

GPT4All: Training an Assistant-style Chatbot with Large Scale Data Distillation from GPT-3.5-Turbo

Yuvanesh Anand
yuvanesh@nomic.ai

Zach Nussbaum
zanussbaum@gmail.com

Brandon Duderstadt
brandon@nomic.ai

Benjamin Schmidt
ben@nomic.ai

Andriy Mulyar
andriy@nomic.ai

Abstract

This preliminary technical report describes the development of GPT4All, a chatbot trained over a massive curated corpus of assistant interactions including word problems, story descriptions, multi-turn dialogue, and code. We openly release the collected data, data curation procedure, training code, and final model weights to promote open research and reproducibility. Additionally, we release quantized 4-bit versions of the model allowing virtually anyone to run the model on CPU.

1 Data Collection and Curation

We collected roughly one million prompt-response pairs using the GPT-3.5-Turbo OpenAI API between March 20, 2023 and March 26th, 2023. To do this, we first gathered a diverse sample of questions/prompts by leveraging three publicly available datasets:

- The unified_chip2 subset of [LAION OIG](#).
- Coding questions with a random sub-sample of [Stackoverflow Questions](#)
- Instruction-tuning with a sub-sample of [Bigscience/P3](#)

We chose to dedicate substantial attention to data preparation and curation based on commentary in the [Stanford Alpaca project](#) (Taori et al., 2023). Upon collection of the initial dataset of prompt-generation pairs, we loaded data into [Atlas](#) for data curation and cleaning. With Atlas, we removed all examples where GPT-3.5-Turbo failed to respond to prompts and produced malformed output. This reduced our total number of examples to 806,199 high-quality prompt-generation pairs. Next, we decided to remove the entire [Bigscience/P3](#) subset from the final training dataset due to its very

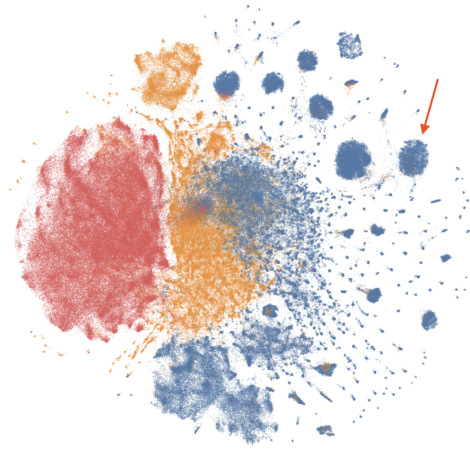


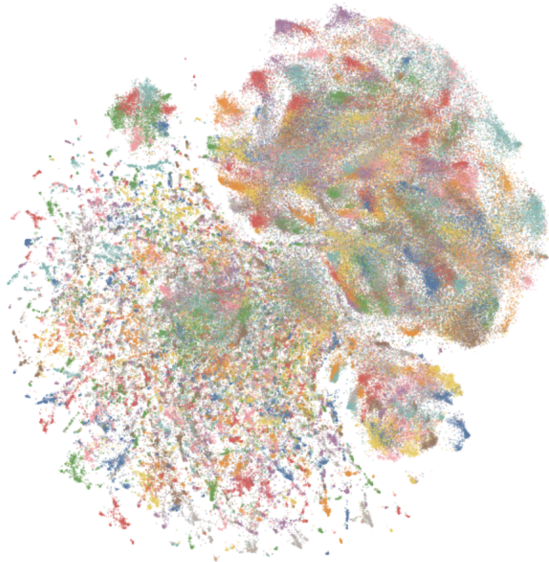
Figure 1: t-SNE ^{PCA} visualization of the candidate training data (Red: Stackoverflow, Orange: chip2, Blue: P3). The large blue balls (e.g. indicated by the red arrow) are highly homogeneous prompt-response pairs.

low output diversity; P3 contains many homogeneous prompts which produce short and homogeneous responses from GPT-3.5-Turbo. This exclusion produces a final subset containing 437,605 prompt-generation pairs, which is visualized in Figure 2. You can interactively explore the dataset at each stage of cleaning at the following links:

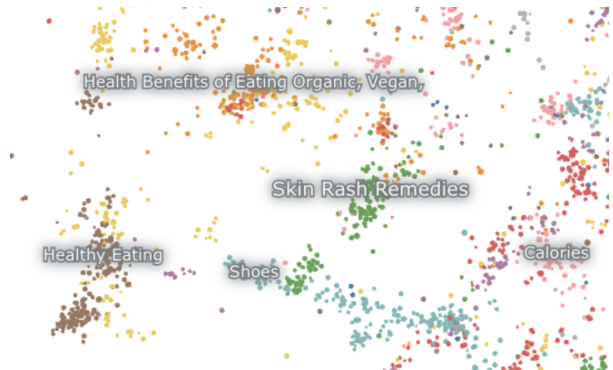
- [Cleaned with P3](#)
- [Cleaned without P3](#) (Final Training Dataset) ✓

2 Model Training

We train several models finetuned from an instance of [LLaMA 7B](#) (Touvron et al., 2023). The model associated with our initial public release is trained with [LoRA](#) (Hu et al., 2021) on the 437,605 post-processed examples for four epochs. Detailed model hyper-parameters and training code can be found in the associated [repository](#) and [model training log](#).



(a) TSNE visualization of the final training data, ten-colored by extracted topic.



(b) Zoomed in view of Figure 2a. The region displayed contains generations related to personal health and wellness.

Figure 2: The final training data was curated to ensure a diverse distribution of prompt topics and model responses.

2.1 Reproducibility

We release all [data](#) (including unused [P3 generations](#)), training code, and model weights for the community to build upon. Please check the [Git repository](#) for the most up-to-date data, training details and checkpoints.

2.2 Costs

We were able to produce these models with about four days work, \$800 in GPU costs (rented from Lambda Labs and Paperspace) including several failed trains, and \$500 in OpenAI API spend. Our released model, `gpt4all-lora`, can be trained in about eight hours on a Lambda Labs DGX A100 8x 80GB for a total cost of \$100.

3 Evaluation

We perform a preliminary evaluation of our model using the [human evaluation data](#) from the Self-Instruct paper (Wang et al., 2022). We report the ground truth perplexity of our model against what is, to our knowledge, the [best openly available alpaca-lora model](#), provided by user chainyo on huggingface. We find that all models have very large perplexities on a small number of tasks, and report perplexities clipped to a maximum of 100.

Models finetuned on this collected dataset exhibit much lower perplexity in the Self-Instruct evaluation compared to Alpaca. This evaluation is in no way exhaustive and further evaluation work

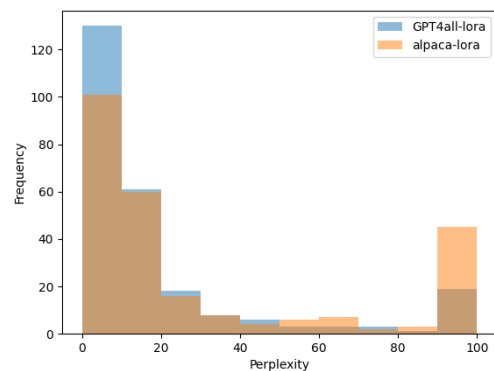


Figure 3: Model Perplexities. Lower is better. Our models achieve stochastically lower ground truth perplexities than alpaca-lora.

remains. We welcome the reader to run the model locally on CPU (see Github for files) and get a qualitative sense of what it can do.

4 Use Considerations

The authors release data and training details in hopes that it will accelerate open LLM research, particularly in the domains of alignment and interpretability. `GPT4All` model weights and data are intended and licensed only for research purposes and any commercial use is prohibited. `GPT4All` is based on LLaMA, which has a non-commercial license. The assistant data is gathered from OpenAI's `GPT-3.5-Turbo`, whose terms of use pro-

hibit developing models that compete commercially with OpenAI.

References

- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. [Lora: Low-rank adaptation of large language models](#).
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [Llama: Open and efficient foundation language models](#).
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hananeh Hajishirzi. 2022. [Self-instruct: Aligning language model with self generated instructions](#).