Do Models Really Learn to Follow Instructions? An Empirical Study of Instruction Tuning

Po-Nien Kung, **Nanyun Peng**
University of California, Los Angeles
{ponienkung,violetpeng}@cs.ucla.edu

Abstract

Recent works on instruction tuning (IT) have achieved great performance with zero-shot generalizability to unseen tasks. With additional context (e.g., task definition, examples) provided to models for fine-tuning, they achieved much higher performance than untuned models. Despite impressive performance gains, what models learn from IT remains understudied. In this work, we analyze how models utilize instructions during IT by comparing model training with altered vs. original instructions. Specifically, we create *simplified task definitions* by removing all semantic components and only leaving the output space information, and *delusive examples* that contain incorrect input-output mapping. Our experiments show that models trained on *simplified task definition* or *delusive examples* can achieve comparable performance to the ones trained on the original instructions and examples. Furthermore, we introduce a random baseline to perform zero-shot classification tasks, and find it achieves similar performance (42.6% exact-match) as IT does (43% exact-match) in low resource setting, while both methods outperform naive T5 significantly (30% per exact-match). Our analysis provides evidence that the impressive performance gain of current IT models can come from picking up superficial patterns, such as learning the output format and guessing. Our study highlights the urgent need for more reliable IT methods and evaluation.

1 Introduction

Recently, instruction tuning(IT) has drawn much attention in the NLP communities, with the rapid growth of new models (Sanh et al., 2021; Wei et al., 2021; Ouyang et al., 2022) and datasets (Wang et al., 2022; Gupta et al., 2022; Finlayson et al., 2022; Mishra et al., 2021; Ye et al., 2021; Bach et al., 2022). Models trained with task instructions demonstrate impressive zero-shot cross-task generalization ability. Despite the remarkable results, how models utilize the instructions during training and inference time remains an open question.

IT Models	Generalize to Unseen Tasks			Generalize to Unseen Instruct.	
models →	TK-Inst	T0	FLAN	Alpaca	Vicuna
Training					
# of tasks	756	39	38	–	
# of instructions	756	390*	380	52K	70K
Testing					
# of tasks	119	11	24	–	
# of instructions	119	110*	240	252	252
Testing on unseen tasks?	✔	✔	✔	✗	✗

Table 1: Comparison between two types of instruction tuning models. Noted that we reported an estimated number of instructions for T0 during training and testing since they have 5 to 10 instructions for each task. Our analysis focuses on the "generalize to unseen task" type.

Prior works have raised the question of whether models really learn to follow the instructions or just capture spurious correlations. Jang et al. (2022), Webson and Pavlick (2021) showed that the current large language models (LLMs) can achieve similar performance with misleading instructions(prompts) in in-context learning(ICL) and few-shot learning scenarios. Min et al. (2022) analyze how model utilize examples in ICL. They observed that (1) Input-output mapping in examples is not important and(2) Output space information is crucial.

Besides ICL and few-shot prompt-tuning, some works raise concerns about instruction following in the instruction tuning field (Finlayson et al., 2022; Gupta et al., 2022; Gu et al., 2022), with a focus on test-time analysis. In contrast, we focus on analyzing how the models utilize instructions during the training process. We compare our analyzing methods and observation with prior works in Appendix A.1.

In this work, we conduct controlled experiments on NatInst-V2 (Wang et al., 2022), the largest open-source instruction learning dataset includes 800+ English tasks with diverse task types, to study how models utilize instructions during IT. Note that existing research on IT can be categorized into two

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 1317–1328
July 9-14, 2023 ©2023 Association for Computational Linguistics

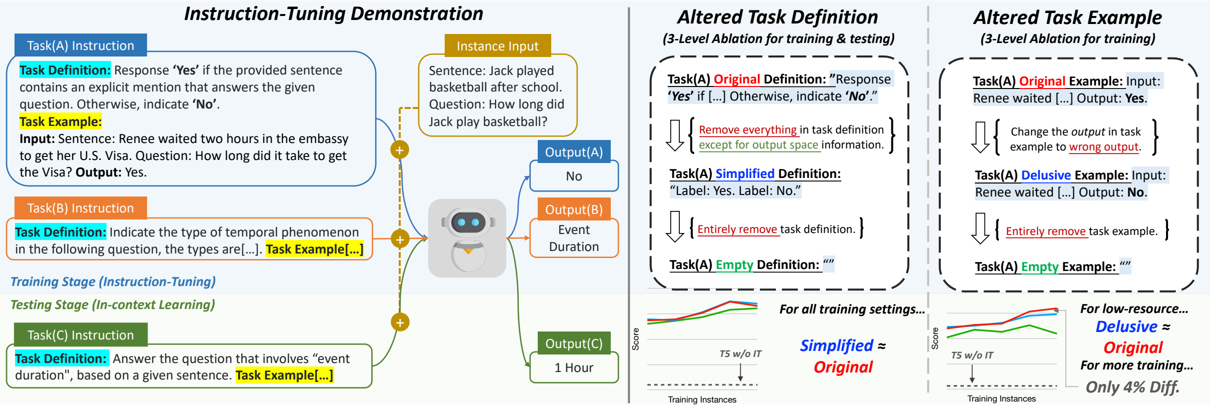
Instruction-Tuning Demonstration
Task(A) Instruction
Task Definition: Response 'Yes' if the provided sentence contains an explicit mention that answers the given question. Otherwise, indicate 'No'.
Task Example:
Input: Sentence: Renee waited two hours in the embassy to get her U.S. Visa. Question: How long did it take to get the Visa? Output: Yes.
Instance Input
Sentence: Jack played basketball after school. Question: How long did Jack play basketball?
Output(A)
No
Task(B) Instruction
Task Definition: Indicate the type of temporal phenomenon in the following question, the types are[...]. Task Example[...]
Output(B)
Event Duration
Training Stage (Instruction-Tuning)
Testing Stage (In-context Learning)
Task(C) Instruction
Task Definition: Answer the question that involves "event duration", based on a given sentence. Task Example[...]
Output(C)
1 Hour
Altered Task Definition
(3-Level Ablation for training & testing)
Task(A) Original Definition: "Response 'Yes' if [...] Otherwise, indicate 'No'."
Remove everything in task definition except for output space information.
Task(A) Simplified Definition: "Label: Yes. Label: No."
Entirely remove task definition.
Task(A) Empty Definition: ""
Score
T5 w/o IT
Training Instances
For all training settings...
Simplified ≈ Original
Altered Task Example
(3-Level Ablation for training)
Task(A) Original Example: Input: Renee waited [...] Output: Yes.
Change the output in task example to wrong output.
Task(A) Delusive Example: Input: Renee waited [...] Output: No.
Entirely remove task example.
Task(A) Empty Example: ""
Score
T5 w/o IT
Training Instances
For low-resource...
Delusive ≈ Original
For more training...
Only 4% Diff.

Figure 1: The left sub-figure demonstrates a two-stage pipeline where the model first trains on a set of tasks and then evaluates other unseen tasks. The model inputs *task definition*, *examples*, and *instance input* together to make a prediction. The two right sub-figures show how we create *Simplified task definition* and *Delusive task example* for ablation studies. We also demonstrate the results at the bottom with *T5 w/o IT*(Untuned models) results. It is shown that models can still achieve significant performance gain compared to *T5 w/o IT* while training on *Simplified* task definition and *Delusive examples*.

major camps: **generalize to unseen tasks** and **generalize to unseen instructions**, based on their objectives. Table 1 shows the comparison. Our analysis focuses on the former with more background and justifications provided in section 2. We strategically alter the instructions and compare them with original instructions for IT. Specifically, for task definition, we create *simplified versions* by removing all semantic components in the instructions and only leaving the output space information. For task examples, we create *delusive examples* with incorrect input-output mapping, where the examples' input and output spaces are correct, but the input-output mappings are wrong. Figure 1 demonstrates specific examples of these altered instructions.

Our experiments show that models trained with *simplified task definitions* achieve performances on par with the original IT models with different numbers of training examples ranging from 10 to 800 per task. We also observe that instruction-tuned models are sensitive to input-output mapping during the testing ICL stage, but not during the instruction-tuning (training) stage, especially in low resource settings (i.e., ≤ 50 training instance per task). To further understand why instruction tuning improves performance for zero-shot test tasks, we establish a random baseline that only knows the correct output format (label space) for classification and multi-choice tasks. We discover that the random baseline can get 30% absolute exact-match score improvement over an untuned model, almost comparable to some IT models in low resource settings.

Our results suggest that the impressive performance gains of IT may just come from models learning superficial patterns, such as the output space and format. We suggest future research on IT more carefully analyze their performance gains and benchmark against trivial baselines.

2 Background

Recently, many instruction tuning work train and test the models with instructions to achieve better zero-shot generalizability toward unseen tasks/instructions. We categorize these works by their objectives: **generalize to unseen tasks** and **generalize to unseen instructions**, and show the comparison in Table 1.

Instruction tuning to generalize to unseen tasks. Figure 1 illustrates a two-stage instruction tuning pipeline used in many IT models, such as T0 (Sanh et al., 2021), FLAN (Wei et al., 2021), and TK-Instruct (Wang et al., 2022). In the first stage, the models are trained on a set of training tasks with instructions (task-definition and task-examples). After training, the models are evaluated on a set of unseen testing tasks for zero-shot generalizability. By incorporating instructions during training, the models are shown to significantly improve performance over untuned models. The impressive performance gains led people to believe that models learned to follow instructions via instruction tuning. The goal of our analysis is to verify this belief.

Instruction tuning to generalize to unseen instructions. Different from T0, FLAN, and TK-Instruct training and testing the model with clear task boundaries and focusing on cross-task generalizability, Instruct-GPT (Ouyang et al., 2022), Alpaca (Taori et al., 2023) and Vicuna (Chiang et al., 2023) focus more on instruction generalizability, which they train their model without clear task boundary but with diverse instructions, and further test on user-oriented instructions. These models show very different behavior compared with instruction tuning models that aim to generalize to unseen tasks.

Since Instruct-GPT is not open-sourced and distilled IT models such as Alpaca and Vicuna come up after our submission, we focus our analysis on the first category using the TK-instruct model and NatInst-V2 dataset. However, we also conduct additional experiments and discuss the Alpaca model's instruction following ability in Table 2.

3 Analysis Method

Task definition manipulation. To analyze whether models really "understand" and utilize the semantic meaning of task definitions, we conduct controlled experiments to remove semantic information in task definitions. Specifically, we conduct instruction-tuning with task definitions at 3 levels of granularity: **Original**, **Simplified**, and **Empty**. The **Original** version uses human-crafted human-readable task definitions provided in NatInst-V2 (Wang et al., 2022). The **Simplified** task definitions remove all semantic components in the original task definition and only leave the output space information. Specifically, we only provide possible output labels as task definitions for classification tasks, and completely remove task definitions for other tasks (mostly generative tasks) during IT. Figure 1 shows an example of **Simplified** task definition. More details can be found in Appendix A.2. For **Empty**, we don't provide task definition during instruction-tuning.

Task example manipulation. Finlayson et al. (2022) show that by providing a few task examples, both humans and models can guess and perform a task. We thus design a controlled experiment to study whether models learn the input-output mapping from task examples. Specifically, we compare models trained with 3 types of task examples: **Original**, **Delusive**, and **Empty**. For the **Original** setup, we provide one positive example in NatInst-V2 (Wang et al., 2022). For **Delusive** examples, we sample negative examples from NatInst-V2, which have correct input and output formats, but incorrect input-output mappings. For **Empty**, we do not provide task examples during training.

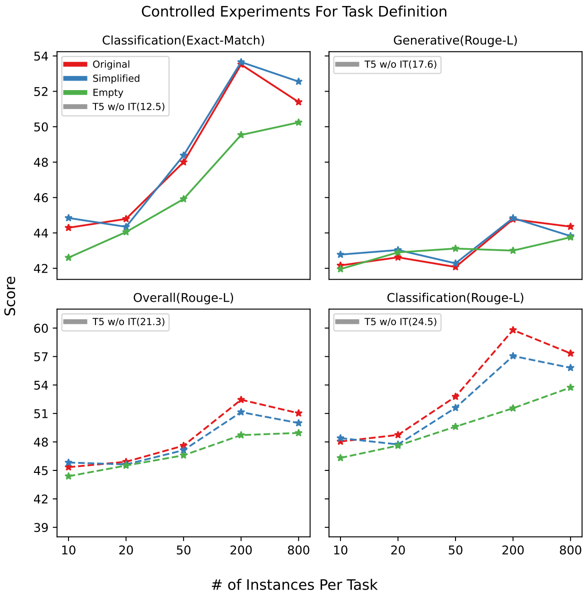

Figure 2: Controlled experiments for task definition. **Original**, **Simplified**, and **Empty** represent what type of task-definition the model is trained and tested with. **T5 w/o IT(12.5) shows the score(12.5) of the baseline T5-large model.** The top two subfigures show the main results evaluating classification tasks using Exact-Match (accuracy) and Generative tasks using Rouge-L. The bottom two sub-figures are supplementary results evaluating rouge-L for *All tasks* and *classification tasks*.

4 Experimental Setup

Dataset. We conduct experiments on the NatInst-V2 (Wang et al., 2022), the largest open-source instruction learning dataset, including over 800+ English tasks with diverse task types. The instructions include human-crafted human-readable *Task Definition*, *Positive Task Examples*, *Negative Task Examples*, and *Explanation*. We focus on studying task definition and task examples, which were shown to be most useful in the original paper.

Model. we conduct experiments on TK-Instruct, the current SOTA model provided in NatInst-V2 paper. The model significantly outperformed previous SOTA models, such as T0 (62.0 v.s. 32.3 rouge-L for 11B model). We follow the seq-to-seq instruction-tuning method used in TK-Instruct, and train a T5-large-lm-adapt (770M parameters) model (Raffel et al., 2020) with performance comparable to the larger model (3B parameters) reported in Wang et al. (2022).[1]

Evaluation Metrics. For task definition, we separately evaluate *Classification* and *Generative* tasks using exact match and rouge-L respectively. For

[1] For task definition experiment, we follow the best performance settings from Wang et al. (2022) to use task definition and two examples as instructions. For task examples experiments, due to the lack of negative examples, we conduct ablation studies using task definition and one example.

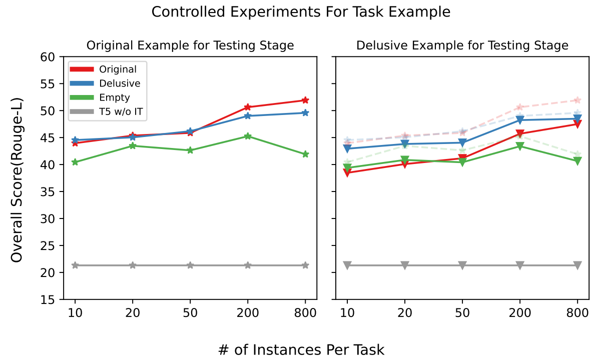

Figure 3: Controlled experiments for task examples. The left sub-figure shows the main results, where **Original** task examples are used for testing (in-context learning). **Original**, **Delusive**, and **Empty** represent what type of task examples are used for training and the **T5 w/o IT** is the baseline T5-large model. The right sub-figure shows supplementary results using **Delusive** examples for testing. The faint dashed lines are copied from the left sub-figure for comparison purposes.

task examples, we follow Wang et al. (2022) to report the overall rouge-L score for both classification and generative tasks. To understand the impact of training examples, we report model performances with varying numbers of training instances per task (i.e., 10, 20, 50, 200, 800).

5 Results

Task Definition Experiments. Figure 2 shows experimental results for task definitions. In the top sub-figures, we can see that the models trained with **Simplified** instructions achieve almost the same results as models trained with **Original** definitions both on Classification and Generative tasks. Note that **Simplified** task definitions remove all semantic components in task definitions and only retain output space information for Classification tasks and remove task definitions altogether for Generative tasks. This indicates that models may only utilize output space information during instruction tuning. The bottom-left sub-figure in Figure 2 shows the overall rouge-L score for classification tasks, where models trained on the **Original** task definition slightly outperform the **Simplified** ones. A closer examination reveals that models trained on the **Original** task definitions are more likely to predict partially correct answers that help with the ROUGE-L score in some tasks. We provide further details in Appendix A.5. In addition, we also observe that training with **Simplified** prompts can yield comparable performance to the T0 model trained with **Original** prompts on T0 dataset. Please refer to Appendix A.6 for details.

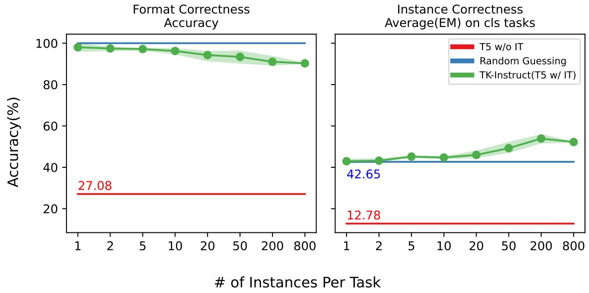

Figure 4: Results for the **Random Guessing** baseline which randomly guesses an answer from the output space (labels). The left figure shows the format correctness, which calculates the accuracy of model predictions lied in the label space for classification (CLS) tasks. The right figure shows the average exact-match score of CLS tasks.

Task Examples Experiments. Figure 3 shows the experimental results for task examples. The left sub-figure shows overall ROUGE-L scores. It shows that models trained with **Delusive** task examples can achieve almost the same performance as **Original** task examples when the number of training instances per task is small (≤ 50). When the data per task goes to 200, the **Original** models started to outperform **Delusive** ones slightly. Combined with the previous results for task definition, we observe that comparing to the untuned models(*T5 w/o IT*), the IT models may achieve significant performance gain (Rouge-L from 22 to 46) with (1)*Simplified* task definition and (2)*Delusive* task example, indicating that the current impressive improvement of IT models can come from the models learning superficial patterns without utilizing (following) the instructions like human do.

For the right sub-figure, we show the results using **Delusive** task examples during test time via in-context learning. We see the performance drops for all three models, indicating that the input-output mapping matters for in-context learning on instruction-tuned models. This observation seems to misalign with previous work (Min et al., 2022), which they found input-output mapping is unimportant for in context learning for classification tasks. However, a closer investigation found that most tasks suffer from significant performance drop are analogical tasks rather than classification tasks as studied in Min et al. (2022).[2]

6 Additional Analysis

Random baseline. While our experiments suggest that models do not utilize most information in the instructions, we still observe huge performance gains via instruction tuning. To understand where the gains come from, we introduce a **Random** baseline that simply guesses within the cor-

[2] See examples of analogical tasks in Appendix A.4.

Model / Metric	CLS (EM)	Δ	GEN (Rouge-L)	Δ
LLaMA				
Test w/ **Original**	4.40		14.31	
Train w/ **Original**	59.19	-2.58	48.80	-3.05
Train w/ **Simplified**	56.61		45.75	
Alpaca				
Test w/ **Original**	45.08	-3.42	44.40	**-9.6**
Test w/ **Simplified**	41.66		34.80	
Train w/ **Original**	59.33	-3.16	48.69	-3
Train w/ **Simplified**	56.17		45.69	

Table 2: Altered task definition experiment for LLaMA and Alpaca model. **Original** and **Simplified** specify the provided task definition type. *Test w/* specify how the model is tested with provided task definition type. *Train w/* specify how the model is trained and tested with provided task definition type. We provide task definition and one example as instruction for both training and testing.

rect output space. Figure 4 shows the results. First, IT improves format correctness from 27% to 97% by training with only one instance per task, and the exact-match score improves from 12.78% to 43%. Further providing more training instances per task(200) can improve exact-match score to 52%. However, while the performance gains seem impressive, the **Random Guessing** baseline can also achieve 42.6% exact-match score, on par with TK-Instruct trained in low resource setting (less than five instances per task). This suggests that the majority of score improvement from IT may come from model learning the output format, especially in low-resource settings.

Fair comparison for IT models. Existing studies on instruction tuning often introduce changes to both models and datasets simultaneously, which can obscure fair comparisons. To address this issue, we conduct experiments comparing different models (T0, TK-Instruct) on the same dataset (NatInst-V2) and emphasize the importance of careful evaluation. In Table 3, when evaluating using the NatInst-V2 evaluation method and considering only the overall Rouge-L score, the TK-Instruct model appears to outperform T0 significantly. However, upon closer examination of the classification (CLS) and generative (GEN) tasks separately, we observe that T0's classification score is even lower than the Random baseline, primarily due to its format correctness being only 64%. To ensure a fairer comparison between these models, we employ constrained decoding techniques to align the model's predictions with the label space. By adopting this approach, we observe a substantial performance improvement for T0 in CLS tasks (34.03 to 51.31). T0 surpasses both the TK-Instruct model and the random baseline, indicating that it is indeed superior to these models in CLS tasks.

Metric → Model ↓	Format (Acc)	CLS (EM)	GEN (Rouge-L)	Overall (Rouge-L)
Random	100	42.65	–	–
T0	64.61	34.03	27.36	32.28
w/ CD	100	**51.31**	27.36	40.7
TK	96.23	44.29	**42.16**	45.34
w/ CD	100	47.12	**42.16**	**45.93**

Table 3: Careful evaluation of the NatInst-V2 dataset. The *Format* metric is the same as the format correctness in Figure 4. The **w/ CD** indicates that the model's decoding is constrained to match the label choices for CLS tasks. The **TK** is the TK-Instruct(770M) model trained with 10 instances per task.

7 Discussion

Do Alpaca better follow the instruction on NatInst-V2 dataset? After our submission, new instruction tuning models, like Alpaca and Vicuna, are trained on distilled data from Chat-GPT and exhibit behavior closer to it. To investigate their instruction utilization, we conduct the "Altered Task Definition" experiment on LLaMA-7B (Touvron et al., 2023) and Alpaca-7B models using the NatInst-V2 test set. In Table 2, training the LLaMA model on the NatInst-V2 dataset using the **Original** task definition leads to substantial performance enhancements than zero-shot. However, the **Simplified** task definition also achieves comparable performance, with a minimal decrease of 3 (EM/Rouge-L)scores. This finding is consistent with our previous observations on the TK-Instruct and T0 models. Even without tuning on NatInst-V2, the Alpaca model demonstrates strong performance on the NatInst-V2 test set. However, when the model is tested using a **simplified** task definition, there is a significant decrease in performance for generative tasks (but not for classification tasks). This highlights the importance of a well-written task definition for the Alpaca model to effectively perform generative tasks.

8 Conclusion

We constructed controlled experiments on NatInst-V2 to compare model training with altered vs. original instructions (task definitions and examples). Our findings indicate that some current IT models do not fully utilize instructions, and the impressive performance gains of IT may come from models learning superficial patterns, such as the output space and format. We suggest future research on instruction tuning to analyze their performance gains with more comprehensive evaluation and benchmark against trivial baselines.

9 Limitations

While our analysis suggests that IT models do not fully utilize instructions but instead learn superficial patterns from instructions, there are some limitations to our experiments. First, we only analyze a SOTA IT method on the NatInst-V2 dataset and T0 dataset. Though Wang et al. (2022) showed that their model can outperform other large models such as Instruct-GPT (Ouyang et al., 2022) and T0 (Sanh et al., 2021), we did not analyze other IT methods, such as RLHF (Reinforcement Learning from Human Feedback) in Instruct-GPT. Secondly, since our analysis is conducted in the training stage, we cannot analyze private models such as Chat-GPT. Also, we did not explore models larger than 7B parameters due to our computation resource limitation. This may miss some emergent abilities of large language models (LLMs) (Wei et al., 2022). Lastly, while we observe the models do not utilize the majority of the instructions by IT, a certain degree of instruction understanding may already exist in pre-trained LLMs, which we did not study in this work. In conclusion, our work is a concentrated analysis to illuminate the potential vulnerability of the current IT models and evaluation metrics. We encourage future works to conduct more comprehensive studies on larger models and propose more reliable IT methods and evaluation frameworks.

10 Ethical Considerations

We will go through the computation resources and models we used to conduct our experiments. All of our models run on 4 48GB NVIDIA A6000 GPUs, along with 48 TB disk storage and AMD EPYC 7413 24-Core Processor. The experiment take around 1200 GPU hours for one 48GB NVIDIA A6000 GPU. Our experiments do not need to leverage model or data parallelism. For the model, we use Huggingface T5-large-lm-adapt models for our experiments, and will release our code once the paper been accepted.

Acknowledgements

Many thanks to Zefan Cai for implementing the altered task definition experiment on the T0 dataset and model. We would also like to thank Te-Lin Wu and Da Yin for their valuable insights during discussion, paper reviews, and constructive comments. We thank the anonymous reviewers for their feedback. This work was partially supported by AFOSR MURI via Grant #FA9550- 22-1-0380, Defense Advanced Research Project Agency (DARPA) grant #HR00112290103/HR0011260656.

References

Stephen H Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, et al. 2022. Promptsource: An integrated development environment and repository for natural language prompts. *arXiv preprint arXiv:2202.01279*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Matthew Finlayson, Kyle Richardson, Ashish Sabharwal, and Peter Clark. 2022. What makes instruction learning hard? an investigation and a new challenge in a synthetic environment. *arXiv preprint arXiv:2204.09148*.

Yuxian Gu, Pei Ke, Xiaoyan Zhu, and Minlie Huang. 2022. Learning instructions with unlabeled data for zero-shot cross-task generalization. *arXiv preprint arXiv:2210.09175*.

Prakhar Gupta, Cathy Jiao, Yi-Ting Yeh, Shikib Mehri, Maxine Eskenazi, and Jeffrey P Bigham. 2022. Improving zero and few-shot generalization in dialogue through instruction tuning. *arXiv preprint arXiv:2205.12673*.

Joel Jang, Seonghyeon Ye, and Minjoon Seo. 2022. Can large language models truly understand prompts? a case study with negated prompts. *arXiv preprint arXiv:2209.12711*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

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

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*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, A. Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, M. Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddharth Deepak Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hanna Hajishirzi, and Daniel Khashabi. 2022. Supernaturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks.

Albert Webson and Ellie Pavlick. 2021. Do prompt-based models really understand the meaning of their prompts? *arXiv preprint arXiv:2109.01247*.

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, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021. Crossfit: A few-shot learning challenge for cross-task generalization in nlp. *arXiv preprint arXiv:2104.08835*.

Fan Yin, Jesse Vig, Philippe Laban, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. 2023. Did you read the instructions? rethinking the effectiveness of task definitions in instruction learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

A Appendix

A.1 Related Analysis.

Min et al. (2022) found input-output mapping in examples is irrelevant for in-context learning (ICL) on *classification tasks*. However, we observe that it matters to ICL but is irrelevant to IT training on *analogical generative tasks*. Webson and Pavlick (2021) analyzed prompt-based models in few-shot learning scenarios and observed that models learn as fast using irrelevant or misleading prompts, which aligned with our findings. For instruction tuning, prior works raised concerns about models not following instructions. Gu et al. (2022); Gupta et al. (2022) analyze how models utilize instructions by removing them during inference stages. However, they did not address how models use instructions during training. Wei et al. (2021); Wang et al. (2022) observe performance drop when removing task definition during IT and conclude that task definition is helpful, which we found true but only in terms of providing output space information. Additionally, a concurrent study (Yin et al., 2023) has undertaken a comprehensive analysis of how models employ task definition in the process of instruction tuning on the NatInst-V2 dataset. They observed that by removing a majority of components from the task definition and retaining only the user intent, the model can attain comparable or even superior performance compared to utilizing the complete task definition.

A.2 Simplified Task Definition

To remove all semantic components and only leave the output space information within the task definition, we first manually look through all tasks to verify how each task definition describes their output space and further categorize all task definitions into four types: **(1) Exact Mentioned, (2) Combined Mentioned, (3) Keyword Mentioned, and (4) No Mentioned**. For **Exact Mentioned, Combined Mentioned** and **Keyword Mentioned**, there is a description of output space in the original task definition. For **No Mentioned**, The original task definition doesn't directly describe the labels or keywords in output space. This includes all the generative tasks and some classification tasks(We observe a few classification tasks in which task definitions do not describe output space information). Further details and examples are shown in Table 4.

A.3 Hyper-parameter tuning results

Before we conduct analysis, we follow the model settings in Wang et al. (2022) to perform the hyper-parameter search. Prior works trained the TK-Instruct(770M) models from T5-Large-lm-adapt(770M) with a learning rate 1e-5, batch size 16, and 100 training instances per task for two epochs. We found out that (1) learning rate 1e-4 can converge faster while performance remains; (2) Higher batch size($\geq$ 128) leads to much lower loss and better performance; (3) more training instances per task($\geq$ 200) leads to better performance; and (4) the loss will converge with 4 to 6 epochs. Following the hyper-parameter search results, we conducted our experiment with the following setting: learning rate 1e-4, batch size 128, [10, 20, 50, 200*, 800] training instance per task, and trained for six epochs. Our best results(200 instances) achieve a 52.8 Rouge-L score, which is better than TK-Instruct-770M(48 Rouge-L) from Wang et al. (2022) and comparable to their TK-Instruct-3B(54 Rouge-L) model.

A.4 Analogical Tasks

We look into a set of models training with *Original* task examples and find out a list of tasks with the most performance drop(Drop more than 20% score) when using *Delusive* examples during testing(in-context learning). We show the list of tasks in Table 6 and some of their details in Table 5. It is seen that these types of tasks have short input and output lengths, where input and output have direct word-level relations.

A.5 Performance gap between rouge-L and exact match

In the Results section, we observed that there's a slight performance gap on *Classification* tasks between model training with *Original* and *Simplified* task definition. By further examining the data, we observed that this could happen to some *Keyword Mentioned* tasks we described in Appendix A.2. Table 4 shows the example tasks in **Keyword Mentioned**. This task is a 7-class classification task with a special label "REFERENCE". The ground truth with "REFERENCE" will be combined with other text in the input, and both *Original* and *Simplified* models struggles(0% exact match) to predict the correct answer for this class. However, while both models failed to predict exactly correct answers, we observed that the *Original* model could achieve better partially correct answers by simply predicting more "REFERENCE". When we look into the testing set, we observe that 94 percent of ground truth is in "REFERENCE" class. Also, when we look into the predictions, we observe *Original* model will predict 55 percent of "REFERENCE" while *Simplified* only predicts 4 percent, achieving a 33.8 higher rouge-L score. We hypothesized that this happened because the word "reference" has explicitly been mentioned numerous times(8) in the *Original* task definition while mentioning other labels less than twice, leading to *Original* model's tendency to predict "REFERENCE".

A.6 Simplified task definition for T0.

Besides analyzing on NatInst-V2 dataset, we also conduct the simplified task definition experiment on T0 training stages. We follow the T0 training settings and changed the prompts to **Simplified** prompt, leaving only labels in the prompt for classification tasks and removing the entire prompt for generative tasks. We further train the T0-3B model using learning rate 1e-4, batch size 1024 for 10000 steps. The T0 model training and testing with **Simplified** prompts achieve a 60.69 overall score, which is comparable to training with **Original Prompt**(61.93) and aligns with our observation on the NatInst-V2 dataset.

	Exact Mentioned
Description	For tasks labeled as **Exact Mentioned**, the task definition describes the finite output space, which means all the labels within the output space are directly written in the definition.
Original Definition	Definition: In this task, you will be shown a short story with a beginning, two potential middles, and an ending. Your job is to choose the middle statement that makes the story incoherent / implausible by indicating 1 or2 in the output. If both sentences are plausible, pick the one that makes less sense.
Output Space	Finite Set: ["1", "2"]
Simplified Definition	"Label: 1. Label: 2."
	Combined Mentioned
Description	For tasks labeled as **Combined Mentioned**, the task definition describes a set of keyword labels that construct an infinite output space with all possible combinations of these keyword labels.
Original Definition	Given a command in a limited form of natural language, provide the correct sequence of actions that executes the command to thus navigate an agent in its environment. [...] There are only six actions: , ’I_WALK’, ’I_RUN’, ’I_JUMP’, ’I_TURN_LEFT’, and ’I_TURN_RIGHT’. [...]
Output Space	Infinite Set: ["I_LOOK", "I_LOOK I_WALK", "I_JUMP I_RUN", ... ∞]
Simplified Definition	"Combined Label: I_LOOK. Combined Label: I_WALK. Combined Label: I_RUN. Combined Label: I_JUMP. Combined Label: I_TURN_LEFT. Combined Label: I_TURN_RIGHT."
	Keyword Mentioned
Description	For tasks labeled as **Keyword Mentioned**, the task definition describes a set of keyword labels that construct an infinite output space combined with the input text.
Original Definition	In this task, you will use your knowledge about language (and common sense) to determine what element the marked number refers to. [...] Options to choose from are: REFERENCE: Some object which is being mentioned in the text before or after the target number. The reference answer has a higher priority than any other. If both Reference and another answer are possible, prioritize the Reference. YEAR: Describing a calendric year AGE: Describing someone’s age CURRENCY: Reference to some monetary value e.g dollar, euro etc. PEOPLE: Describing a single/plural persons TIME: Describing a time of the day. Usually you can add the word o’clock after those numbers. OTHER: Some other option, which isn’t listed here.
Output Space	Infinite Set: ["YEAR", "AGE", "CURRENCY", "PEOPLE", "TIME", "OTHER", "REFERENCE phone number", "REFERENCE crooler" ... ∞]
Simplified Definition	"Keyword Label: YEAR. Keyword Label: AGE. Keyword Label: CURRENCY. Keyword Label: PEOPLE. Keyword Label: TIME. Keyword Label: OTHER. Keyword Label: REFERENCE."
	No Mentioned
Description	For tasks labeled as **No Mentioned**, the task definition does not describe the output space by providing keyword labels.
Original Definition	In this task, you’re expected to write answers to questions involving multiple references to the same entity. The answer to the question should be unambiguous and a phrase in the paragraph. Most questions can have only one correct answer.
Output Space	Infinite Set: [∞]
Simplified Definition	""

Table 4: We describe how we created *Simplified* task definition from *Original* task definition for four task definition types: **Exact Mentioned**, **Combined Mentioned**, **Keyword Mentioned**, and **No Mentioned**. For each task definition type, *Description* describes how the task definition provides the output space information; *Original Definition* shows an example of a task definition within this definition type, which are all retrieved from real tasks in NatInst-V2 dataset; *Output Space* describes the set of the output space; *Simplified Definition* shows an example of how we simplified the Original Task Definition into the simplified version.

task036_qasc_topic_word_to_generate_related_fact	
Task Definition	In this task, you need to write a topic word from the given fact. The topic word must have at least one word overlap with the given fact. The topic word often involves adding a new word from a related concept. In your topic word, use at least one word from the given fact. Topic words with two or more words work best.
Task Example	**Input:** Fact: pesticides cause pollution. **Output:** pollution harms.
task1152_bard_analogical_reasoning_causation	
Task Definition	Two analogies that relate actions with their consequences are given in the form "A : B. C : ?". The phrase "A : B" relates action A to consequence B. Your task is to replace the question mark (?) with the appropriate consquence of the given action C, following the "A : B" relation. Your answer should be a single verb, without further explanation.
Task Example	**Input:** throw : fly. aspire : ? **Output:** attain
task1159_bard_analogical_reasoning_containers	
Task Definition	Two analogies that relate items to the associated containers is given in the form "A : B. C : ?". "A : B" relates item A to its associated container B. Your task is to replace the question mark (?) with the appropriate container for the given item C, following the "A : B" relation.
Task Example	**Input:** soda : can. water : ? **Output:** bottle

Table 5: We provide several examples of these analogical tasks.

task036_qasc_topic_word_to_generate_related_fact
task1152_bard_analogical_reasoning_causation
task1154_bard_analogical_reasoning_travel
task1157_bard_analogical_reasoning_rooms_for_containers
task1158_bard_analogical_reasoning_manipulating_items
task1159_bard_analogical_reasoning_containers

Table 6: List of tasks with the most performance drop when using *Delusive* examples for *Original* model.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
7

☑ A2. Did you discuss any potential risks of your work?
7

☑ 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?

Left blank.

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

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We are using NatInst-V2 which is an open-source dataset open to everyone. Also our code base is based on the their published repository.

☑ 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)?
3

☒ 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?
We're using a well-known open-source dataset. We've look into the dataset and don not see these issues.

☑ 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.
Left blank.

C ☑ Did you run computational experiments?

3

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
7

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

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
3

☑ 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?
3

☑ 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.)?
3

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.?
Not applicable. Left blank.

□ 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)?
Not applicable. Left blank.

□ 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?
Not applicable. Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.