

Self-Instruct: Framework for Creating Self-Generated Instruction Datasets to Align Language Models

Ignacio Talavera Cepeda
Euskal Herriko Unibertsitatea
italavera002@ikasle.ehu.eus

Abstract

Large Language Models (LLMs) exhibit remarkable zero-shot generalization abilities, largely due to their instruction-tuned alignment, which typically relies on human-generated instruction data. However, obtaining such data is challenging and expensive. To overcome this limitation, researchers have introduced SELF-INSTRUCT, a framework aimed at enhancing the instruction-following capabilities of pre-trained LLMs by generating instruction datasets from a small initial set of human-created seeds. SELF-INSTRUCT operates through a semi-automated iterative bootstrapping algorithm, augmenting the seed set with instructions and completions generated by the LLM. The framework incorporates heuristics to filter out low-quality or repeated instructions iteratively. Evaluations of SELF-INSTRUCT demonstrate its effectiveness in producing a diverse range of creative tasks, diverging from conventional NLP task distributions. Fine-tuning a version of GPT3 with SELF-INSTRUCT significantly improves performance across NLP tasks, surpassing the original model by a significant margin. Human evaluation confirms the broadened instruction-following capabilities of the enhanced LLM. Additionally, we have created an instruction dataset based on the medical field using SELF-INSTRUCT and the LLM Phi-2 that is ready to be used for aligning a LLM to better answer instructions related to diseases, medicament and drugs.

1 Introduction

Large Language Models have a remarkable ability to generalize to new tasks in a zero-shot fashion, mostly thanks to their instruction-tuned alignment. These instructions, however, depend heavily on humans, and obtaining this kind of data is difficult and costly. To address this limitation, several researchers from different universities and the Allen Institute for AI introduced SELF-INSTRUCT

(Wang et al., 2022a), a framework for improving the instruction-following capabilities of pre-trained language models by creating instructions datasets from a relatively small set of seeds, which have to be made by humans.

Therefore, SELF-INSTRUCT is a semi-automated process for aligning a pre-trained LLM using instructional signals that are generated in an iterative bootstrapping algorithm, that starts with a limited, manually-written set of seeds. The framework also creates completions to these instructions. Several heuristics are also included to filter low-quality or repeated instructions in the final step of each of the iterations. The process is repeated for several iterations until a desired set of instructions and completions is reached. This process can be seen in Figure 1.

The evaluations that the authors made on SELF-INSTRUCT showed that the obtained data provided a diverse range of creative tasks, deviating from the distribution of typical NLP tasks. They fine-tuned a GPT3 version with SELF-INSTRUCT, and evaluated this version with other LLMs in NLP tasks included in SUPERNI (Wang et al., 2022b) and a set of new instructions. The GPT3 version outperformed the original by a large margin, and after human evaluation, it showed a broad range of instructing-following abilities.

2 Related Work

Reinforcement Learning from Human Feedback

The current state of the art of aligning LLMs with human-written instructions and preferences builds on previous techniques to align models for human use, particularly reinforcement learning from human feedback (RLHF). It was originally developed for training robots in simulated environments and video games (Christiano et al., 2023; Ibarz et al., 2018). Still, it was applied to the fine-tuning phase of LLMs with positive results in several NLP tasks (Jaques et al., 2019; Kreutzer et al., 2018; Zhou and

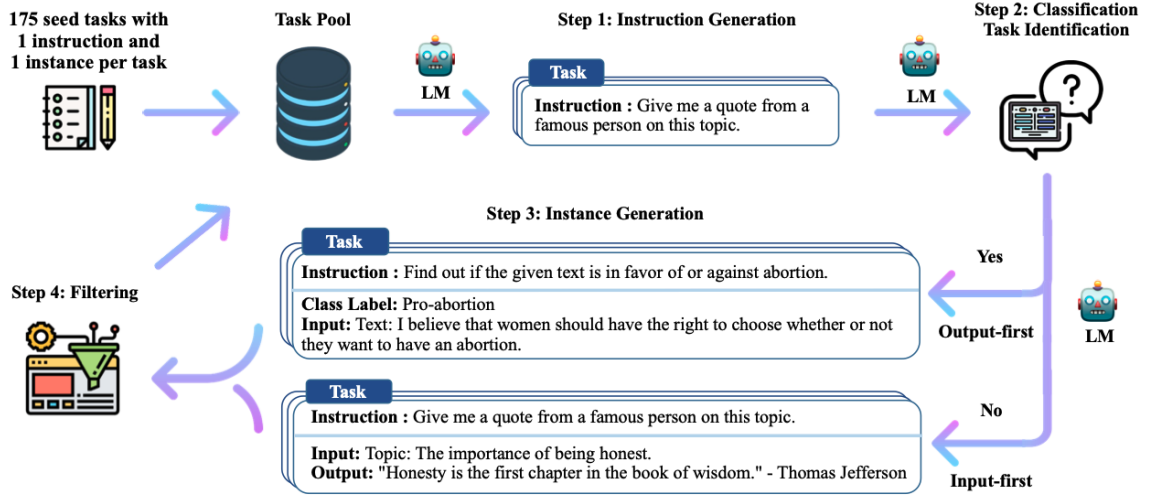


Figure 1: Overview of the SELF-INSTRUCT process. It starts with a set of seeds and the Task Pool. Random tasks are sampled from the pool and used to prompt an LM to generate instructions and completions to those instructions. After filtering repeated and low-quality instructions, these instructions are added to the original pool. This process is iterative.

Xu, 2020; Cho et al., 2019; Madaan et al., 2023). Lately, the question of aligning a model and what a misaligned model can produce, with special focus on potentially harmful content, has also been discussed. Currently, pre-training an LLM with a high number of parameters is not enough, as the NLP community also focuses on this alignment to avoid misuse, either intentional or unintentional, of LLMs.

Training language models to follow instructions

SELF-INSTRUCT capitalises on the interest in human-written instructions datasets, as it has been found to be effective in cross-task generalisation of LLMs. A consistent finding across studies is that fine-tuning LLMs with instruction datasets to perform well on a variety of NLP tasks improves their downstream performance in zero-shot and few-shot learning (Ouyang et al., 2022).

Cost and use of human-written instructions

The NLP community seeks models that are helpful, honest and harmless, and the combination of RLHF and instruction datasets has been proven to be an effective way to achieve so (Askell et al., 2021). However, for the initial work that trained language models to follow instructions with human feedback by OpenAI (Ouyang et al., 2022), a team of 40 contractors was needed to label the database. Very few organisations can assemble these annotation resources.

3 Self-Instruct Process Overview

Here, the process for SELF-INSTRUCT, for creating instruction datasets by generating tasks with a vanilla pretrained language model, filtering the generated data and conducting instruction tuning is explained.

3.1 Pipeline for data generation

The SELF-INSTRUCT data generation pipeline consists of steps that generate task instructions, determine if the instructions are a classification task or not, generate instances and filter those that do not meet certain quality requirements.

In the first step, SELF-INSTRUCT generates new instructions from a small set of human-written instructions that define a task, called seeds. These seeds are introduced into a task pool, from where a sample of 8 is retrieved. In the original paper, they retrieved 6 human-written seeds and 2 that had been synthetically generated in previous steps. Afterwards, they identify whether the generated instruction represents a classification task or not. It was found that pretrained LLMs can understand the target task in an **Input-first approach**, in which the LLM is asked to come up with the input fields first based on the instruction, and then produce the corresponding output. However, it was found that this approach can generate a bias towards one label, especially in classification tasks. Therefore, if the seed is considered to belong to a classification task,

it follows an **Output-first approach**, in which the possible class label is generated first, and then the input on each class label is obtained.

Once the instances are generated, they undergo a filtering process. To encourage diversity, new instructions are added to the pool of seeds only if their ROUGE-L (Lin, 2004) similarity with any existing instruction is less than 0.7. Other instances that are exactly the same or that have invalid generating are filtered out based on heuristics. This whole pipeline can be seen in Figure 1.

4 Use Case: Generating a Medical Instruction Dataset

For showcasing the SELF-INSTRUCT framework, I have created my own synthetic instruction dataset from a set of seeds. The code for this use case can be found in its GitHub repository¹.

The objective of this practical part was to generate an instruction dataset related to the medical field, containing instructions about illnesses, medical procedures and drugs. As I could not get the help from any expert in the field to create the seeds, I used the medical terms and descriptions from the dataset gamino/wiki_medical_terms² that can be found on the HuggingFace Hub. This dataset contains over 6,000 medical terms and their Wikipedia text. It is intended to be used on a downstream task that requires medical terms and their Wikipedia explanation.

Both the instructions and the responses to those instructions are created using distilabel³, an AI Feedback (AIF) framework for building datasets with and for LLMs. The datasets are then stored and visualized with Argilla⁴, a collaboratory platform that allows to upload and annotate datasets. It can be done locally or, in this case, using a HuggingFace Space. For the inference step, the GPUs available in Google Colab were used.

Firstly, for preprocessing the dataset, I generated a subset of the original dataset made by the medical terms and the first 300 characters of the description related to those terms, to avoid input records being too long and making the inference for the whole process slow. Usually, the most important facts about the term itself

can be found in the first two or three sentences. An example of this can be seen in the Listing below.

Paracetamol poisoning: Paracetamol poisoning, also known as acetaminophen poisoning, is caused by excessive use of the medication paracetamol (acetaminophen). Most people have few or non-specific symptoms in the first 24 hours following overdose. These include feeling tired, abdominal pain, or nausea. This is typically fo

Afterwards, I created a SELF-INSTRUCT task with the following attributes:

- **System Prompt:** “You are an expert prompt writer, writing the best and most diverse prompts for a variety of tasks. You are given a task description and a set of instructions for how to write the prompts for an specific AI application.”
- **Principles:**
 - harmlessness
 - helpfulness
 - truthfulness
 - honesty
- **Application Description:** “A assistant that can answer questions about medications and drugs and what is their use”.
- **Criteria for Query Generation:**
 - Incorporate a diverse range of verbs, avoiding repetition.
 - Ensure queries are compatible with AI model’s text generation functions and are limited to 1-2 sentences.
 - Design queries to be self-contained
 - Blend interrogative and imperative styles.
- **Seeds:** [(Paracetamol poisoning. Also known as acetaminophen poisoning, is caused by excessive use of the medication paracetamol (acetaminophen).), ...]

With these attributes, I asked the LLM to generate two instructions per seed in each step of the iterative process, in a batch size of 4, and one completion per instruction. Given the limitations of the Google Colab environment, I could only perform one step of the iterative process. For this very same reason, the vanilla LLM used to create these instructions and responses was the 2-billion

¹<https://github.com/ignacioct/SelfInstructProjectEHU/>

²https://huggingface.co/datasets/gamino/wiki_medical_terms

³<https://github.com/argilla-io/distilabel>

⁴<https://github.com/argilla-io/argilla>

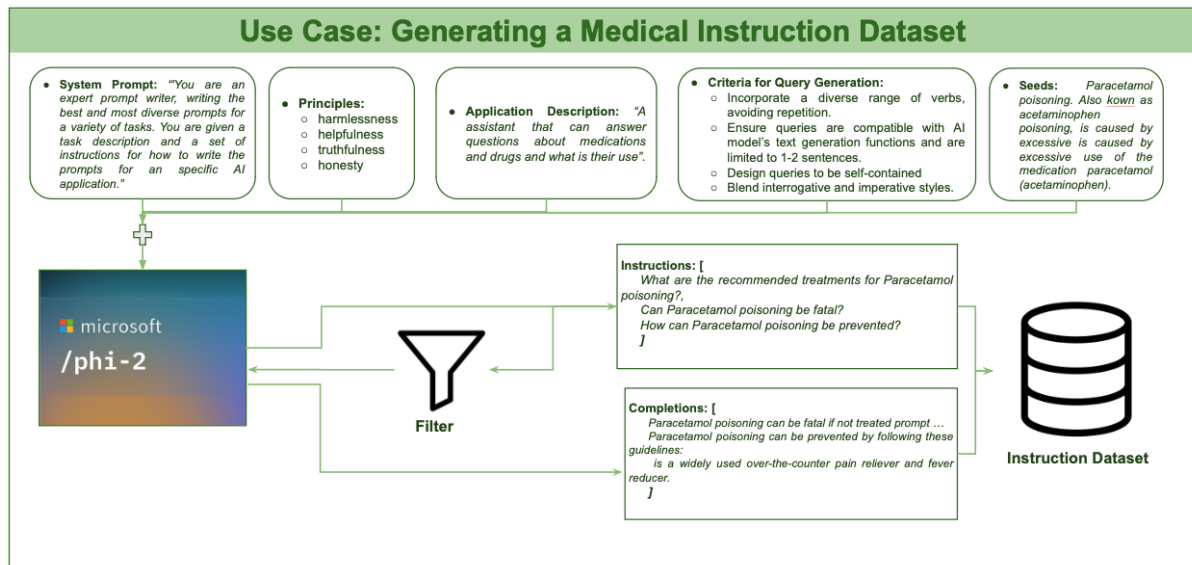


Figure 2: Visual summary of the workflow followed to create the medical instruction-completion dataset.

parameter version of Phi-2⁵, which showcases nearly state-of-the-art performance among models with less than 13 billion parameters and has not been fine-tuned through RLHF. An example of an instruction and its corresponding completion can be seen in the Listing below, and a summary of this process can be seen in Figure 2. After filtering out the instructions that did not meet the quality criteria, the final dataset consisted on 1147 rows.

Instruction: 'What are the common side effects of paracetamol overdose?' Completion: 'Paracetamol overdose can cause serious liver damage and even be fatal. The most common side effects of paracetamol overdose include nausea, vomiting, abdominal pain, and dark urine. Other symptoms may include loss of appetite, fever, and jaundice (yellowing of the skin and eyes). If you suspect that you or someone else has taken too much paracetamol, seek medical attention immediately. It is important to note that the symptoms of paracetamol overdose can be similar to those of other illnesses, so it is always best to consult a healthcare professional if you are unsure.'

After a visual analysis of the obtained results, it can be seen that the model generates correct completions for the most common medical terms, but fails to be precise when the term is more obscure or when it can be mistaken (for example, it mistakes AWS for Amazon Web Services, even though we specified that we are working in the medical environment). However, with no medical experts that can validate this instruction dataset, we cannot be sure that the content has enough quality or safe enough to be used to align an LLM, and neither can

we use another LLM to compare this completions given the limitation in computation resources.

5 Conclusions & Future Work

We have introduced SELF-INSTRUCT, a method to align LLMs with synthetically-generated instruction datasets. The authors of the paper showed state-of-the-art improvements when fine-tuning GPT3 using this framework, while highlighting the cost that obtaining an equivalent dataset labelled by humans would have supposed. SELF-INSTRUCT can serve as the first step to align pretrained LLMs to follow human instructions, and future work can build on top of this data to improve results. As use case, I have created an instruction dataset using SELF-INSTRUCT and the LLM Phi 2 about the medical environment, with instructions about diseases, drugs and medical terms.

Beyond the immediate focus of this project, I believe that further improvements can be done by working on evaluating the obtained datasets in a way that is also cost-efficient and may not require expert labelers, as I did not have resources to make a proper evaluation over the obtained datasets. Other frameworks for automatically generating similar datasets could then be compared thanks to these evaluations techniques.

References

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas

⁵<https://huggingface.co/microsoft/phi-2>

- Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. 2021. [A general language assistant as a laboratory for alignment](#).
- Woon Sang Cho, Pengchuan Zhang, Yizhe Zhang, Xiu-jun Li, Michel Galley, Chris Brockett, Mengdi Wang, and Jianfeng Gao. 2019. [Towards coherent and cohesive long-form text generation](#).
- Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martić, Shane Legg, and Dario Amodei. 2023. [Deep reinforcement learning from human preferences](#).
- Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. 2018. [Reward learning from human preferences and demonstrations in atari](#).
- Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. 2019. [Way off-policy batch deep reinforcement learning of implicit human preferences in dialog](#).
- Julia Kreutzer, Shahram Khadivi, Evgeny Matusov, and Stefan Riezler. 2018. [Can neural machine translation be improved with user feedback?](#)
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2023. [Memory-assisted prompt editing to improve gpt-3 after deployment](#).
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#).
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022a. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.
- Yizhong Wang, Swaroop Mishra, Pegah Alipoor-molabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krma Doshi, Maitreya Patel, Kuntal Kumar Pal, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Noah A. Smith, Hannaneh Hajishirzi, and Daniel Khashabi. 2022b. [Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks](#).
- Wangchunshu Zhou and Ke Xu. 2020. [Learning to compare for better training and evaluation of open domain natural language generation models](#).