How Many Data Samples is an Additional Instruction Worth?

Ravsehaj Singh Puri **Swaroop Mishra** **Mihir Parmar** **Chitta Baral**

Arizona State University

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 large datasets with lots of task instances or architectural/training changes in 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 these significantly improve 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 SNLI (Bowman et al., 2015) and architectural development in models such as CNNs (Amari et al., 2003) and transformers (Vaswani et al., 2017; Devlin et al., 2018; Raffel et al., 2019; Brown et al., 2020) have propelled our progress in deep learning. However, creating high quality benchmark by controlling its artifacts(Gururangan et al., 2018; Mishra et al., 2020; Swayamdipta et al., 2020), developing new models and training them are 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. Even though instruction-paradigm has led to 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 improves 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 instruction-augmentation is 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 instruction in many different ways.

The expanded version of NATURAL INSTRUCTIONS (Mishra et al., 2021b)[2] provides a rich collection of diverse category of tasks that covers a variety of reasoning skills, domains, and languages. This is a constantly evolving benchmark which is growing in size with respect to time. We take 426 tasks [3] and create variant instructions for each task.

[1] Code and data available at link

[2] https://github.com/allenai/natural-instructions

[3] These were the accepted tasks in NATURAL INSTRUC-

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 counterpart by 17%, 11% and 11% respectively when averaged over all experiments across the evaluation tasks. Interestingly, instruction augmentation is more effective on low-data regime [4] as we see performance gain of 26%, 16% and 11% in TS, MT and CT setting. We also quantify the contribution of each of the additional instruction and find that an an additional instruction can be equivalent to $\sim$200 data samples on average across tasks.

In summary, our contributions are as follows: (a) we augment a subset of NATURAL INSTRUCTIONS with variant instruction tasks that contain alternate definitions and new instances. In NATURAL INSTRUCTIONS, number of instances were limited to 6500 to reduce massive data imbalance, we leverage remaining instances of source datasets in constructing instances of our variant instruction tasks; (b) we show the efficacy of instruction augmentation across a variety of learning scenarios; (c) we conduct experiment across varying number of instances and tasks that empirically quantify the equivalence between an additional instruction and data samples.

2 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 which are often longer than prompts.

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 focus on developing NLP systems that solve new tasks after reading their descriptions. Mishra et al. (2021b) has proposed natural language instructions to improve the performance of LMs in cross-task generalization. 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. 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 on a large set of training tasks. Ouyang et al. (2022) propose the InstructGPT model that is fine-tuned with human feedback to follow instructions. Wang et al. (2022) has developed instruction-based multi-task framework for few-shot Named Entity Recognition (NER) task. Motivated by these previous works, we introduce instruction augmentation and its impact on learning in LMs.

3 Multi-Variant Instruction Dataset

We construct our Multi-Variant Instruction dataset on top of various tasks in NATURAL INSTRUCTIONS. In total, our dataset has 426 different NLP tasks; each of which contain multi-variant instructions. Here we first define variant instruction tasks, then we describe the dataset creation process and finally we shed light on various properties and statistics of our dataset.

3.1 Variant Instruction Task

An instruction task in NATURAL INSTRUCTIONS contains definition of the task, positive examples, negative examples and instances. While constructing a variant instruction task, we alter the definition and instances of the instruction task[5]. Figure 1 shows the schematic representation of variant instruction task where the blue boxes show the parts that differentiate variant instruction tasks with their original counterparts in NATURAL INSTRUCTIONS.

TIONS in September 2021

[4] Average across 1%, 5% and 10% data

[5] We do this wherever there exist extra instances in the source dataset that did not get used in the instruction task of NATURAL INSTRUCTIONS

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

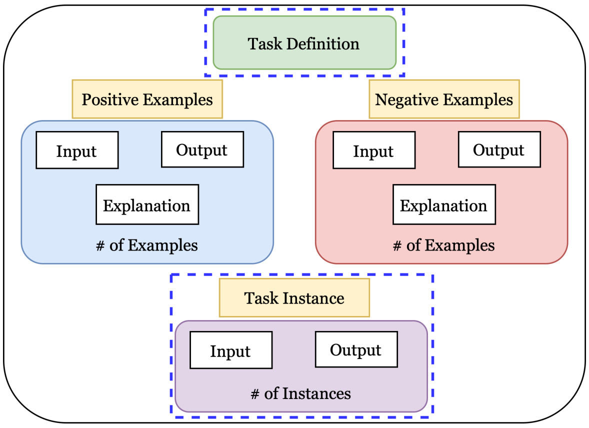


Figure 1: Schematic representation of instructional-prompts (Mishra et al., 2021b) - Dotted blue box represents entities which are changed in constructing variant instruction task.

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 2: Multi-Variant Instructions dataset statistics

3.2 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 definition (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 are 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 1 illustrates an example of alternate definitions across variant instruction tasks for AFS (Misra et al., 2017) classification task in NATURAL INSTRUCTIONS. We provide an additional example of alternate definitions across variant instruction tasks for QASC (Khot et al., 2020) question answering task in Table 3

3.3 Dataset Properties and Statistics

Table 2 shows the statistics of our meta-dataset. Note that, variant instruction task still contain a max of 6500 instances, following the instance constraint of tasks in NATURAL INSTRUCTIONS. However, variant instruction tasks can have different instances and so the average number of instances per task is higher than 6500. We explain various dataset properties as follows:

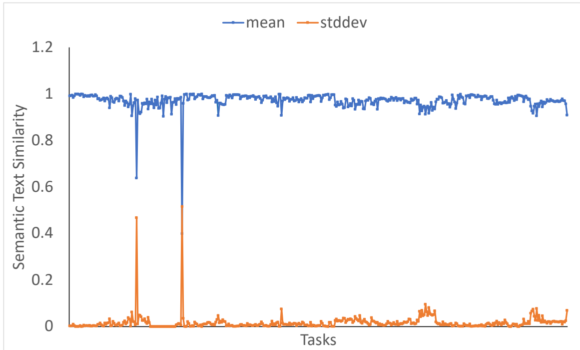


Figure 2: Semantic text similarity between original instruction and its variants - Variant instructions have high semantic similarity with each other (represented through high mean) across all the tasks in our dataset.

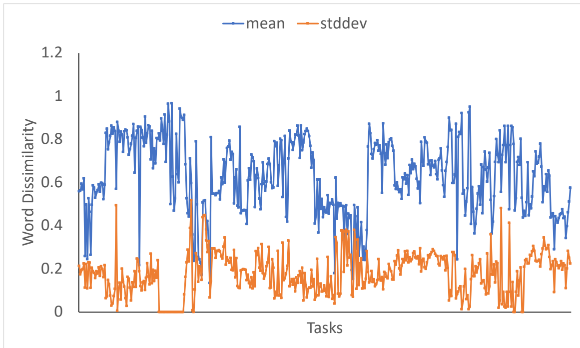


Figure 3: Word-level dissimilarity between original instruction and its variants - Variant instructions have high diversity amongst them in terms of words used to define them (represented through high mean) for most of the tasks in our dataset.

3.3.1 Semantic Textual Similarity

It is necessary to understand the semantic similarity between the original instruction and its variants. Semantic similarity should be high between original instruction and augmented instructions as they represent the same task. To achieve this, we compute the pair-wise Semantic Text Similarity (STS) score between definitions of original instruction and variant instructions since we have only augmented the definition. 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. Figure 2 shows the mean and

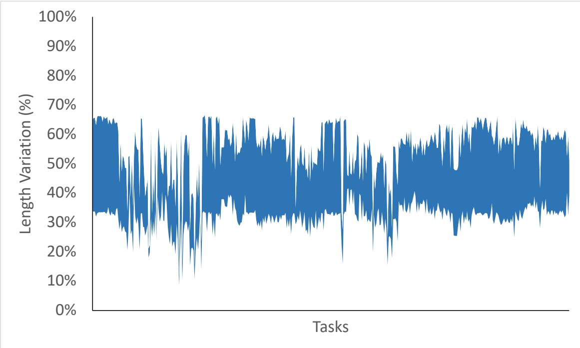


Figure 4: Definition length variation between original instruction and its variants - Variants instructions differ in task definition length (represented through high % variation between # of tokens) across all the tasks in our dataset.

SD of STS score between original instruction and its variants across 426 tasks.

3.3.2 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 3.

3.3.3 Length Diversity

It is necessary to see how task definition lengths varies between original instructions and its 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 4.

Analysis of dataset properties From all dataset properties, we can observe that STS score is higher for almost all the tasks. This indicate that all augmented variants are semantically similar with original instruction. Moreover, we can see a significant variation in terms of word dissimilarity and length of definitions. From this, we can conclude that the variants created in our meta-dataset for each tasks have sufficient variations in terms of words and length yet sustaining semantic similarity with original instruction.

4 Experimental Setup

We design three different experiments: (1) single-task, (2) multi-task, and (3) cross-task. BART-base model (Lewis et al., 2019) is used with default hyper parameters from huggingface to perform experiments. All experiments are performed using V100 and GTX-1080 GPUs. We use single instruction learning as baseline where only original instruction is used to fine-tune the model. Even though the variant instruction tasks often contain different instances than the instances in the original instruction task, we use the exact same instances for both original and variant instruction task to accurately the gauge the importance of additional instruction.[6]

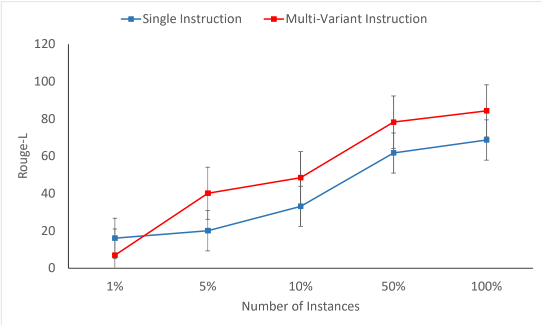
Single-Task In this experiment, we fine-tune baseline and our model on one task and evaluate on the same task. Experiments are performed by using 1%, 5%, 10%, 50% and 100% instances from the task for fine-tuning. Here, we divide instances into train, test and dev splits by randomly sampling in the ratio 70%, 20% and 10% respectively. We have performed single-task learning on 3 different tasks - winogrande_answer_generation, winogrande_question_modification_person and qasc_answer_generation.

Multi-Task To perform multi-task learning, we use 8 different tasks (see Table 4). We have selected these 8 tasks spanning across 4 different categories: (1) question generation, (2) answer generation, (3) text modification and (4) classification. In this setting, we fine-tune baseline and our model on all 8 tasks combined and evaluate on each task. We use similar setting for experiments suggested by (Mishra et al., 2021b). However, we use only two positive and two negative examples to satisfy the maximum token limit of BART-base. Here, we divide instances into train, test and dev splits by randomly sampling in the ratio 70%, 20% and 10% respectively. We perform fine-tuning experiments by random sampling of 1%, 5%, 10%, 50% and 100% instances from each of the 8 tasks.

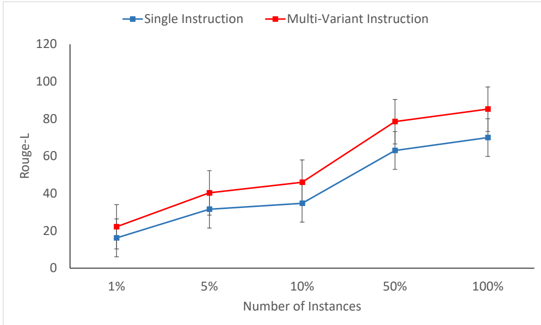
Cross-Task In this setting, we fine-tune the BART on a set of tasks and evaluate on different set of tasks. Here, we use 274 different tasks [7] for training by sampling 10% instances from each task and evaluate on a set of 8 tasks which are same as in

[6] Utilizing additional instances in variant instruction tasks will be part of our future work.

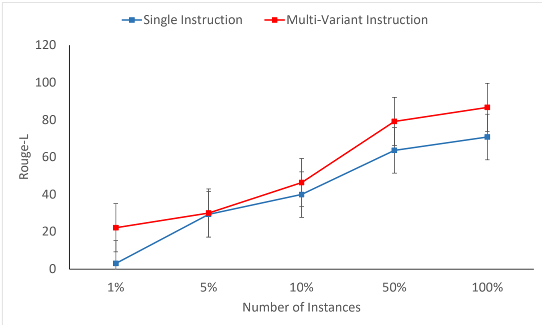
[7] Rest of the tasks are non-english tasks which will be part of our future work



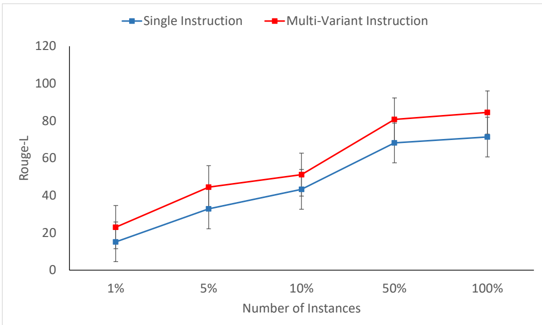
(a) fixing number of tasks to 1%



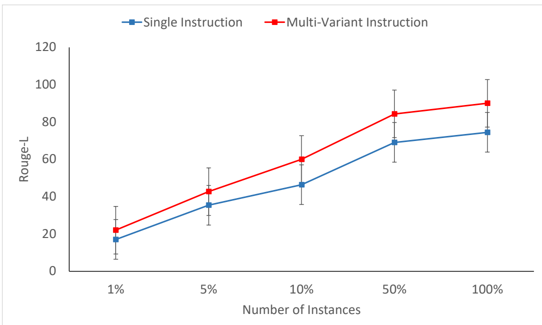
(b) fixing number of tasks to 5%



(c) fixing number of tasks to 10%



(d) fixing number of tasks to 50%



(e) fixing number of tasks to 100%

Figure 5: Comparison of rouge-L 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.

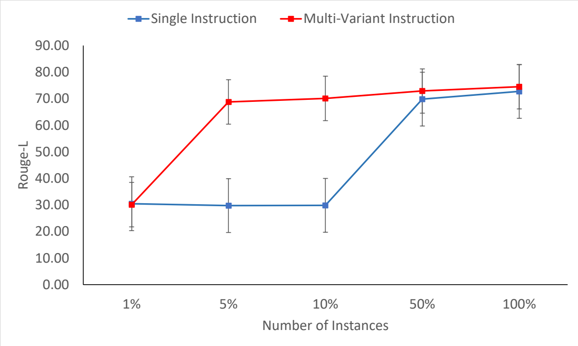


Figure 6: Comparison of Rouge-L across SI and MVI learning in single-task setting; this is done by varying number of instances averaged over 3 tasks. Evaluation is performed on the test set of original instructions.

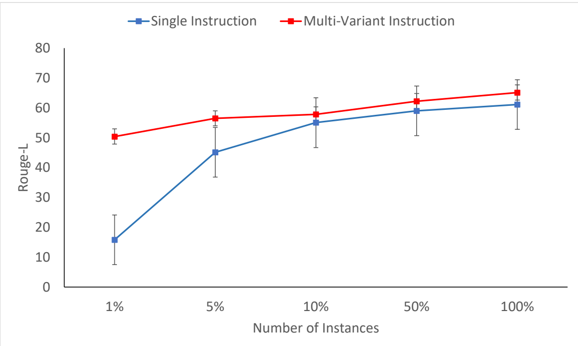


Figure 7: Comparison of rouge-L across SI and MVI learning in multi-task setting by varying number of instances. Evaluation is performed on the test set of original instructions.

the multi-task setup. We use a similar approach like the previous section but instead sampling instances, we 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 regime; we do this by randomly sampling 1%, 5%, 10% instances for fine-tuning.

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

5 Results and Analysis

5.1 Single-Task Results

We present results averaged over the three tasks. Figure 6 shows the comparison between SI and MVI across different number of instances sampled for fine-tuning. From this, we can observe that MVI outperforms SI by 17% on an average. The performance difference between MVI and SI increases to 26% in low data regime (average performance with 1%, 5% and 10% instances for fine-tuning). Appendix C contains more details.

5.2 Multi-Task Results

Figure 7 presents the comparison of rouge-L between SI and MVI learning. We can observe that MVI outperforms SI by 11% margin. Moreover, we can see higher improvement in low data regime (16%). In particular, our model achieves high performance boost (~35% gain) at 1% instances setting. Appendix D contains more details.

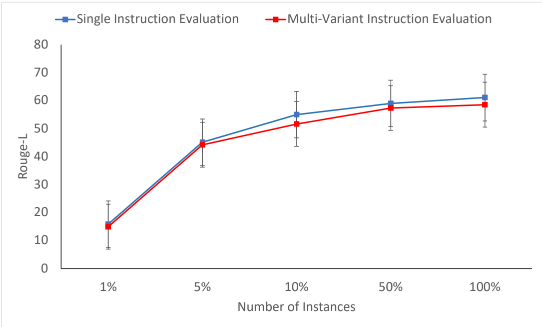
5.3 Cross-Task Setting

Figure 5 shows the rouge-L comparison between SI and MVI. We can observe that MVI outperforms SI by 9% margin. Moreover, we can see higher improvement in low resource settings. In particular, our model achieves high performance boost (~15% gain) at 1% instances setting. Appendix E contains more details.

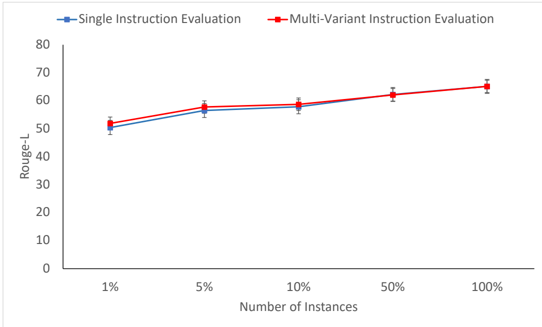
5.4 Analysis

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.

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 in MVI with 5% instances. We interpolate model performance plot in SI to find out the percentage of instances needed to match the model 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 the worth of an additional instruction in terms of data samples. On an average across TS, MT and CT, we conclude that an additional variant instruction alone is worth ~200 instances (data samples).



(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.

6 Conclusion

We introduced instruction augmentation in order 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 significantly improves model performance in single-task and multi-task and cross-task learning paradigms. We also find that instruction augmentation is more effective in low-date regime (up to 35% improvement in model performance). Our results further indicate that an additional instruction can be equivalent to ~200 data samples on an average. We hope our work will bring more attention to developing unconventional techniques (beyond dataset creation and model training) in order to empower non-expert users leverage NLP resources and teach a task without having domain knowledge.

Acknowledgements

We thank CSE 576 NLP class students at Arizona State University for helping with the data creation process.

References

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

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8082–8090.

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.

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*.

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 Sachdeva, Chris Bryan, and Chitta Baral. 2020. Dqi: A guide to benchmark evaluation. *arXiv preprint arXiv:2008.03964*.

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*.

Amita Misra, Brian Ecker, and Marilyn A Walker. 2017. Measuring the similarity of sentential arguments in dialog. *arXiv preprint arXiv:1709.01887*.

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*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

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*.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293.

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. 2022. Instructionner: A multi-task instruction-based generative framework for few-shot ner. *arXiv preprint arXiv:2203.03903*.

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*.

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*.

A Example of Variants

Table 3 shows the examples of different variants created from the task018_qasc_answer_generation.

B Number of variants per task

Here, we have shown the number of variant instructions for 8 different tasks used for performing various multi-task and cross-task experiments. Table 4 shows the different number of variant instructions for 8 tasks and their categories.

C Single-Task Results

Table 5 shows the results for single-task experiments for task010_winogrande_answer_generation, task012_winogrande_question_modification_person and task018_qasc_answer_generation.. From the average results, we can observe that multi-variant instruction learning helps model to improve performance in single-task learning.

D Multi-Task Results

The results for multi-task learning experiments are shown in Table 6.

E Cross-Task Results

The results for cross-task learning experiments are shown in Table 7.

	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 3: Example of an instruction for an answer generation task with its variant instructions - task018_qasc_answer_generation

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 4: Number of variant instructions for 8 different tasks

# of Instances	SI		MVI	
	Original	Ours	Original	Ours
		task_010		
1%	0.00	0.00	0.00	0.02
5%	0.00	36.75	0.06	37.07
10%	0.23	39.17	0.15	38.26
50%	37.00	43.02	25.40	42.54
100%	41.97	45.65	33.84	45.50
		task_012		
1%	84.48	83.54	75.45	82.66
5%	84.73	90.68	74.52	90.68
10%	84.81	90.61	75.47	90.60
50%	90.29	90.49	85.65	90.48
100%	90.84	90.50	88.47	90.52
		task_018		
1%	7.05	6.92	4.36	5.27
5%	4.65	79.07	3.42	79.55
10%	4.72	80.59	3.68	80.95
50%	82.43	85.23	81.36	85.20
100%	85.58	87.37	84.90	87.52
		Average		
1%	30.51	30.15	26.60	29.32
5%	29.79	68.83	26.00	69.10
10%	29.92	70.12	26.43	69.94
50%	69.91	72.91	64.14	72.74
100%	72.80	74.51	69.07	74.51

Table 5: Comparison of LM Rouge-L performance in single-task setting across single-instruction and multi-variant instruction learning. SI: Single-Instruction, MVI: Multi-Variant Instruction.

# of Instances	SI		MVI	
	Original	Ours	Original	Ours
1%	15.84	50.40	14.97	51.88
5%	45.13	56.49	44.24	57.71
10%	55.03	57.80	51.67	58.70
50%	59.01	62.21	57.37	62.06
100%	61.08	65.13	58.58	65.09

Table 6: Comparison of LM Rouge-L performance in multi-task setting across single-instruction and multi-variant instruction learning. SI: Single-Instruction, MVI: Multi-Variant Instruction.

# of Instances	SI		MVI	
	Original	Ours	Original	Ours
		1% tasks		
1%	16.00	6.94	10.93	10.16
5%	20.04	40.14	19.51	31.09
10%	33.09	48.43	31.83	47.66
50%	61.70	78.22	58.53	78.43
100%	68.66	84.22	64.39	84.87
		5% tasks		
1%	16.23	22.17	3.32	18.78
5%	31.58	40.3	29.81	33.12
10%	34.73	46.02	34.38	49.15
50%	63.06	78.48	60.5	79.76
100%	69.93	85.2	67.41	86.68
		10% tasks		
1%	2.98	22.16	2.46	19.98
5%	29.27	30.06	28.03	30.9
10%	39.95	46.38	36.3	50.4
50%	63.58	79.13	59.98	79.81
100%	70.82	86.66	69.11	87.86
		50% tasks		
1%	15.18	23.06	17.08	26.2
5%	32.88	44.5	33.88	44.64
10%	43.33	51.2	42.5	54.62
50%	68.18	80.8	66.42	81.29
100%	71.35	84.52	68.85	84.65
		100% tasks		
1%	17.04	22	19.2	24.95
5%	35.4	42.68	36.42	45.06
10%	46.4	60	45.33	59.3
50%	69.06	84.32	67.29	84.47
100%	74.45	90.01	72.26	90.35

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