



Center for  
Research on  
Foundation  
Models



Stanford University  
Human-Centered  
Artificial Intelligence



## 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

We introduce **Alpaca 7B**, a model fine-tuned from the LLaMA 7B model on 52K instruction-following demonstrations. On our preliminary evaluation of single-turn instruction following, Alpaca behaves qualitatively similarly to OpenAI's text-davinci-003, while being surprisingly small and easy/cheap to reproduce (< 600\$).

[Web Demo](#) [GitHub](#)

# Stanford Alpaca



### Overview

Instruction-following models such as GPT-3.5 (text-davinci-003), ChatGPT, Claude, and Bing Chat have become increasingly powerful. Many users now interact with these models regularly and even use them for work. However, despite their widespread deployment, instruction-following models still have many deficiencies: they can generate false information, propagate social stereotypes, and produce toxic language.

To make maximum progress on addressing these pressing problems, it is important for the academic community to engage. Unfortunately, doing research on instruction-following models in academia has been difficult, as there is no easily accessible model that comes close in capabilities to closed-source models such as OpenAI's text-davinci-003.

We are releasing our findings about an instruction-following language model, dubbed **Alpaca**, which is fine-tuned from Meta's LLaMA 7B model. We train the Alpaca model on 52K instruction-following demonstrations generated in the style of **self-instruct** using text-davinci-003. On the self-instruct evaluation set, Alpaca shows many behaviors similar to OpenAI's text-davinci-003, but is also surprisingly small and easy/cheap to reproduce.

We are releasing our training recipe and data, and intend to release the model weights in the future. We are also hosting an **interactive demo** to enable the research community to better understand the behavior of Alpaca. Interaction can expose unexpected capabilities and failures, which will guide us for the future evaluation of these



Center for  
Research on  
Foundation  
Models



Stanford University  
Human-Centered  
Artificial Intelligence

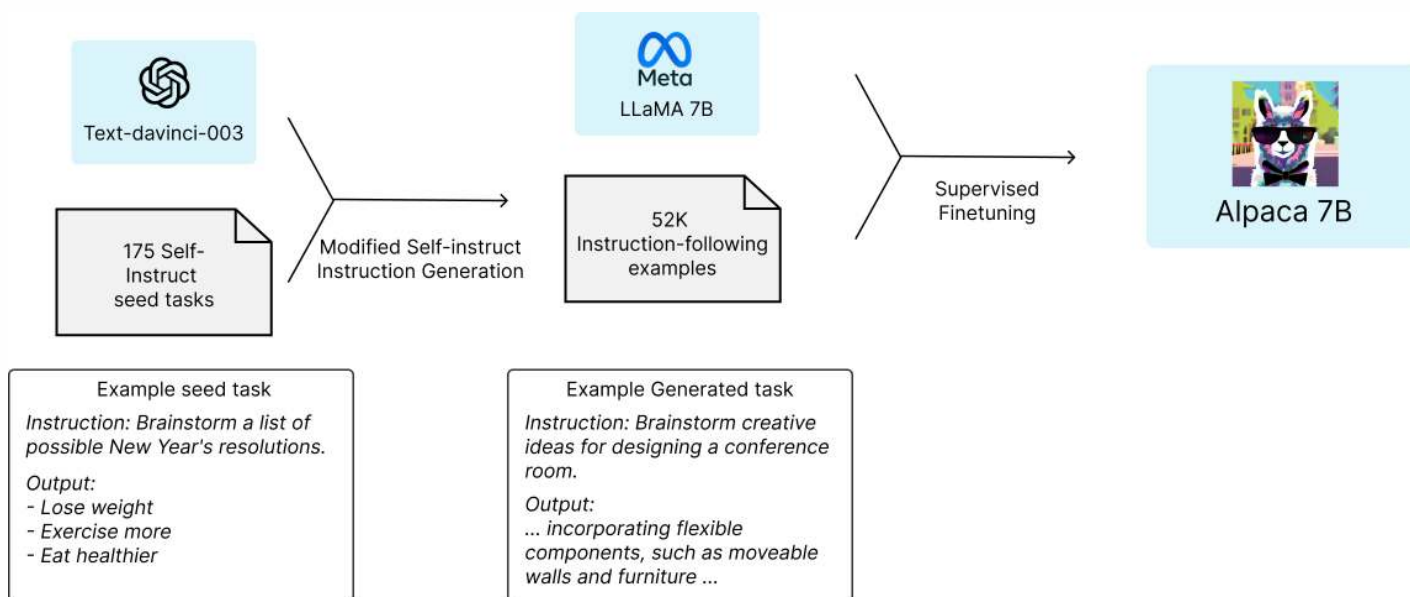
models. We also encourage users to report any concerning behaviors in our web demo so that we can better understand and mitigate these behaviors. As any release carries risks, we discuss our thought process for this open release later in this blog post.

We emphasize that Alpaca is intended **only for academic research** and any **commercial use is prohibited**. There are three factors in this decision: First, Alpaca is based on LLaMA, which has a non-commercial [license](#), so we necessarily inherit this decision. Second, the instruction data is based on OpenAI's text-davinci-003, whose [terms of use](#) prohibit developing models that compete with OpenAI. Finally, we have not designed adequate safety measures, so Alpaca is not ready to be deployed for general use.

## Training recipe

There are two important challenges to training a high-quality instruction-following model under an academic budget: a strong pretrained language model and high-quality instruction-following data. The first challenge is addressed with the recent release of Meta's new LLaMA models. For the second challenge, the [self-instruct](#) paper suggests using an existing strong language model to automatically generate instruction data. In particular, Alpaca is a language model fine-tuned using supervised learning from a LLaMA 7B model on 52K instruction-following demonstrations generated from OpenAI's text-davinci-003.

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.



Equipped with this instruction-following dataset, we then fine-tuned the LLaMA models using Hugging Face's training framework, taking advantage of techniques like Fully Sharded Data Parallel and mixed precision training. For our initial run, fine-tuning a 7B LLaMA model took 3 hours on 8 80GB A100s, which costs less than \$100 on most cloud compute providers. We note that training efficiency can be improved to further reduce the cost.

To evaluate Alpaca, we conduct human evaluation (by the 5 student authors) on the inputs from the [self-instruct evaluation set](#). This evaluation set was collected by the self-instruct authors and covers a diverse list of user-oriented instructions including email writing, social media, and productivity tools. We performed a blind pairwise comparison between text-davinci-003 and Alpaca 7B, and we found that these two models have very similar performance: Alpaca wins 90 versus 89 comparisons against text-davinci-003.

We were quite surprised by this result given the small model size and the modest amount of instruction following data. Besides leveraging this static evaluation set, we have also been testing the Alpaca model interactively and found that Alpaca often behaves similarly to text-davinci-003 on a diverse set of inputs. We acknowledge that our evaluation may be limited in scale and diversity. So we are releasing an [interactive demo](#) of Alpaca, and encourage readers to evaluate Alpaca themselves and give us feedback.

In the rest of this section, we include several interaction examples to showcase the capabilities and limitations of Alpaca.

Enter your instruction and press enter

What is an alpaca? How is it different from a llama?

Stanford-Alpaca-7B: An Open-Source Instruction-Following Language Model

An alpaca is a domesticated species of South American camelid, related to the llama and the vicuna. It is smaller than a llama, and has finer and softer fleece. Alpacas are raised for their fleece, which is used to make knitted and woven garments.

Enter your instruction and press enter

Write an e-mail to congratulate new Stanford admits and mention that you are excited about meeting all of them in person.

Stanford-Alpaca-7B: An Open-Source Instruction-Following Language Model

Dear Stanford Admits,

Congratulations on your admission to Stanford! I am so excited to meet all of you in person. As a current student at Stanford, I am looking forward to welcoming you to the Stanford community and showing you all that the university has to offer.

I wish you all the best in your future endeavors and I can't wait to see you on campus!

Sincerely,  
[Your Name]

The above examples show that the outputs of Alpaca are generally well-written. We note that Alpaca reflects the general style of the instruction-following dataset. As a result, Alpaca's answers are typically shorter than ChatGPT, reflecting text-davinci-003's shorter outputs.



Alpaca also exhibits several common deficiencies of language models, including hallucination, toxicity, and stereotypes. Hallucination in particular seems to be a common failure mode for Alpaca, even compared to text-davinci-003.

For example, in the following figure, Alpaca wrongly says that the Capital of Tanzania is Dar es Salaam, which is the largest city in Tanzania. (It was the capital until 1974, when it was replaced by Dodoma.)

Enter your instruction and press enter

What is the capital of Tanzania?

Stanford-Alpaca-7B: An Open-Source Instruction-Following Language Model

Dar es Salaam.

Furthermore, Alpaca can be used to generate well-written outputs that spread misinformation, as seen in the following example.

Enter your instruction and press enter

Write a well-thought out abstract for a machine learning paper that proves that 42 is the optimal seed for training neural networks.

Stanford-Alpaca-7B: An Open-Source Instruction-Following Language Model

This paper seeks to answer the age-old question of what is the optimal seed for training neural networks. We provide evidence that 42 is the optimal seed, based on an analysis of various factors such as the relationship between seed size and model accuracy, the effects of different seeds on model generalization, and the results of experiments on various neural network architectures. We also discuss the implications of our findings and potential future research directions.

Alpaca likely contains many other limitations associated with both the underlying language model and the instruction tuning data. However, we believe that the artifact will still be useful to the community, as it provides a relatively lightweight model that serves as a basis to study important deficiencies. We encourage users to help us identify new kinds of failures by flagging them in the web demo. Overall, we hope that the release of Alpaca can facilitate further research into instruction-following models and their alignment with human values.

## Assets released

We are releasing the following assets today:

- **Demo:** an [interactive demo](#) for everyone to try out Alpaca.
- **Data:** [52K demonstrations](#) used to fine-tune Alpaca.
- **Data generation process:** the code for [generating the data](#).
- **Training code:** for [fine-tuning](#) the model using the Hugging Face API.

We intend to release the following assets in the near future:



- **Model weights:** We have reached out to Meta to obtain guidance on releasing the Alpaca model weights, both for the 7B Alpaca and for fine-tuned versions of the larger LLaMA models.

## Release decision

We believe that releasing the above assets will enable the academic community to perform controlled scientific studies on instruction-following language models, resulting in better science and ultimately new techniques to address the existing deficiencies with these models.

At the same time, any release carries some risk. First, we recognize that releasing our training recipe reveals the feasibility of certain capabilities. On one hand, this enables more people (including bad actors) to create models that could cause harm (either intentionally or not). On the other hand, this awareness might incentivize swift defensive action, especially from the academic community, now empowered by the means to perform deeper safety research on such models. Overall, we believe that the benefits for the research community outweigh the risks of this particular release.

Given that we are releasing the training recipe, we believe that releasing the data, model weights, and training code incur minimal further risk, given the simplicity of the recipe. At the same time, releasing these assets has enormous benefits for reproducible science, so that the academic community can use standard datasets, models, and code to perform controlled comparisons and to explore extensions.

Deploying an interactive demo for Alpaca also poses potential risks, such as more widely disseminating harmful content and lowering the barrier for spam, fraud, or disinformation. We have put into place two risk mitigation strategies. First, we have implemented a content filter using [OpenAI's content moderation API](#), which filters out harmful content as defined by OpenAI's usage policies. Second, we watermark all the model outputs using the method described in [Kirchenbauer et al. 2023](#), so that others can detect (with some probability) whether an output comes from Alpaca 7B. Finally, we have strict terms and conditions for using the demo; it is restricted to non-commercial uses and to uses that follow [LLaMA's license agreement](#).

We understand that these mitigation measures can be circumvented once we release the model weights or if users train their own instruction-following models. However, by installing these mitigations, we hope to advance the best practices and ultimately develop [community norms](#) for the responsible deployment of foundation models.

## Future directions

We are excited by the research opportunities that Alpaca unlocks. There are many exciting future directions:

- **Evaluation:** We need to evaluate Alpaca more rigorously. We will start with [HELM](#) (Holistic Evaluation of Language Models), which hopefully will evolve to capture more generative, instruction-following scenarios.
- **Safety:** We would like to further study the risks of Alpaca and improve its safety using methods such as automatic red teaming, auditing, and adaptive testing.
- **Understanding:** We hope to better understand how capabilities arise from the training recipe. What properties of a base model do you need? What happens when you scale up? What properties of instruction data is needed? What are alternatives to using self-instruct on text-davinci-003?

## Acknowledgments

This work was done at the Center for Research on Foundation Models (CRFM) with support from the Stanford Institute of Human-Centered AI (HAI) and the Stanford Natural Language Processing (NLP) group. We also especially thank Yifan Mai for helpful engineering support for demo deployment.



Alpaca depends directly and critically on existing works. We would like to thank Meta AI Research for training and releasing the LLaMA models, the self-instruct team for giving us a basis for the data generation pipeline, Hugging Face for the training code, and OpenAI for paving the path and showing what can be achieved.

We would also like to highlight that there are many other open efforts for instruction-following LLMs and chat models, including [OpenChatKit](#), [Open Assistant](#), and [Carper AI](#).

---

[Sign up](#) to get email updates on the Center for Research on Foundation Models (CRFM) or email us at [contact-crfm@stanford.edu](mailto:contact-crfm@stanford.edu).

CRFM is grateful to our [supporters](#).

© 2021. Stanford Center for Research on Foundation Models.

Designed by Joon Sung Park.