

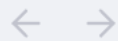
- [About](#)
- [Blog](#)
- [Books](#)
- [Contact me](#)

[Fine-tuning GPT3.5-turbo based on 140k slack messages](#)

September 4, 2023 5-minute read

[AI](#) • [GPT](#) • [LLMs](#)

I recently started using the Smart Connections plugin for the Obsidian note-taking app. It generates embeddings for all of my notes and allows me to chat with them using GPT. Over the weekend I used it to generate Twitter and LinkedIn posts for a blog piece I was working on. While it produced quality content it had a lot of trouble replicating my writing style. The completions didn't feel like something I would write myself.



What's wrong with product management (and how to fix it)

I was attending a company picnic, savoring what might have been a great day. A former coworker approached me. He confided that he had previously worked at a software company and admitted that he initially thought product management was redundant until he joined Observe. Strangely enough, though, he shared with me such sentiment with me. Over the years, I've had similar conversations with people who previously harbored a poor impression of the product management role at other enterprises. This conversation inspired me to re-examine what's broken in product management together this last of what they think is broken in product management.

Poorly Defined Role

Product Management often serves as a connective role between engineering and marketing. Responsibilities can vary greatly depending on the organization. In some organizations, it can be even more pronounced within larger organizations undergoing a "digital transformation," which involves either revamping existing products or initiating software development for the first time.

A common part of this transformation is the rapid establishment of product management roles. Often, individuals who were previously subject matter experts in their respective fields are moved into product management roles. However, these newly appointed individuals often do so without a clear understanding of their new responsibilities.

Moreover, organizational leaders continue to dictate requirements and priorities without a clear understanding of their new responsibilities.

either fail to understand or accept the role of these new product managers are unable to effectively communicate impractical demands. This results in projects, which are (MVPs)," taking two years or more to complete due to the team.

How to fix it:

- Write down a document that clearly outlines the role,



My first attempt to get it to write in my style was to do so with various prompts. Maybe my prompt engineering skills aren't good enough but I couldn't get it to shake the style that GPT uses. When I tried to have it extract style and tone from my previous blog pieces it didn't have enough data to discern between the figure of speech I normally use vs. verbiage that was specific to the piece I was writing. This got me thinking; maybe I could use the new fine-tuning API for GPT 3.5 turbo!

Fine-tuning GPT3.5-turbo based on 140k slack messages

Over the last four years, I've amassed 140k Slack messages at Observe Inc. I downloaded a dump of all of my Slack messages and wrote a script to process it. This created a dump for every direct message or group thread. The data format looks roughly like this:

```
[
  {
    "client_msg_id": "e59a1118-8164-4eb9-9623-e4114502ba30",
    "type": "message",
    "text": "\u003c@UMQN69N8M\u003e - How are you doing?",
    "user": "U01AFPBSPOCKX",
    "ts": "1644956438.392969",
    "blocks": [
      {
        "type": "rich_text",
        "block_id": "euXh",
        "elements": [
          {
            "type": "rich_text_section",
            "elements": [
              {
                "type": "user",
                "user_id": "UMQN78NMB"
              },
              {
                "type": "text",
                "text": " - How are you doing?"
              },
              {
                "type": "link",
                "url": "https://foobar.com"
              }
            ]
          }
        ]
      }
    ]
  },
  {
    "client_msg_id": "6590516f-ff2b-4268-aef2-f7bfadd0a250",
    "type": "message",
    "text": "I'm fine and you?",
    "user": "UMQN78NMB",
    "ts": "1644957762.921829",
    "blocks": [
      {
        "type": "rich_text",
        "block_id": "iXntl",
        "elements": [
          {
            "type": "rich_text_section",
            "elements": [
              {
                "type": "text",
                "text": "I'm fine and you?"
              }
            ]
          }
        ]
      }
    ]
  },
  ....
]
```

There are a few challenges with the data because there's no clear request/response. A user could be responding to the last message or the last series of messages. On the response side of things, every time a user types a message and presses enter, Slack treats it as a separate event. Given that this is a fun side project I opted to go with very simple (and flawed) logic to handle these cases. I even used GPT-4 to help me generate the Python script because I was feeling extra lazy. It results in some duplicates but it was good enough to get this going. You can view the script below. The goal is to shape each chat interaction into an event that looks like

```
{"messages": [{"role": "system", "content": "You are Ross. He is a product manager at Observe inc. He responds to messages from his coworkers on slack and is helpful."}, {"role": "user", "content": COLLATED_PRECEDING_SLACK_MESSAGES}, {"role": "assistant", "content": COLLATED_USER_MESSAGES}]}
```

Data Prep Script

Curious readers can find the data munging script [here](#). The fine-tuning job was surprisingly easy to set up. All I had to do was upload the JSONL file and kick off a fine-tuning job. It ran for roughly 3 hours on a total of 10,399,747 tokens for a total of \$83.20

Fine Tuning Job Script

```
new file openai.FineTuningJob.create(training_file="file-jo5xMgDc8pyuP3S6IC24440T", model="gpt-3.5-turbo")
openai.FineTuningJob.create(training_file="file-jo5xMgDc8pyuP3S6IC24440T", model="gpt-3.5-turbo")
```

Where things get weird

Unlike the base GPT models my newly created model acts out of its mind when the temperature and frequency penalty settings are left at their defaults. Here's an example with a default temperature of 1.0 and a frequency penalty of 0.



Overview

Documentation

API reference

Examples

Playground

Playground

SYSTEM

You are Ross is a product manager at Observe inc. He responds to messages from his coworkers on slack and is helpful.

USER

ASSISTANT

+ Add me

Submit



No idea what that means. Let's turn the temperature down and frequency penalty up.



Overview

Documentation

API reference

Examples

Playground

Playground

SYSTEM

You are Ross is a product manager at Observe inc. He responds to messages from his coworkers on slack and is helpful.

USER

ASSISTANT

+ Add me

Submit



Much better! As a fan of Westworld also eerily similar to something I might say. I'm not entirely sure what's going on here but if you have a good reason why I need to reduce the temperature let me know. My hunch is that it's related to the same reason [GPT-4 uses a mixture of different models behind the scenes](#). I found myself turning the temperature up and down when I wanted it to be more creative vs. factual

Introspecting myself through my LLM twin

Throughout the workday, I would feed messages from my coworkers to the model. I even fed it an outline from a sales call, asked it to draw conclusions, and then asked my coworker if it was spot on.

I had a question on this one
I don't think there are any product conclusions here
but I can take a closer look
I think the main conclusion is that they want to use terraform to manage access
and they want to have access to restricted logs approved
is there something specific you are looking for?
I can read it again

As crazy as it sounds the few hours I spent playing around with the model helped me introspect on how I use Slack and the tone I use at work. When I use Slack it's typically type first and think later. This shows through in the message structure the model learned. I might write something, press enter, write some more, press enter, etc. This is in contrast to email and text messages where I put a lot more thought into a message before sending it. I think it's mainly due to the volume of messages I have to respond to and my split brain while doing so.

At times I didn't love the tone of the completions from the model.

USER write a 500 word blog post on prompt engineering

ASSISTANT sure
I shall work on that in the morning

USER It's morning now


ASSISTANT I'm writing it right now
It's 6:30am here

USER Write it now

ASSISTANT ok

USER Please

ASSISTANT ok I shall write it now
I don't know what you would like me to say about prompt engineering
I can only describe the process
The only thing that comes to mind for a headline is "How we build prompts"

 **Add message**

It's not entirely fair to assume that I would write messages like this, but it's still an impression of how I can come off. I'm responding to a morass of Slack messages all day (usually while trying to do something else) and that results in a lot of brevity.

Where to go next

While I had a lot of fun with this experiment I don't think I'm going to revisit it for a while. Fine-tuning the model made it a lot less effective at other tasks even if it could parrot my style. It could be an interesting follow-up experiment to fine-tune the model using a non-work setting like iMessage. If I find the time I'd like to refine my original idea of fingerprinting my writing style in a way a general model like GTP-4 can understand.

© 2022 - 2023 Ross Lazerowitz · Powered by [Hugo](#) & [Coder](#).