Fine-tuning LLMs on LLM-generated Data

Yury Svirschevsky

HSE University

NATIONAL RESEARCH UNIVERSITY

March 12, 2024

Problem

- Vanilla LLMs are trained to predict the next word, but they can struggle with following instructions.
- To solve this problem, we want to fine-tune LLMs on datasets of instructions and responses (instruction-tuning).
- However, collecting human-generated instruction datasets is costly and time-consuming. Available data suffers from limited quantity, diversity and creativity.

Solution (General Strategy)

- **Generation.** LLM generates training data.
- **Filtration.** LLM filters the generated data.
- **Fine-tuning.** LLM is fine-tuned on the filtered training data.

SELF-INSTRUCT: Aligning Language Models with Self-Generated Instructions

Yizhong Wang[♣] **Yeganeh Kordi**[◇] **Swaroop Mishra**[♡] **Alisa Liu**[♣]
Noah A. Smith[♣+] **Daniel Khashabi**[♠] **Hannaneh Hajishirzi**[♣+]
[♣]University of Washington [◇]Tehran Polytechnic [♡]Arizona State University
[♠]Johns Hopkins University [+]Allen Institute for AI
yizhongw@cs.washington.edu

Published December 2022
828 citations

Self-Instruct Method I

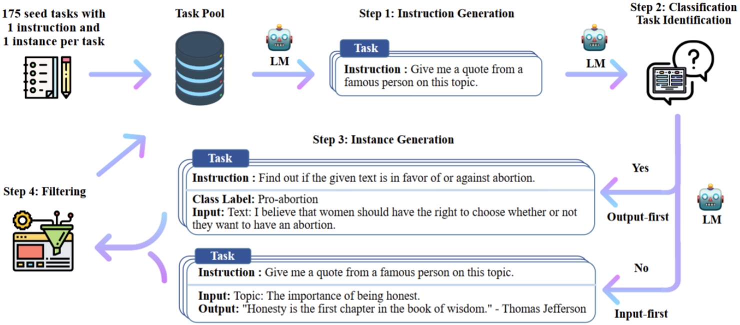


Figure 2: A high-level overview of SELF-INSTRUCT. The process starts with a small seed set of tasks as the task pool. Random tasks are sampled from the task pool, and used to prompt an off-the-shelf LM to generate both new instructions and corresponding instances, followed by filtering low-quality or similar generations, and then added back to the initial repository of tasks. The resulting data can be used for the instruction tuning of the language model itself later to follow instructions better. Tasks shown in the figure are generated by GPT3.

Self-Instruct Method II

- Initialize task pool with a small set of human-written instructions.
- For every step, sample 6 human-written and 2 model-generated instructions as in-context examples.
- Instances for each instruction are generated independently.
- An instruction is only added to the pool if its ROUGE-L similarity with any existing instruction is less than 0.7.
- Invalid input-output instances are filtered heuristically.

Self-Instruct. Diversity and Quality of Generated Instructions

Figure 3: The top 20 most common root verbs (inner circle) and their top 4 direct noun objects (outer circle) in the generated instructions. Despite their diversity, the instructions shown here only account for 14% of all the generated instructions because many instructions (e.g., "Classify whether the user is satisfied with the service.") do not contain such a verb-noun structure.

Quality Review Question	Yes %
Does the instruction describe a valid task?	92%
Is the input appropriate for the instruction?	79%
Is the output a correct and acceptable response to the instruction and input?	58%
All fields are valid	54%

Table 2: Data quality review for the instruction, input, and output of the generated data. See Table 10 and Table 11 for representative valid and invalid examples.

Self-Instruct Evaluation I

Model	# Params	ROUGE-L
Vanilla LMs		
T5-LM	11B	25.7
GPT3	175B	6.8
Instruction-tuned w/o SuperNI		
T0	11B	33.1
GPT3 + T0 Training	175B	37.9
$\text{GPT3}_{\text{SELF-INST}}$ (Ours)	175B	39.9
InstructGPT_{001}	175B	**40.8**
Instruction-tuned w/ SuperNI		
Tk-Instruct	11B	46.0
GPT3 + SuperNI Training	175B	49.5
$\text{GPT3}_{\text{SELF-INST}}$ + SuperNI Training (Ours)	175B	**51.6**

Table 3: Evaluation results on *unseen* tasks from SuperNI (§4.3). From the results, we see that 1 Self-Instruct can boost GPT3 performance by a large margin (+33.1%) and 2 nearly matches the performance of InstructGPT_{001}. Additionally, 3 it can further improve the performance even when a large amount of labeled instruction data is present.

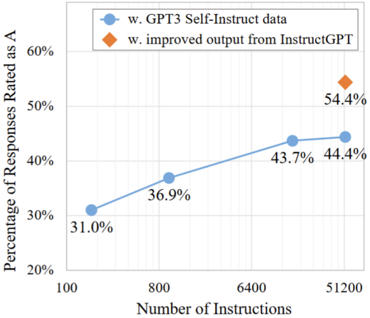


Figure 7: Human evaluation performance of $\text{GPT3}_{\text{SELF-INST}}$ models tuned with different sizes of instructions. x-axis is in log scale. The smallest size is 175, where only the seed tasks are used for instruction tuning. We also evaluate whether improving the data quality will further improve the performance by distilling the outputs from InstructGPT_{003}. We see consistent improvement from using larger data with better quality.

Self-Instruct Evaluation II

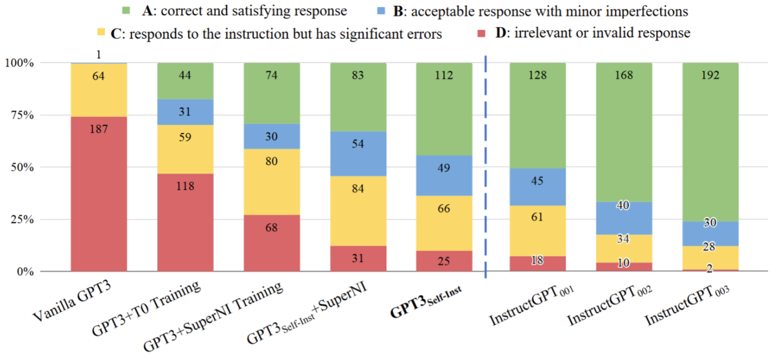


Figure 6: Performance of GPT3 model and its instruction-tuned variants, evaluated by human experts on our 252 user-oriented instructions (§4.4). Human evaluators are instructed to rate the models' responses into four levels. The results indicate that $\text{GPT3}_{\text{SELF-INST}}$ outperforms all the other GPT3 variants trained on publicly available instruction datasets. Additionally, $\text{GPT3}_{\text{SELF-INST}}$ scores nearly as good as InstructGPT_{001} (cf. footnote 1).

Alpaca: A Strong, Replicable Instruction-Following Model

Authors: Rohan Taori* and Ishaan Gulrajani* and Tianyi Zhang* and Yann Dubois* and Xuechen Li* and Carlos Guestrin and Percy Liang and Tatsunori B. Hashimoto

Released March 2023

INSTRUCTION TUNING WITH GPT-4

Baolin Peng*, Chunyuan Li*, Pengcheng He*, Michel Galley, Jianfeng Gao
Microsoft Research
{bapeng,chunyl,penhe,mgalley,jfgao}@microsoft.com

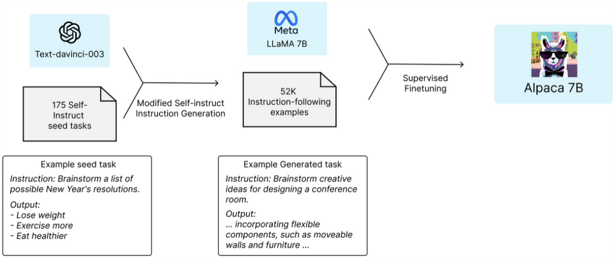
Published April 2023
384 citations

Instruction Tuning with GPT

- Alpaca is a LLaMA 7B Model fine-tuned on 52K instruction-following demonstrations generated by GPT-3.5 (text-davinci-003).
- LLaMA-GPT4 is fine-tuned by GPT-4 and gives better results.
- Fine-tuning models on examples generated by other models is a form of distillation.

Alpaca Model

The figure below illustrates how we obtained the Alpaca model. For the data, we generated instruction-following demonstrations by building upon the self-instruct method. We started with the 175 human-written instruction-output pairs from the self-instruct seed set. We then prompted text-davinci-003 to generate more instructions using the seed set as in-context examples. We improved over the self-instruct method by simplifying the generation pipeline (see details in GitHub) and significantly reduced the cost. Our data generation process results in 52K unique instructions and the corresponding outputs, which costed less than $500 using the OpenAI API.



Evaluation

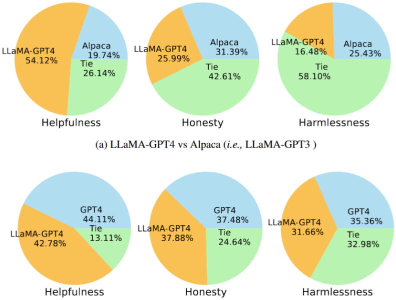


Figure: Helpfulness, Honesty and Harmlessness evaluated on 252 user-oriented questions

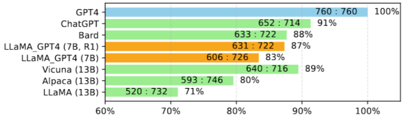


Figure: Chatbots against GPT-4. Evaluated by GPT-4 on 80 questions.

Self-Alignment with Instruction Backtranslation

Xian Li **Ping Yu** **Chunting Zhou** **Timo Schick**
Luke Zettlemoyer **Omer Levy** **Jason Weston** **Mike Lewis**

Meta AI

Published August 2023
59 citations

Instruction Backtranslation I

- **Self-augmentation.** Using a backward model fine-tuned on (output, instruction) seed pairs, generate instructions for unlabelled data, i.e. the web corpus, to produce candidate training data of (instruction, output) pairs for instruction tuning.
- **Self-curation** Self-select high quality demonstration examples as training data to finetune the base model to follow instructions. This approach is done iteratively where a better intermediate instruction-following model can improve on selecting data for finetuning in the next iteration.

Instruction Backtranslation II

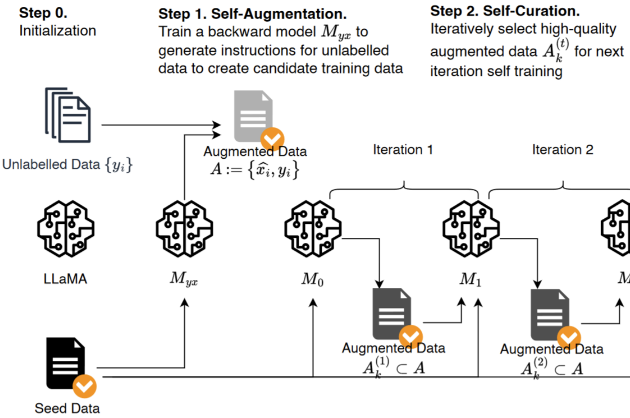


Figure 1: An overview of our **instruction backtranslation** method. We start from a base language model, e.g. LLaMa, a small amount of seed examples of (instruction, output) pairs, and a collection of unlabelled documents which are considered candidate outputs for unknown instructions. **Self-augmentation**: the base model is finetuned with (output, instruction) pairs from the seed examples as an instruction prediction model M_{yx}, which is used to generate candidate instructions for outputs from the unlabelled data. **Self-curation**: starting from an intermediate instruction-following model M_0 finetuned from seed examples only, it selects high-quality (instruction, output) pairs $\mathcal{A}_k^{(1)}$ from the candidates from the previous step, and uses them as finetuning data for the next intermediate model M_1, which is in turn used to select training data for obtaining M_2.

Instruction Backtranslation Evaluation I

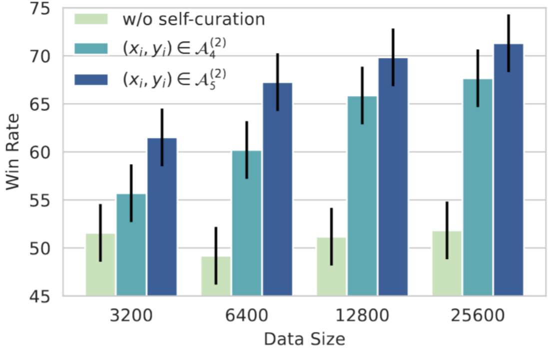


Figure 3: Evaluating self-augmented data of different data size and quality using self-curation. The y-axis is the win rate against text-davinci-003 when finetuning 7B LLaMa with the given data size and quality. We compare three augmentation datasets: without self-curation, $\mathcal{A}_4^{(2)}$ and $\mathcal{A}_5^{(2)}$ that are progressively smaller augmentation sets but of higher data quality (see Table 2 for statistics). Similar to observations in LIMA using human-annotated data [Zhou et al., 2023], improving the quality of the training data dramatically improves the quality of the model, despite the smaller dataset size.

Instruction Backtranslation Evaluation II

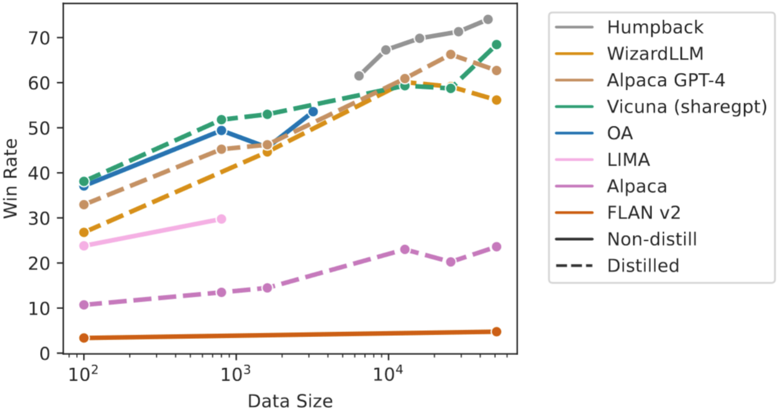


Figure 4: Comparing data efficiency of different instruction tuning datasets. The y-axis is the win rate against text-davinci-003 when finetuning 7B LLaMa with the given instruction tuning dataset. Dashed lines depict models that use distillation from more powerful models to construct data, and methods with solid lines do not.

Questions?

Thank you for listening!