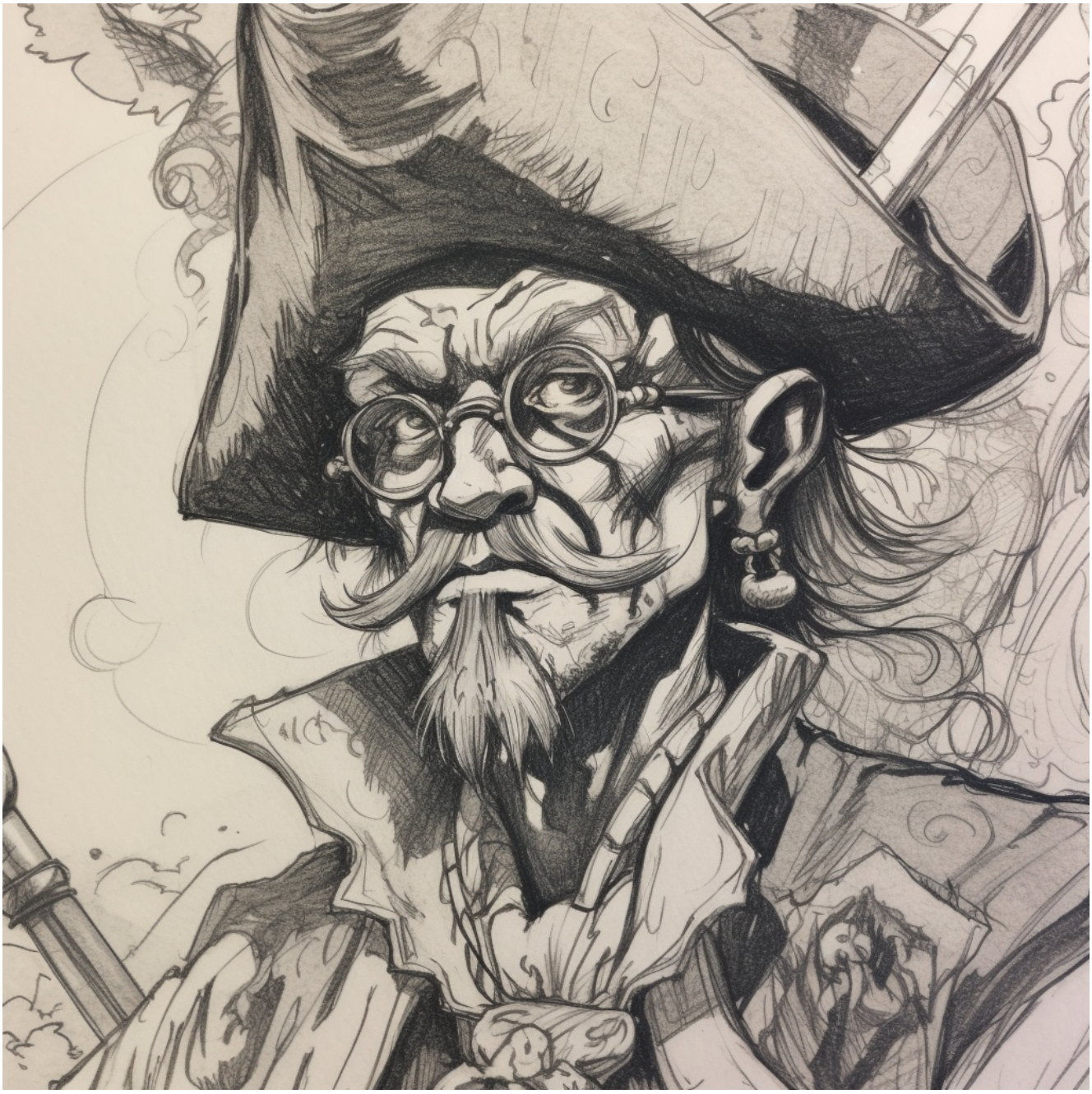
LLM Fine-tuning (Prompt Engineering)

written by DaltonRuer | November 7, 2023

Purpose

If I were to sit down at a prompt window with a Generative AI Art tool like MidJourney and ask for a Pirate it might render something like below. That’s what it does.

While that is a classic pirate look for sure, it’s not exactly what I wanted. So I would need to refine my prompt and instead of just asking for a “pirate” I might prompt it for a “**rough pencil sketch of 1960’s comic book style pirate.”**



“rough pencil sketch of 1960’s comic book style pirate”

As magical as that may be, if I were to ask it to generate a “**Mozart inspired pirate concerto**” it will let my music loving ears down. Because that isn’t what it’s purpose is. It’s purpose is to generate art, not music.

Persona

In my previous post, I asked you to watch a video about the various personas that might interact with Qlik’s Application Automation solution. So that you would understand why I was

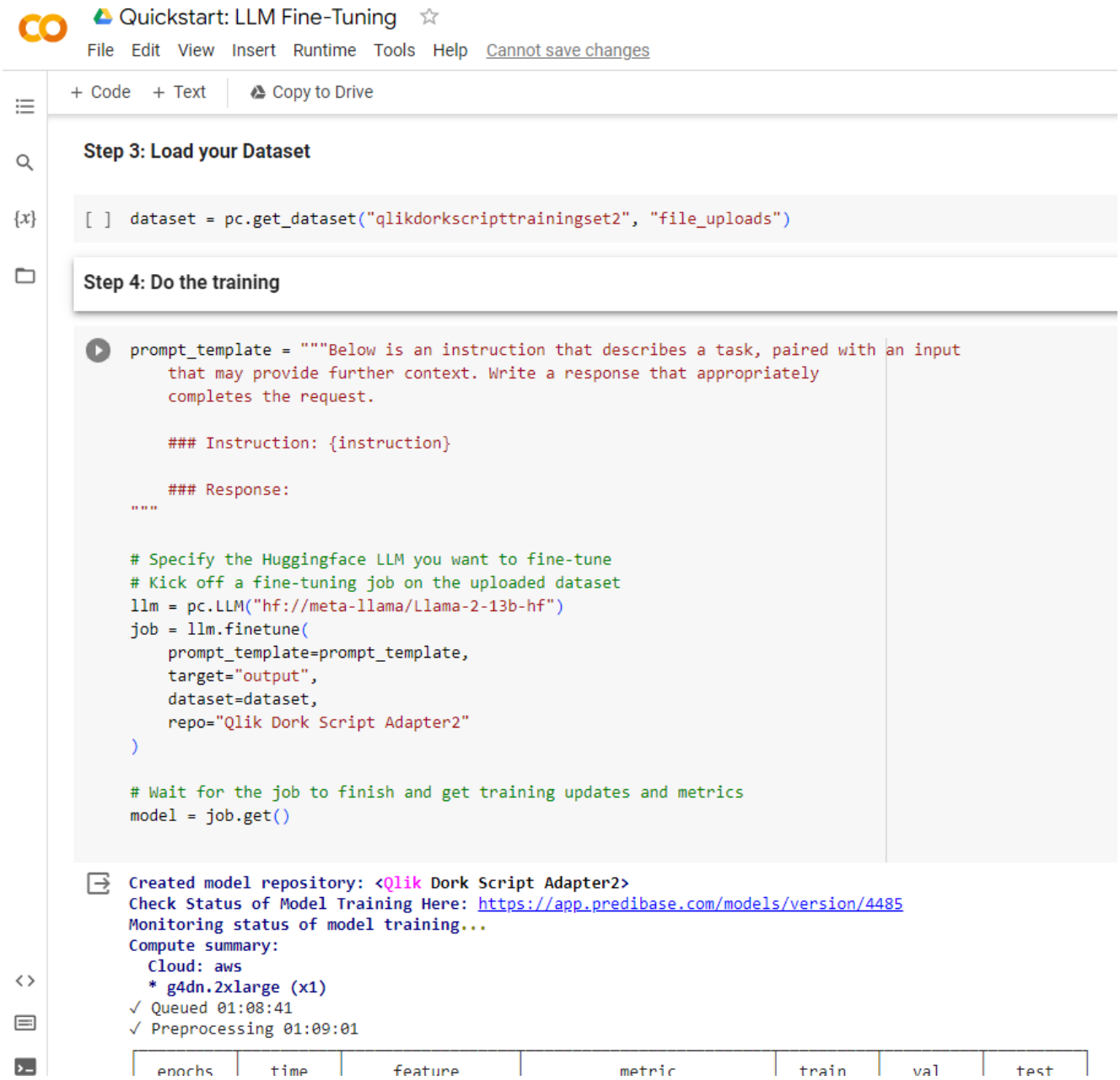
trying to generate very bespoke JSON workflows that met a users prompt. Actually, the goal would really be a copilot type chat, but on my budget and time constraints, I settled for copy and paste.

Initially the output was perfect. The more complicated the solution I was asking for, the more I struggled to get exactly what was needed. 1,217 right out of 1,220 characters wasn't enough.

I didn't realize the most important persona in the picture was "The model I was fine tuning" not the end user. I tweaked, and tuned and refined my training data set for both my previous posts, and I tweaked and tuned the way I asked for my desired workflow. But I never tweaked, or tuned, or refined the purpose I was giving the model I was trying to tune. I wasn't giving it a specific purpose. I wasn't providing it with a concise persona.

Prompt Engineering

As I look back at my initial LLM Fine-tuning post, I realize now that I actually got very lucky in that I was trying such a simple task. I had focused my energies on my training data set. I had focused my energies on naming my model. What I hadn't even paid any attention to, was the prompt template that was being used to provide the purpose, the persona, the reason for the model to consume energy. I simply copied the default prompt that Predibase had provided me in a training exercise.



The Llam2 base model already knew how to generate code. While I needed to provide a training data set so it understood my Qlik Dork Script language it was still within the generic purpose of that model. While it worked for my simple workflow requests. It didn't work for the more complex cases I continued asking for after publishing my workflow piece.

While you can tweak, refine, and tune the prompts you provide to MidJourney they have to fit within it's purpose.

When you provide the proper "prompt template" to a model for fine-tuning, to provide it's persona, it's purpose, it's reason for using electricity, that's called **prompt**

engineering. Predibase actually provides lots of examples, that in my haste I completely ignored.

1. Sentiment Analysis:

```
Given the following text, classify the sentiment as positive, negative, or neutral.
Text: {input_text}
Sentiment:
```

2. Text Summarization:

```
Summarize the following passage into a concise paragraph.
Paragraph: {paragraph}
Summary:
```

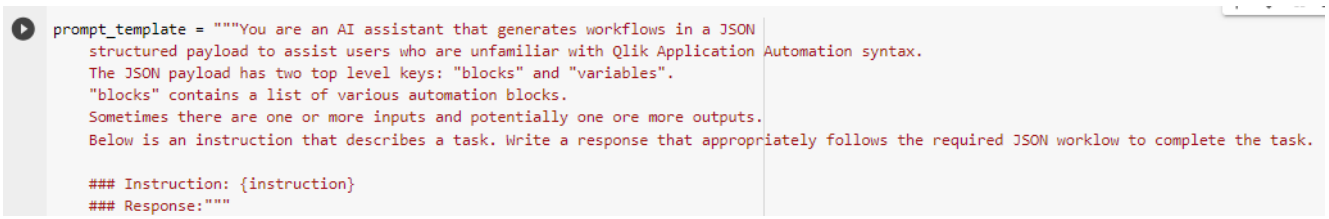
3. Named Entity Recognition:

```
Identify and label the named entities in the following sentence.
Sentence: {sentence}
Named Entities:
```

4. Language Translation:

```
Translate the following sentence from English to French.
Sentence: {sentence}
French Translation:
```

Those examples gave me something to work with, so I gave it a shot and tried my hand at prompt engineering:



```
prompt_template = """You are an AI assistant that generates workflows in a JSON
    structured payload to assist users who are unfamiliar with Qlik Application Automation syntax.
    The JSON payload has two top level keys: "blocks" and "variables".
    "blocks" contains a list of various automation blocks.
    Sometimes there are one or more inputs and potentially one ore more outputs.
    Below is an instruction that describes a task. Write a response that appropriately follows the required JSON worklow to complete the task.

    ### Instruction: {instruction}
    ### Response:"""
```

My buddy Connor McCormick also provided a very robust prompt template that actually includes some examples to really guide the tuning:

```
prompt_template = """You are an AI assistant that generates workflows in a JSON
    structured payload. The JSON payload has two top level keys: "blocks" and "variables".
    "blocks" contains a list of input and output blocks. Sometimes there are only inputs or a single
    input, but sometimes there are multiple inputs and outputs.

    Here is an example with only one input block and no ouput blocks:

    Instruction: Write a QAA that captures Dalton

    Workflow JSON: {"blocks":[{"id":"10D0ACE2-E642-48A6-AFEC-13220488E426", "type":"FormBlock","disabled":false,"name":"inputs","displayName":"Inputs","comment"
    {"id":"automations_censor_data","value":false,"type":"checkbox","structure":{}}],"collapsed":[{"name":"loop","isCollapsed":false}],"x":-325,"y":66, "form":[
    "persistData":"no"}], "variables":[]}


    Here is an example with 2 input blocks and one output block:

    Instruction: Give me a QAA that captures the value of Field1 and Field2 and outputs Field1

    Workflow JSON: {"blocks":[{"id":"326400C1-4052-4DFE-B69A-3FDC8A6F5E38","type":"FormBlock","disabled":false,"name":"inputs","displayName":"Inputs","comment":
    {"id":"09E99DAA-819C-4913-AA88-F68E7A1BBA10","type":"ShowBlock","disabled":false,"name":"output2","displayName":"Output 2","comment":"","childId":null,"inpu
     "variables":[]}

    ### Instruction: {instruction}

    ### Workflow JSON:"""
```

Lessons Learned

I was happy to be able to generate very simple single lines of code that were unique in the world. Even more ecstatic to get very, very close to generating the very bespoke JSON workflow syntax. Yet humbled by the fact that in my rush to do those things, I hadn't taken the time to realize the importance of the prompt template. Surely, by this stage in my career I should have realized that training a model for a specific task deserved a template that identified the reason for it to exist and consume power. But I hadn't. This process continues to intrigue and excite me and this week I learned:

1. Users prompts can be tweaked, refined and tuned, but they must fit within the persona of the model.
2. In addition to providing training data, fine-tuning involves prompt engineering, which means providing it a focused prompt template during the training process.
3. Fine-tuning requires patience, patience and patience. Some of my prompt templates helped me produce better output, others didn't. While I can tweak my "pirate" images within 1 minute, it was taking me between 17-24 hours to try and hone my prompt templates. My project was only for my own learning purposes. If your purpose is great enough, that time will be easily justified. Because it will mean you are taking one more bite out of that Generative AI elephant in the room.

4. Fine-tuning requires a combination of starting with the right base model, the right training data set and the right prompt template. Take in the entire picture, don’t just focus on 1 element of it.
5. While it didn’t generate actual music, the “Mozart inspirate Pirate concerto” prompt did produce the stunning featured image for this post. Which is simply a reminder, that every now and then we may get lucky even when we are doing the wrong thing. Which is a good thing because this Generative AI stuff is new for all of us. So keep trying. Keep failing. Keep learning. Keep adjusting.