How Many Data Samples is an Additional Instruction Worth?

Ravsehaj Singh Puri[*] **Swaroop Mishra**[*] **Mihir Parmar** **Chitta Baral**
Arizona State University, Tempe, USA
{rpuri8, srmishr1, mparmar3, chitta}@asu.edu

Abstract

Recently introduced *instruction-paradigm* empowers non-expert users to leverage NLP resources by defining a new task in natural language. Instruction-tuned models have significantly outperformed multitask learning models (without instruction); however they are far from state-of-the-art task-specific models. Conventional approaches to improve model performance via creating datasets with large number of task instances or architectural changes in the model may not be feasible for non-expert users. However, they can write alternate instructions to represent an instruction task. *Is Instruction-augmentation helpful?* We augment a subset of tasks in the expanded version of NATURAL INSTRUCTIONS with additional instructions and find that it significantly improves model performance (up to 35%), especially in the low-data regime. Our results indicate that an additional instruction can be equivalent to ∼200 data samples on average across tasks.[1]

1 Introduction

Large-scale benchmarks such as Imagenet (Russakovsky et al., 2015), SQuAD (Rajpurkar et al., 2018) and architectural development in models such as CNNs (Amari et al., 2003) and transformers (Vaswani et al., 2017) have propelled our progress in deep learning. However, creating high-quality benchmarks by controlling its artifacts (Gururangan et al., 2018; Mishra et al., 2020), developing new models, and training them is hard for non-expert users. Recently introduced *instruction-paradigm* empowers non-expert users, practitioners, and domain experts in other fields to leverage NLP resources (Weller et al., 2020) as they now can describe their tasks in natural language without requiring to create task-specific datasets or developing models[2]. Even though the instruction paradigm

has led to the development of models that significantly outperform multitasking baselines, model performance has remained far behind the supervised learning model trained with task-specific data (Efrat and Levy, 2020; Mishra et al., 2021b).

Non-expert users can write multiple instructions per task each of which covers multiple perspectives spanning over a variety of linguistic features; many of these can be created automatically by replacing certain words with their synonyms without changing the overall semantics of instruction. Can the relatively inexpensive process of instruction augmentation improve the model's performance in the *instruction-paradigm*, similar to the role data-augmentation has played conventionally in machine learning (Feng et al., 2021)? *Instruction-paradigm* is pivotal where it is expensive or infeasible to gather training data. How effective is instruction augmentation in low-data regimes?

Multi-variant instructions (original + augmented instructions) also can help evaluate the robustness of instruction-following models to respond to variant instructions. This is similar to the model robustness evaluation (Jia et al., 2019) that is done by creating variant data instances. Multi-variant instruction-based setup will also help gauge the true potential of instruction-following systems since in a real-world setting, users can write task instructions in many different ways.

The expanded version of NATURAL INSTRUCTIONS (Mishra et al., 2021b; Wang et al., 2022b)[3] provides a rich collection of the diverse category of tasks that covers a variety of reasoning skills, domains, and languages. This constantly evolving benchmark is growing in size with respect to time. We take 426 tasks[4] and creates variant instructions

---

[3]https://github.com/allenai/natural-instructions
[4]These were the accepted tasks in the expanded version of NATURAL INSTRUCTIONS in September 2021. The expanded dataset is also known as NATURAL INSTRUCTIONS v2 or SUPER-NATURALINSTRUCTIONS.

for each task. In NATURAL INSTRUCTIONS, the number of instances was limited to 6500 to reduce massive data imbalance, we leverage the remaining instances of source datasets in constructing instances of our variant instruction tasks. We experiment with 3 types of learning scenarios (i) task-specific (TS), (ii) multi-task (MT), and (iii) cross-task (CT) and observe that instruction augmented models outperform their single-instruction counterparts by 17%, 11%, and 11%, respectively when averaged over all experiments across the evaluation tasks. Interestingly, instruction augmentation is more effective on the low-data regime (average across 1%, 5%, and 10% data) as we see a performance gain of 26%, 16%, and 11% in TS, MT, and CT settings, respectively. We also quantify the contribution of each of the additional instructions and find that an additional instruction can be equivalent to ∼200 data samples on average across tasks.

2 Multi-Variant Instruction Dataset

We construct a Multi-Variant Instruction dataset on top of various tasks in NATURAL INSTRUCTIONS. In total, our dataset has 426 different NLP tasks; each of which contains multi-variant instructions.

2.1 Variant Instruction Task

An instruction task in NATURAL INSTRUCTIONS contains the definition of the task, positive examples, negative examples, and instances. Figure 1 shows the schematic representation of variant instruction tasks where the blue boxes show the parts that differentiate variant instruction tasks from their original counterparts in NATURAL INSTRUCTIONS. While constructing a variant instruction task, we alter the definition and instances of the instruction task.

Parameter	Value
Avg. # of variants per task	4.59
Avg. # of instances per task	9510.64
Avg. # of positive examples per task	3.15
Avg. # of negative examples per task	2.30

Table 1: Multi-Variant Instructions dataset statistics

2.2 Dataset Creation Process

Computer Science graduate students who participated in the data creation process are asked to create as many variant instruction tasks as possible. They are instructed to change the definition
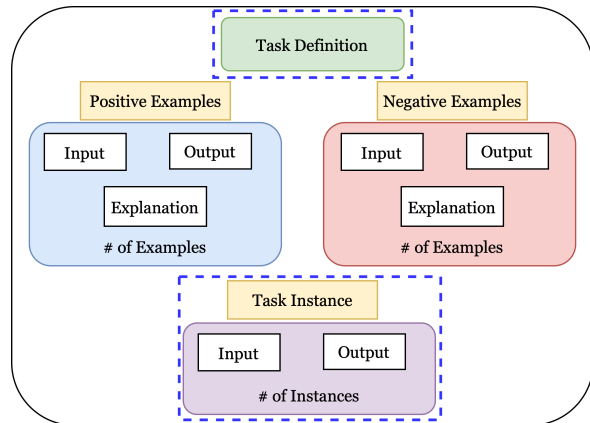

Figure 1: Schematic representation of instructional-prompts (Wang et al., 2022b) - Dotted blue box represents entities that are changed in constructing variant instruction task.

(without changing the semantic meaning of the definition in the original task) and instances (by random sampling from the set of instances in the source dataset which is not part of instruction tasks in NATURAL INSTRUCTIONS. They are allowed to use automated tools such as Semantic Control (Ross et al., 2021), Text Style Transfer (Reif et al., 2021), NL-Augmenter (Dhole et al., 2021). Sometimes, the participants create variant instruction tasks manually. Table 5 and Table 6 in App. B illustrates examples of alternate definitions across variant instructions created for our dataset.

2.3 Dataset Properties and Statistics

Table 1 shows the statistics of our meta-dataset. Note that, variant instruction tasks contain all instances from NATURAL INSTRUCTIONS, so the average number of instances per task is higher than 6500 (which is a constraint in NATURAL INSTRUCTIONS). We describe various attributes of our dataset in the following.

2.3.1 Semantic Textual Similarity

Semantic Textual Similarity (STS) should be high between original instruction and augmented instructions as they represent the same task. We compute the pair-wise STS score between definitions of original instruction and variant instructions. Figure 2 shows the mean and SD of STS score between original instruction and its variants across 426 tasks. More detail is presented in App. C.

Analysis of dataset properties From all dataset properties, we can observe that STS score is higher for almost all the tasks. This indicates that all aug-

Task ID	Task Name	Task Category	# of Variants
task010	winogrande_answer_generation	Answer Generation	8
task011	winogrande_question_modification_object	Text Modification	8
task012	winogrande_question_modification_person	Text Modification	8
task017	qasc_question_generation	Question Generation	8
task018	qasc_answer_generation	Answer Generation	8
task020	essential_terms_answering_incomplete_questions	Classification	8
task028	multirc_correct_answer_single_sentence	Answer Generation	3
task058	babi_t1_single_supporting_fact_answer_generation	Answer Generation	5

Table 2: Number of variant instructions for 8 different tasks
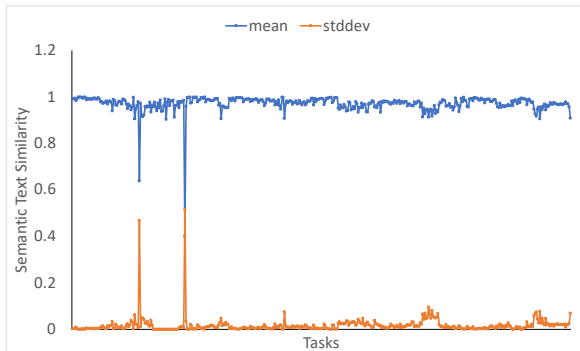

Figure 2: Semantic text similarity between original instruction and its variants.

mented variants are semantically similar to the original instruction. Moreover, we can see a significant variation in terms of word dissimilarity and length of definitions (see App. C). From this, we can conclude that the variants created in our meta-dataset for each task have sufficient variations in terms of words and length yet sustain semantic similarity with original instruction.

3 Experimental Setup

3.1 Models

BART-base (Lewis et al., 2019) and T5-base (Raffel et al., 2020) models are used with default hyper parameters from Huggingface (Wolf et al., 2019) to perform experiments. We use Single Instruction (SI) learning as baseline where only original instruction is used to fine-tune the model. We propose Multi-Variant Instruction (MVI) learning where variants are used to fine-tune models. We use the same number of instances for both original and variant instruction learning to accurately gauge the importance of additional instructions.

3.2 Experiments

We perform three experiments: (1) Task-Specific, (2) Multi-Task, and (3) Cross-Task. All experiments are performed using 1%, 5%, 10%, 50% and 100% instances from the task for fine-tuning. Here, we divide instances into train, test and dev splits by randomly sampling in the ratio 70%, 20% and 10%, respectively. Evaluation is performed on the test set of original instructions. As SI is dependent on NATURAL INSTRUCTIONS which has exactly one instruction per task, this limits our experiments to use only one instruction in the SI setting while comparing it with MVI which has multiple variant instructions.

Task-Specific Here, we fine-tune the baseline and our model on one task and evaluate on the same task. We have performed task-specific learning on 3 different tasks - winogrande_answer_generation, winogrande_question_modification_person, and qasc_answer_generation. In addition, we also analyze two different tasks in other task categories like tweetqa_question_generation and odd-man-out_classification_no_category for generation and classification tasks respectively.

Multi-Task To perform multi-task learning, we use 8 different tasks spanning across 4 different categories. Table 2 shows the different number of variant instructions for 8 tasks and their categories. In this setting, we fine-tune the baseline and our model on all 8 tasks combined and evaluate on each task. However, we use only two positive and two negative examples to satisfy the maximum token limit of the BART-base.

Cross-Task Here, we fine-tune the model on a set of tasks and evaluate on a different set of tasks. Here, we use 274 different tasks for training by sampling 10% instances from each task and evaluate on a set of 8 tasks which are the same as in the

multi-task setup. In addition to sampling instances, we also sampled number of tasks by taking 1%, 5%, 10%, 50%, and 100% tasks. We also investigate the extent of cross-task generalization in low-data regimes; we do this by randomly sampling 1%, 5%, and 10% instances for fine-tuning.

Metric   We use the Rouge-L metric (Lin, 2004) for evaluation in all our experiments, following the evaluation in NATURAL INSTRUCTIONS.

4   Results and Analysis

4.1   Experimental Results

Task-Specific   Figure 3 shows the comparison between SI and MVI across a different number of instances sampled for fine-tuning. From this, we can observe that MVI outperforms SI by 17% on average. The performance difference between MVI and SI increases to 26% in a low data regime (average performance with 1%, 5%, and 10% instances for fine-tuning). We observe similar results for the additional 2 tasks we have analyzed (present in App. D).
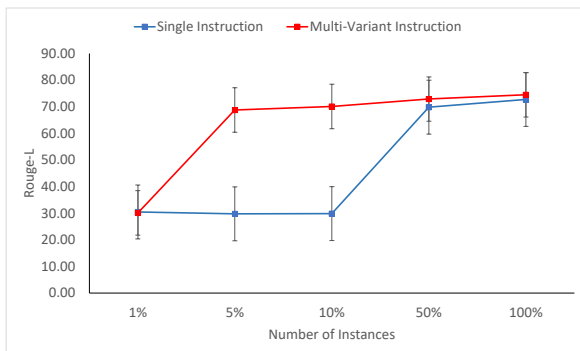

Figure 3: Comparison across SI and MVI learning in task-specific setting; Results are averaged over 3 tasks.

Multi-Task   Figure 4 presents the comparison between SI and MVI for multi-task setting. We can observe that MVI outperforms SI by 11% on an average. Moreover, we can see higher improvement in low data regime ($\sim 16\%$). Our model achieves high performance boost ($\sim$35%) at 1% instances setting. App. E contains more details.

Cross-Task   Figure 5 shows a comparison between SI and MVI for 100% tasks in cross-task setting (see Figure 9 in App. F for other settings). We can observe that MVI outperforms SI by 9% on an average. App. F contains more details.
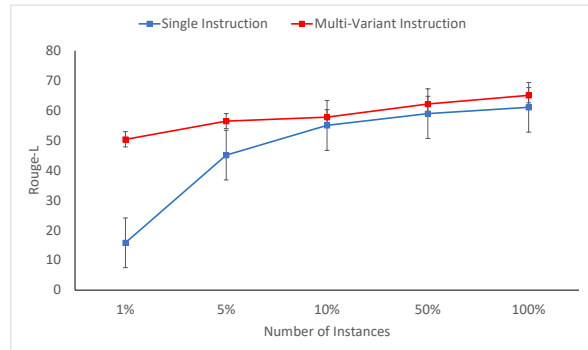

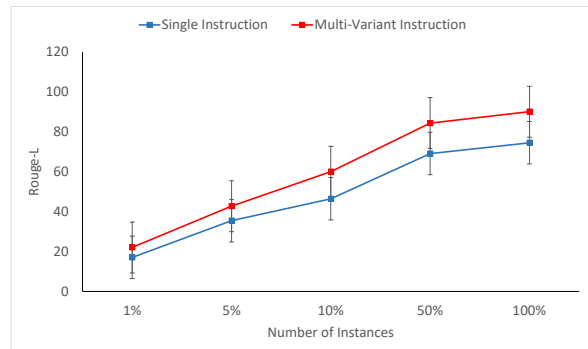Figure 4: Comparison across SI and MVI learning in multi-task setting by varying number of instances.

Figure 5: Comparison between SI and MVI learning in cross-task setting by varying number of instances and fixing number of tasks to 100%.

4.2   Analysis

How Many Data Samples is a Variant Instruction Worth?   We calculate the contribution of an additional instruction with respect to data samples in the following way: we calculate model performance for BART-base in MVI with 5% instances. We interpolate the model performance plot in SI to find out the percentage of instances needed to match performance in MVI (with 5% instances). We divide the average number of instance difference by average number of instruction variants to get the number that indicates worth of an additional instruction in terms of data samples. Using the above described procedure, we calculate the contribution for additional instruction in all three settings and summarize the results in Table 3. We use MVI performance with 5% instances as the base because a typical instruction-paradigm is designed in a "low-data regime" where non-expert users can teach a task to a model without requiring to create a dataset. However, we also calculated the instruction-equivalence using MVI with 10% instances as the base and report the results in Table

3. On an average across TS, MT and CT, we conclude that an additional variant instruction alone is worth ∼200 instances.

Base	Task-Specific	Multi-Task	Cross-Task	Average
5%	456.2	94.1	152.3	234.2
10%	460.4	58.2	279.6	266.1

Table 3: Weight of each additional instruction in terms of number of data samples across task-specific, multi-task and cross-task settings.

Equal Data Analysis   We believe that each instruction variant is equivalent to ∼200 data instances. To show this by experiment, we perform equal data analysis and observe that model trained using our approach shows competitive performance compared to single-instruction learning by using only N/V instances where N is the total number of instances in the original task and V is the number of instruction variants for this task. See App. G for more details.

Is Model Robust to Instruction Perturbations?
Here, we introduce 3 perturbations while testing SI and MVI: (1) we perturb the instruction by removing the task definition, (2) we perturb the instruction by changing the order of positive and negative examples by placing positive examples followed by negative different from training setup, and (3) we perturb the instruction by removing all positive and negative examples from the test set. We evaluate the model's robustness across these perturbations (performance change while the change in instruction) which are excluded from the training data. Here, Table 4 for task-specific setting on T5-base (see Table 11 in App. H for multi-task results). We can clearly observe that our approach is robust to all three instruction perturbations whereas model trained with single-instruction learning is not able to perform equally well on perturbed test sets compared to its original test counterpart. A similar trend is observed in the multi-task setting as well (see App. H).

5   Conclusion

We introduced instruction augmentation to improve existing LMs in terms of improving performance and usability to non-expert users. To this extent, we created multi-variant instructions for 426 NLP tasks. Our experiment results show that instruction augmentation improves model performance in task-specific, multi-task and cross-task learning

# of Instances	SI Original	SI Ours	Perturbation 1 Original	Perturbation 1 Ours	Perturbation 2 Original	Perturbation 2 Ours	Perturbation 3 Original	Perturbation 3 Ours
1%	0.90	25.21	1.60	18.03	1.02	23.16	5.12	9.71
5%	0.98	75.72	2.18	75.32	1.36	75.50	5.52	74.26
10%	50.88	78.20	20.76	78.07	50.49	78.37	40.31	77.22
50%	76.55	82.16	68.88	82.15	76.50	82.16	75.34	81.92
100%	79.38	83.16	73.51	82.97	79.34	83.12	78.71	82.40

Table 4: Comparison of performance in task-specific setting across SI and MVI learning.

paradigms. We find that instruction augmentation is more effective in low-data regime. Our results further indicate that an additional instruction can be equivalent to ∼200 instances on an average. We hope our work will bring more attention to developing unconventional techniques (beyond dataset creation and model training) to empower non-expert users to leverage NLP resources and teach a task without having domain knowledge.

Limitations

We use BART-base and T5-base for all our experiments, however, we wish to experiment with different language models in future to show the benefit of our approach. Our analysis includes only tasks in English language, hence, it is important to see if our approach can be extended to non-English tasks as well. We feel that developing diverse instruction augmentation techniques will be pivotal to achieving more improvements as future research.

References

Shunichi Amari et al. 2003. *The handbook of brain theory and neural networks*. MIT press.

Xiang Chen, Xin Xie, Ningyu Zhang, Jiahuan Yan, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2021. Adaprompt: Adaptive prompt-based finetuning for relation extraction. *arXiv e-prints*, pages arXiv–2104.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using bart. *arXiv preprint arXiv:2106.01760*.

Kaustubh D Dhole, Varun Gangal, Sebastian Gehrmann, Aadesh Gupta, Zhenhao Li, Saad Mahamood, Abinaya Mahendiran, Simon Mille, Ashish Srivastava, Samson Tan, et al. 2021. Nl-augmenter: A framework for task-sensitive natural language augmentation. *arXiv preprint arXiv:2112.02721*.

Avia Efrat and Omer Levy. 2020. The turking test: Can language models understand instructions? *arXiv preprint arXiv:2010.11982*.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*.

Tanmay Gupta, Amita Kamath, Aniruddha Kembhavi, and Derek Hoiem. 2021. Towards general purpose vision systems. *arXiv preprint arXiv:2104.00743*.

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.

Peter Hase and Mohit Bansal. 2021. When can models learn from explanations? a formal framework for understanding the roles of explanation data. *arXiv preprint arXiv:2102.02201*.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4129–4142.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Kirby Kuznia, Swaroop Mishra, Mihir Parmar, and Chitta Baral. 2022. Less is more: Summary of long instructions is better for program synthesis. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4532–4552, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR.

Teven Le Scao and Alexander M Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, et al. 2021. Few-shot learning with multilingual language models. *arXiv preprint arXiv:2112.10668*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pretrain, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Man Luo, Sharad Saxena, Swaroop Mishra, Mihir Parmar, and Chitta Baral. 2022. Biotabqa: Instruction learning for biomedical table question answering. *arXiv preprint arXiv:2207.02419*.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*.

Swaroop Mishra, Anjana Arunkumar, Bhavdeep Singh Sachdeva, Chris Bryan, and Chitta Baral. 2020. Dqi: Measuring data quality in nlp. *ArXiv*, abs/2005.00816.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2021a. Reframing instructional prompts to gptk's language. *arXiv preprint arXiv:2109.07830*.

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

Swaroop Mishra and Elnaz Nouri. 2022. Help me think: A simple prompting strategy for non-experts to create customized content with models. *arXiv preprint arXiv:2208.08232*.

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. *Preprint*.

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

Pruthvi Patel, Swaroop Mishra, Mihir Parmar, and Chitta Baral. 2022. Is a question decomposition unit all we need? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4553–4569, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2021. A recipe for arbitrary text style transfer with large language models. *arXiv preprint arXiv:2109.03910*.

Alexis Ross, Tongshuang Wu, Hao Peng, Matthew E Peters, and Matt Gardner. 2021. Tailor: Generating and perturbing text with semantic controls. *arXiv preprint arXiv:2107.07150*.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

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. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Liwen Wang, Rumei Li, Yang Yan, Yuanmeng Yan, Sirui Wang, Wei Wu, and Weiran Xu. 2022a. Instructionner: A multi-task instruction-based generative framework for few-shot ner. *arXiv preprint arXiv:2203.03903*.

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

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*.

Orion Weller, Nicholas Lourie, Matt Gardner, and Matthew E Peters. 2020. Learning from task descriptions. *arXiv preprint arXiv:2011.08115*.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–22.

Qinyuan Ye and Xiang Ren. 2021. Zero-shot learning by generating task-specific adapters. *arXiv e-prints*, pages arXiv–2101.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161*.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. *arXiv preprint arXiv:2104.04670*.

1048

A Related Work

Prompt Learning Due to the success of large LMs, research paradigm in ML/DL has been shifted to prompt-based learning to achieve generalization and eliminate the need of creating task-specific models and large scale datasets (Liu et al., 2021). Past attempts have been made using prompt-based learning to solve various tasks including text classification (Yin et al., 2019), Natural Language Inference (NLI) (Schick and Schütze, 2020), Question Answering (QA) (Jiang et al., 2020), Information Extraction (IE) (Chen et al., 2021; Cui et al., 2021) and many more (Liu et al., 2021). Recently, T0 model (Sanh et al., 2021) is proposed which uses prompts to achieve zero-shot generalization across various NLP tasks. We were motivated by the work of Le Scao and Rush (2021) which shows that prompting is often worth 100s of data points on average. Our work instead focuses on instructions that are often different in terms of length, language, and capacity to represent a task (Wang et al., 2022b). Additionally, in contrast to prior works, we focus on the use of automatic methods for instruction augmentation and evaluate its efficacy across low-data to high-data regime in task-specific, multi-task, cross-task setups.

Instruction Learning Efrat and Levy (2020) studies whether existing LMs understands instructions. After that, many works have been proposed to show that models follow language instructions (Hase and Bansal, 2021; Ye and Ren, 2021; Gupta et al., 2021; Zhong et al., 2021). Furthermore, (Weller et al., 2020) has developed a framework that focuses on developing NLP systems that solve new tasks after reading their descriptions. Mishra et al. (2021b) has proposed natural language instructions for cross-task generalization of LMs. Along with that, PromptSource and FLAN (Wei et al., 2021; Sanh et al., 2021) were built for leveraging instructions and achieving zero-shot generalization on unseen tasks. Moreover, Parmar et al. (2022) shows the effectiveness of instructions in multi-task settings for the biomedical domain. Mishra et al. (2021a) discuss the impact of task instruction reframing on model response. Min et al. (2021) introduce a framework to better understand in-context learning. Ouyang et al. (2022) propose the InstructGPT model that is fine-tuned with human feedback to follow instructions. Wang et al. (2022a) has developed instruction-based multi-task

framework for few-shot Named Entity Recognition (NER) tasks. In addition, many approaches have been proposed to improve model performance using instructions (Wu et al., 2022; Lin et al., 2021; Wang et al., 2022b; Luo et al., 2022; Kuznia et al., 2022; Patel et al., 2022; Mishra and Nouri, 2022).

B Example of Variants

Table 5 and Table 6 show the examples of different variants created from the task117_afs_argument_similarity_gun_control and task018_qasc_answer_generation respectively.

C Multi-Variant Dataset Additional Details

C.1 Semantic Textual Similarity

We use en_core_web_md semantic similarity model of SpaCy to compute STS in our experiments. We also calculate STS score between definitions of variants of the same task. At the end, we calculate their mean and Standard Deviation (SD) for each task.

In the plot, the two exception points are task058 (Answer generation task based on babi dataset (Weston et al., 2015)) and task097 (Structured text generation task based on SCAN dataset (Lake and Baroni, 2018)) where the original instructions are very long and the variant task contains a short definition which causes the strong variation in STS. We also discuss the Word-Level Dissimilarity and Length Diversity properties of our dataset below.

C.2 Word-Level Dissimilarity

To show the quality and diversity of variant instructions, we calculate the pair-wise edit distance between the definition of the original instruction and its variant instructions. We also calculate distance between definitions of variant instructions of the same task, further normalize by the highest distance to obtain a dissimilarity score. We compute the mean and SD of these scores for each task and show it in Figure 6.

C.3 Length Diversity

It is necessary to see how task definition lengths vary between original instructions and their variants. To understand this, we compute the percentage difference between the length of the maximum instruction definition and the minimum instruction definition for each task and show it in Figure 7.

	Original instruction along with its augmented variant instructions
ORIGINAL INSTRUCTION	***Definition:*** *We would like you to classify each of the following sets of argument pairs (discussing Gun Control) into either SIMILAR or NOT SIMILAR. A pair of arguments is considered SIMILAR if the arguments are about the same FACET (making the same argument), and is considered NOT SIMILAR if they do not have the same FACET. A FACET is a low level issue that often reoccurs in many arguments in support of the author's stance or in attacking the other author's position.* ***Negative Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\> ***Positive Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\>
VARIANT INSTRUCTION 1	***Definition:*** *Each of the following sets of argument pairs (on the topic of Gun Control) should be classified as SIMILAR or NOT SIMILAR. If the arguments are about the same FACET (making the same argument), they are deemed SIMILAR; otherwise, they are NOT SIMILAR. A FACET is a low-level problem that appears frequently in many arguments in favor of the author's position or in opposition to the position of the other author.* ***Negative Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\> ***Positive Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\>
VARIANT INSTRUCTION 2	***Definition:*** *Please classify the following sets of argument pairs (discussing the Gun Control) as SIMILAR or NOT SIMILAR. If the arguments are about the same FACET (making the same argument), they are regarded SIMILAR; if they are not, they are considered NOT SIMILAR. A FACET is a low-level problem that frequently recurs in numerous arguments in favor of the author's position or in opposition to the position of the other author.* ***Negative Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\> ***Positive Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\>
VARIANT INSTRUCTION 3	***Definition:*** *Two arguments are SIMILAR if they are making the same case related to author's position, else they are NOT SIMILAR. Your task is to classify any 2 arguments as SIMILAR or NOT SIMILAR.* ***Negative Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\> ***Positive Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\>
VARIANT INSTRUCTION 4	***Definition:*** *Each of the following sets of argument pairs (discussing the Gun Control) should be classified as SIMILAR or NOT SIMILAR. If the arguments are about the same FACET (making the same argument), they are regarded SIMILAR; otherwise, they are NOT SIMILAR. A FACET is a low-level issue that appears frequently in many arguments in support of the author's position or in opposition to the position of the other author.* ***Negative Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\> ***Positive Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\>

Table 5: Example of an instruction for a classification task with its variant instructions; these belong to the task117_afs_argument_similarity_gun_control.

	Original instruction along with its augmented variant instructions
ORIGINAL INSTRUCTION	***Definition:*** *Write a correct answer to the given question based on its associated fact. Make sure that your answer is contained in the associated fact. Things to avoid: Don't be creative and introduce any new word that is not mentioned in the associated fact! Remember that, the associated fact has been rearranged to form the question. So, the correct answer words must lie within the associated fact. Emphasis & Caution: The correct answer can be a word, phrase, or even a sentence.* ***Negative Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\> ***Positive Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\>
VARIANT INSTRUCTION 1	***Definition:*** *Handwriting a rectify reply to the given issue based on its related fact. Make sure that your replying is contained in the associated fact. Aspects to avoidance: Don't be creativity and introduces any nouveau word that is not alluded in the associated doing! Recall that, the linked doing has been restructured to forma the question. Thus, the corrects replying words needs lie within the associated doing. Focuses & Discretion: The exact replying can be a word, phrase, or even a penalties.* ***Negative Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\> ***Positive Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\>
VARIANT INSTRUCTION 2	***Definition:*** *Write a correcting responding to the gave question bases on its associated fact. Make persuaded that your answering is contained in the associated facto.Matters to shirk: Don't be inventive and introduce any nouveau word that is not referred in the associated fact! Recollect that, the associated fact has been redesigned to forma the issue. Therefore, the accurate responses words owes lying inside the associated doing. Concentrating & Circumspect: The correcting responses can be a word, phrase, or even a punishments.* ***Negative Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\> ***Positive Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\>
VARIANT INSTRUCTION 3	***Definition:*** *Write a corrects answer to the afforded issue founded on its associated fact. Deliver sure that your replied is contain in the linked fact. Things to shirk: Don't be creative and introduce any novel word that is not alluded in the associated fact! Remind that, the associated doing has been redesigned to forme the question. Accordingly, the correcting reply phrases needs lied indoors the linked fact. Concentrates & Caveat: The corrects response can be a word, phrase, or even a condemnation.* ***Negative Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\> ***Positive Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\>
VARIANT INSTRUCTION 4	***Definition:*** *Writing a accurate responded to the yielded matter founded on its associated fact. Deliver sure that your reply is contained in the associated doing. Aspects to avoidance: Don't be creative and introduce any newer word that is not talked in the associated facto! Recall that, the associated fact has been rearranged to form the issue. Thereby, the corrects responding phrase gotta lie within the related doing. Focus & Circumspect: The correct responding can be a word, expression, or even a sentences.* ***Negative Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\> ***Positive Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\>
VARIANT INSTRUCTION 5	***Definition:*** *Writing a correct answers to the granted question bases on its associated doing. Make sure that your respond is contained in the associated doing. Matters to shirk: Don't be creative and introduces any novo word that is not referenced in the associated facto! Remind that, the associated fact has been reconfigured to forms the question. So, the corrects respond words ought lies within the related doing. Concentrate & Careful: The accurate reply can be a word, phrase, or yet a sentences.* ***Negative Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\> ***Positive Examples:*** **Input**: \<input\> **Output**: \<output\> **Explanation**: \<explanation\>

Table 6: Example of an instruction for an answer generation task with its variant instructions - task018_qasc_answer_generation
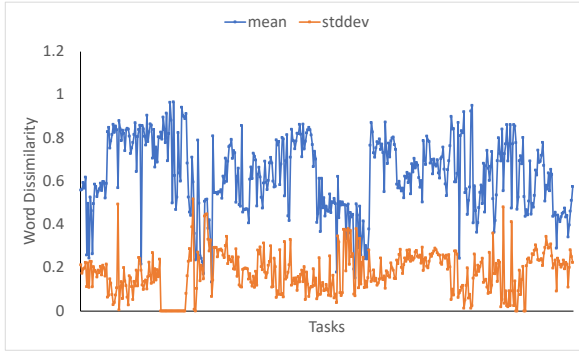
Figure 6: Word-level dissimilarity between original instruction and its variants.
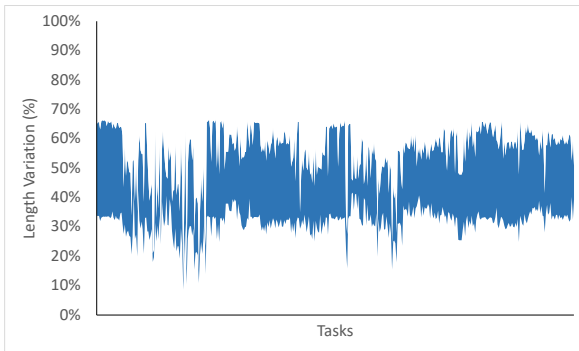

Figure 7: Definition length variation between original instruction and its variants.

D Task-Specific Results

Table 7 shows the results for task-specific experiments for task010_winogrande_answer_generation, task012_winogrande_question_modification_person and task018_qasc_answer_generation. We also performed experiments for other task categories like task210_tweetqa_question_generation and task113_odd-man-out_classification_no_category for generation and classification tasks respectively and summarize our results in Table 8. From the average results, we can observe that multi-variant instruction learning helps model to improve performance in task-specific learning.

E Multi-Task Results

The results for multi-task learning experiments are shown in Table 9.

F Cross-Task Results

The results for cross-task learning experiments are shown in Table 12. Figure 9 compares single-instruction learning and our approach in cross-task setting.

G Equal Data Analysis

We keep the original number of instances in SI learning, however, reduce the number of instances used in MVI learning by sampling N/V number of instances randomly for each task where N is the total number of instances in the original task and V is the number of instruction variants for this task. We perform these experiments in both task-specific and multi-task settings using BART-base. Table 10 summarizes the results of these experiments, and we can observe that the model trained using our approach shows competitive performance compared to single-instruction learning by using only N/V instances.

The results for cross-task learning experiments are shown in Table 12. Figure 9 compares single-instruction learning and our approach in cross-task setting.
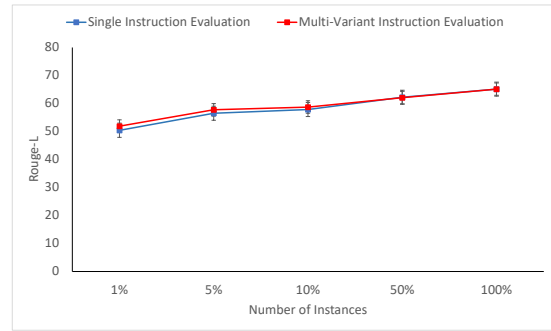
H Robustness Analysis

Is single-instruction learning robust? As Figure 8 illustrates, LM fine-tuned with single-instruction learning or original setting is not robust to instructions written in a different way; this includes transformation techniques like paraphrasing, adding spelling mistakes, grammatical mistakes etc. Our experiment results show that model trained using the proposed multi-variant instruction learning technique is able to perform reasonably well and is robust to variant instructions in both multi-task setting, as evident by lower performance difference between single instruction evaluation and multi-variant instruction evaluation setup.

I Contribution of Individual Variants

Do each of the variant instructions contribute equally towards performance gain? To analyse the contribution of each of the variant instructions, we study the performance gain by adding a single variant instruction at one time. We perform this analysis in TS setting (task_010) and MT setting and summarize the results in Table 13 and Table 14 respectively. We observe that all variants do not contribute equally, e.g. MVI_All above are often smaller than individual MVIs. Identifying optimal variants, however, will be a scope for future work.
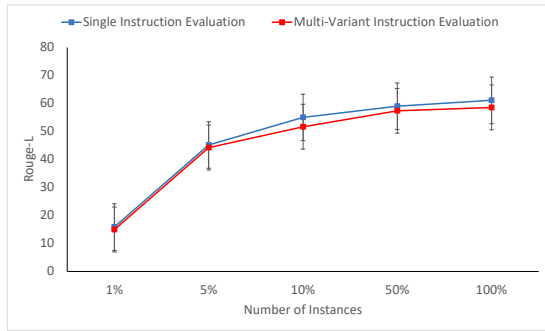
| (a) Multi-task SI learning | (b) Multi-task MVI learning |

Figure 8: Robustness comparison of SI *vs.* MVI in multi-task setting - LM fine-tuned using MVI learning is more robust to variants as compared to SI learning.

# of Instances	BART-base				T5-base			
	SI		MVI		SI		MVI	
	Original	Ours	Original	Ours	Original	Ours	Original	Ours
task_010								
1%	0.00	0.00	0.00	0.02	0.04	13.71	0.16	11.26
5%	0.00	36.75	0.06	37.07	0.01	46.44	0.14	44.69
10%	0.23	39.17	0.15	38.26	12.03	53.03	9.05	52.60
50%	37.00	43.02	25.40	42.54	48.11	64.94	46.01	64.80
100%	41.97	45.65	33.84	45.50	55.67	67.49	53.74	66.92
task_012								
1%	84.48	83.54	75.45	82.66	0.07	0.00	6.20	6.17
5%	84.73	90.68	74.52	90.68	0.05	90.90	6.17	90.87
10%	84.81	90.61	75.47	90.60	79.62	90.99	62.69	90.99
50%	90.29	90.49	85.65	90.48	90.92	90.77	90.81	90.81
100%	90.84	90.50	88.47	90.52	91.02	90.75	90.87	90.80
task_018								
1%	7.05	6.92	4.36	5.27	2.57	61.92	3.02	58.53
5%	4.65	79.07	3.42	79.55	2.89	89.84	3.80	89.99
10%	4.72	80.59	3.68	80.95	61.00	90.57	56.28	90.56
50%	82.43	85.23	81.36	85.20	90.63	90.76	90.86	90.79
100%	85.58	87.37	84.90	87.52	91.44	91.25	91.41	91.11
Average								
1%	30.51	30.15	26.60	29.32	0.90	25.21	3.12	25.32
5%	29.79	68.83	26.00	69.10	0.98	75.72	3.37	75.18
10%	29.92	70.12	26.43	69.94	50.88	78.20	42.67	78.05
50%	69.91	72.91	64.14	72.74	76.55	82.16	75.89	82.13
100%	72.80	74.51	69.07	74.51	79.38	83.16	78.68	82.94

Table 7: Comparison of performance in single-task setting across single-instruction and multi-variant instruction learning. SI: Single-Instruction, MVI: Multi-Variant Instruction.

# of Instances	SI	MVI	SI	MVI
	task_210		task_113	
1%	13.37	12.25	3.00	3.85
5%	13.50	25.92	4.77	15.26
10%	14.67	27.14	4.00	30.77
50%	27.88	41.06	41.72	81.80
100%	37.24	44.10	66.73	98.10

Table 8: Comparison of performance in task-specific setting across single-instruction and multi-variant instruction learning. SI: Single-Instruction

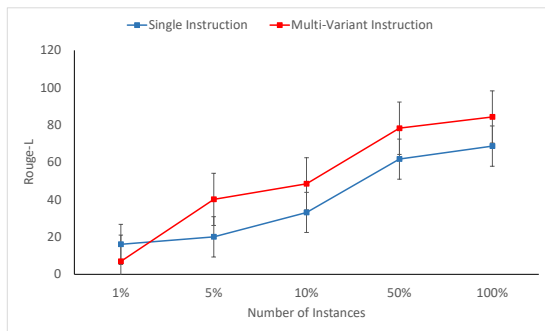| # of Instances	BART-base				T5-base			
	SI		MVI		SI		MVI	
	Original	Ours	Original	Ours	Original	Ours	Original	Ours
1%	15.84	50.40	14.97	51.88	7.34	34.53	6.11	33.61
5%	45.13	56.49	44.24	57.71	32.01	62.61	19.88	62.87
10%	55.03	57.80	51.67	58.70	46.93	63.61	39.76	63.98
50%	59.01	62.21	57.37	62.06	63.38	66.16	57.11	66.76
100%	61.08	65.13	58.58	65.09	64.99	67.15	59.35	67.38

Table 9: Comparison of performance in multi-task setting across single-instruction and multi-variant instruction learning. SI: Single-Instruction, MVI: Multi-Variant Instruction

| # of Instances	Single Task		Multi Task	
	Original	Ours	Original	Ours
1%	10.81	7.32	6.35	0.82
5%	20.86	19.42	4.21	6.31
10%	57.22	51.36	59.95	49.42
50%	76.53	72.75	84.54	79.74
100%	78.36	60.15	86.55	82.02
Average	48.76	42.20	48.32	43.66

Table 10: Comparison of performance in task-specific (average across 3 tasks) and multi-task settings.

# of Instances	SI		Perturbation 1		Perturbation 2		Perturbation 3	
	Original	Ours	Original	Ours	Original	Ours	Original	Ours
1%	7.34	34.53	7.73	39.76	7.23	33.27	3.37	35.32
5%	32.01	62.61	25.90	60.22	29.51	63.52	23.50	69.30
10%	46.93	63.61	46.36	61.70	44.74	63.86	43.28	72.46
50%	63.38	66.16	61.63	64.50	63.73	66.40	71.79	67.99
100%	64.99	67.15	63.12	67.38	65.05	66.02	72.70	68.24

Table 11: Comparison of performance in multi-task setting across single-instruction and multi-variant instruction learning.

(a) fixing number of tasks to 1%
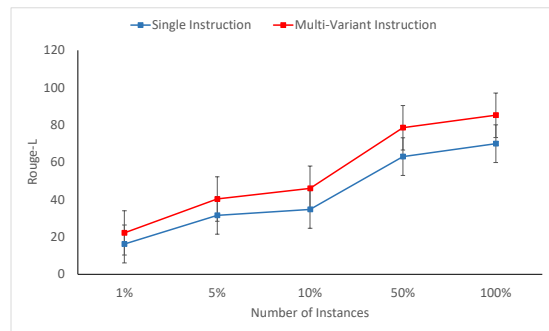
(b) fixing number of tasks to 5%

(c) fixing number of tasks to 10%

(d) fixing number of tasks to 50%

Figure 9: Comparison of performance across SI and MVI learning in cross-task setting by varying number of instances and tasks. Evaluation is performed on the test set of original instructions.

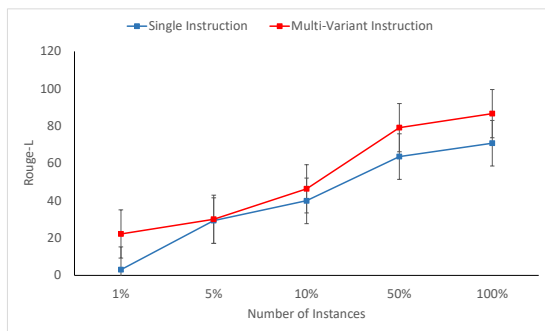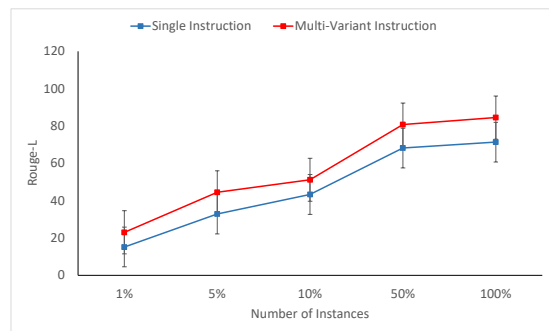| # of Instances	BART-base				T5-base			
	SI		MVI		SI		MVI	
	Original	Ours	Original	Ours	Original	Ours	Original	Ours
				1% tasks				
1%	16.00	6.94	10.93	10.16	0.96	7.36	0.87	7.31
5%	20.04	40.14	19.51	31.09	21.87	29.07	19.89	29.60
10%	33.09	48.43	31.83	47.66	36.17	44.50	33.13	45.28
50%	61.70	78.22	58.53	78.43	64.74	73.94	61.34	73.45
100%	68.66	84.22	64.39	84.87	72.35	83.37	68.9	84.2
				5% tasks				
1%	16.23	22.17	3.32	18.78	1.30	7.55	1.29	7.29
5%	31.58	40.3	29.81	33.12	22.85	29.04	20.44	29.02
10%	34.73	46.02	34.38	49.15	36.01	44.83	33.75	44.93
50%	63.06	78.48	60.5	79.76	65.96	76.25	61.01	76.13
100%	69.93	85.2	67.41	86.68	74.54	83.61	70.2	83.69
				10% tasks				
1%	2.98	22.16	2.46	19.98	3.12	7.89	2.56	7.66
5%	29.27	30.06	28.03	30.9	24.49	29.29	23.41	29.25
10%	39.95	46.38	36.3	50.4	36.76	45.22	36.23	44.81
50%	63.58	79.13	59.98	79.81	66.07	73.49	62.56	73.54
100%	70.82	86.66	69.11	87.86	71.97	81.16	70.34	81.08
				50% tasks				
1%	15.18	23.06	17.08	26.2	5.58	22.26	5.44	22.21
5%	32.88	44.5	33.88	44.64	33.56	40.37	30.57	38.25
10%	43.33	51.2	42.5	54.62	45.42	44.02	39.01	44.36
50%	68.18	80.8	66.42	81.29	66.62	80.97	63.89	80.93
100%	71.35	84.52	68.85	84.65	72.72	82.82	69.94	82.02
				100% tasks				
1%	17.04	22	19.2	24.95	20.69	22.55	9.02	20.66
5%	35.4	42.68	36.42	45.06	35.18	38.30	30.92	39.51
10%	46.4	60	45.33	59.3	44.70	53.80	44.47	54.15
50%	69.06	84.32	67.29	84.47	71.89	79.20	68.64	79.56
100%	74.45	90.01	72.26	90.35	74.03	81.53	72.34	82.15

Table 12: Comparison of performance in cross-task setting across single-instruction and multi-variant instruction learning. SI: Single-Instruction, MVI: Multi-Variant Instruction.

# of Instances	SI	MVI_1	MVI_2	MVI_3	MVI_4	MVI_5	MVI_6	MVI_7	MVI_All
1%	0.00	17.46	0.92	0.20	0.44	6.92	5.7	6.79	0.00
5%	0.00	34.34	35.84	36.90	37.36	39.96	37.72	37.97	36.75
10%	0.23	37.31	41.03	42.30	42.95	43.59	42.4	41.23	36.75
50%	37.00	44.25	59.30	57.18	59.45	61.82	62.93	44.14	43.02
100%	41.97	44.34	71.02	75.20	80.27	81.74	86.05	53.63	45.65

Table 13: Contribution of each variant instruction towards performance in task-specific setting for task010. SI: Single-Instruction, MVI_k: Multi-Variant Instruction where k equals number of variant instructions used.

# of Instances	SI	MVI_1	MVI_2	MVI_3	MVI_All
1%	15.84	37.03	40.93	64.08	50.4
5%	45.13	55.38	55.80	56.46	56.49
10%	55.03	58.17	58.32	57.70	57.8
50%	59.01	61.62	61.45	62.20	62.21
100%	61.08	62.90	64.08	64.10	65.13

Table 14: Contribution of each variant instruction towards performance in multi-task setting. SI: Single-Instruction, MVI_k: Multi-Variant Instruction where k equals number of variant instructions used.