Exercises for Chapter 11: Advanced Fine Tuning: Drug Classification

Fine-Tuning based on 2000 drug examples from an Excel file

# Step 1: [Preparing the Data and Launching the Fine Tuning](https://hc.labnet.sfbu.edu/~henry/sfbu/course/generative_ai/Advanced_Fine_Tuning_Drug_Classification/slide/Preparing_the_Data_and_Launching_the_Fine_Tuning.html)

# References

# [source](https://hc.labnet.sfbu.edu/~henry/sfbu/course/machine_learning/book/chatgpt/split_by_bookmark/OpenAI_GPT_For_Python_Developers/part70.htm)

## Step 1.1 Preparing the Data and Launching the Fine Tuning

# Use Pandas to transform the data into the desired format.

import pandas as pd

######################################################################

# Read the first n rows from the Excel file

# - The number of rows to read from the Excel file,

# Medicine\_description.xlsx, to 2000.

# + This means that we are going to use a dataset of 2000 drug

# names to fine-tune the model.

# - You can use more.

######################################################################

n = 2000

######################################################################

# [Kaggle data](https://hc.labnet.sfbu.edu/~henry/sfbu/course/generative_ai/Advanced_Fine_Tuning_Drug_Classification/kaggle/index_kaggle.html)

# - [Company\_Name.xlsx](https://hc.labnet.sfbu.edu/~henry/sfbu/course/generative_ai/Advanced_Fine_Tuning_Drug_Classification/kaggle/Company_Name.xlsx)

# - [Medicine\_description.xlsx](https://hc.labnet.sfbu.edu/~henry/sfbu/course/generative_ai/Advanced_Fine_Tuning_Drug_Classification/kaggle/Medicine_description.xlsx) - 3 columns

# + Drug\_Name

# + Reason

# + Description

# - [Ratings.xlsx](https://hc.labnet.sfbu.edu/~henry/sfbu/course/generative_ai/Advanced_Fine_Tuning_Drug_Classification/kaggle/Ratings.xlsx)

######################################################################

# Reading the first n rows of data from the Excel file

# 'Medicine\_description.xlsx’ and stores it in a data frame called df.

df = pd.read\_excel('[Medicine\_description.xlsx](https://hc.labnet.sfbu.edu/~henry/sfbu/course/generative_ai/Advanced_Fine_Tuning_Drug_Classification/kaggle/Medicine_description.xlsx)', sheet\_name='Sheet1',

header=0, nrows=n)

# Get the unique values in the ‘Reason’ column of the data frame,

# stores them in an array called reasons

reasons = df["Reason"].unique()

# Assigns a numerical index to each unique value in the reasons

# array, and stores it in a dictionary called reasons\_dict.

reasons\_dict = {reason: i for i, reason in enumerate(reasons)}

# Add a new line and “Malady:” to the end of each drug name in

# the ‘Drug\_Name’ column of the data frame.

# - The desired format:

# Drug: <Drug\_Name>\nMalady:

df["Drug\_Name"] = "Drug: " + df["Drug\_Name"] + "\n" + "Malady:"

# It concatenates a space and the corresponding numerical index

# from the reasons\_dict to the end of each 'Reason’

# value in the data frame.

df["Reason"] = " " + df["Reason"].apply(lambda x: "" + str(reasons\_dict[x]))

# For this example, we don’t need the ‘Description’ column, that’s

# why the script drops it from the data frame.

df.drop(["Description"], axis=1, inplace=True)

# Renaming the ‘Drug\_Name’ column to ‘prompt’

# and the ‘Reason’ column to ‘completion’.

df.rename(columns={"Drug\_Name": "prompt", "Reason": "completion"}, inplace=True)

# Convert the dataframe to jsonl format

jsonl = df.to\_json(orient="records", indent=0, lines=True)

# Write the jsonl to a file

#

# - drug\_malady\_data.jsonl has data like

# [..]

# {"prompt":"Drug: Acleen 1% Lotion 25ml\nMalady:","completion":" 0"}

# [..]

# {"prompt":"Drug: Capnea Injection 1ml\nMalady:","completion":" 1"}

# [..]

# {"prompt":"Drug: Mondeslor Tablet 10'S\nMalady:","completion":" 2"}

# [..]

with open("drug\_malady\_data.jsonl", "w") as f:

f.write(jsonl)

A screenshot of a computer program

Description automatically generated

## Step1.2 Command to Prepare Data

Analyze and prepare the data using the OpenAI tools fine\_tunes.prepare\_data command.

A screenshot of a computer program

Description automatically generated

## Step 1.3 Command to train the model

Use the provided command to train the model using fine\_tunes.create.

# Export your OpenAI key

export OPENAI\_API\_KEY="xxxxxxxxxxxx"

openai api fine\_tunes.create \

-t "drug\_malady\_data\_prepared\_train.jsonl" \

-v "drug\_malady\_data\_prepared\_valid.jsonl" \

--compute\_classification\_metrics \

--classification\_n\_classes 3 \

-m ada \

--suffix "drug\_malady\_data"

A screenshot of a computer program

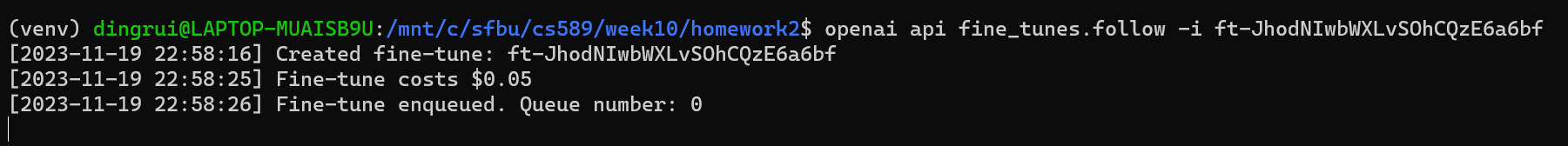
Description automatically generated

## Step 1.4 Checking Job Progress

If the client disconnects during fine-tuning, use the following command to check job progress.

openai api fine\_tunes.follow -i <JOB ID>

The output will display progress information and queue numbers.



## Step 1.5 Completion of Fine-Tuning

When the fine-tuning job is completed, you'll receive an output like this:

Created fine-tune: <JOB ID>

Fine-tune costs $0.03

Fine-tune enqueued

Fine-tune is in the queue. Queue number: 31

Fine-tune is in the queue. Queue number: 30

Fine-tune is in the queue. Queue number: 29

Fine-tune is in the queue. Queue number: 28

[...]

[...]

[...]

Fine-tune is in the queue. Queue number: 2

Fine-tune is in the queue. Queue number: 1

Fine-tune is in the queue. Queue number: 0

Fine-tune started

Completed epoch 1/4

Completed epoch 2/4

Completed epoch 3/4

Completed epoch 4/4

Uploaded model: <MODEL ID>

Uploaded result file: <FILE ID>

Fine-tune succeeded

Job complete! Status: succeeded

Try out your fine-tuned model:

openai api completions.create -m <MODEL ID> -p <YOUR\_PROMPT>

A computer screen with white text

Description automatically generated

# Step 2: [Testing the Fine Tuned Model](https://hc.labnet.sfbu.edu/~henry/sfbu/course/generative_ai/Advanced_Fine_Tuning_Drug_Classification/slide/Testing_the_Fine_Tuned_Model.html)

# References

# [source](https://hc.labnet.sfbu.edu/~henry/sfbu/course/machine_learning/book/chatgpt/split_by_bookmark/OpenAI_GPT_For_Python_Developers/part71.htm)

## Step 2.1 Python Code: Model Testing

A screenshot of a computer program

Description automatically generated