

★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



How to Custom Train and Fine-Tune Models with the ChatGPT API

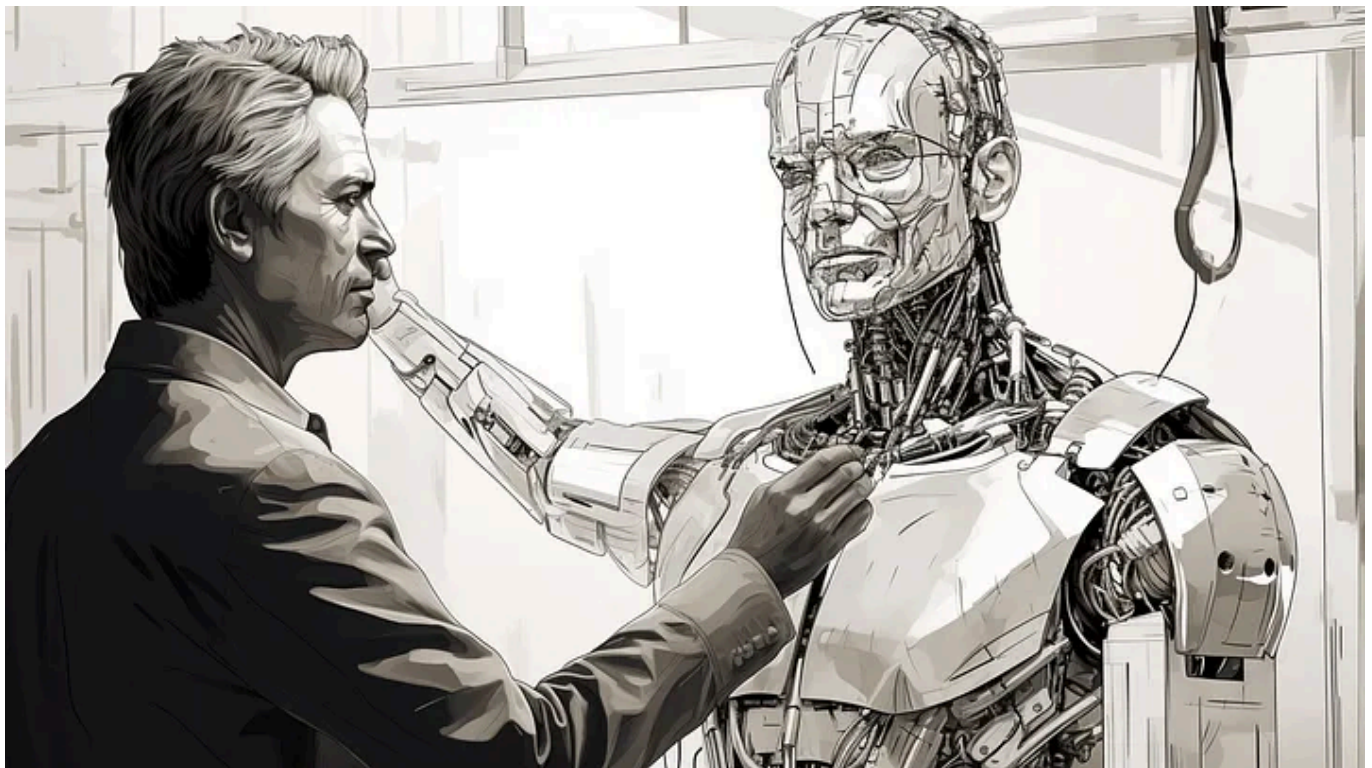


Herman Schutte · [Follow](#)

5 min read · Aug 25, 2023



19



How to Custom Train and Fine-Tune Models with the ChatGPT API

OpenAI made an announcement this week that businesses and developers have the green light to fine-tune GPT-3.5 Turbo with their individual data using the ChatGPT API. This opens up the opportunity for you to craft a custom-trained and fine-tuned GPT model for your company, enhancing its ability to respond to visitor questions accurately and align with your tone.

Fine-Tuning Use Cases

- Improve the model's ability to follow instructions, especially when specific language choices are needed.
- Ensure more reliable output formatting, allowing it to consistently produce JSON that your API can consume.
- Enable a business with a distinct brand voice to make the model's tone more in line with theirs.
- Increase overall performance and allow companies to reduce their prompts without sacrificing effectiveness.

With GPT-3.5-Turbo, fine-tuning can handle up to 4k tokens — double the capacity of previous models. Some early adopters have even cut prompt size by up to 90% by incorporating the fine-tuning instructions directly into the model, speeding up API calls and cutting costs.

For SiteSpeakAI I have been using a combination of vector matching and fine-tuning to produce the most accurate results for SiteSpeakAI's chatbots and virtual assistants.

Creating and using your own basic custom trained ChatGPT model is easy to do though:



Search

Write



M

You'll have to create and supply a dataset to the OpenAI API that's suitable for fine-tuning your model. You'll need at least 10 examples, but you'll probably see better results with more than 50.

Make sure each example in the dataset aligns with the format of our Chat completions API. This means a list of messages, with each one containing a role, content, and maybe a name. Include some training samples that pinpoint situations where the prompted model doesn't respond as you want. The assistant messages in your data should be the ideal answers you hope your model will provide.

Example dataset:

```
{"messages": [{"role": "system", "content": "SiteSpeak AI Assistant is a custome  
{"messages": [{"role": "system", "content": "SiteSpeak AI Assistant is a custome  
{"messages": [{"role": "system", "content": "SiteSpeak AI Assistant is a custome  
{"messages": [{"role": "system", "content": "SiteSpeak AI Assistant is a custome  
{"messages": [{"role": "system", "content": "SiteSpeak AI Assistant is a custome  
{"messages": [{"role": "system", "content": "SiteSpeak AI Assistant is a custome  
{"messages": [{"role": "system", "content": "SiteSpeak AI Assistant is a custome  
{"messages": [{"role": "system", "content": "SiteSpeak AI Assistant is a custome  
{"messages": [{"role": "system", "content": "SiteSpeak AI Assistant is a custome
```

Step 2: Upload your training data

I'll provide a complete script at the end of this post that will allow you to upload and train your model, but if you just need to upload a file:

```
import os
import openai

openai.api_key = os.getenv("OPENAI_API_KEY")

# Upload file
print('Uploading file...')
file = openai.File.create(
    file=open("yourfile.jsonl", "rb"),
    purpose='fine-tune'
)

print('File uploaded: ', file)
```

Step 3: Build Your Custom Fine-Tuned Model

Once you've uploaded your file and seen the status switch from **uploaded** to **processed**, you're ready to kick off your fine-tune training job. Be patient, as training jobs can take some time.

For the short example I provided, the job took about 5 minutes to finish. When the training job is wrapped up, the status will change to **succeeded**. From there, you can grab the **fine_tune_model** from the response and use that model ID for making inferences.

```
import os
import openai

openai.api_key = os.getenv("OPENAI_API_KEY")
```

```
# Create fine-tuning job
print('Creating fine-tuning job...')
job = openai.FineTuningJob.create(
    training_file="file-123abc", model="gpt-3.5-turbo")

print('Job created.')
print(job)
```

Step 4: Use your fine-tuned model

You can now use your newly trained GPT-3.5 Turbo model for inference.

```
import os
import openai
openai.api_key = os.getenv("OPENAI_API_KEY")

completion = openai.ChatCompletion.create(
    model="your-fine-tune-model-id",
    messages=[
        {"role": "system", "content": "SiteSpeak AI Assistant is a customer support assistant."},
        {"role": "user", "content": "What is SiteSpeakAI?"}
    ]
)

print(completion.choices[0].message)
```

The complete python script that uploads a file, waits for it to be processed, creates a training job and outputs the model when it's ready is available here: <https://sitespeak.ai/blog/gpt-3-5-turbo-fine-tuning-custom-model-training>

Need help fine-tuning GPT-3.5 Turbo for your business?

If you don't want to go through the process of setting up your own training data and creating your fine-tuned models, [SiteSpeakAI](#) is a great alternative.

Our chatbots can already be trained on your website content, knowledge base, API's and databases using embeddings and similarity search.

For business plan users we also offer the ability to create a custom fine-tuned GPT model trained on your business data that can exactly match your brand and provide AI powered “human-like” assistance to your visitors and customers.

[Openai Chatgpt](#)[ChatGPT](#)[Ai Chat Bot](#)[Chatgptapi](#)[Customer Support Chatbots](#)

Written by Herman Schutte

125 Followers

za Indie dev, working on [sitespeak.ai](#) and [snapwidget.com](#)

Follow

More from Herman Schutte



Herman Schutte

Enhance Your Business with SiteSpeakAI's Advanced GPT Chatbots

Explore the world of AI-driven customer service with SiteSpeakAI. Our latest blog post highlights the transformative...



1 min read · Jan 11, 2024



See all from Herman Schutte

Recommended from Medium



 Practicing DatScy

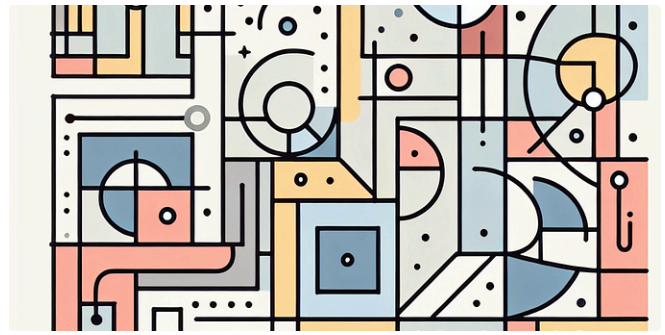
Fine-tuning with OpenAI

I finally was able to test fine-tuning using OpenAI!! Fine-tuning using OpenAI's gpt...

9 min read · Dec 4, 2023



12



 Gareth Cull

Fine Tuning AI Models—A Practical Guide for Beginners

Get started fine tuning OpenAI models such as gpt-3.5-turbo in this easy to understand...

9 min read · Jan 10, 2024



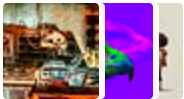
4



1



Lists



What is ChatGPT?

9 stories · 357 saves



The New Chatbots: ChatGPT, Bard, and Beyond

12 stories · 381 saves



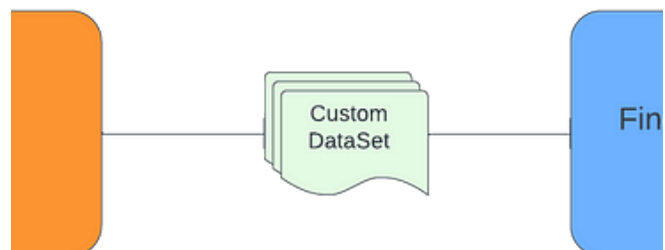
ChatGPT

21 stories · 642 saves



ChatGPT prompts

47 stories · 1578 saves





Yvann in Better Programming

Build a Chatbot on Your CSV Data With LangChain and OpenAI

Chat with your CSV file with a memory chatbot 🤖 — Made with Langchain 🐍...

5 min read · Jun 2, 2023



1.5K



30



Jonathan Yee in Patine Engineering Blog

Prepare JSONL fine-tuning training data in Node.js for OpenAI

A fine-tuned model is essentially a model that undergoes additional training beyond its...

6 min read · Dec 6, 2023



2



Suman Das

Fine Tune Large Language Model (LLM) on a Custom Dataset with...

The field of natural language processing has been revolutionized by large language...

15 min read · Jan 25, 2024



1.2K



15



zkiihne

How to fine-tune an OpenAi model

This article serves as a guide to fine-tuning your Large Language Model (LLM) and...

5 min read · Nov 29, 2023



See more recommendations