.5	**21.3**	70.8	61.3	**45.8** / 48.6	51.0	78.5 / **70.1**	59.7
Qwen 2.5 1.5B	Tülu	35.7	23.4	31.8	7.6	6.6	**73.0**	25.0 / 23.2	12.0	52.4 / 29.5	29.1
	SciRIFF	43.6	**81.8**	45.6	18.9	**71.2**	67.8	**47.0** / **51.4**	**52.7**	78.8 / 70.5	57.2
	SciRIFF +Tülu	**46.5**	79.0	**78.3**	**19.4**	70.2	63.8	40.4 / 49.7	51.7	**80.9** / **70.6**	**59.1**
Llama 3.2 3B-`Instruct`	-	42.9	35.9	61.0	11.2	47.3	**86.0**	43.9 / 35.8	20.8	59.5 / 40.0	44.0
	SciRIFF	42.7	**84.0**	**83.4**	**25.5**	71.4	64.8	50.0 / 57.1	**58.2**	**86.8** / **70.5**	**63.1**
	SciRIFF +Tülu	**43.0**	83.3	82.9	21.7	**72.2**	69.0	**51.9** / **58.2**	53.3	85.6 / 70.3	62.8
Llama 3.2 3B	Tülu	35.5	30.1	46.7	3.1	44.0	75.6	47.4 / 34.4	20.3	55.4 / 36.6	39.0
	SciRIFF	43.6	84.2	83.2	**25.2**	71.7	64.3	46.0 / **57.2**	**57.2**	81.6 / 65.8	61.8
	SciRIFF+Tülu	**46.0**	**84.3**	**83.3**	24.6	**72.7**	65.5	**47.7** / 56.3	57.0	**82.7** / **71.2**	**62.8**
Llama 3.1 8B-`Instruct`	-	43.7	48.8	62.2	17.8	48.8	**88.3**	**54.0** / 43.0	30.6	66.7 / 51.7	50.5
	SciRIFF	45.9	**86.0**	**83.7**	25.0	**71.4**	70.5	53.3 / 54.1	**56.8**	**85.8** / **72.5**	**64.1**
	SciRIFF+Tülu	**48.8**	84.7	83.6	**26.6**	71.3	66.0	50.9 / **55.2**	54.4	85.5 / 70.2	63.4
Llama 3.1 8B	Tülu	44.4	42.8	51.8	1.1	39.4	**80.8**	42.8 / 28.6	24.3	50.0 / 33.6	40.0
	SciRIFF	**46.2**	84.2	**83.9**	23.5	71.0	68.5	**49.8** / 52.2	56.2	83.3 / **71.9**	62.8
	SciRIFF+Tülu	41.6	**85.2**	78.7	**28.2**	**71.6**	70.5	47.9 / **61.0**	**58.1**	**87.4** / 71.2	**63.8**

Table 2: Performance on SciRIFF-Eval tasks across model families and training configurations (§3.2). Best performance per model group is **bolded**. Columns with a "/" indicate two evaluation metrics as described in §3.1.

Config	**Ours**	**Simple**	**Synthetic**
Llama-3.1-8B	**62.8**	42.2	28.0
Qwen-2.5-1.5B	**57.2**	33.1	19.1

Table 3: Average SciRIFF-Eval scores across selected configurations. Columns use their matching evaluations, SciRIFF-Eval, SciRIFF-Eval-Simple, and SciRIFF-Eval-Synthetic for fair comparisons. See Appendix B and Table 8 for details.

summarization task (MuP) generally decreases after SciRIFF finetuning. This suggests that while SciRIFF is particularly effective for enhancing IE capabilities, it may not provide additional benefits for summarization tasks that are likely well-covered in general instruction-following training. The fact that frontier models our strong finetuned models achieve only an average score of around 60 highlights the difficulty of SciRIFF-Eval. Model performance remains relatively low on tasks like EI; This is due to a combination of task difficulty and evaluation challenges, which we discuss in §6.

Balancing scientific and general data As shown in Table 2, combining SciRIFF and Tülu V2 Mix training data (SciRIFF+Tülu) yields the best performance on SciRIFF-Eval for *base* models. This suggests that incorporating general instruction-following data may provide some broader capability transfer, which base models particularly benefit from, though the impact remains limited (within 2.2%). On the other hand, training "–instruct" models exclusively on SciRIFF data proves to be slightly more effective (within 1% on average).

Comparing with domain-specialized baselines Models trained on in-domain scientific corpora in continual pretraining, followed by instruction-tuning for science literature tasks, can be very competitive (e.g., SciLitLLM at 60.3 on SciRIFF-Eval). In contrast, models specialized for biomedical and general science tasks (e.g., BioMedical, BioMistral) consistently underperform in literature understanding. Llama 2 7B achieves an average score of only 19.5, with near-zero performance on IE tasks (BioRED and SciERC) partly due to its inability to follow JSON output requirements. We also observe that CodeLlama, likely benefiting from exposure to JSON and code-based reasoning improvements, outperforms Llama 2 and BioMistral. None of the specialized models match the performance of our approach, which uniquely leverages SciRIFF training to enhance scientific literature understanding.

Grounded Attribution vs. General Reasoning While highly capable at strong general reasoning, DeepSeek-V3.1 and Kimi-K2 show lower performance on tasks requiring grounded attribution. Specifically, Table 2 shows their evidence-finding scores on Qasper and SciFact are lower than other frontier models, as is their performance on MultiCite. This suggests a distinction between general problem-solving and the specific skill of finding and attributing evidence from a given text. This finding, also discussed in concurrent work (Li et al., 2025), indicates that strong abstract reasoning does not guarantee proficiency in document-grounded tasks. This reinforces the value of SciRIFF-Eval as a specialized benchmark for measuring this crucial, evidence-based capability in scientific literature understanding.

4.2 Template Ablation

We compare our standard expert-written templates with (1) simple templates that mirror FlanV2 (Chung et al., 2024) and (2) templates generated by GPT-4o. We conduct the analysis on selected (due to compute constraints) Base models with SciRIFF only training data, to exclude confounding factors (see templating details in Appendix B.) While prompt ablations are more meaningful for general-purpose language models rather than supervised-finetuned models (Voronov et al., 2024; Kung and Peng, 2023), we present the experiments to validate our design decisions to rely on expert human-written templates for the emerging and complex domain of instruction-following for scientific literature understanding and synthesis. Table 3 shows that expert-written templates, which carefully specify task requirements and output structures, outperform the alternatives. We argue, along with detailed descriptions in §2 and prompt examples at Appendix §F, that expert-written template is preferred. These ablations, while not central to our main contributions and objectives, provide signals on the importance of careful template design for scientific literature understanding tasks.

Model	Data	7B	70B
Llama 2	Tülu	36.7	47.5
	SciRIFF	**48.0**	**51.1**
	SciRIFF+Tülu	46.0	50.8
Tülu V2	SciRIFF	**47.0**	48.8
	SciRIFF+Tülu	**47.0**	**50.7**

Table 4: Comparison of SciRIFF-Eval (Sci.) performance for models finetuned from Llama 2 base and Tülu V2 (science-decontaminated).

4.3 Continual Finetuning Analysis

In early phase of our study, we explore strategies for efficient adaptation. Specifically, we examined whether starting from an existing instruction-tuned checkpoint (on general domain instructions) could provide compute advantages over training from scratch, without hurting SciRIFF-Eval performance. For this controlled experiment, we selected two starting points: (1) Llama 2 base and (2) the same model already finetuned on science-decontaminated Tülu V2 Mix (referred as Tülu V2). We explored different training approaches:For Llama 2 base, we train on all available Tülu V2 Mix demonstrations, combined with *1000* instances per SciRIFF task, given the *empirical findings* in § 4.4. For the Tülu V2 starting point, we perform continual finetuning using 1000 instances per SciRIFF task, together with a *matching number* (1000) of instances sampled from Tülu V2 Mix.

Table 4 reports average SciRIFF-Eval performance for our two starting checkpoints using three data configurations. Starting from Tülu V2 performs comparably to Llama 2 base while requiring only 20% of the compute (Table 5). When trained on SciRIFF+Tülu data, models from both starting points achieve similar performance: Tülu V2 is slightly better on science at 7B and nearly identical at 70B. Given that finetuning Tülu V2 requires only 20% of the data, this highlights a compute-efficient adaptation for scientific domains, aligning with prior findings (Dong et al., 2024; Shi et al., 2023). While our main experiments (§3.2) use

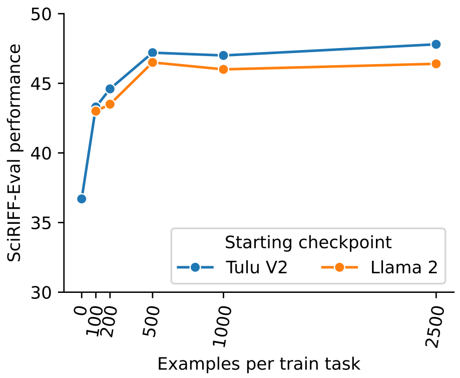


Figure 3: Performance on SciRIFF-Eval vs. n_{sci} (instances/task). Gains saturate at $n_{sci} = 1000$ (see §3.2)

newer architectures,[9] this analysis, along with the results in § 4.4, illustrates how practitioners can optimize training for SciRIFF under fixed model architectures.

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

Table 5: SciRIFF and Tülu V2 Mix instances used for finetuning described in §4.3, with $n_{sci} = 1000$.

4.4 Instruction Data Scale

We define n_{sci} as the number of instances per SciRIFF task. Figure 3 shows that performance on SciRIFF-Eval increases sharply as n_{sci} rises from 100 to 500 and levels off subsequently. We found that 1,000 instances per science task are sufficient for peak performance for Llama 2 models. Therefore, we set $n_{sci} = 1000$ across our experiments in the continual finetuning analysis (§4.3).

5 Related Work

Strategies for creation of instruction-following resources. Related work has explored a number of methods for curating instruction-following resources such as repurposing existing datasets using human-written templates (Wei et al., 2022; Chung et al., 2024), crowdsourcing instructions Databricks (2023); Zhou et al. (2023); Mishra et al. (2021), ShareGPT[10] and generating synthetic data (Lambert et al., 2024a). Broadly, synthetic 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) for quality assurance. We refer the readers to see template examples in Appendix F for evidence.

Instruction-following resources for scientific literature. Despite many instruction-following collections, few resources focus on scientific literature, which are crucial for assisting researchers and accelerating 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). 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. While recent work such as Li et al. (2024) explores improving language models' scientific understanding through continuous pretraining and SFT, our work specifically contributes a diverse, high-quality instruction dataset for this domain. Some instruction-tuning resources have explored structured output formats (Zhang et al., 2023b; Wang et al., 2023a; Jiao et al., 2023; Gao et al., 2023), but not with a focus on science. Finally, most datasets in SciRIFF have longer instruction contexts than prior works (see Appendix Table 7 for a comparison).

Other scientific literature benchmarks. Prior works have 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 (Singh et al., 2023; Taylor et al., 2022; Wei et al., 2023) cover domains beyond biomedicine, but are not targeted for training instruction-following models. SciASSESS (Cai et al., 2024) evaluates LLMs' proficiency in sci-

[9] Due to compute constraints, we do not extend this analysis to all models.

[10] https://sharegpt.com/

entific literature analysis, focusing on memorization and reasoning abilities. Complementary to our static benchmark, SciArena (Zhao et al., 2025) provides a dynamic platform that evaluates models via ongoing expert preference voting. In contrast, SciRIFF provides fully *expert-written* instructions, serving both as a benchmark and training resource.

Concurrent with our work, Li et al. (2025) introduce SciReas, a meta-benchmark for scientific problem-solving that includes a subset of SciRIFF tasks. Their analysis characterizes SciRIFF as focusing on grounded literature comprehension, distinguishing it from abstract reasoning benchmarks. This distinction is supported by their findings that performance on SciRIFF has low correlation with reasoning-focused benchmarks like GPQA (Rein et al., 2023), validating the unique contribution of our resource for measuring essential skills in evidence-based literature understanding.

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 significant improvement in model performance on held-out scientific tasks (on average 70.6% over baselines). The large improvements we observe, especially on tasks requiring structured extraction and evidence-finding, underscore the value of targeted data for building practical tools for researchers.

As observed in §4.1, neither our best finetuned models nor the proprietary frontier models are sufficiently strong on SciRIFF-Eval (around 60%), which demonstrates the difficulty of our tasks. Utilizing LLMs to perform more flexible and fine-grained evaluations (Kim et al., 2024) represents a promising direction. Future work could focus on reliably generating multiple templates for such complex tasks in a more controlled and principled manner to help models improve their generalization to unseen tasks. Incorporating reliable synthetic data generation techniques and preference data (Lambert et al., 2024b) for scientific literature understanding tasks is also a promising avenue. In conclusion, we are optimistic that the SciRIFF data and evaluations SciRIFF-Eval, as well as the model checkpoints, will serve as valuable resources to build systems for scientific researchers.

Limitation

While we demonstrated the effectiveness of SciRIFF and the value of SciRIFF-Eval, we note the following limitations about our work: Although we included a wide range of datasets, this still could limit the open-ended tasks that could involve literature understanding. For example, more sophisticated iterative or chat-style interactions mimicking interactions with a research assistant are not captured with SciRIFF. Finally, computational constraints prevented us to experiment with largest open-source models; we suspect that training larger open-source models (such as Llama 3.1 405B) can provide even further improvements over state-of-the-art commercial models.

Ethics Statement

The ethical risks associated with this work are minimal. As we source the data from existing datasets and we work in the science domain, we do not suspect major risks are involved in the curation of our dataset. However, potential biases might still exist in some datasets. For example, one of the source datasets is paper summarization which is sourced from OpenReview.net peer reviews by the original authors. And peer reviews might inherently occasionally include biases or unhelpful languages. As with all LLMs, our trained models are still prone to issues such as hallucinations, so users should exercise caution when interpreting model outputs, particularly in downstream applications in science.

Author contributions

David Wadden and Kejian Shi contributed equally and led the project. Jacob Morrison, Alan Li, and Aakanksha Naik were among the core contributors and substantially contributed to the experiments and data collection. Shruti Singh, Nitzan Barzilay, Kyle Lo, Tom Hope, and Luca Soldaini contributed ideas and provided additional support with experiments. Shannon Shen, Doug Downey, Hanna Hajishirzi, and Arman Cohan provided core mentorship and advising.

Acknowledgements

We thank Minyi Chen, Yicheng Gao, Kaiyuan Guan, and Yujie Qiao for their data sourcing contributions during early phase of the project. We are grateful to Google's TRC program for compute support.

References

Griffin Adams, Bichlien Nguyen, Jake Smith, Yingce Xia, Shufang Xie, Anna Ostropolets, Budhaditya Deb, Yuan-Jyue Chen, Tristan Naumann, and Noémie Elhadad. 2023. What are the desired characteristics of calibration sets? identifying correlates on long form scientific summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10520–10542, Toronto, Canada. Association for Computational Linguistics.

Fan Bai, Junmo Kang, Gabriel Stanovsky, Dayne Freitag, Mark Dredze, and Alan Ritter. 2024. Schema-driven information extraction from heterogeneous tables. *Preprint*, arXiv:2305.14336.

Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. 2008. The ACL Anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Elliot Bolton, Abhinav Venigalla, Michihiro Yasunaga, David Hall, Betty Xiong, Tony Lee, Roxana Daneshjou, Jonathan Frankle, Percy Liang, Michael Carbin, and Christopher D. Manning. 2024. Biomedlm: A 2.7b parameter language model trained on biomedical text. *arXiv preprint arXiv: 2403.18421*.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. 2020. TLDR: Extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4766–4777, Online. Association for Computational Linguistics.

Hengxing Cai, Xiaochen Cai, Junhan Chang, Sihang Li, Lin Yao, Changxin Wang, Zhifeng Gao, Hongshuai Wang, Yongge Li, Mujie Lin, Shuwen Yang, Jiankun Wang, Mingjun Xu, Jin Huang, Fang Xi, Jiaxi Zhuang, Yuqi Yin, Yaqi Li, Changhong Chen, Zheng Cheng, Zifeng Zhao, Linfeng Zhang, and Guolin Ke. 2024. Sciassess: Benchmarking llm proficiency in scientific literature analysis. *arXiv preprint arXiv: 2403.01976*.

Qingyu Chen, Alexis Allot, Robert Leaman, Rezarta Islamaj, Jingcheng Du, Li Fang, Kai Wang, Shuo Xu, Yuefu Zhang, Parsa Bagherzadeh, et al. 2022. Multi-label classification for biomedical literature: an overview of the biocreative vii litcovid track for covid-19 literature topic annotations. *Database*, 2022:baac069.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3586–3596, Minneapolis, Minnesota. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Arman Cohan, Guy Feigenblat, Tirthankar Ghosal, and Michal Shmueli-Scheuer. 2022. Overview of the first shared task on multi perspective scientific document summarization (MuP). In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 263–267, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Kevin Bretonnel Cohen, Karin Verspoor, Karën Fort, Christopher Funk, Michael Bada, Martha Palmer, and Lawrence E. Hunter. 2017. The colorado richly annotated full text (craft) corpus: Multi-model annotation in the biomedical domain. In *Handbook of Linguistic Annotation*, pages 1379–1394. Springer.

Nigel Collier, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Jin-Dong Kim. 2004. Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 73–78, Geneva, Switzerland. COLING.

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

Databricks. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm. Blog post.

Jay DeYoung, Iz Beltagy, Madeleine van Zuylen, Bailey Kuehl, and Lucy Lu Wang. 2021. MS^2: Multi-document summarization of medical studies. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7494–7513, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jay DeYoung, Eric Lehman, Benjamin Nye, Iain Marshall, and Byron C. Wallace. 2020. Evidence inference 2.0: More data, better models. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 123–132, Online. Association for Computational Linguistics.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024. How abilities in large language models are affected by supervised fine-tuning data composition.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv: 2407.21783*.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. 2024. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. In *International Conference on Learning Representations* (ICLR).

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. *Preprint*, arXiv:2406.09098.

Jason Fries, Leon Weber, Natasha Seelam, Gabriel Altay, Debajyoti Datta, Samuele Garda, Sunny Kang, Rosaline Su, Wojciech Kusa, Samuel Cahyawijaya, Fabio Barth, Simon Ott, Matthias Samwald, Stephen Bach, Stella Biderman, Mario Sänger, Bo Wang, Alison Callahan, Daniel León Periñán, Théo Gigant, Patrick Haller, Jenny Chim, Jose Posada, John Giorgi, Karthik Rangasai Sivaraman, Marc Pàmies, Marianna Nezhurina, Robert Martin, Michael Cullan, Moritz Freidank, Nathan Dahlberg, Shubhanshu Mishra, Shamik Bose, Nicholas Broad, Yanis Labrak, Shlok Deshmukh, Sid Kiblawi, Ayush Singh, Minh Chien Vu, Trishala Neeraj, Jonas Golde, Albert Villanova del Moral, and Benjamin Beilharz. 2022. Bigbio: A framework for data-centric biomedical natural language processing. In *Advances in Neural Information Processing Systems*, volume 35, pages 25792–25806. Curran Associates, Inc.

Chang Gao, Wenxuan Zhang, Guizhen Chen, and Wai Lam. 2023. Jsontuning: Towards generalizable, robust, and controllable instruction tuning. *arXiv preprint arXiv:2310.02953*.

Team Gemini-2.5. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.20534.

Martin Gerner, Goran Nenadic, and Casey M. Bergman. 2010. Linnaeus: A species name identification system for biomedical literature. *BMC Bioinformatics*, 11:85.

Tomas Goldsack, Zhihao Zhang, Chenghua Lin, and Carolina Scarton. 2022. Making science simple: Corpora for the lay summarisation of scientific literature. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10589–10604, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23.

Tanishq Gupta, Mohd Zaki, Devanshi Khatsuriya, Kausik Hira, N M Anoop Krishnan, and Mausam. 2023. DiSCoMaT: Distantly supervised composition extraction from tables in materials science articles. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13465–13483, Toronto, Canada. Association for Computational Linguistics.

Tianyu Han, Lisa C. Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexander Löser, Daniel Truhn, and Keno Kyrill Bressem. 2023. Medalpaca - an open-source collection of medical conversational ai models and training data. *ArXiv*, abs/2304.08247.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. 2013. The ddi corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of Biomedical Informatics*, 46(5):914–920.

Rezarta Islamaj, Robert Leaman, Sun Kim, Dongseop Kwon, Chih-Hsuan Wei, Donald C. Comeau, Yifan Peng, David Cissel, Cathleen Coss, Carol Fisher, Rob Guzman, Preeti Gokal Kochar, Stella Koppel, Dorothy Trinh, Keiko Sekiya, Janice Ward, Deborah Whitman, Susan Schmidt, and Zhiyong Lu. 2021a. Nlm-chem, a new resource for chemical entity recognition in pubmed full text literature. *Scientific Data*, 8:61.

Rezarta Islamaj, Chih-Hsuan Wei, David Cissel, Nicholas Miliaras, Olga Printseva, Oleg Rodionov, Keiko Sekiya, Janice Ward, and Zhiyong Lu. 2021b. Nlm-gene, a richly annotated gold standard dataset for gene entities that addresses ambiguity and multi-species gene recognition. *Journal of Biomedical Informatics*, 118:103779.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, 47:1–10.

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

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

Yizhu Jiao, Ming Zhong, Sha Li, Ruining Zhao, Siru Ouyang, Heng Ji, and Jiawei Han. 2023. Instruct and extract: Instruction tuning for on-demand information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10030–10051, Singapore. Association for Computational Linguistics.

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

Marcin Kardas, Piotr Czapla, Pontus Stenetorp, Sebastian Ruder, Sebastian Riedel, Ross Taylor, and Robert Stojnic. 2020. AxCell: Automatic extraction of results from machine learning papers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8580–8594, Online. Association for Computational Linguistics.

Tetsu Kasanishi, Masaru Isonuma, Junichiro Mori, and Ichiro Sakata. 2023. SciReviewGen: A large-scale dataset for automatic literature review generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6695–6715, Toronto, Canada. Association for Computational Linguistics.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*.

Team Kimi. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv: 2507.20534*.

Martin Krallinger, Obdulia Rabal, Saber Ahmad Akhondi, Martín Pérez Pérez, Jesus Santamaría, Gael Pérez Rodríguez, Georgios Tsatsaronis, Ander Intxaurrondo, José Antonio Baso López, Umesh K. Nandal, Erin M. van Buel, Ambika Chandrasekhar, Marleen Rodenburg, Astrid Lægreid, Marius A. Doornenbal, Julen Oyarzábal, Anália Lourenço, and Alfonso Valencia. 2017. Overview of the biocreative vi chemical-protein interaction track.

Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M. Lowe, Roger A. Sayle, Riza Theresa Batista-Navarro, Rafal Rak, Torsten Huber, Tim Rocktäschel, Sérgio Matos, David Campos, Buzhou Tang, Hua Xu, Tsendsuren Munkhdalai, Keun Ho Ryu, S. V. Ramanan, Senthil Nathan, Slavko Žitnik, Marko Bajec, Lutz Weber, Matthias Irmer, Saber A. Akhondi, Jan A. Kors, Shuo Xu, Xin An, Utpal Kumar Sikdar, Asif Ekbal, Masaharu Yoshioka, Thaer M. Dieb, Miji Choi, Karin Verspoor, Madian Khabsa, C. Lee Giles, Hongfang Liu, Komandur Elayavilli Ravikumar, Andre Lamurias, Francisco M. Couto, Hong-Jie Dai, Richard Tzong-Han Tsai, Caglar Ata, Tolga Can, Anabel Usié, Rui Alves, Isabel Segura-Bedmar, Paloma Martínez, Julen Oyarzabal, and Alfonso Valencia. 2015. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of Cheminformatics*, 7(S1):S2.

Po-Nien Kung and Nanyun Peng. 2023. Do models really learn to follow instructions? an empirical study of instruction tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1317–1328, Toronto, Canada. Association for Computational Linguistics.

Fabrício Kury, Alex Butler, Chi Yuan, Li-heng Fu, Yingcheng Sun, Hao Liu, Ida Sim, Simona Carini, and Chunhua Weng. 2020. Chia, a large annotated corpus of clinical trial eligibility criteria. volume 7, page 281.

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

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, P. Gourraud, Mickael Rouvier, and Richard Dufour. 2024. Biomistral: A collection of open-source pretrained large language models for medical domains. *Annual Meeting of the Association for Computational Linguistics*.

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

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024b. RewardBench: Evaluating reward models for language modeling. *arXiv preprint arXiv: 2403.13787*.

Anne Lauscher, Brandon Ko, Bailey Kuehl, Sophie Johnson, Arman Cohan, David Jurgens, and Kyle Lo. 2022. MultiCite: Modeling realistic citations requires moving beyond the single-sentence single-label setting. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1875–1889, Seattle, United States. Association for Computational Linguistics.

Yoonjoo Lee, Kyungjae Lee, Sunghyun Park, Dasol Hwang, Jaehyeon Kim, Hong-In Lee, and Moontae Lee. 2023. QASA: Advanced question answering on scientific articles. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19036–19052. PMLR.

Alan Li, Yixin Liu, Arpan Sarkar, Doug Downey, and Arman Cohan. 2025. Demystifying scientific problem-solving in llms by probing knowledge and reasoning. *arXiv preprint arXiv: 2508.19202*.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016:baw068.

Sihang Li, Jian Huang, Jiaxi Zhuang, Yaorui Shi, Xiaochen Cai, Mingjun Xu, Xiang Wang, Linfeng Zhang, Guolin Ke, and Hengxing Cai. 2024. Scilitllm: How to adapt llms for scientific literature understanding. *arXiv preprint arXiv: 2408.15545*.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason Weston, and Mike Lewis. 2023. Self-alignment with instruction backtranslation. *arXiv preprint arXiv: 2308.06259*.

Kyle Lo, Joseph Chee Chang, Andrew Head, Jonathan Bragg, Amy X. Zhang, Cassidy Trier, Chloe Anastasiades, Tal August, Russell Authur, Danielle Bragg, Erin Bransom, Isabel Cachola, Stefan Candra, Yoganand Chandrasekhar, Yen-Sung Chen, Evie (Yu-Yen) Cheng, Yvonne Chou, Doug Downey, Rob Evans, Raymond Fok, F.Q. Hu, Regan Huff, Dongyeop Kang, Tae Soo Kim, Rodney Michael Kinney, Aniket Kittur, Hyeonsu B Kang, Egor Klevak, Bailey Kuehl, Michael Langan, Matt Latzke, Jaron Lochner, Kelsey MacMillan, Eric Stuart Marsh, Tyler C. Murray, Aakanksha Naik, Ngoc-Uyen Nguyen, Srishti Palani, Soya Park, Caroline Paulic, Napol Rachatasumrit, Smita R Rao, Paul Sayre, Zejiang Shen, Pao Siangliulue, Luca Soldaini, Huy Tran, Madeleine van Zuylen, Lucy Lu Wang, Christopher Wilhelm, Caroline M Wu, Jiangjiang Yang, Angele Zamarron, Marti A. Hearst, and Daniel S. Weld. 2023. The semantic reader project: Augmenting scholarly documents through ai-powered interactive reading interfaces. *ArXiv*, abs/2303.14334.

Renze Lou, Kai Zhang, Jian Xie, Yuxuan Sun, Janice Ahn, Hanzi Xu, Yu su, and Wenpeng Yin. 2024. MUFFIN: Curating multi-faceted instructions for improving instruction following. In *The Twelfth International Conference on Learning Representations*.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Yao Lu, Yue Dong, and Laurent Charlin. 2020. Multi-XScience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8068–8074, Online. Association for Computational Linguistics.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv: 2308.09583*.

Ling Luo, Po-Ting Lai, Chih-Hsuan Wei, Cecilia N Arighi, and Zhiyong Lu. 2022. Biored: a rich biomedical relation extraction dataset. *Briefings in Bioinformatics*, 23(5).

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

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Cross-Task Generalization via Natural Language Crowdsourcing Instructions. *arXiv*.

Sunil Mohan and Donghui Li. 2019. Medmentions: A large biomedical corpus annotated with umls concepts. *Preprint*, arXiv:1902.09476.

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

Sheshera Mysore, Zachary Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew McCallum, and Elsa Olivetti. 2019. The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic structures. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 56–64, Florence, Italy. Association for Computational Linguistics.

Nihal V. Nayak, Yiyang Nan, Avi Trost, and Stephen H. Bach. 2024. Learning to generate instruction tuning datasets for zero-shot task adaptation. *arXiv preprint arXiv: 2402.18334*.

Benjamin Nye, Junyi Jessy Li, Roma Patel, Yinfei Yang, Iain Marshall, Ani Nenkova, and Byron Wallace. 2018. A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 197–207, Melbourne, Australia. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Srishti Palani, Aakanksha Naik, Doug Downey, Amy X. Zhang, Jonathan Bragg, and Joseph Chee Chang. 2023. Relatedly: Scaffolding literature reviews with existing related work sections. *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*.

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

Mihir Parmar, Swaroop Mishra, Mirali Purohit, Man Luo, Murad Mohammad, and Chitta Baral. 2022. In-BoXBART: Get instructions into biomedical multi-task learning. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 112–128, Seattle, United States. Association for Computational Linguistics.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. 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)*.

Sampo Pyysalo and Sophia Ananiadou. 2014. Anatomical entity mention recognition at literature scale. *Bioinformatics*, 30(6):868–875.

Machel Reid, Victor Zhong, Suchin Gururangan, and Luke Zettlemoyer. 2022. M2d2: A massively multi-domain language modeling dataset.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv: 2311.12022*.

Omid Rohanian, Mohammadmahdi Nouriborji, and David A. Clifton. 2023. Exploring the effectiveness of instruction tuning in biomedical language processing. *arXiv preprint arXiv: 2401.00579*.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv: 2308.12950*.

Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. COVID-fact: Fact extraction and verification of real-world claims on COVID-19 pandemic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2116–2129, Online. Association for Computational Linguistics.

Mourad Sarrouti, Asma Ben Abacha, Yassine Mrabet, and Dina Demner-Fushman. 2021. Evidence-based fact-checking of health-related claims. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3499–3512, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300.

Chufan Shi, Yixuan Su, Cheng Yang, Yujiu Yang, and Deng Cai. 2023. Specialist or generalist? instruction tuning for specific NLP tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15336–15348, Singapore. Association for Computational Linguistics.

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2023. SciRepEval: A multi-format benchmark for scientific document representations. In *Proceedings of the 2023 Conference* *on Empirical Methods in Natural Language Processing*, pages 5548–5566, Singapore. Association for Computational Linguistics.

Yu Song, Santiago Miret, Huan Zhang, and Bang Liu. 2023. HoneyBee: Progressive instruction finetuning of large language models for materials science. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5724–5739, Singapore. Association for Computational Linguistics.

Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. 2024. Mathscale: Scaling instruction tuning for mathematical reasoning. *arXiv preprint arXiv: 2403.02884*.

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. *arXiv preprint arXiv: 2211.09085*.

Aryeh Tiktinsky, Vijay Viswanathan, Danna Niezni, Dana Meron Azagury, Yosi Shamay, Hillel Taub-Tabib, Tom Hope, and Yoav Goldberg. 2022. A dataset for n-ary relation extraction of drug combinations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3190–3203, Seattle, United States. Association for Computational Linguistics.

Augustin Toma, Patrick R. Lawler, Jimmy Ba, Rahul G. Krishnan, Barry Rubin, and Bo Wang. 2023. Clinical camel: An open-source expert-level medical language model with dialogue-based knowledge encoding. *ArXiv*, abs/2305.12031.

Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. 2024. Openmathinstruct-1: A 1.8 million math instruction tuning dataset. *arXiv preprint arXiv: Arxiv-2402.10176*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023a. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

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

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

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16:138.

Vijay Viswanathan, Luyu Gao, Tongshuang Wu, Pengfei Liu, and Graham Neubig. 2023. DataFinder: Scientific dataset recommendation from natural language descriptions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10288–10303, Toronto, Canada. Association for Computational Linguistics.

Anton Voronov, Lena Wolf, and Max Ryabinin. 2024. Mind your format: Towards consistent evaluation of in-context learning improvements. *Annual Meeting of the Association for Computational Linguistics*.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Byron C. Wallace, Sayantan Saha, Frank Soboczenski, and Iain J. Marshall. 2021. Generating (factual?) narrative summaries of rcts: Experiments with neural multi-document summarization. In *AMIA Joint Summits on Translational Science Proceedings*, pages 605–614.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023a. Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv: 2304.08085*.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. 2023b. SCIBENCH: Evaluating college-level scientific problem-solving abilities of large language models. In *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*.

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

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. 2015. Gnormplus: An integrative approach for tagging genes, gene families, and protein domains. *BioMed Research International*, 2015:918710.

Jason Wei, Maarten Bosma, Vincent 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. *International Conference on Learning Representations*.

Shufa Wei, Xiaolong Xu, Xianbiao Qi, Xi Yin, Jun Xia, Jingyi Ren, Peijun Tang, Yuxiang Zhong, Yihao Chen, Xiaoqin Ren, Yuxin Liang, Liankai Huang, Kai Xie, Weikang Gui, Wei Tan, Shuanglong Sun, Yongquan Hu, Qinxian Liu, Nanjin Li, Chihao Dai, Lihua Wang, Xiaohui Liu, Lei Zhang, and Yutao Xie. 2023. Academicgpt: Empowering academic research. *Preprint*, arXiv:2311.12315.

Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Weidi Xie, and Yanfeng Wang. 2024. PMC-LLaMA: toward building open-source language models for medicine. *Journal of the American Medical Informatics Association*, page ocae045.

Tong Xie, Yuwei Wan, Wei Huang, Zhenyu Yin, Yixuan Liu, Shaozhou Wang, Qingyuan Linghu, Chunyu Kit, Clara Grazian, Wenjie Zhang, Imran Razzak, and Bram Hoex. 2023. Darwin series: Domain specific large language models for natural science. *arXiv preprint arXiv: 2308.13565*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, et al. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Da Yin, Xiao Liu, Fan Yin, Ming Zhong, Hritik Bansal, Jiawei Han, and Kai-Wei Chang. 2023. Dynosaur: A dynamic growth paradigm for instruction-tuning data curation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4031–4047, Singapore. Association for Computational Linguistics.

Botao Yu, Frazier N. Baker, Ziqi Chen, Xia Ning, and Huan Sun. 2024. Llasmol: Advancing large language models for chemistry with a large-scale, comprehensive, high-quality instruction tuning dataset. *arXiv preprint arXiv: 2402.09391*.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2024a. MAmmoTH: Building math generalist models through hybrid instruction tuning. In *The Twelfth International Conference on Learning Representations*.

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

Dan Zhang, Ziniu Hu, Sining Zhoubian, Zhengxiao Du, Kaiyu Yang, Zihan Wang, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024a. Sciglm: Training scientific language models with self-reflective instruction annotation and tuning. *arXiv preprint arXiv: 2401.07950*.

Di Zhang, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyu Yue, Wanli Ouyang, Dongzhan Zhou, Shufei Zhang, Mao Su, Han-Sen Zhong, and Yuqiang Li. 2024b. Chemllm: A chemical large language model. *arXiv preprint arXiv: 2402.06852*.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023a. Instruction tuning for large language models: A survey. *arXiv preprint arXiv: 2308.10792*.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2023b. Tablellama: Towards open large generalist models for tables. *arXiv preprint arXiv: 2311.09206*.

Xinlu Zhang, Chenxin Tian, Xianjun Yang, Lichang Chen, Zekun Li, and Linda Ruth Petzold. 2023c. Alpacare:instruction-tuned large language models for medical application. *arXiv preprint arXiv: 2310.14558*.

Yilun Zhao, Kaiyan Zhang, Tiansheng Hu, Sihong Wu, Ronan Le Bras, Taira Anderson, Jonathan Bragg, Joseph Chee Chang, Jesse Dodge, Matt Latzke, Yixin Liu, Charles McGrady, Xiangru Tang, Zihang Wang, Chen Zhao, Hannaneh Hajishirzi, Doug Downey, and Arman Cohan. 2025. Sciarena: An open evaluation platform for foundation models in scientific literature tasks. *arXiv preprint arXiv: 2507.01001*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, L. Yu, Susan Zhang, Gargi Ghosh, M. Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. In *Advances in Neural Information Processing Systems* (NeurIPS).

A SciRIFF Provenance

In this section, we provide additional details for SciRIFF introduced in the main body of our paper (§1, §2).

A.1 SciRIFF Task and Schema

We provide detailed information on all tasks–including citations, URLs to source websites, and licensing information where available–in Table 6. SciRIFF task taxonomy is visualized in Figure 4. Where convenient, we use datasets as preprocessed by the BigBio resource (`https://huggingface.co/bigbio`); details will also be provided in the dataset card upon release.

A.2 Task Length Distribution

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

Table 7 compares SciRIFF with selected instruction-following datasets, including canonical collections commonly used for general fine-tuning and selected datasets specialized in scientific domains. Our dataset features longer input contexts than existing resources.

A.3 Instruction Mix Statistics

We further describe our data mixture following the main discussion in §2. Figure 2 presents an overview of the SciRIFF training set distribution over task categories and domains. The domain distribution reflects the current landscape of available high-quality scientific datasets (e.g., Reid et al., 2022), with a notable representation from the biomedicine and AI domain. This aligns with our dataset selection criteria, which prioritize well-documented resources with permissive licenses.

Given the significant presence of information extraction tasks, 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.4). 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.

A.4 Truncation Strategy

In §A.3, 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 Template Ablation

We created two variants of templates for comparison: (1) simple templates adapted from previous work FlanV2, a collection of datasets, templates, and methods for general-purpose instruction tuning (Chung et al., 2024), and (2) LLM-generated templates with GPT-4o.

B.1 Evaluation under Alternative Template

For fair evaluation, we develop corresponding variants of our evaluation templates (for SciRIFF-Eval tasks; §3.1) to ensure that models trained on alternative templates are evaluated on prompts of matching distribution.

B.2 Simple Template

We adapted the style of FlanV2's basic instruction format **while maintaining essential task requirements**. For example, we transformed complex templates into basic input-output patterns (e.g., `Summarize:text\n \n Summary:  \n`) while preserving necessary variable substitutions using "variable" syntax in Jinja. To ensure valid comparison and prevent complete task failure, we maintained minimal but crucial specifications such as

SciRIFF name	Source paper	License	Website
acl_arc_intent_classification	ACL ARC (Bird et al., 2008)	-	[Link]
anat_em_ner	AnatEM (Pyysalo and Ananiadou, 2014)	CC BY	[Link]
annotated_materials_syntheses_events	MatSci Text Corpus (Mysore et al., 2019)	MIT	[Link]
bc7_litcovid_topic_classification	LitCOVID (Chen et al., 2022)	-	[Link]
bioasq_{factoid,general,list,yesno}-qa	BioASQ (Tsatsaronis et al., 2015)	CC BY	[Link]
biored_ner	BioRED (Luo et al., 2022)	-	[Link]
cdr_ner	BioCreative V CDR (Li et al., 2016)	-	[Link]
chemdner_ner	CHEMDNER (Krallinger et al., 2015)	-	[Link]
chemprot_{ner,re}	ChemProt (Krallinger et al., 2017)	-	[Link]
chemsum_single_document_summarization	ChemSum (Adams et al., 2023)	-	[Link]
chemtables_te	ChemTables (Bai et al., 2024)	GPL 3.0	[Link]
chia_ner	Chia (Kury et al., 2020)	CC BY	[Link]
covid_deepset_qa	COVID-QA (Möller et al., 2020)	Apache 2.0	[Link]
covidfact_entailment	CovidFact (Saakyan et al., 2021)	-	[Link]
craftchem_ner	CRAFT-Chem (Cohen et al., 2017)	-	[Link]
data_reco_mcq_{mc,sc}	DataFinder (Viswanathan et al., 2023)	Apache 2.0	[Link]
ddi_ner	DDI (Herrero-Zazo et al., 2013)	CC BY	[Link]
discomat_te	DISCoMaT (Gupta et al., 2023)	CC BY-SA	[Link]
drug_combo_extraction_re	Drug Combinations (Tiktinsky et al., 2022)	-	[Link]
evidence_inference	Evidence inference (DeYoung et al., 2020)	MIT	[Link]
genia_ner	JNLPBA (Collier et al., 2004)	CC BY	[Link]
gnormplus_ner	GNormPlus (Wei et al., 2015)	-	[Link]
healthver_entailment	HealthVer (Sarrouti et al., 2021)	-	[Link]
linnaeus_ner	LINNAEUS (Gerner et al., 2010)	CC BY	[Link]
medmentions_ner	MedMentions (Mohan and Li, 2019)	CC 0	[Link]
mltables_te	AxCell (Kardas et al., 2020)	Apache 2.0	[Link]
mslr2022_cochrane_multidoc_summarization	Cochrane (Wallace et al., 2021)	Apache 2.0	[Link]
mslr2022_ms2_multidoc_summarization	MS^2 (DeYoung et al., 2021)	Apache 2.0	[Link]
multicite_intent_classification	MultiCite (Lauscher et al., 2022)	CC BY-NC	[Link]
multixscience_multidoc_summarization	Multi-XScience (Lu et al., 2020)	MIT	[Link]
mup_single_document_summarization	MUP (Cohan et al., 2022)	Apache 2.0	[Link]
ncbi_ner	NCBI Disease (Islamaj Doğan et al., 2014)	CC 0	[Link]
nlmchem_ner	NLM-Chem (Islamaj et al., 2021a)	CC 0	[Link]
nlmgene_ner	NLM-Gene (Islamaj et al., 2021b)	CC 0	[Link]
pico_ner	EBM-NLP PICO (Nye et al., 2018)	-	[Link]
pubmedqa_qa	PubMedQA (Jin et al., 2019)	MIT	[Link]
qasa_abstractive_qa	QASA (Lee et al., 2023)	MIT	[Link]
qasper_{abstractive,extractive}_qa	Qasper (Dasigi et al., 2021)	CC BY	[Link]
scicite_classification	SciCite (Cohan et al., 2019)	-	[Link]
scientific_lay_summarisation_{elife,plos}_single_doc_summ	Lay Summarisation (Goldsack et al., 2022)	-	[Link]
scientific_papers_summarization__single_doc_{arxiv,pubmed}	Scientific Papers (Cohan et al., 2018)	-	[Link]
scierc_{ner,re}	SciERC (Luan et al., 2018)	-	[Link]
scifact_entailment	SciFact (Wadden et al., 2020)	CC BY-NC	[Link]
scireviewgen_multidoc_summarization	SciReviewGen (Kasanishi et al., 2023)	CC BY-NC	[Link]
scitldr_aic	SciTLDR (Cachola et al., 2020)	Apache 2.0	[Link]

Table 6: Overview of source datasets repurposed for SciRIFF (§2). SciRIFF is licensed under `ODC-By` and is derived from existing scientific literature understanding datasets. { } indicates subsets belonging to the same source.

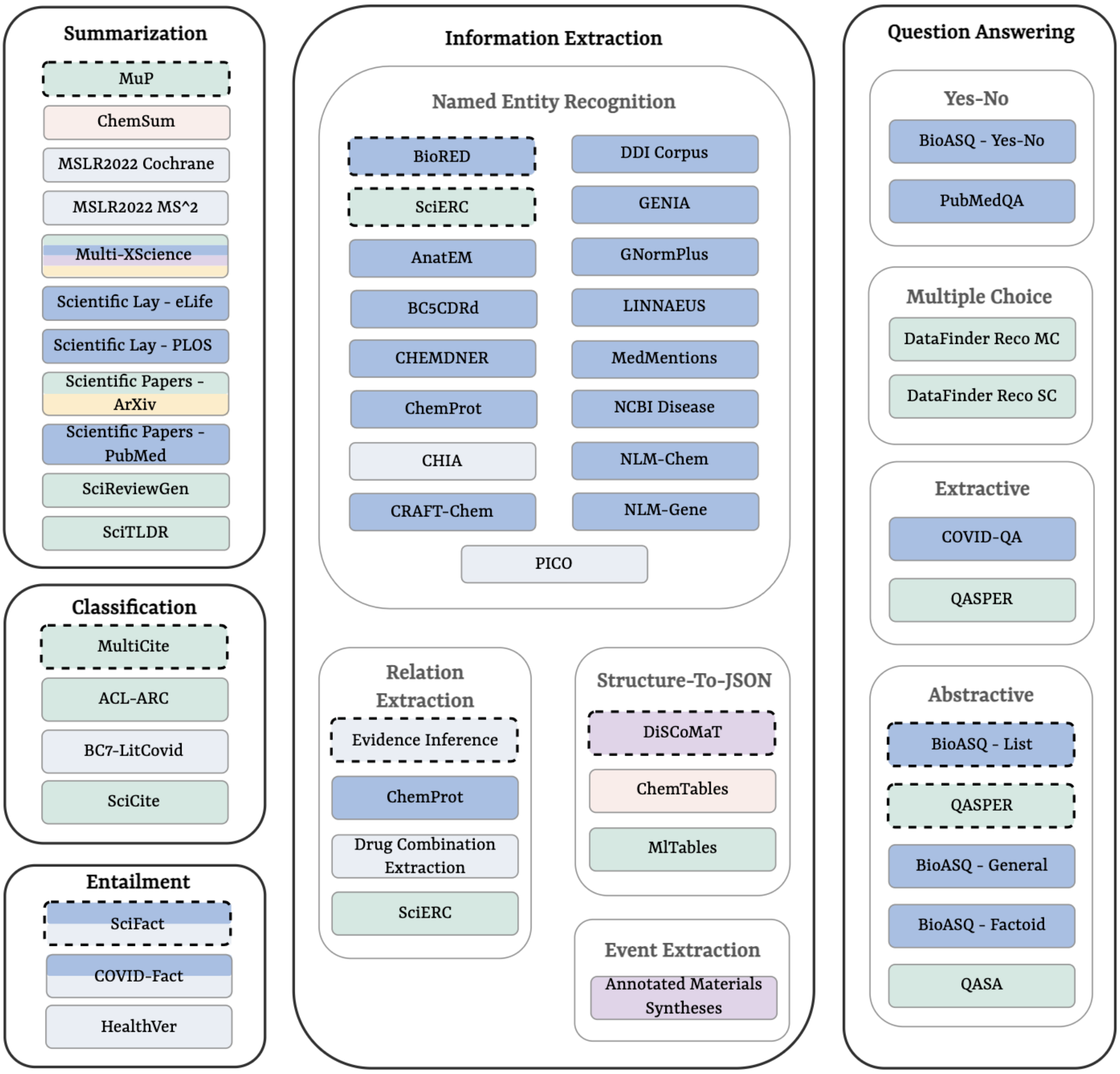


Figure 4: Overview of SciRIFF dataset. Dashed black lines indicate that a task is included in SciRIFF-Eval and held out during model training. Scientific domains are colored as follows: ■Biomedicine; ■AI; ■Clinical Medicine; ■Chemistry; ■Materials Science; ■Miscellaneous.

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., 2023b)	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 7: 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.

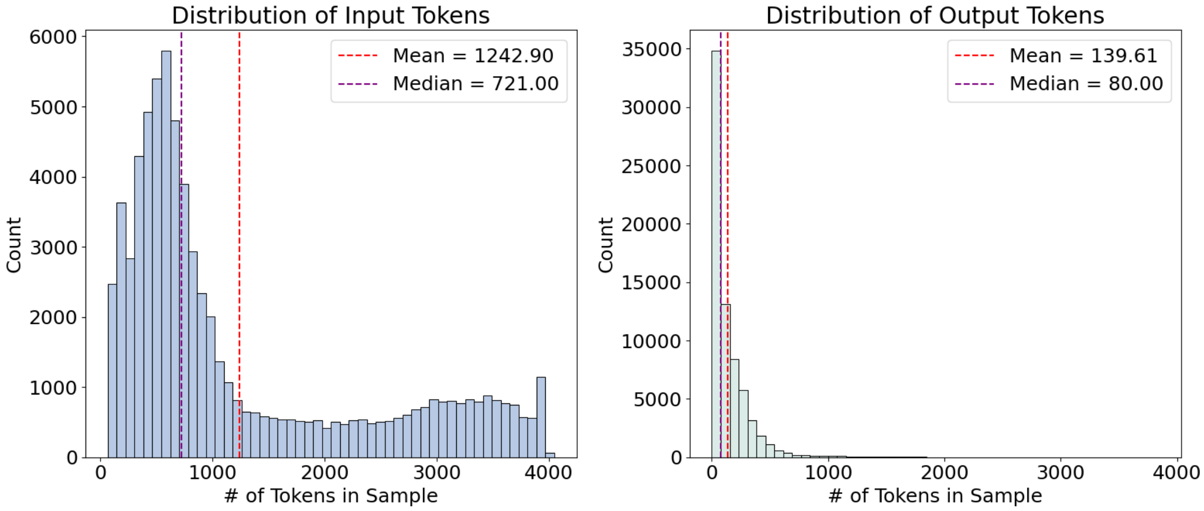


Figure 5: Distribution of input (left) and output (right) token lengths over SciRIFF training instances.

output format requirements (e.g., JSON structure) and output constraints where necessary.

The simple templates still require careful adaptation since previous work typically handles simpler scenarios - their templates rarely need to ground instructions in scientific papers or specify structured output formats. We made deliberate choices to preserve these critical requirements while simplifying the instruction language and reducing template complexity. Figures 6 and 7 show examples of simple templates.

For evaluation, we created `SciRIFF-Eval-Simple`, a variant of our evaluation suite using simple prompts. This ensures that models trained on simple templates are not unfairly evaluated on complex instructions, while still testing the core capabilities required for scientific literature understanding tasks.

B.3 Synthetic Template

We also explored using GPT-4o to generate instruction templates[11].

For each task category, we provided GPT-4o with a canonical example template and detailed specifications including task requirements, input-output structures, and available variables ({{ anchors}}) from our prior post-processing steps (See §2). Generating templates for diverse scientific literature understanding tasks proved challenging. The complexity of our tasks–ranging from evidence-based question answering to structured information extraction– makes it difficult to create a universal prompting strategy.

We provide our prompt template for synthetic template generation in Figure 8.

For evaluation, similar to the approach in §B.2, we created `SciRIFF-Eval-Synthetic`, following the same principle of matching training and evaluation distribution.

B.4 Results and Discussion

Table 8 shows that expert-written templates, which carefully specify task requirements and output structures, outperform the alternatives. Additionally, we observe that the (in-distribution) evaluation for `Synthetic` variants show zero performance on QASPER and SciERC tasks (See Figure 9 and Figure 10–our expert-crafted template–for reference). Upon inspection, we found that GPT-4o[12] failed to specify the required output format correctly, thus the evaluation fails. Nevertheless, when we drop the two tasks, we still see that expert-written templates perform much stronger than the alternatives.

C Training Details

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

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

[11] Initial attempts at naive prompting failed to produce usable templates.

[12] Note that the effort to prompt GPT4o to generate template for diverse and different scientific literature understanding tasks is non-trivial in itself; See Figure 8.

SciERC-simple

Extract all unique entities from the paper abstract.

Output a JSON object where keys are entity types and values are lists of extracted entities.

Abstract:
{{paper}}

|||

{{ ner_dict | tojson }}

Figure 6: simple template for `SciERC` task.

QASPER-simple

Read the following paper excerpts and answer the question. Output a JSON object with "answer" and "evidence" fields.

Paper: {{paper}}

Question: {{question}}

|||

{{output}}

Figure 7: simple template for `QASPER` task.

Model	Data	BioASQ	BioR	DiscMT	EI	MC	MuP	Qasper	SciERC	SciFact	Sci.	Sci.Selected
Llama-3.1-8B	Ours	46.2	84.2	83.9	23.5	71.0	68.5	49.8 / 52.2	56.2	83.3 / 71.9	62.8	65.0
	Simple	57.3	64.4	19.6	4.1	9.4	42.3	49.8 / 65.0	33.7	65.8 / 52.6	42.2	36.6
	Synthetic	41.0	58.1	38.7	0.3	9.1	57.5	0.0	0.0	63.4 / 39.7	28.0	36.7
Qwen2.5-1.5B	Ours	43.6	81.8	45.6	18.9	71.2	67.8	47.0 / 51.4	52.7	78.8 / 70.5	57.2	57.7
	Simple	36.2	43.1	38.5	0.8	2.9	42.5	32.5 / 48.3	28.8	47.9 / 42.2	33.0	29.9
	Synthetic	33.9	39.7	40.9	0.3	7.7	68.5	0.0	0.0	7.2 / 11.8	19.1	28.6

Table 8: Performance on evaluation tasks (SciRIFF-Eval, `SciRIFF-Eval-Simple`, and `SciRIFF-Eval-Synthetic` respectively, across ablations for §4.2 and Appendix B. This table accompanies Table 3. Sci.Selected represents the average score dropping QASPER and SciERC tasks (representative of complex output in SciRIFF), where synthetic templates failed to specify the required complex output formats. We show that our templates show stronger performances under either comparison scheme.

- Effective batch size: 128

For context, each training run of 7B-sized models requires approximately 40 GPU hours on H100 GPUs, making comprehensive ablation studies (on e.g. task mixing ratios) prohibitively expensive for most research labs. We have prioritized our computational resources for experiments that directly address core research questions while maintaining reproducibility for typical computing budgets.

D Evaluation Details

The following pages show full input / output examples for all SciRIFF-Eval tasks, along with details on metric calculations. This information will be available on our project GitHub page. We use gpt-4o-2024-08-06 model for tasks using an LLM judge as evaluation.

Prompt Generation

Today, you will write instruction templates (in Jinja) to format an instruction-following task that a researcher might reasonably ask about scientific literature.

You will be writing templates in Jinja formats. Input field and output field are separated by "|||". Since our Jinja template will likely be a multiline string, please use a block scalar "|" to indicate a multiline string in Jinja. For example:
"""
jinja: |
<input part: most of your instructions will be in this part>
|||
<output part>
"""

Here is the task that you are about to create template for:

{{TASK_DESCRIPTION}}

{{RELEVANT_CONTEXT}}

<—start_of_author_notes—NOT IN ACTUAL PROMPT>
Author notes: We optionally provide “relevant context” in this general format. In actual generation, we drop fields that do not apply.

- task_family: The category to which this task belongs. Options include summarization, ie, qa, entailment, and classification.
- domain: Scientific field(s) that the task covers like "artificial intelligence"
- input_context: Whether the input is full paper text, a table, etc.
- source_type: Indicates whether the input comes from a single paper or multiple sources.
- output_context: Clear text descriptions for output requirements like "Yes or No", json, jsonlines.

<—end_of_author_notes—NOT IN ACTUAL PROMPT>

You should clearly and concisely specify task requirements and any special output structures (if applicable). For tasks that require JSON (or JSON array) outputs, explicitly mention the output requirement in your template.

Here is the list of anchor variables for this task, which are prepared for you:
{{VARIABLES_POSTPROCESSED_BY_EXPERT_ANNOTATORS}}

Important: the content enclosed by "{{" and "}}": should NOT change. You should re-use the verbatim texts for anchor variables.

Here is a template example belonging to the same task category. You should only study the overall structure and the style, but do not copy the content:

{{DEMONSTRATION_FROM_STANDARD_SCIRIFF_INSTRUCTION}}

Make sure your generated template prompt is clear and not verbose.

Figure 8: Template generation prompt for GPT-4o for synthetic templates §B.3. We adapt the prompt for individual tasks. We note that GPT-4o often generate vague and under-specified instructions for our use case.

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.

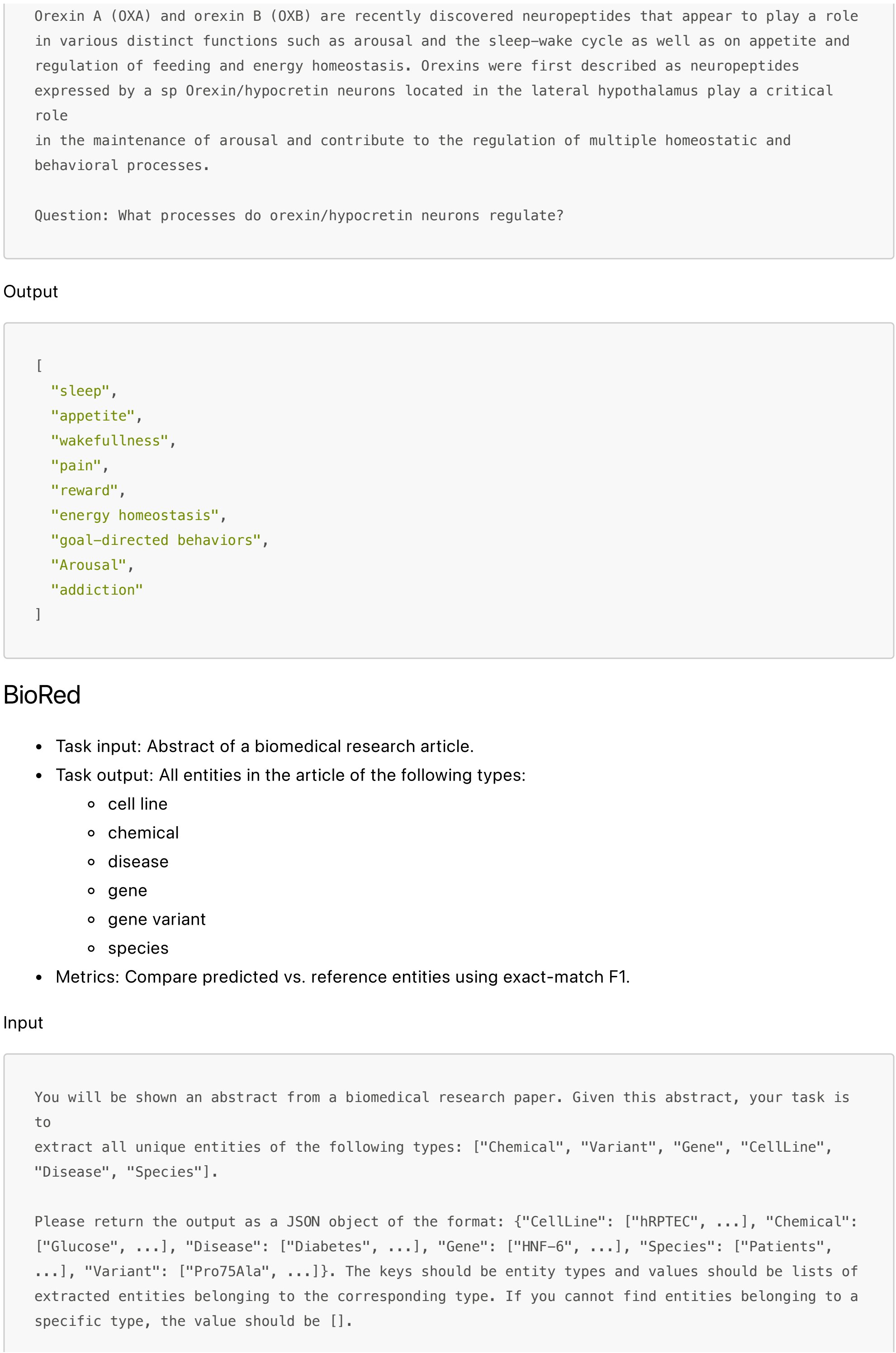
Input

```
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)."
  ]
}
```

E Instruction Template Creation

Instruction templates are written in (Pallets, 2024), Guidelines and best practices" for prompt-writing will be available at our GitHub repository. Each prompt was double-checked by an additional paper author for clarity and correctness.

F Sample template

In this section, we provide examples of our expert-written templates that demonstrate the complexity and precision required for scientific literature understanding tasks, described in §1 and §2.1. These templates are carefully designed to elicit structured outputs while requiring sophisticated capabilities such as information extraction with attribution, multi-step reasoning, and adherence to specific output schemas. The templates shown –QASPER (QA, Figure 9), SciERC (IE, Figure 10), HealthVer (Fact-checking, Figure 11), DiSCoMaT (IE over tabular data, Figure 12), and DataFinder Reco MC (Multiple Choice QA, Figure 13) – demonstrates how our instruction format guides models to perform challenging tasks like answering questions with evidence attribution, extracting nested entity relationships, and verifying scientific claims with supporting rationales.[13]

G Information About Use of AI Assistants

We use OpenAI ChatGPT and Anthropic Claude for grammar checking in manuscript preparation.

[13] Our preliminary experiments showed that even strong proprietary models like GPT-4o struggled to reliably generate such structured outputs without explicit templates. This observation motivated our decision to use expert-written templates.

```
QASPER

You will be shown sections from a scientific research paper, together with a question about the paper. This is an extractive question-answering task, where you must find and extract relevant text spans directly from the paper to answer the question. Your response should strictly be a json object with two fields:

- "answer": An array of strings extracted directly from the paper which, collectively, answer the question.
- "evidence": An array of strings. Each should be an excerpt from the paper, in which one or more of the extracted answers can be found.

For example, for the question "What baselines did the authors compare against?", a sample response might be:
{
answer”: [BERT”,RoBERT”],
evidence”: [In our experiments, we compare the performance of our model against BERT and RoBERTa.”]
}
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".
Paper: {{paper}}

Question: {{question}}

|||

{% if unanswerable %} null
{% else %}
{{ {"answer": answer, "evidence": evidence} | tojson }}
{% endif %}
```

Figure 9: Canonical template for QASPER task in Figure 4. See §F for description.

```
SciERC

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:

{{ org_text }}

|||

{{ ner_dict | tojson }}
```

Figure 10: Canonical template for SciERC task in Figure 4. See §F for description.

HealthVer

You will be shown a claim about public health 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 "wearing masks can prevent the spread of COVID", the output might be:

```
{
"verdict": "SUPPORT",
"evidence": ["Our findings indicate that mass mask-wearing reduces the transmission rate for COVID-19."]
}
```

Claim: {{ claim }}

Abstract:
{{ abstract_with_newlines }}

|||

{{ output_json_with_sentences }}

Figure 11: Canonical template for `HealthVer` task in Figure 4. See §F for description.

DiSCoMaT

{{ table_code_text }}

You are provided with the table above from a materials 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:

|||

{{ json_records }}

Figure 12: Canonical template for `DiSCoMaT` task in Figure 4. See §F for description.

DataFinder Reco MC

You are provided with a research question, keyphrases about the question, a description of candidate datasets and dataset options. Read the description of popular datasets provided below and select the ones that can be used to validate the following research question. Use your knowledge of machine learning datasets to make the best judgement.

Your response should be formatted as a json array. For instance, for the query "Semi supervised image classification", a sample response might be: ["CIFAR-10", "CIFAR-100"]. Do not include any extra text in the response other than the answer array.

Query: {{ query }}

Keyphrases: {{ keyphrase_query }}

Dataset description:
{{ context }}

Options:- {{ options }}

|||

{%- set ans_list = answer.split(", ") %}
{{ ans_list | tojson }}

Figure 13: Canonical template for `DataFinder Reco MC` (QA-multiple choice) task in Figure 4. See §F for description.