Unnatural Instructions: Tuning Language Models with (Almost) No Human Labor

Or Honovich$^{\tau}$ **Thomas Scialom**$^{\mu}$ **Omer Levy**$^{\tau\mu}$ **Timo Schick**$^{\mu}$
$^{\tau}$ Tel Aviv University
$^{\mu}$ Meta AI

Abstract

Instruction tuning enables pretrained language models to perform new tasks from inference-time natural language descriptions. These approaches rely on vast amounts of human supervision in the form of crowdsourced datasets or user interactions. In this work, we introduce *Unnatural Instructions*: a large dataset of creative and diverse instructions, collected with virtually no human labor. We collect 64,000 examples by prompting a language model with three seed examples of instructions and eliciting a fourth. This set is then expanded by prompting the model to rephrase each instruction, creating a total of approximately 240,000 examples of instructions, inputs, and outputs. Experiments show that despite containing a fair amount of noise, training on Unnatural Instructions rivals the effectiveness of training on open-source manually-curated datasets, surpassing the performance of models such as T0++ and Tk-Instruct across various benchmarks. These results demonstrate the potential of model-generated data as a cost-effective alternative to crowdsourcing for dataset expansion and diversification.

1 Introduction

Instruction tuning enables pretrained language models to generalize to unseen tasks in a zero-shot setting (Sanh et al., 2021; Wei et al., 2021). One way to collect examples of instructions and their execution is to reformulate existing NLP datasets in an explicit instruction-input-output format via prompt engineering (Mishra et al., 2022; Wang et al., 2022). However, the resulting data is limited to existing academic benchmarks, even though the instruction paradigm can describe any text-based task (Efrat and Levy, 2020). Alternatively, Ouyang et al. (2022) collect user-generated prompts and manually annotate their expected outputs, reflecting a different (and arguably more desirable) distribution of the instruction space, but requiring a live application with existing users and major investments in human annotation. Can we create a large dataset of instructions that is diverse in tasks, content, and phrasing, *without* human labor?

We introduce **Unnatural Instructions**, a dataset of natural language instructions and their corresponding inputs and outputs. Inspired by recent work on utilizing language models for data generation (Schick and Schütze, 2021b; Lee et al., 2021; Liu et al., 2022a), we collect data in a fully automatic manner by prompting a pretrained language model with three examples from the Super-Natural Instructions[1] dataset (Mishra et al., 2022; Wang et al., 2022) and asking the model to generate a fourth (Figure 1). We repeat this process with 5 different seeds – i.e. the entire process requires only 15 instruction examples – to automatically produce 64,000 diverse triplets of instructions, inputs, and outputs.[2] We further diversify the dataset's format by generating additional natural language paraphrases of each instruction, while preserving the contents of any input arguments and outputs, expanding the dataset to approximately 240,000 examples. Although the dataset contains noise, our analysis reveals that more than 50% of generated examples are indeed correct, and that even incorrect examples typically contain valuable information for instruction tuning. At the same time, we find that Unnatural Instructions contains highly creative tasks – some of which are very different from "classic" NLP tasks – and has a more diverse set of instructions than Super-Natural Instructions.

Experiments show that fine-tuning an 11B-parameter T5 model (Raffel et al., 2020) on Unnatural Instructions can outperform both T0++ (Sanh et al., 2021) and Tk-Instruct (Wang et al., 2022) across several benchmarks, including Super-Natural Instructions (Wang et al., 2022), BIG-

[1] Also known as Natural Instructions v2.

[2] In practice, we collected 68,478 examples, but only used subsets of 64,000 examples for training.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 14409–14428

bench Hard (Suzgun et al., 2022), and LMentry (Efrat et al., 2022). When controlling for all variables besides the data, we find that a model trained on Unnatural Instructions performs competitively with a baseline model trained on Super-Natural Instructions. In particular, we observe an 18-point gain on BIG-bench Hard (original task formulation) and a 16-point gain on LMentry, suggesting that Unnatural Instructions is particularly useful for generalizing to instructions that deviate from the distribution of classic NLP tasks. These improvements become even more pronounced when the cost of generating examples is amortized; in this case, training on Unnatural Instructions substantially outperforms our baseline on all benchmarks. We observe a log-linear relationship between the number of generated examples and downstream task performance, suggesting that performance of models trained on Unnatural Instructions can further be improved simply by increasing its size.

Beyond the immediate implications on instruction tuning, this work demonstrates the viability of automatic dataset expansion using language models as an alternative to crowdsourcing. Unnatural Instructions highlights the ability of language models to produce creative and diverse data, a trait that is difficult to obtain with crowd workers, who lack the intrinsic motivation to create novel examples and typically collapse into predictable heuristics to form annotation artifacts (Gururangan et al., 2018). At the same time, language models are faster and cheaper than human labor, opening up new possibilities for scaling up data annotation.

2 Data Collection

We introduce Unnatural Instructions, a dataset of 240,670 diverse natural language instructions. Each example contains a natural language instruction as input and its expected execution as output. Table 2 displays examples from the dataset.

Unnatural Instructions is collected in a completely automatic process, requiring a seed of only 15 manually-constructed examples, which can be produced in about one hour of human labor. We first collect a core set of 68,478 examples (§2.1) by prompting a pretrained language model M with a seed of 3 manually-annotated examples to produce a new (fourth) example. This phase uses a structured instruction format and filtering heuristics to ensure data quality. We then expand the core dataset by rephrasing the structured instructions in

Example 1

Instruction: You are given a science question (easy-level) and four answer options (associated with "A", "B", "C", "D"). Your task is to find the correct answer based on scientific facts, knowledge, and reasoning. Do not generate anything else apart from one of the following characters: 'A', 'B, 'C', 'D'. There is only one correct answer for each question.

Input: Which part of a bicycle BEST moves in a circle? (A) Seat (B) Frame (C) Foot pedal (D) Kickstand

Constraints: The output should be one of the following characters: 'A', 'B, 'C', 'D'.

Example 2

Instruction: You are given a negative review and your task is to convert it to a positive review by one or more making minimal changes. Avoid changing the context of the review.

Input: we stood there in shock, because we never expected this.

Constraints: None.

Example 3

Instruction: In this task, you are given two sentences taken from a conversation, and your job is to classify whether these given sentences are sequential or not. We will mark the given sentence pair as 'True' if it's sequential, otherwise 'False'. The two sentences are spoken by two different people.

Input: Noah: When and where are we meeting? :), Madison: I thought you were busy...?

Constraints: None.

Example 4

Instruction: In this task, you will be given a profile of someone and your job is to generate a set of interesting questions that can lead to a conversation with the person.

Input: Yvonne has been playing the violin since she was four years old. She loves all kinds of music, but her favorite composer is Bach.

Constraints: None.

Figure 1: Our data generation prompt. **Blue**: The meta-prompt, which contains the number of the in-context example, as well as the constant fields of each example: instruction, input, and constraints. **Black**: The in-context examples. We show here one of our 5 in-context seeds. **Pink**: One of the model's generations for the given prompt.

free-form natural language (§2.2). This expansion is performed automatically by prompting a language model with manually-constructed examples, scaling up the dataset more than 3-fold. Throughout this section, we use OpenAI's text-davinci-002 as M. See §6 for experiments with other models.

2.1 Core Dataset Generation

The core dataset consists of examples in a structured format, making it easier for the generating model M to predict and for us to filter automatically. We use stochastic decoding to generate example inputs (to promote creativity), followed by deterministic decoding to generate their outputs (for accuracy). Figure 2 illustrates the process.

Format Each example in the core dataset contains four fields: (1) An **instruction** describing the task. The instruction can be a generic template (e.g. "Write whether the following review is positive or

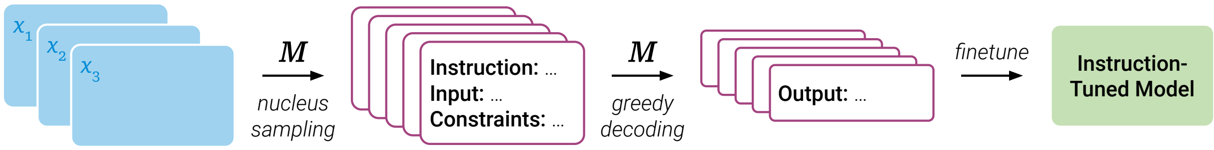

Figure 2: The core Unnatural Instructions generation pipeline. We use a seed of three in-context demonstrations x_1, x_2, x_3 to create a large dataset of NLP tasks with instructions, inputs and outputs. As a first step, we sample instructions, inputs, and constraints from a language model M. In the next step, we use M to deterministically generate the corresponding outputs. Finally, the data can be used for instruction tuning.

negative") that can be instantiated by a particular input argument (e.g. the review itself). (2) The **input** argument that instantiates the instruction, creating a specific example of the task. (3) Output space **constraints**, which detail the restrictions on the task's output space. Constraints are mainly relevant for classification tasks; for tasks with no specific output space constraints, this field is "None." (4) A textual **output** reflecting a correct execution of the instruction given the input arguments and output space constraints. The first three fields (instruction, input argument, constraints) are the model's input, and the output field acts as the reference for training and/or evaluation. The constraints field is meant to guide M during output generation and is discarded after generating the outputs (see next). In Appendix D we provide data-driven evidence for selecting this particular format.

Input Generation We first generate examples of instruction-input-constraints by prompting a model with three task demonstrations x_1, x_2, x_3, each presented in the structured format (without outputs). These demonstrations are wrapped by a simple meta-prompt that incentivizes the model to create a fourth example x_4, as illustrated in Figure 1.

We use 5 seeds of 3 demonstrations each to generate the core dataset; i.e., the whole process requires only 15 examples. Demonstrations are taken from the Super-Natural Instructions (Wang et al., 2022) train set. To obtain various examples using the same prompt, decoding is done by nucleus sampling with $p = 0.99$ (Holtzman et al., 2020).

Filtering We apply three automatic filters to the generated examples to remove: (1) model generations that do not include the three input fields (instruction, input argument, and constraints), (2) instructions and inputs that are identical to those demonstrated in the prompt, (3) duplicate examples, i.e. two different examples that have the same instruction and input argument.

Output Generation Given a generated example x, we generate the corresponding output y by conditioning a pretrained language model with the instruction, input argument, and constraints (if not none), followed by an "Output:" prompt. Here we apply greedy decoding to prioritize correctness over creativity. We ignore examples for which the generated output is an empty string.

2.2 Template Expansion

Examples in our core dataset have a strict instruction-input-output format. To increase the format diversity and obtain tasks phrased in free-form natural language (Schick and Schütze, 2021a; Sanh et al., 2021), we collect alternative formulations that preserve the content of the original instructions. Specifically, we prompt a language model to reformulate the core dataset tasks and collect two alternative formulations for each generated task.[3] Alternative formulations are often shorter and less formal than the original instructions. The rephrasing prompt contains two examples of instructions and their alternative formulation. We do not include inputs, constraints, and outputs in the rephrasing prompt; instead, we utilize the already-generated inputs and outputs to complement the rephrased instruction. Unlike the examples in the core dataset, the input is often embedded into the task description. We achieve that by adding an "{INPUT}" placeholder, which marks the position for input insertion (Figure 3).

In some cases, the model generates two identical reformulations, or it copies the original instruction. Some alternative formulations may also have an invalid format - e.g., not containing the "{INPUT}" placeholder. When such failures occur we continue to sample reformulations, stopping after five unsuccessful attempts. Consequently, some instructions have only one alternative formulation, while others have none. Overall, more than 97.5% of the

[3]The seed reformulations in each prompt are inspired and partially taken from PromptSource (Bach et al., 2022).

Example 1

Instruction: In this task, you are given an article. Your task is to summarize the article in a sentence.

Input: {INPUT}

Alternative formulation: My college roommate asked me what this article means: "{INPUT}". So I recapped it in layman's terms:

Example 2

Instruction: This task is about writing a correct answer for the reading comprehension task. Based on the information provided in a given passage...

Input: {INPUT}

Alternative formulation: {INPUT} Based on the given context, the answer to the question is

Example 3

Instruction: In this task, you are asked to determine whether the given recipe is for a savory or sweet dish. If it is for a savory dish, output "SAVORY". If the recipe is for a sweet dish, output "SWEET".

Input: {INPUT}

Alternative formulation: Given the following recipe, {INPUT}, is the dish savory or sweet? Your output should be "SAVORY" or "SWEET"

Figure 3: Our template expansion prompt. **Black**: Few-shot demonstrations of instructions and alternative formulations. **Blue**: The instruction we wish to paraphrase. **Pink**: Model-generated task reformulation.

instructions have two distinct, valid reformulations.

In fact, some instructions end up with more than two paraphrases because we generate two paraphrases per *example* (i.e. instruction-input-output pair) and the core dataset contains examples that share the exact same instruction but not the same input argument. Therefore, by cross-referencing each instruction's alternative phrasings with all of its input arguments, we can extend the data even further and arrive at a total of 240,670 examples without additional cost.

3 Data Analysis

We first demonstrate the *creativity* of Unnatural Instructions, and then manually analyze 200 randomly-sampled examples from our core dataset, focusing on *correctness* and *diversity*. We also compare our data's distribution to Super-Natural Instructions, and find our inputs to be more diverse.

Creativity A major challenge when creating an instruction dataset is task creativity. Crowd workers typically collapse into predictable heuristics to form annotation artifacts (Gururangan et al., 2018). While the high performance of models trained on Unnatural Instructions (see §5) suggests that it is indeed diverse and creative, we also present in Table 1 some cherry-picked examples, providing a glimpse at their creativity.

Correctness When evaluating correctness, we test whether (1) the generated instructions are logical and executable, (2) the input arguments correspond to the task described in the instruction, and (3) the outputs are correct, given the instruction and input. Although our data filtering process is minimal, 113 of the 200 analyzed examples (56.5%) are correct. Of the 87 incorrect examples, 9 (4.5%) had incomprehensible instructions, 35 (17.5%) had an input that did not match the task description, and 43 (21.5%) had incorrect outputs. Table 2 shows some correct and incorrect examples from our analysis.

While the amount of noise in the data may raise concerns regarding its usability, many of the examples that were marked as incorrect can still be considered informative. For example, one erroneous example had the instruction *"In this task, you will be provided with a list of countries and their corresponding capital cities. You are also given a list of clues...For each clue, determine which country it is referring to and write down that country's name..."* The input argument was *"Clue 1: This capital city is on two different continents."* This example is incorrect since the input does not conform with the format described by the instruction – a list of countries and their capitals is not provided, only a clue. However, the output is *Istanbul, Turkey*, which indeed lies in both Europe and Asia and therefore corresponds with the input clue. In §5 we show that, despite being noisy, Unnatural Instructions provides a highly informative training signal.

Diversity We manually cluster the instructions into tasks and measure the number of unique types. Out of the 200 examples tested, we identify 117 distinct tasks. While many tasks are classical NLP tasks, such as sentiment analysis, question answering, and summarization, others are not quite canonical, and some are very specific, such as detecting a recipe given a list of ingredients. Table 3 shows the most commonly generated tasks from the set of 200 analyzed examples. Other tasks appeared 3 times or less, with 85 tasks appearing only once.

We also analyze how similar each pair of examples is, as a general proxy for diversity. Specifically, we sample 10,000 pairs of examples from Unnatural Instructions, and compute the similarity of their inputs using BERTScore (Zhang et al.,

Instruction	Category
You need to answer the question 'Is this a good experiment design?', given an experiment scenario. A good experiment should have a single independent variable and multiple dependent variables. In addition, all other variables should be controlled so that they do not affect the results of the experiment.	Experiment Verification
You are given a recipe for baking muffins that contains some errors. Your task is to correct the errors in the instructions by replacing each underlined word with the correct one from the options provided.	Recipe Correction
You will be given a piece of text that contains characters, places, and objects. For each character in the text, you need to determine whether they are static or dynamic. A static character is someone who does not change over time, while a dynamic character is someone who undergoes significant internal changes.	Character Categorization
In this task, you are asked to generate a limerick given two rhyming words. A limerick is a five-line poem with the following rhyme scheme: AABBA. The first, second and fifth lines must be of three beats, while the third and fourth lines must be of two beats each. Additionally, all poems should have the same meter (e.g., iambic pentameter)	Poem Generation
I'm not sure what this idiom means: "{INPUT}". Could you give me an example?	Idiom Explanation
{INPUT} By analyzing the writing styles of the two passages, do you think they were written by the same author?	Author Classification
I need to invent a new word by combining parts of the following words: {INPUT}. In what order should I put the parts together?	Word Invention
What is the punchline to the following joke? {INPUT}	Humor Understanding

Table 1: Examples of eight interesting generated instructions and their corresponding category. The first four examples are taken from the core dataset, while the last four were generated during the template expansion phase.

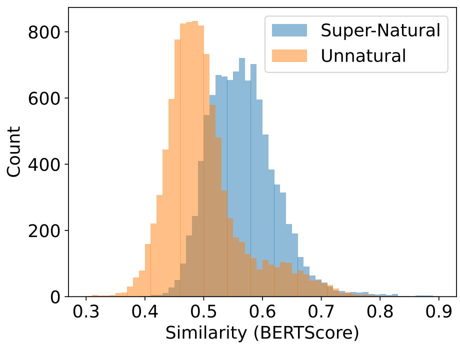

Figure 4: Similarity scores distribution for Super-Natural Instructions and for Unnatural Instructions, obtained by sampling 10,000 pairs of examples from each dataset and computing their similarity.

2020). We repeat this process for Super-Natural Instructions, producing two empirical distributions. Figure 4 shows that the inputs of Unnatural Instructions tend to be less similar to each other than the inputs of Super-Natural Instructions. This result comes as a surprise considering the fact that the entire Unnatural Instructions dataset was constructed by conditioning only on 15 original examples.

4 Experimental Setup

We describe model fine-tuning on Unnatural Instructions and our evaluation protocol.

4.1 Fine-Tuning on Unnatural Instructions

We fine-tune T5-LM, the language-model-adapted variant of T5-11B (Raffel et al., 2020; Lester et al., 2021). We follow standard practice for fine-tuning, using a batch size of 16 examples over 3 epochs. For training on our core dataset, we use the same template as Wang et al. (2022) for formatting instructions and inputs. Our full set of training hyperparameters is available in Appendix A. We create a small validation set of 1,000 examples for model selection following the methodology proposed by Wang et al. (2022): we randomly select 10 examples from 100 random tasks of the Super-Natural Instructions training set.

4.2 Baselines

We measure the relative utility of Unnatural Instructions by comparing it to a variety of models, all based on T5-11B, which were fine-tuned with different types and quantities of manually-annotated instruction data.

T0++ (Sanh et al., 2021) is an instruction-tuned variant of T5-LM, trained on tasks in the PromptSource (Bach et al., 2022) prompt formats.

Tk-Instruct Wang et al. (2022) fine-tune T5 v1.1 on Super-Natural Instructions, using a subsample

Example
Instruction: In this task, you are asked to come up with a design for an office lobby. The client has provided specific instructions on what they want in the lobby, and it is your job to create a floor plan that meets all of their requirements. (✓) **Input:** Client wants a welcoming environment with space for people to wait comfortably without feeling cramped. They also would like some sort of area where people can do work if they need to kill some time before their next meeting. (✓) **Output:** [...] Based on these requirements, a possible floor plan [...] The lobby would have a reception area [...] with access to outlets, and a coffee station. (✓)
Instruction: You will be given several pieces of information about an event, and you have to determine whether or not it is a cause-and-effect relationship. If the given statements are related by cause and effect, then output 'True'. Otherwise, output 'False'. (✓) **Input:** Statements: ['The tornado damaged the city', 'Many people were left homeless']. (✓) **Output:** True (✓)
Instruction: You are to determine if the statement is true or false. (✓) **Input:** Text: Santa Claus lives at the North Pole. (✓) **Output:** False (✗)
Instruction: You are given a scientific experiment scenario and your job is to determine which variable(s) the scientist should control in order to test the hypothesis. (✓) **Input:** The hypothesis is "If a plant receives more sunlight, then it will grow taller." Which variable(s) should the scientist control? (✗) **Output:** A (–)

Table 2: Examples of generated instructions, inputs and outputs in our core dataset. For the first two examples, the entire pair of instruction, input and output is valid. The third example has an incorrect output; in the fourth example, the experiment is not described in the input.

of 757 tasks with 100 examples each. Tk-Instruct is trained with a batch size of 1,024 examples for 1,000 steps. Since our evaluation focuses on zero-shot instruction understanding, we use the definition-only version of Tk-Instruct.

FLAN-T5 Chung et al. (2022) fine-tune T5 on a collection of tasks phrased as instructions in multiple prompting setups (zero-shot, few-shot, Chain-of-Thought (Wei et al., 2022)), achieving impressive zero-shot generalization capabilities.

T5-LM on Natural Instructions Our main point of comparison is the utility of the original manually-curated instructions in Super-Natural Instructions. We therefore train a model which is identical to ours in all aspects but data. Specifically, we fine-tune the LM-adapted variant of T5-11B on a subsample of 64,000 examples from Super-Natural Instructions training set, excluding examples from any task that participates in the validation set. This model differs from Tk-Instruct along three aspects: the dataset subsample, the base model (T5-LM), and some training hyperparameters (batch size 16 for 3 epochs).

Task	#Examples
Question Answering	11
Sentiment Analysis	10
Arithmetic	8
Geometry	8
Event Ordering	7
Fact Verification	5
Fill-in-the-Blank	5
General Math Puzzles	4
Identifying Overlapping Strings	4
Array Manipulations and Puzzles	4

Table 3: Top 10 tasks by #examples, out of the 200 manually-analyzed Unnatural Instructions examples.

4.3 Evaluation

We evaluate models on four different benchmarks, measuring a range of capabilities. All evaluations are carried out in a zero-shot setting, without few-shot demonstrations, unless explicitly provided in the instructions. See the full evaluation details in Appendix B.

Natural Instructions We evaluate models on the test set of Super-Natural Instructions (Mishra et al., 2022; Wang et al., 2022). As in the original papers, outputs are generated using greedy decoding, and performance is measured using Rouge-L.

T0: Zero-Shot We evaluate models on the held-out set of T0 (Sanh et al., 2021), using rank classification for decoding and accuracy as a metric. For fair comparison, we remove tasks supersets of which are present in the Tk-Instruct training set. The final set contains six tasks: ANLI R1-R3, CB, COPA and RTE. We refer to this evaluation set as T0: Zero-Shot. Unlike Super-Natural Instructions, T0: Zero-Shot tasks do not have a strict format and are phrased in a rather free-form manner, including inputs that can be embedded into the task description. We therefore expect models trained on our core dataset (without instruction paraphrases) to perform poorly under these conditions, while adding the task reformulation data should boost performance on T0: Zero-Shot.

BIG-bench: Hard The "hard" subset of BIG-bench (Suzgun et al., 2022) contains 23 challenging tasks from BIG-Bench (Srivastava et al., 2022). We investigate two different formats for all tasks:

their original format in BIG-bench, and the format of Suzgun et al. (2022), who reformulate each task as question answering with manually added instructions; for the latter, we remove all few-shot demonstrations. For both formats, we use greedy decoding and exact match with the reference for evaluation.

LMentry LMentry (Efrat et al., 2022) is a benchmark that tests basic language abilities, designed to complement common approaches for evaluating large language models. Outputs are generated by applying greedy decoding and evaluated using high-accuracy regular expressions. The benchmark's metric is the LMentry score, which combines accuracy with multiple aspects of robustness.

5 Results

Our main results are shown in Table 4, which reports the performance of each model on every benchmark. Remarkably, T5-LM finetuned on Unnatural Instructions outperforms several strong instruction-tuned baselines such as T0++ and Tk-Instruct; the only exception to this is BIG-bench: Hard (Orig), where T0++ performs better. Retraining a model on Super-Natural Instructions using our exact setup reveals a significantly better baseline than Tk-Instruct, using the same data. However, even in this direct comparison, Unnatural Instructions leads to stronger or equal performance for every dataset except Super-Natural Instructions itself. While T5-LM finetuned on Unnatural Instructions is outperformed by FLAN-T5, that model was trained on approximately 60 times more data. These results demonstrate that automated data generation with pretrained LMs is a viable and cost-effective alternative to human-curated data.

5.1 Performance with Template Expansion

We evaluate the contribution of template expansion (§2.2) to the performance of models trained on Unnatural Instructions. To this end, we finetune a single model on our full dataset with paraphrases; results are shown in the bottom row of Table 4.

Adding instruction paraphrases boosts performance on T0: Zero-Shot (+3.3), Big-bench: Hard in its original format (+12.1) and LMentry (+8.7). We surmise that this improvement is largely because examples in our core dataset were generated based on demonstrations from Super-Natural Instructions only and therefore have their exact format and style. Accordingly, models trained on our core dataset rely too much on this specific format and cannot generalize well to different formats found in other benchmarks. Obtaining more format diversity through template expansion successfully addresses this issue. On the other hand, over-reliance on the format of Super-Natural Instructions is probably preferable when testing on this dataset itself, which explains the performance drop when adding paraphrases compared to the boost in performance on other benchmarks.

While some of the performance gains observed may also be attributed to the fact that adding paraphrases simply increases the data, in §5.2 we show that template expansion is helpful even when controlling for dataset size.

5.2 Performance Scaling by Dataset Size

As all of our data is generated from the same model using the same set of prompts, scaling up the amount of generated examples might lead to numerous repetitions and, as a consequence, diminishing returns in terms of downstream task performance. To investigate whether this is an issue, we analyze how the amount of training examples affects the performance of our finetuned models. To this end, we train models on subsets of both Super-Natural Instructions and Unnatural Instructions, ranging from 250 to 64,000 examples. As shown in Figure 5, our core and full data as well as Super-Natural Instructions all exhibit log-linear scaling laws, suggesting that even for subsets of Unnatural Instructions containing thousands of examples, simply generating more examples still adds a valuable signal to our training data.

Results for LMentry (Figure 5) show that our template expansion process is still beneficial when controlling for dataset size. The added value of the paraphrases is therefore likely to be in terms of format diversity rather than solely as a method for increasing the amount of data.

5.3 Performance Scaling by Cost

In practical scenarios with fixed annotation budgets, the actual *cost* associated with a certain level of performance is even more relevant than the number of required examples. We therefore measure model performance as a function of the cost for obtaining the training data. Based on OpenAI's pricing as of December 2022, the cost for generating an example is estimated at \$0.02 for our core dataset, and \$0.01 for the expanded dataset. Kiela et al. (2021) estimate human annotation cost at \$0.50–\$1.00 per

Model	#Examples	Super-Natural Instructions	T0: Zero-Shot	BIG-bench: Hard (Orig/QA)	LMentry
Prior Work					
T5-LM	0	24.3	40.2	0.0 / 0.7	20.6
T0++	12,492,800	40.3	NHO	20.2 / 13.9	38.3
Tk-Instruct	75,417	45.6	41.4	5.8 / 11.8	35.7
FLAN-T5	14,336,000	NHO	NHO	<u>39.3</u> / <u>40.0</u>	<u>52.2</u>
Direct Comparison Baseline					
T5-LM on Super-Natural Instructions	64,000	<u>**54.0**</u>	44.0	10.2 / **29.7**	34.6
Our Approach					
T5-LM on Unnatural Instructions	64,000	51.9	45.7	16.0 / 29.5	42.0
+ Instruction Paraphrases	240,670	49.3	<u>**49.0**</u>	**28.1** / 29.4	**50.7**

Table 4: Model performance on four benchmarks. Best results in our direct comparison setup are bold, best results overall are underlined. NHO indicates that a benchmark's data is *not held out* because it was used for training.

example, excluding indirect costs such as task design and UX development; for comparison with our automatic data collection method, we assume the lower-bound human annotation cost of $0.50.

As shown in Figure 5, Unnatural Instructions is clearly more cost-efficient than manually curated data. This is true even for the Super-Natural Instructions test set, where a model trained on Unnatural Instructions is weaker than a model trained on Super-Natural Instructions for a fixed number of examples, but better when controlling for cost, showing that our automatic approach outperforms crowdsourcing for a fixed annotation budget.

6 Generative Model Ablations

As a data generation model, we used text-davinci-002, an instruction-tuned variant of GPT-3 (Brown et al., 2020). However, our approach is not limited to this specific model. We experiment with original (untuned) GPT-3 model by using it as the model M in both the input generation and output generation phases (see §2). We train models for 1,500 steps using 2,000 examples and evaluate the Super-Natural Instructions validation set performance as a proxy, averaged across three different random seeds.

Table 5 shows how replacing an instruction-tuned model with a vanilla model affects the quality of the data. We observe that while the quality of generated *inputs* does drop by 4.5 points, it is well within the range of other prompt ablations (see Appendix D). In other words, informative and diverse *instructions* can be generated by untuned language models. However, generating *outputs* does seem to require instruction tuning. A manual analysis reveals that outputs generated by GPT-3 mainly suffer from the model's inability to stop, often starting with the correct answer, but then degenerating into repetitions or tangents. While this may be remedied through various post-processing heuristics, we leave exploration of such methods to future work.

Model Used to Generate: Input	Output	Super-Natural Instructions
text-davinci-002	text-davinci-002	**48.7** $\pm$ 0.3
GPT-3	text-davinci-002	44.2 $\pm$ 0.7
GPT-3	GPT-3	4.1 $\pm$ 0.1

Table 5: Performance of 11B T5-LM models trained on 2,000 examples, generated with different models, on the Super-Natural Instructions validation set.

7 Related Work

Instruction Tuning Efrat and Levy (2020) propose the Instruction Paradigm, where models learn new tasks from natural language instructions alone. Mishra et al. (2022); Wang et al. (2022) construct the first large-scale instruction benchmarks by collecting crowdsourcing instructions used to create NLP datasets and converting them into a uniform format. Sanh et al. (2021); Wei et al. (2021) further extend the usability of instructions by suggesting *instruction tuning*, where a language model is trained on many natural language instructions in the hope that it will generalize to new, unseen instruction tasks. Chung et al. (2022) advance instruction tuning by scaling the number of tasks, scaling the model size, and adding chain-of-thought (Wei et al., 2022), while Ouyang et al. (2022) propose a reinforcement learning approach for instruction tuning from comparative human judgements.

Automatic Data Generation Obtaining large-scale supervised data can be expensive and time-consuming, making automatic data generation appealing. A common approach is to automatically augment existing datasets (Anaby-Tavor et al., 2020; Andreas, 2020; Yang et al., 2020; Kaushik

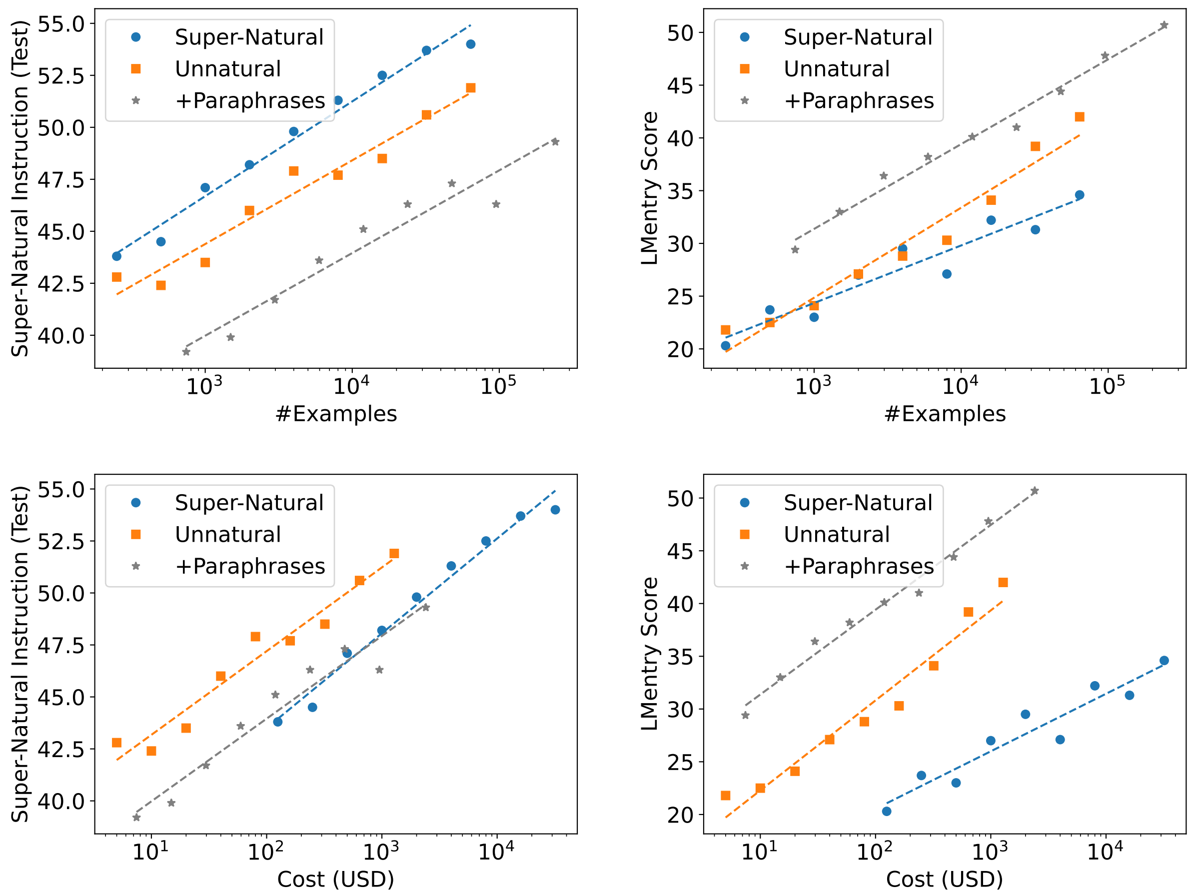

Figure 5: Scaling experiments comparing Unnatural Instructions with Super-Natural Instructions. **Top row:** Model performance when controlling for *dataset size*, tested on Super-Natural Instructions (left) and LMentry (right). **Bottom row:** Model performance when controlling for the *cost of obtaining data*, tested on Super-Natural Instructions (left) and LMentry (right).

et al., 2020; Lee et al., 2021, *inter alia*). Kiela et al. (2021) suggest a human-and-model-in-the-loop dataset creation; In the same manner, Nie et al. (2020) apply a process to create training data for the task of NLI (Dagan et al., 2006; Bowman et al., 2015). Liu et al. (2022a) combine human annotators and GPT-3, create challenging NLI examples.

Other work suggested creating datasets entirely automatically, without the need for labeled data. Schick and Schütze (2021b) and Ye et al. (2022) propose to leverage pretrained language models to generate entire labeled datasets from scratch, for a given, predefined task. Agrawal et al. (2022) use pretrained language models to automatically construct multilingual QA data using only five examples per language.

8 Conclusion

We introduce Unnatural Instructions, an automatically generated dataset of natural language instructions and their corresponding inputs and outputs. To the best of our knowledge, this is the first general-purpose NLP dataset that was automatically generated. Our experiments show that models trained on Unnatural Instructions outperforms models trained on manually annotated datasets across several benchmarks. Unnatural Instructions is not only cost-effective, we also provide evidence of enhanced diversity in the instructions produced and a high level of creativity in the tasks devised, a trait difficult to obtain with crowd workers. Ablations show that even weaker models without instruction tuning can generate useful instructions, though they may struggle with producing the corresponding outputs. However, coming up with interesting tasks and writing diverse instructions is arguably the main challenge of the data collection process, whereas given instructions and inputs, outputs are often far easier to annotate through crowdsourcing. Our findings incentivize utilizing models for general-purpose data generation, which we view as an intriguing direction for future research.

9 Limitations

We point at some directions for future improvements in automatic instruction generation.

First, as shown in §3, Unnatural Instructions contains noisy examples, in which either the instruction, input, or output are invalid. Future work may focus on developing better filters for such examples - e.g., by annotating a subset of examples as either valid or not and training a classifier for determining the correctness of generated instances (West et al., 2022; Liu et al., 2022a).

Second, future work may employ a human-in-the-loop approach, where humans should recognize challenging patterns, encouraging models to generate more complex examples (Liu et al., 2022a). In another human-in-the-loop scenario, models trained on Unnatural Instructions can be queried by humans to find examples on which these models fail, thus collecting harder examples (Nie et al., 2020).

Finally, language models are known to sometimes reflect undesirable biases present in their training data. Automatically generated data may therefore contain such content. We note that during our manual analysis, we did not notice any harmful examples. Still, future work may consider applying a filtering mechanism to reduce the risk of having biased content.

References

Priyanka Agrawal, Chris Alberti, Fantine Huot, Joshua Maynez, Ji Ma, Sebastian Ruder, Kuzman Ganchev, Dipanjan Das, and Mirella Lapata. 2022. Qameleon: Multilingual qa with only 5 examples. *arXiv preprint arXiv:2211.08264*.

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, N. Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *AAAI Conference on Artificial Intelligence*.

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566, Online. Association for Computational Linguistics.

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. 2022. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, Dublin, Ireland. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Avia Efrat, Or Honovich, and Omer Levy. 2022. Lmentry: A language model benchmark of elementary language tasks.

Avia Efrat and Omer Levy. 2020. The turking test: Can language models understand instructions?

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. 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 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Sawan Kumar and Partha Talukdar. 2021. Reordering examples helps during priming-based few-shot learning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4507–4518, Online. Association for Computational Linguistics.

Kenton Lee, Kelvin Guu, Luheng He, Tim Dozat, and Hyung Won Chung. 2021. Neural data augmentation via example extrapolation.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022a. Wanli: Worker and ai collaboration for natural language inference dataset creation.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022b. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. 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.

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

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3505–3506, New York, NY, USA. Association for Computing Machinery.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Timo Schick and Hinrich Schütze. 2021a. Few-shot text generation with natural language instructions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 390–402, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. Generating datasets with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6943–6951, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Super-naturalinstructions:generalization via declarative instructions on 1600+ tasks. In *EMNLP*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. Symbolic knowledge distillation: from general language models to commonsense models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4602–4625, Seattle, United States. Association for Computational Linguistics.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020. Generative data augmentation for commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1008–1025, Online. Association for Computational Linguistics.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. ZeroGen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *ICLR 2020*.

A Fine-Tuning Hyperparameters

We use the same set of hyperparameters for fine-tuning experiments with T5-LM (Raffel et al., 2020; Lester et al., 2021). All models are trained for up to $\max(3 \text{ epochs}, 3000 \text{ steps})$ and the final model is chosen based on Rouge-L on our validation set, where we evaluate every 100 steps. We use a batch size of 16, a maximum learning rate of $1 \cdot 10^{-5}$ with warm-up for the first 10% of training and a weight decay of 0.01. We truncate inputs at 1,024 tokens and outputs at 128 tokens. All models are trained using DeepSpeed's ZeRO-3 (Rasley et al., 2020). Training on up to 64,000 examples is performed on 32 NVIDIA Tesla V100 16GB Volta GPUs using FP32; for bigger training datasets, we used 8 NVIDIA A100 40GB GPUs with BF16. For computing Rouge-L and exact match scores, we use the implementation of Wang et al. (2022).

B Evaluation Details

For evaluating model performance on Super-Natural Instructions, T0: Zero-Shot and LMEntry, we use their official evaluation scripts. For evaluation on BIG-bench: Hard, we lowercase outputs, remove punctuation characters and trim extra whitespace before computing exact match scores. The only exception to this is the task `dyck_languages`, where the target output consists entirely of punctuation characters.

C Data Generation Prompts

Table 6 presents the in-context demonstrations we used, taken from Wang et al. (2022).

In-Context Demonstrations

Seed 1

Example 1
Instruction: In this task, you're given passages that contain mentions of names of people, places, or things. Some of these mentions refer to the same person, place, or thing. Your job is to write questions that evaluate one's understanding of such references. Good questions are expected to link pronouns (she, her, him, his, their, etc.) or other mentions to people, places, or things to which they may refer. Do not ask questions that can be answered correctly without understanding the paragraph or having multiple answers. Avoid questions that do not link phrases referring to the same entity. For each of your questions, the answer should be one or more phrases in the paragraph, and it should be unambiguous.
Input: Passage: Nearing London, Oliver encounters Jack Dawkins, a pickpocket more commonly known by the nickname the "Artful Dodger", and his sidekick, a boy of a humorous nature named Charley Bates, but Oliver's innocent and trusting nature fails to see any dishonesty in their actions. The Dodger provides Oliver with a free meal and tells him of a gentleman in London who will "give him lodgings for nothing, and never ask for change". Grateful for the unexpected assistance, Oliver follows the Dodger to the "old gentleman's" residence. In this way Oliver unwittingly falls in with an infamous Jewish criminal known as Fagin, the gentleman of whom the Artful Dodger spoke. Ensnared, Oliver lives with Fagin and his gang of juvenile pickpockets in their lair at Saffron Hill for some time, unaware of their criminal occupations. He believes they make wallets and handkerchiefs.
Constraints: None.

Example 2
Instruction: You will be given a piece of text either about an everyday event, or a general statement. If the event seems a plausible event to you, or the general statement makes sense matches your commonsense, output 'True', otherwise output 'False'.
Input: Text: The glass fell of a three-story building, so it broke into pieces.
Constraints: The output should be one of the two: 'True' or 'False'.

Example 3
Instruction: You need to answer the question 'Are the given steps in order?', given a set of steps describing a process. Your answer must be either Yes or No. If the answer is No, that means the steps are out of order and do not make sense in the order they are in. If the answer is Yes, that means the steps are in order and make sense in the order that they are in. A set of steps are not in order if the steps reference information that is introduced in a later step.
Input: Steps: ['The seeds are dispersed by wind, animals, etc', 'The seeds reach the ground', 'Grow into new trees', 'The process repeats itself over and over', 'A tree produces seeds','These new trees produce seeds']
Constraints: The output should be one of the two: 'Yes' or 'No'.

Example 4

Seed 2

Example 1
Instruction: In this task, you are given two phrases: Head and Tail, separated with <sep>. The Head and the Tail events are short phrases possibly involving participants. The names of specific people have been replaced by generic words (e.g., PersonX, PersonY, PersonZ). PersonX is always the subject of the event. You have to determine whether the Head is used for the Tail or not. The usage describes everyday affordances or uses of objects and includes both typical and atypical uses. For example, a popcorn bucket can typically be used to hold popcorn, but it could also serve as a hat in atypical situations. Classify your answers into "Yes" and "No". The phrase may also contain "-", a placeholder that can be an object, a person, and/or an action.
Input: Head: floor mats<sep>Tail: wipe off one's boots
Constraints: The output should be 'Yes' or 'No'.

Example 2
Instruction: In this task, you're given a short story of five sentences written in natural language. However, the order of the given story is not correct. Your job is to return the correct order for the given five sentences to create a coherent short story with the new order that has the correct flow. Generate your answer using the number of sentences in the correct order, such as '23415'.
Input: Sentence1: He is happy now. Sentence2: Rick grew up in a troubled household. Sentence3: It wasn't long before Rick got shot in a robbery. Sentence4: The incident caused him to turn a new leaf. Sentence5: He never found good support in family, and turned to gangs.
Constraints: The output must consist of the numbers representing the sentences.

Example 3
Instruction: In this task, you are given Yelp reviews. The task is to classify a review as "POSITIVE" if the overall sentiment of the review is positive or as "NEGATIVE" if the overall sentiment of the review is negative.
Input: This is my go to place to get the best chicken Pad Thai! Also the price is super good, considering most places have high prices and poor quality. Love this place, its definitely a top 5 fav for take out.
Constraints: The output should be 'POSITIVE' or 'NEGATIVE'.

Example 4

Seed 3

Example 1
Instruction: You are given a question title. You need to classify it into one of the 10 topics: 0 - 'Society & Culture', 1 - 'Science & Mathematics', 2 - 'Health', 3 - 'Education & Reference', 4 - 'Computers & Internet', 5 - 'Sports', 6 - 'Business & Finance', 7 - 'Entertainment & Music', 8 - 'Family & Relationships', 9 - 'Politics & Government'.
Input :What's the best way to clean a keyboard?
Constraints: The output should be a number between 0 and 9.

Example 2
Instruction: You are given a target person and two sentences. Your task is to determine whether the second sentence conveys either a stereotype, anti-stereotype, or is unrelated with respect to the first sentence and the target. A stereotype is a widely held but oversimplified notion of a group of people. An anti-stereotype is an idea that goes against the widely held but oversimplified notion of a group of people. Sentences and targets are unrelated if there is no relevance between them.
Input: Target: mother. The mother walked into her kitchen. She pulled a beer out of the fridge and leaned against the doorway, sipping it slowly.
Constraints: The output should be one of the three: 'stereotype', 'anti-stereotype' or 'unrelated'.

Example 3
Instruction: In this task, you are given an article. Your task is to summarize the article in a sentence.
Input: Ms Bourne, who became Sussex's first PCC in 2012, was declared winner after the vote went to a second count. Three candidates were eliminated in the first count, leaving only Ms Bourne and Labour's Michael Jones. In the first count, Ms Bourne had 114,570 votes and Mr Jones had 61,017. The second count brought their totals to 139,335 and 86,392 respectively. She said: "I'm absolutely honoured and very privileged to be elected." She said she needed to "find extra savings because budgets are still reducing" and "to invest in frontline policing because I know that is really important to people in Sussex". Voter turnout was 22.5% compared with 15.3% in 2012. The three eliminated in the first count were Green Party candidate James Doyle, UKIP's Patrick Lowe and James Walsh from the Liberal Democrats. Results listed alphabetically by surname are as follows. BBC News App users: tap here to see the results.
Constraints: None.

Example 4

Seed 4

Example 1
Instruction: In this task, you are given Wikipedia articles on a range of topics as passages and a question from the passage. We ask you to answer the question by classifying the answer as 0 (False) or 1 (True).
Input: Passage: Property tax – Property tax or 'house tax' is a local tax on buildings, along with appurtenant land. It is and imposed on the Possessor (not the custodian of property as per 1978, 44th amendment of constitution). It resembles the US-type wealth tax and differs from the excise-type UK rate. The tax power is vested in the states and is delegated to local bodies, specifying the valuation method, rate band, and collection procedures. The tax base is the annual rental value (ARV) or area-based rating. Owner-occupied and other properties not producing rent are assessed on cost and then converted into ARV by applying a percentage of cost, usually four percent. Vacant land is generally exempt. Central government properties are exempt. Instead a 'service charge' is permissible under executive order. Properties of foreign missions also enjoy tax exemption without requiring reciprocity. The tax is usually accompanied by service taxes, e.g., water tax, drainage tax, conservancy (sanitation) tax, lighting tax, all using the same tax base. The rate structure is flat on rural (panchayat) properties, but in the urban (municipal) areas it is mildly progressive with about 80% of assessments falling in the first two brackets. Question: is house tax and property tax are same.
Constraints: The output should be 0 or 1.

Example 2
Instruction: Rewrite each original sentence in order to make it easier to understand by non-native speakers of English. You can do so by replacing complex words with simpler synonyms (i.e. paraphrasing), deleting unimportant information (i.e. compression), and/or splitting a long complex sentence into several simpler ones. The final simplified sentences need to be grammatical, fluent, and retain the main ideas of their original counterparts without altering their meanings.
Input: From its inception, it was designated a duty-free port and vied with the neighboring Sultanate of Pattani for trade.
Constraints: None.

Example 3
Instruction: You are provided with an arithmetic question. Your task is to compute the solution using the given arithmetic operations. The only arithmetic operators needed to answer the questions are'+'(addition) and'-'(subtraction). The answer should be correct to one decimal place.
Input: Joan found 70 seashells on the beach. She gave Sam some of her seashells, after which she has 27 seashell left. How many seashells did she give to Sam?
Constraints: None.

Example 4

Seed 5

Example 1
Instruction: You are given a science question (easy-level) and four answer options (associated with "A", "B", "C", "D"). Your task is to find the correct answer based on scientific facts, knowledge, and reasoning. Do not generate anything else apart from one of the following characters: 'A', 'B, 'C', 'D'. There is only one correct answer for each question.
Input: Which part of a bicycle BEST moves in a circle? (A) Seat (B) Frame (C) Foot pedal (D) Kickstand
Constraints: The output should be one of the following characters: 'A', 'B, 'C', 'D'.

Example 2
Instruction: You are given a negative review and your task is to convert it to a positive review by one or more making minimal changes. Avoid changing the context of the review.
Input: we stood there in shock, because we never expected this.
Constraints: None.

Example 3
Instruction: In this task, you are given two sentences taken from a conversation, and your job is to classify whether these given sentences are sequential or not. We will mark the given sentence pair as 'True' if it's sequential, otherwise 'False'. The two sentences are spoken by two different people.
Input: Noah: When and where are we meeting? :), Madison: I thought you were busy...?
Constraints: The output should be 'True' or 'False'.

Example 4

Table 6: The in-context demonstrations used in our experiments.

Minimal	Enumeration	Verbose
Instruction: ... Input: ... Constraints: ... Instruction: ... Input: ... Constraints: ... Instruction: ... Input: ... Constraints: ... Instruction: ...	Example 1 Instruction: ... Input: ... Constraints: ... Example 2 Instruction: ... Input: ... Constraints: ... Example 3 Instruction: ... Input: ... Constraints: ... Example 4	Below are examples of instructions describing a diverse set of textual tasks and their inputs. Instruction: ... Input: ... Constraints: ... Instruction: ... Input: ... Constraints: ... Instruction: ... Input: ... Constraints: ... Write instructions and inputs for other textual tasks.

Figure 6: The meta-prompts used in our ablations.

D Structural Prompt Ablations

We explore the effect of the different components of our data collection pipeline by conducting structural prompt ablations. Throughout this section, we train models for 1,500 steps using 2,000 examples and evaluate the Super-Natural Instructions validation set performance, averaged across three different random seeds.

D.1 Meta-Prompts

Language models are known to be sensitive to the *meta-prompt* – i.e., the text wrapping the in-context demonstrations, which can include task description or additional guidance regarding the desired output. We therefore experiment with three different meta-prompt styles: *minimal*, *enumeration*, and *verbose* (Figure 6).

Table 7 presents the results obtained from fine-tuning on datasets generated with different meta-prompts. We observe that the simple enumeration approach elicits more informative examples than either the minimalistic or verbose approaches. Perhaps surprisingly, the verbose meta-prompt performs worse than the minimalistic one, possibly because the last line (the command) interrupts the pattern, and does not align well with patterns in the pretraining corpus.[4]

[4]While our core dataset was created using the enumeration meta-prompt, the remaining ablation experiments in this section were run using the verbose meta-prompt.

Meta-Prompt	Super-Natural Instructions
Minimal	47.5 ± 0.6
Enumeration	$\mathbf{48.7} \pm \mathbf{0.3}$
Verbose	46.9 ± 0.3

Table 7: Performance of 11B T5-LM models trained on 2,000 examples, generated with each meta-prompt, on the Super-Natural Instructions validation set.

Seed Demonstrations	Super-Natural Instructions
1	46.9 ± 0.3
2	46.1 ± 0.3
3	46.8 ± 0.4
4	41.9 ± 1.0
5	46.0 ± 0.2
Mix	46.1 ± 0.3

Table 8: Performance of 11B T5-LM models trained on 2,000 examples, generated with various sets of three in-context demonstrations (seeds), on the Super-Natural Instructions validation set. *Mix* samples 400 examples from each of the five single-seed datasets.

D.2 In-Context Examples

Models such as GPT-3 are known to be sensitive to slight variations in prompt content, resulting in performance differences when provided with different demonstrations sampled from the same dataset (Liu et al., 2022b) and when permuting the in-context demonstrations (Kumar and Talukdar, 2021; Lu et al., 2022). To account for the effect of the provided demonstrations on the quality of the generated data, we experiment with each of our five demonstration sets separately.[5] Table 8 shows that the data generation pipeline is largely robust to variations in the in-context demonstrations, with one outlier (seed 4). Inspecting the differences between these groups, we find that seed 4 led to less constrained instructions: 1,376 out of 2,000 examples do not have constraints, whereas that number is between 28 and 880 for all other sets. Indeed, in seed 4, only one out of three prompt demonstrations had constraints, while in other sets, at least two demonstrations had constraints.

D.3 Constraints

As mentioned in §2, each instruction-input demonstration is accompanied by an additional *constraints* field, which details the task's output space restrictions (e.g., "entailment", "contradiction" or "neutral" for NLI). We note that, in all demonstrations, the instruction itself lists the output space

[5]See Appendix C for all demonstration sets.

Use "Constraints:" for Input Gen	Output Gen	Super-Natural Instructions
✓	✓	**46.9 ± 0.3**
✓		43.9 ± 0.7
		41.7 ± 0.2

Table 9: Performance of 11B T5-LM models trained on 2,000 examples, generated with and without the *constraints* field, on the Super-Natural Instructions validation set.

constraints. We hypothesize that adding the constraints field may emphasize these restrictions, ultimately steering the output generation model to produce outputs in the correct format. We verify our hypothesis by conducting two ablation experiments. First, we keep the constraints field when generating the instructions and inputs, but only use instructions and input arguments for the output generation step (i.e., without concatenating generated constraints). Second, we completely remove the constraints field from the data generation pipeline, leaving the instruction field as the only source of information for output space constraints. Table 9 shows that the constraints field has a positive effect both on the quality of the generated outputs *and* inputs. Removing constraints from the output generation step reduces performance by 3 points, and removing the field from the instructions-inputs generation phase decreases performance by an additional 2.2 points.

D.4 Two-Step Process

An alternative to our two-step pipeline is to generate instruction-input-output triplets in one pass. To test this approach, we provide the model with the same prompt used for the instruction-input-constraints generation, only with an additional *output* field, added after the constraints field. As Table 9 shows, one-step generation obtains a score that is lower by 1.7 than the default two-step process. We suspect that this gap is a result of using stochastic decoding in the unified input-output generation phase, which is critical for obtaining diverse inputs. In contrast, when generating outputs in a separate phase, we can use deterministic decoding algorithms to maximize accuracy.

Data Generation Process	Super-Natural Instructions
Separate I/O Steps	**46.9 ± 0.3**
Unified I/O Step	45.2 ± 0.6

Table 10: Performance of 11B T5-LM models trained on 2,000 examples, generated either using separate input and output steps or a single unified step, on the Super-Natural Instructions validation set.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
9

☑ A2. Did you discuss any potential risks of your work?
9

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

2

☑ B1. Did you cite the creators of artifacts you used?
1, 2, 4

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We verified that all the data and code used is publicly open - we verified license details for each, and we provided citation to all relevant resources, where license details can also be found.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
4

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
As for existing datasets we used, we didn't discuss that, but other than the fact that we used published datasets that are already used by the research community - we also sampled examples and manually verified their content. As for data we collected, we did discuss that in section 9, and additionally provided data analysis in section 3.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
3

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
2

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

4, 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
4, Appendix A

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4, Appendix A

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
5, 6, Appendix D

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4, Appendix A, Appendix B

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.