**GPT-3 Fine Tuning Prompts & Code**

By Liam Ottley - <https://www.youtube.com/channel/@LiamOttley>

**Prompt 1:**

I have a spreadsheet of basketball data.

These are the column headings in csv format:

GP Games played,GS Games started,MPG Minutes Per Game,PPG Points Per Game,FGM Field Goals Made,FGA Field Goals Attempted,FG% Field Goal Percentage,3FGM Three-Point Field Goals Made,3FGA Three-Point Field Goals Attempted,3FG% Three-Point Field Goal Percentage,FTM Free Throws Made,FTA Free Throws Attempted,FT% Free Throw Percentage,Player,Position,Team

Here are example rows of data in csv format:

19,19,37.4,33.5,218,432,50.5,46,146,31.5,154,213,72.3,Luka Doncic,PG,DAL

20,20,34.7,31.4,214,413,51.8,103,236,43.6,97,107,90.7,Stephen Curry,PG,GS

20,20,35.8,31.1,215,425,50.6,21,62,33.9,170,185,91.9,Shai Gilgeous-Alexander,SG,OKC

16,16,32.6,30.9,182,341,53.4,11,48,22.9,120,200,60,Giannis Antetokounmpo,PF,MIL

Do you understand?

**Prompt 2:**

I want to use this data to fine tune GPT-3.

Write me a python script to create prompt and completion pairs in this format:

{"prompt": "<prompt text>", "completion": "<ideal generated text>"}

{"prompt": "<prompt text>", "completion": "<ideal generated text>"}

{"prompt": "<prompt text>", "completion": "<ideal generated text>"}

I want each completion to contain a written summary of all data for that player using string interpolation.

When inserting the data remember to use the full name of the column from the csv,

eg 'GP Games played' instead of just 'GP'. You must use all of the text for each column as written below (each column name is separated by a comma):

GP Games played,GS Games started,MPG Minutes Per Game,PPG Points Per Game,FGM Field Goals Made,FGA Field Goals Attempted,FG% Field Goal Percentage,3FGM Three-Point Field Goals Made,3FGA Three-Point Field Goals Attempted,3FG% Three-Point Field Goal Percentage,FTM Free Throws Made,FTA Free Throws Attempted,FT% Free Throw Percentage,Player,Position,Team

The prompt and completion pairs need to be exported to a json file.

**Prompt & Completion Generation Script:**

import csv

import json

# Read CSV file

with open('basketball\_data.csv', 'r') as f:

data = list(csv.reader(f))

headers = data[0]

data = data[1:]

# Create prompt and completion pairs

pairs = []

for player in data:

prompt = f"Write a summary of {player[headers.index('Player')]}'s statistics:"

completion = f"{player[headers.index('Player')]} played {player[headers.index('GP Games played')]} games, starting {player[headers.index('GS Games started')]} of them. He had an average of {player[headers.index('MPG Minutes Per Game')]} minutes per game, scoring {player[headers.index('PPG Points Per Game')]} points per game. He made {player[headers.index('FGM Field Goals Made')]} out of {player[headers.index('FGA Field Goals Attempted')]} field goals, for a field goal percentage of {player[headers.index('FG% Field Goal Percentage')]}. He made {player[headers.index('3FGM Three-Point Field Goals Made')]} out of {player[headers.index('3FGA Three-Point Field Goals Attempted')]} three-point field goals, for a three-point field goal percentage of {player[headers.index('3FG% Three-Point Field Goal Percentage')]}. He made {player[headers.index('FTM Free Throws Made')]} out of {player[headers.index('FTA Free Throws Attempted')]} free throws, for a free throw percentage of {player[headers.index('FT% Free Throw Percentage')]}. He plays as {player[headers.index('Position')]} for the {player[headers.index('Team')]}."

pairs.append({"prompt": prompt, "completion": completion})

# Export to JSON file

with open('prompt\_completion\_pairs.json', 'w') as f:

json.dump(pairs, f)

**Application Script (GUI):**

(Replace your “model name” with yours from your terminal after fine tuning!)

import tkinter as tk

import openai

# Replace FINE\_TUNED\_MODEL with the name of your fine-tuned model

model\_name = "curie:ft-personal-2023-01-21-10-47-08"

def on\_submit():

# Get the prompt from the input field

prompt = input\_field.get()

# Make the completion request

completion = openai.Completion.create(model=model\_name, prompt=prompt)

# Clear the input field

input\_field.delete(0, "end")

# Get the completion text from the first choice in the choices list

text = completion.choices[0]["text"]

# Display the completion in the result text area

result\_text.config(state="normal")

result\_text.delete("1.0", "end")

result\_text.insert("end", text)

result\_text.config(state="disabled")

# Create the main window

window = tk.Tk()

window.title("Fine-tuned GPT-3")

# Create the input field and submit button

input\_field = tk.Entry(window)

submit\_button = tk.Button(window, text="Submit", command=on\_submit)

# Create the result text area

result\_text = tk.Text(window, state="normal", width=80, height=20)

# Add the input field, submit button, and result text area to the window

input\_field.pack()

submit\_button.pack()

result\_text.pack()

# Run the main loop

window.mainloop()