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

Khoi Duong
Prof. Chang
CS589
4/9/2024

1. Fine-Tuning based on 2000 drug examples from an Excel file .
    - Project Implementation
        - Step 1: Preparing the Data and Launching the Fine Tuning
            - References
                - source
        - Step 2: Testing the Fine Tuned Model
            - References
                - source
    - Process for the project documentation
        - Step 1: Adding the project to your portofolio
            1. Please use Google Slides to document the project
                - Copy from a Google Slides file and mofigy the file, but still keep the original Google Slides file.
            2. Please link your presentation on GitHub using this structure

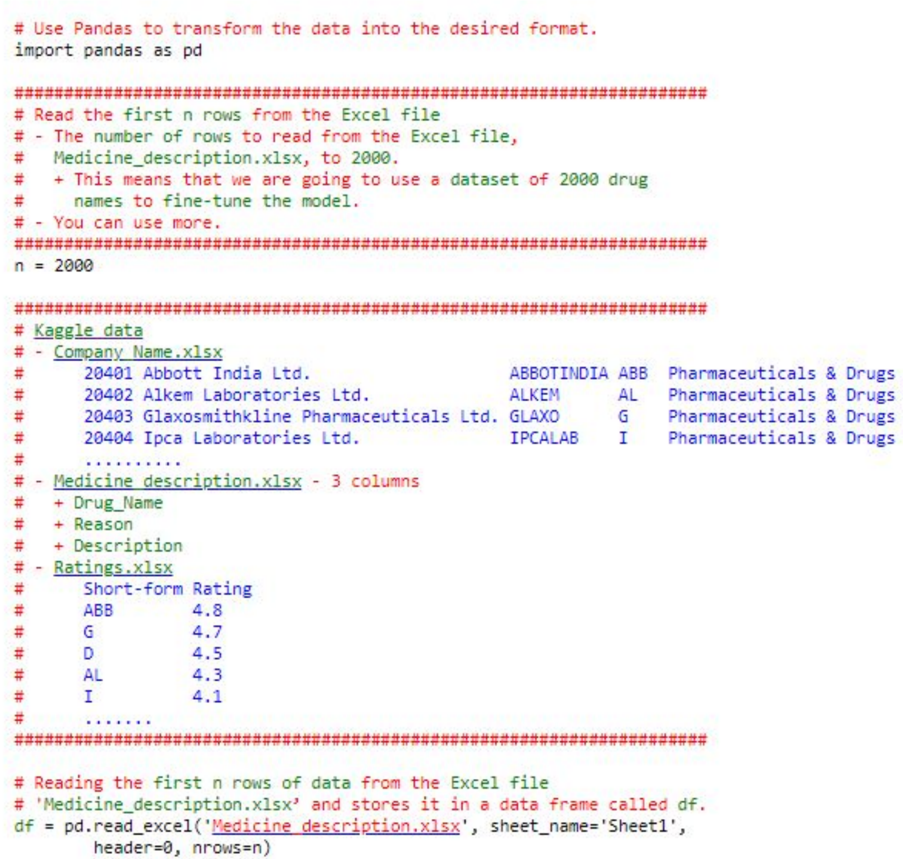
```
Generative AI
  - Fine-Tuning
    + 2000 Drug Examples
```

        - Step 2: Submit
            1. The URLs of the Google Slides and GitHub web pages related to this project.
            2. A PDF file of your Google Slides

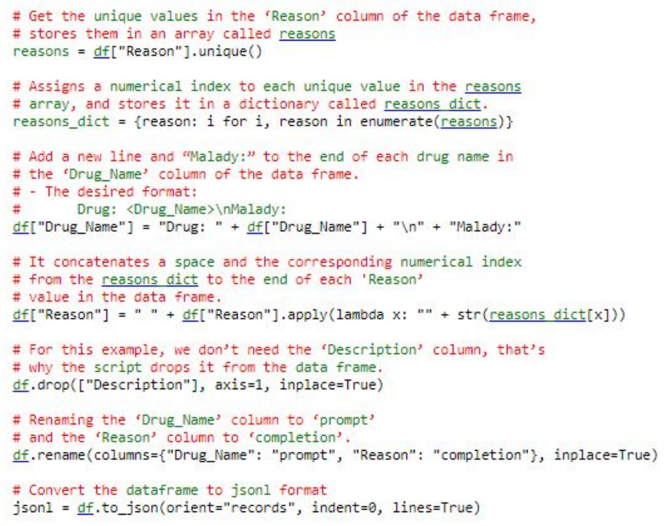
Project Implementation

- Step 1: Preparing the Data and Launching the Fine Tuning
    - Ratings.xlsx
    - Company_Name.xlsx
    - Medicine_description.xlsx
- Step 2: Testing the Fine Tuned Model

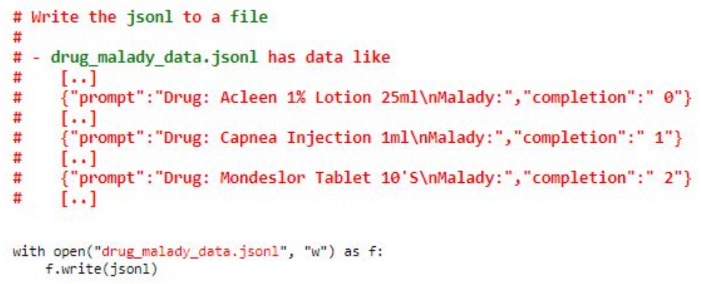
Step 1: Preparing the Data and Launching the Fine Tuning



```
# 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
# - Company_Name.xlsx
#      20401 Abbott India Ltd.                  ABBOTINDIA ABB  Pharmaceuticals & Drugs
#      20402 Alkem Laboratories Ltd.            ALKEM      AL   Pharmaceuticals & Drugs
#      20403 Glaxosmithkline Pharmaceuticals Ltd. GLAXO    G    Pharmaceuticals & Drugs
#      20404 Ipca Laboratories Ltd.             IPCALAB    I    Pharmaceuticals & Drugs
#      ..........
# - Medicine_description.xlsx - 3 columns
#   + Drug_Name
#   + Reason
#   + Description
# - Ratings.xlsx
#      Short-form Rating
#      ABB        4.8
#      G          4.7
#      D          4.5
#      AL         4.3
#      I          4.1
#      .......
######################################################################

# 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', sheet_name='Sheet1',
        header=0, nrows=n)
```



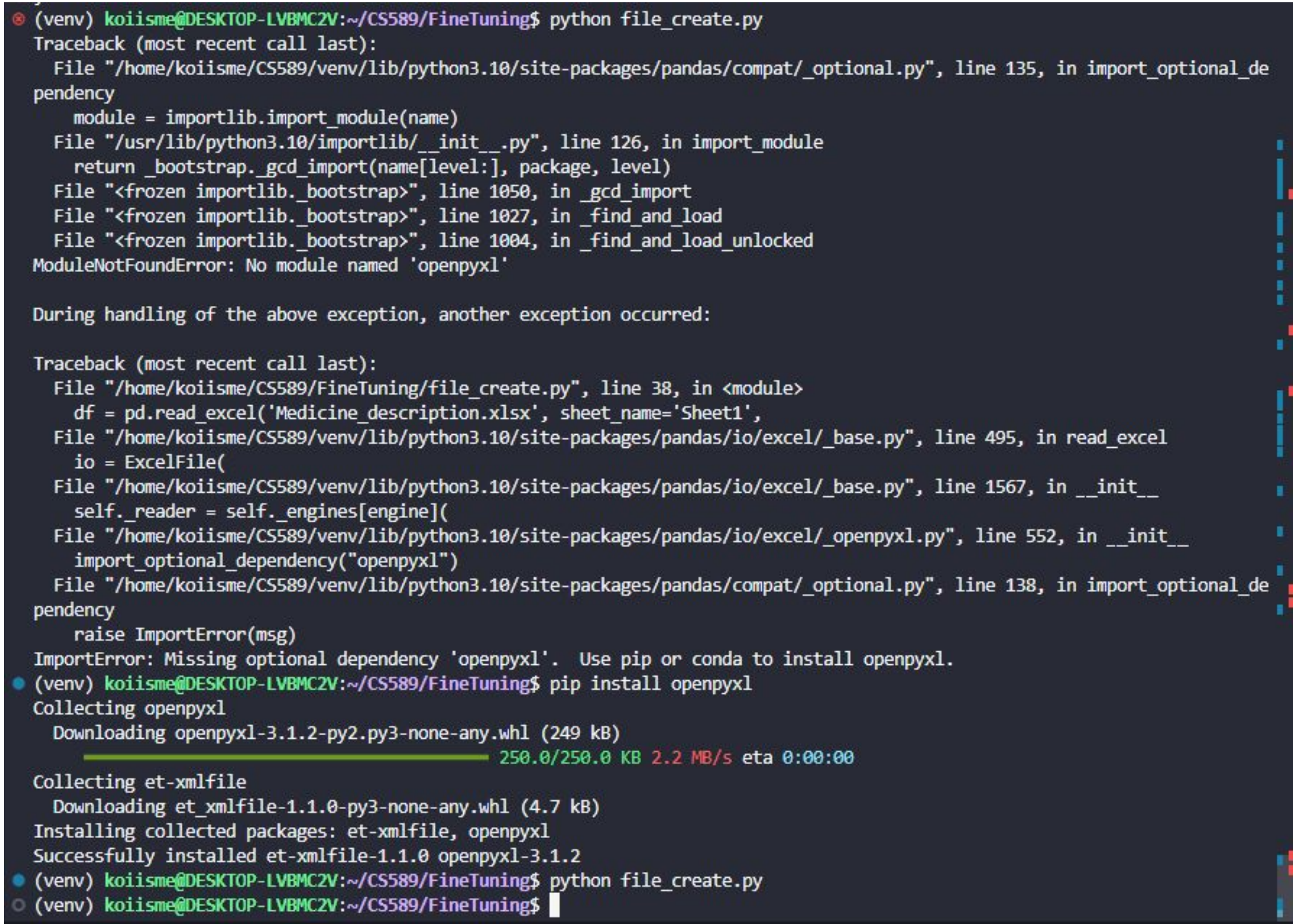
```
# Get the unique values in the 'Reason' column of the data frame,
# stores them in an array called reasons
reasons = df["Reason"].unique()

# 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)
```

Step 1

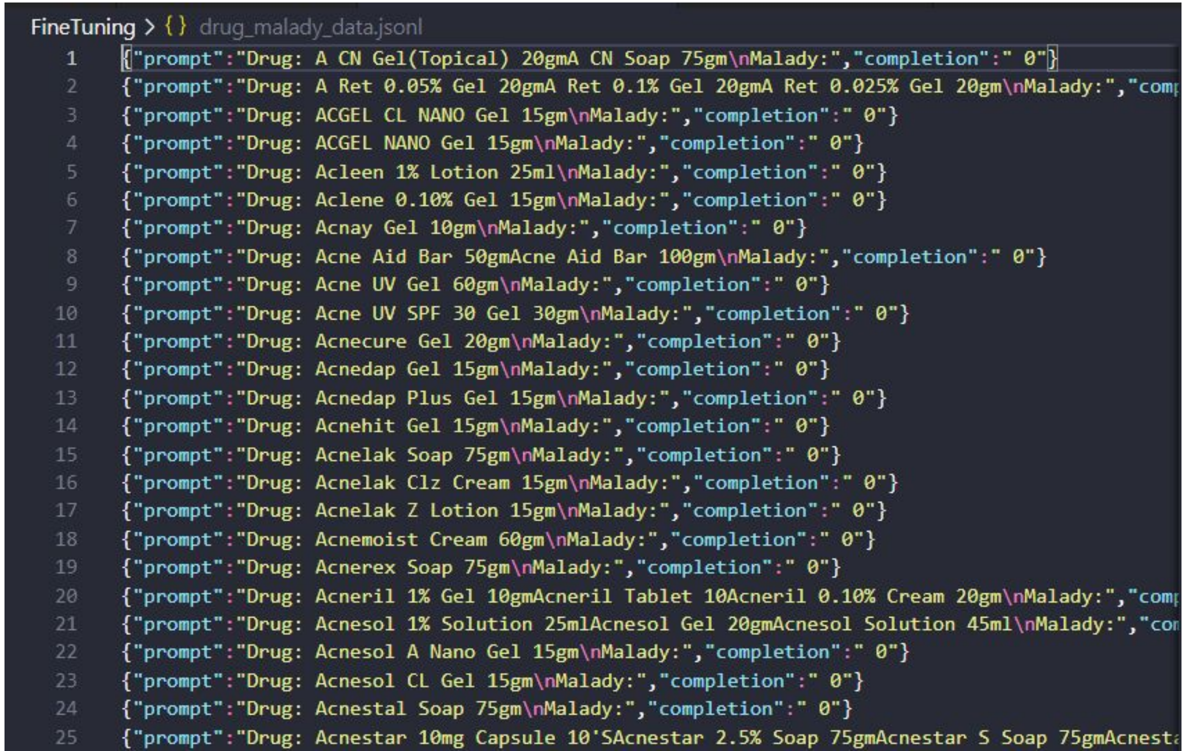
Run the file, if there is any missing dependencies, try to install with pip



```
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$ python file_create.py
Traceback (most recent call last):
  File "/home/koiisme/CS589/venv/lib/python3.10/site-packages/pandas/compat/_optional.py", line 135, in import_optional_de
pendency
    module = importlib.import_module(name)
  File "/usr/lib/python3.10/importlib/__init__.py", line 126, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
  File "<frozen importlib._bootstrap>", line 1050, in _gcd_import
  File "<frozen importlib._bootstrap>", line 1027, in _find_and_load
  File "<frozen importlib._bootstrap>", line 1004, in _find_and_load_unlocked
ModuleNotFoundError: No module named 'openpyxl'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/home/koiisme/CS589/FineTuning/file_create.py", line 38, in <module>
    df = pd.read_excel('Medicine_description.xlsx', sheet_name='Sheet1',
  File "/home/koiisme/CS589/venv/lib/python3.10/site-packages/pandas/io/excel/_base.py", line 495, in read_excel
    io = ExcelFile(
  File "/home/koiisme/CS589/venv/lib/python3.10/site-packages/pandas/io/excel/_base.py", line 1567, in __init__
    self._reader = self._engines[engine](
  File "/home/koiisme/CS589/venv/lib/python3.10/site-packages/pandas/io/excel/_openpyxl.py", line 552, in __init__
    import_optional_dependency("openpyxl")
  File "/home/koiisme/CS589/venv/lib/python3.10/site-packages/pandas/compat/_optional.py", line 138, in import_optional_de
pendency
    raise ImportError(msg)
ImportError: Missing optional dependency 'openpyxl'.  Use pip or conda to install openpyxl.
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$ pip install openpyxl
Collecting openpyxl
  Downloading openpyxl-3.1.2-py2.py3-none-any.whl (249 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 250.0/250.0 KB 2.2 MB/s eta 0:00:00
Collecting et-xmlfile
  Downloading et_xmlfile-1.1.0-py3-none-any.whl (4.7 kB)
Installing collected packages: et-xmlfile, openpyxl
Successfully installed et-xmlfile-1.1.0 openpyxl-3.1.2
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$ python file_create.py
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$
```

1. Preparing the Data and Launching the Fine Tuning

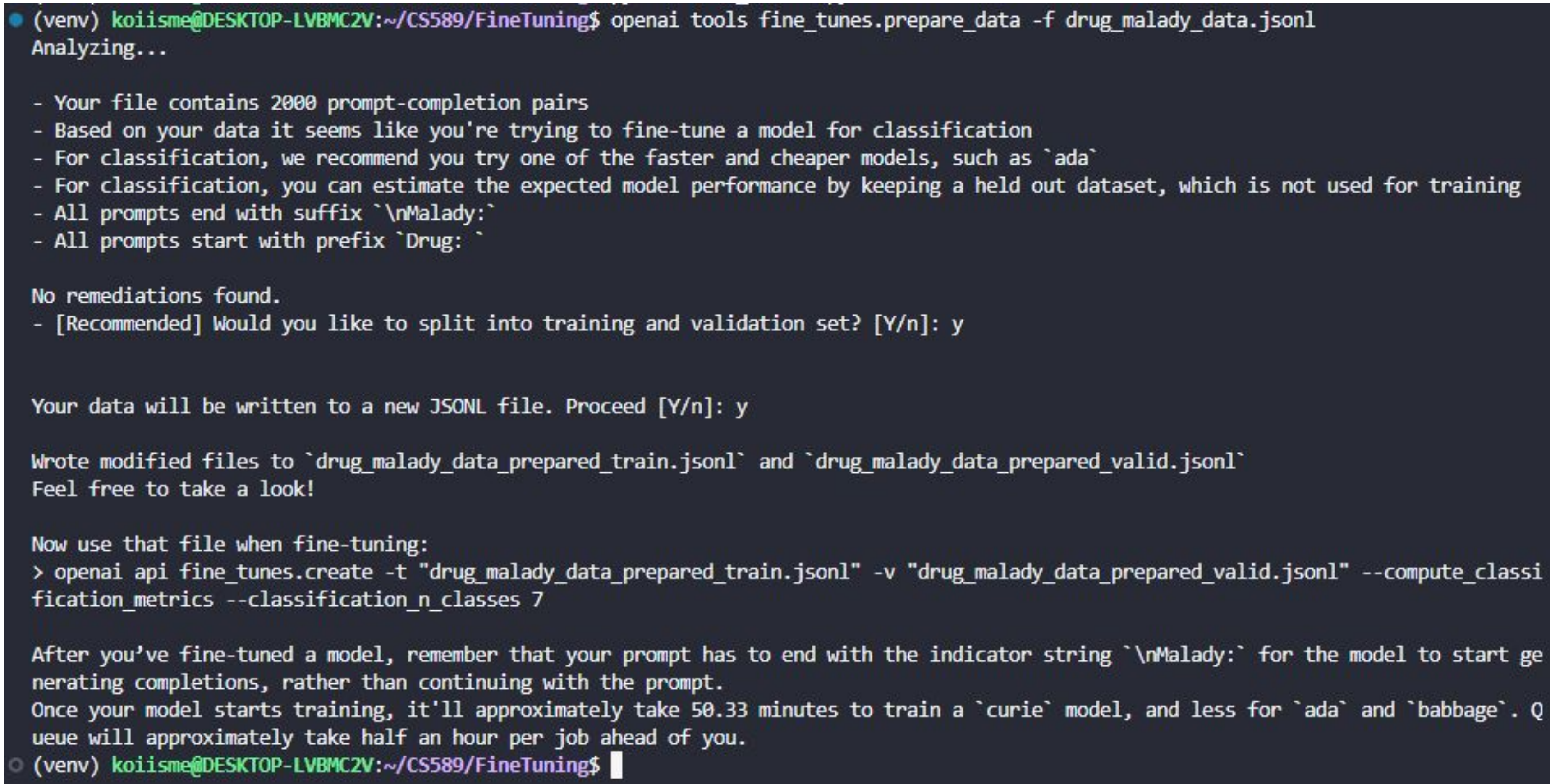
We then have a sample of our jsonl file created



FineTuning > {} drug_malady_data.jsonl

```
{"prompt":"Drug: A CN Gel(Topical) 20gmA CN Soap 75gm\nMalady:","completion":" 0"}
{"prompt":"Drug: A Ret 0.05% Gel 20gmA Ret 0.1% Gel 20gmA Ret 0.025% Gel 20gm\nMalady:","comp
{"prompt":"Drug: ACGEL CL NANO Gel 15gm\nMalady:","completion":" 0"}
{"prompt":"Drug: ACGEL NANO Gel 15gm\nMalady:","completion":" 0"}
{"prompt":"Drug: Acleen 1% Lotion 25ml\nMalady:","completion":" 0"}
{"prompt":"Drug: Aclene 0.10% Gel 15gm\nMalady:","completion":" 0"}
{"prompt":"Drug: Acnay Gel 10gm\nMalady:","completion":" 0"}
{"prompt":"Drug: Acne Aid Bar 50gmAcne Aid Bar 100gm\nMalady:","completion":" 0"}
{"prompt":"Drug: Acne UV Gel 60gm\nMalady:","completion":" 0"}
{"prompt":"Drug: Acne UV SPF 30 Gel 30gm\nMalady:","completion":" 0"}
{"prompt":"Drug: Acnecure Gel 20gm\nMalady:","completion":" 0"}
{"prompt":"Drug: Acnedap Gel 15gm\nMalady:","completion":" 0"}
{"prompt":"Drug: Acnedap Plus Gel 15gm\nMalady:","completion":" 0"}
{"prompt":"Drug: Acnehit Gel 15gm\nMalady:","completion":" 0"}
{"prompt":"Drug: Acnelak Soap 75gm\nMalady:","completion":" 0"}
{"prompt":"Drug: Acnelak Clz Cream 15gm\nMalady:","completion":" 0"}
{"prompt":"Drug: Acnelak Z Lotion 15gm\nMalady:","completion":" 0"}
{"prompt":"Drug: Acnemoist Cream 60gm\nMalady:","completion":" 0"}
{"prompt":"Drug: Acnerex Soap 75gm\nMalady:","completion":" 0"}
{"prompt":"Drug: Acneril 1% Gel 10gmAcneril Tablet 10Acneril 0.10% Cream 20gm\nMalady:","comp
{"prompt":"Drug: Acnesol 1% Solution 25mlAcnesol Gel 20gmAcnesol Solution 45ml\nMalady:","com
{"prompt":"Drug: Acnesol A Nano Gel 15gm\nMalady:","completion":" 0"}
{"prompt":"Drug: Acnesol CL Gel 15gm\nMalady:","completion":" 0"}
{"prompt":"Drug: Acnestal Soap 75gm\nMalady:","completion":" 0"}
{"prompt":"Drug: Acnestar 10mg Capsule 10'SAcnestar 2.5% Soap 75gmAcnestar S Soap 75gmAcnesta
```

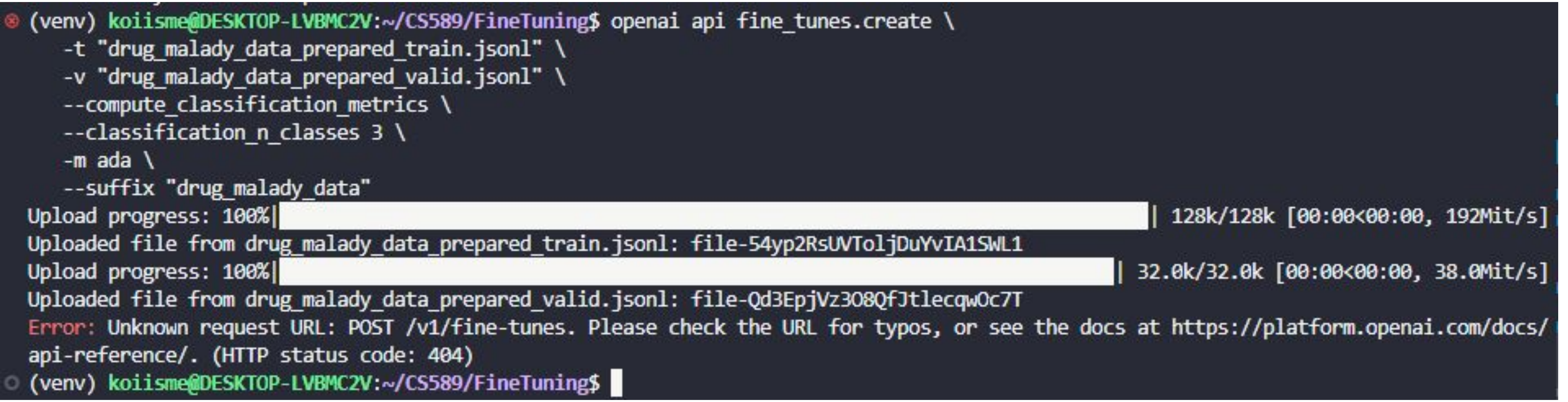
2. Command to prepare data



```
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$ openai tools fine_tunes.prepare_data -f drug_malady_data.jsonl
Analyzing...

- Your file contains 2000 prompt-completion pairs
- Based on your data it seems like you're trying to fine-tune a model for classification
- For classification, we recommend you try one of the faster and cheaper models, such as `ada`
- For classification, you can estimate the expected model performance by keeping a held out dataset, which is not used for training
- All prompts end with suffix `\nMalady:`
- All prompts start with prefix `Drug: `

No remediations found.
- [Recommended] Would you like to split into training and validation set? [Y/n]: y


Your data will be written to a new JSONL file. Proceed [Y/n]: y

Wrote modified files to `drug_malady_data_prepared_train.jsonl` and `drug_malady_data_prepared_valid.jsonl`
Feel free to take a look!

Now use that file when fine-tuning:
> openai api fine_tunes.create -t "drug_malady_data_prepared_train.jsonl" -v "drug_malady_data_prepared_valid.jsonl" --compute_classi
fication_metrics --classification_n_classes 7

After you've fine-tuned a model, remember that your prompt has to end with the indicator string `\nMalady:` for the model to start ge
nerating completions, rather than continuing with the prompt.
Once your model starts training, it'll approximately take 50.33 minutes to train a `curie` model, and less for `ada` and `babbage`. Q
ueue will approximately take half an hour per job ahead of you.
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$
```

3. Command to Train the Model

We try to execute the command below with OpenAI version 0.28.1

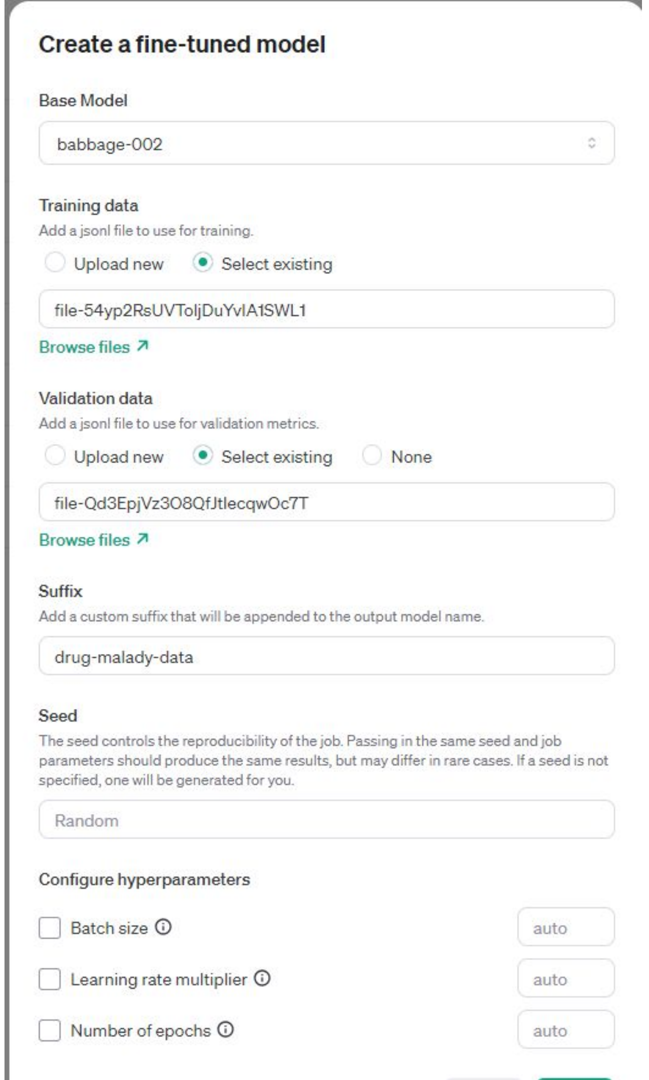


```
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$ openai api fine_tunes.create \
    -t "drug_malady_data_prepared_train.jsonl" \
    -v "drug_malady_data_prepared_valid.jsonl" \
    --compute_classification_metrics \
    --classification_n_classes 3 \
    -m ada \
    --suffix "drug_malady_data"
Upload progress: 100%|                                                    | 128k/128k [00:00<00:00, 192Mit/s]
Uploaded file from drug_malady_data_prepared_train.jsonl: file-54yp2RsUVToljDuYvIA1SWL1
Upload progress: 100%|                                                  | 32.0k/32.0k [00:00<00:00, 38.0Mit/s]
Uploaded file from drug_malady_data_prepared_valid.jsonl: file-Qd3EpjVz3O8QfJtlecqwOc7T
Error: Unknown request URL: POST /v1/fine-tunes. Please check the URL for typos, or see the docs at https://platform.openai.com/docs/
api-reference/. (HTTP status code: 404)
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$
```

We can see that the command is deprecated since they no longer support the old version command. Therefore, we can use the platform to fine-tune the model

3. (alternative) Form to Train the Model

Here is how we can set up a form to create a fine-tuned model on OpenAI platform.



Create a fine-tuned model

Base Model

babbage-002

Training data

Add a jsonl file to use for training.

Upload new [✓ Select existing]

file-54yp2RsUVToljDuYvIA1SWL1

Browse files ↗

Validation data

Add a jsonl file to use for validation metrics.

Upload new [✓ Select existing] None

file-Qd3EpjVz3O8QfJtlecqwOc7T

Browse files ↗

Suffix

Add a custom suffix that will be appended to the output model name.

drug-malady-data

Seed

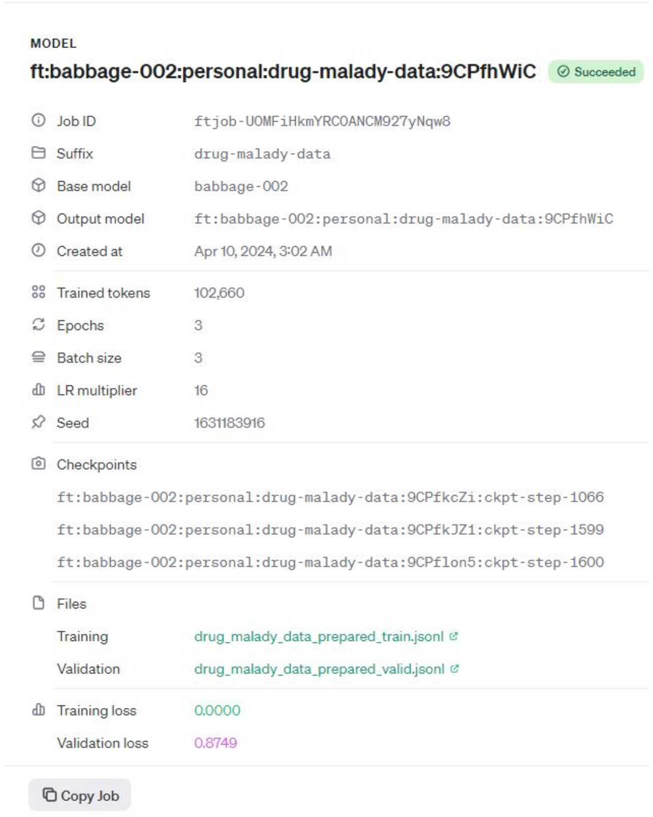
The seed controls the reproducibility of the job. Passing in the same seed and job parameters should produce the same results, but may differ in rare cases. If a seed is not specified, one will be generated for you.

Random

Configure hyperparameters

☐ Batch size	auto
☐ Learning rate multiplier	auto
☐ Number of epochs	auto

After waiting for OpenAI to fine-tune the new model from the base model, we receive the result:



MODEL

ft:babbage-002:personal:drug-malady-data:9CPfhWiC Succeeded

Job ID	ftjob-UOMFiHkmYRCOANCM927yNqw8
Suffix	drug-malady-data
Base model	babbage-002
Output model	ft:babbage-002:personal:drug-malady-data:9CPfhWiC
Created at	Apr 10, 2024, 3:02 AM
Trained tokens	102,660
Epochs	3
Batch size	3
LR multiplier	16
Seed	1631183916

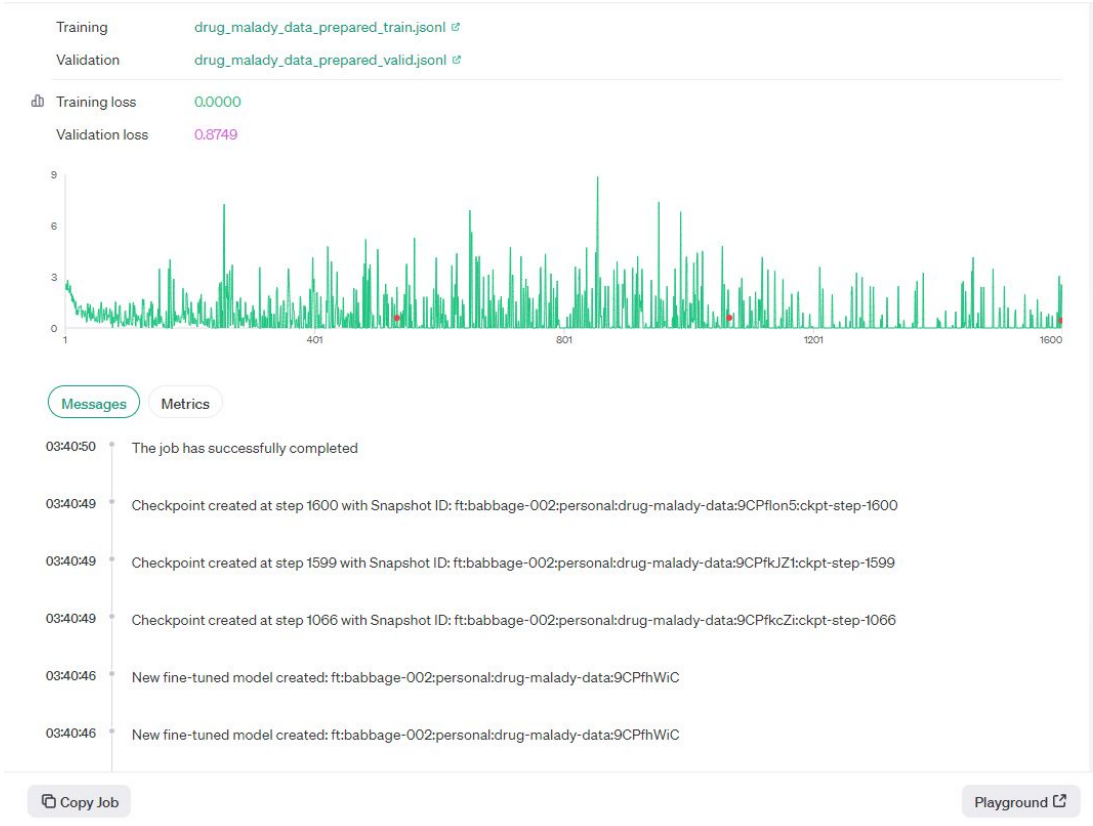
Checkpoints

```
ft:babbage-002:personal:drug-malady-data:9CPfkcZi:ckpt-step-1066
ft:babbage-002:personal:drug-malady-data:9CPfkJZ1:ckpt-step-1599
ft:babbage-002:personal:drug-malady-data:9CPflon5:ckpt-step-1600
```

Files

Training	drug_malady_data_prepared_train.jsonl
Validation	drug_malady_data_prepared_valid.jsonl
Training loss	0.0000
Validation loss	0.8749

Copy Job

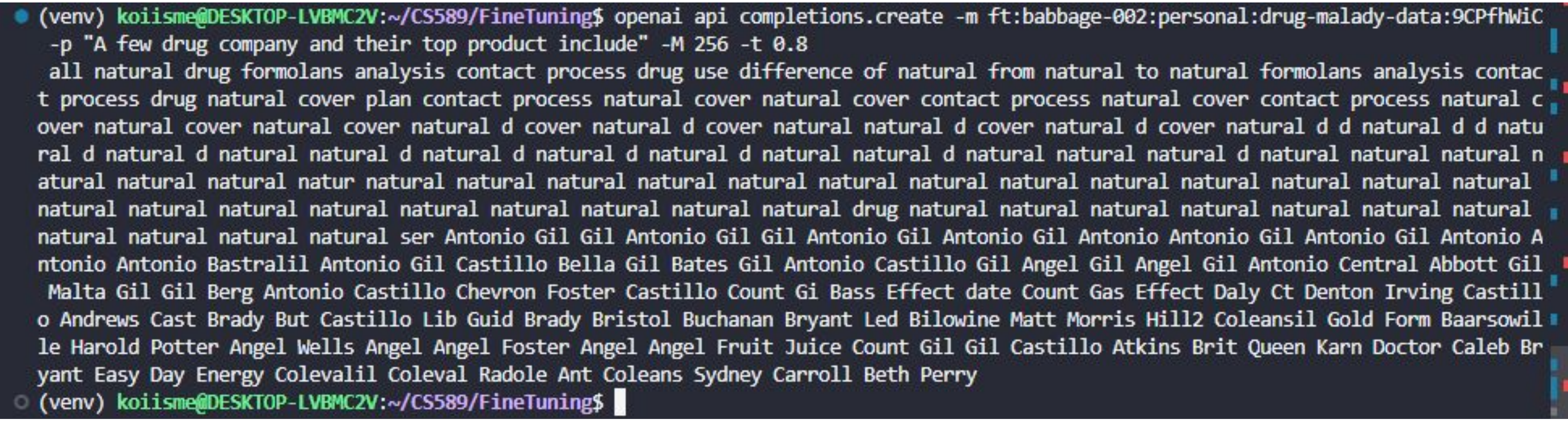


Training	drug_malady_data_prepared_train.jsonl
Validation	drug_malady_data_prepared_valid.jsonl
Training loss	0.0000
Validation loss	0.8749

```
03:40:50  The job has successfully completed
03:40:49  Checkpoint created at step 1600 with Snapshot ID: ft:babbage-002:personal:drug-malady-data:9CPflon5:ckpt-step-1600
03:40:49  Checkpoint created at step 1599 with Snapshot ID: ft:babbage-002:personal:drug-malady-data:9CPfkJZ1:ckpt-step-1599
03:40:49  Checkpoint created at step 1066 with Snapshot ID: ft:babbage-002:personal:drug-malady-data:9CPfkcZi:ckpt-step-1066
03:40:46  New fine-tuned model created: ft:babbage-002:personal:drug-malady-data:9CPfhWiC
03:40:46  New fine-tuned model created: ft:babbage-002:personal:drug-malady-data:9CPfhWiC
```

4. Completion of fine-tuning

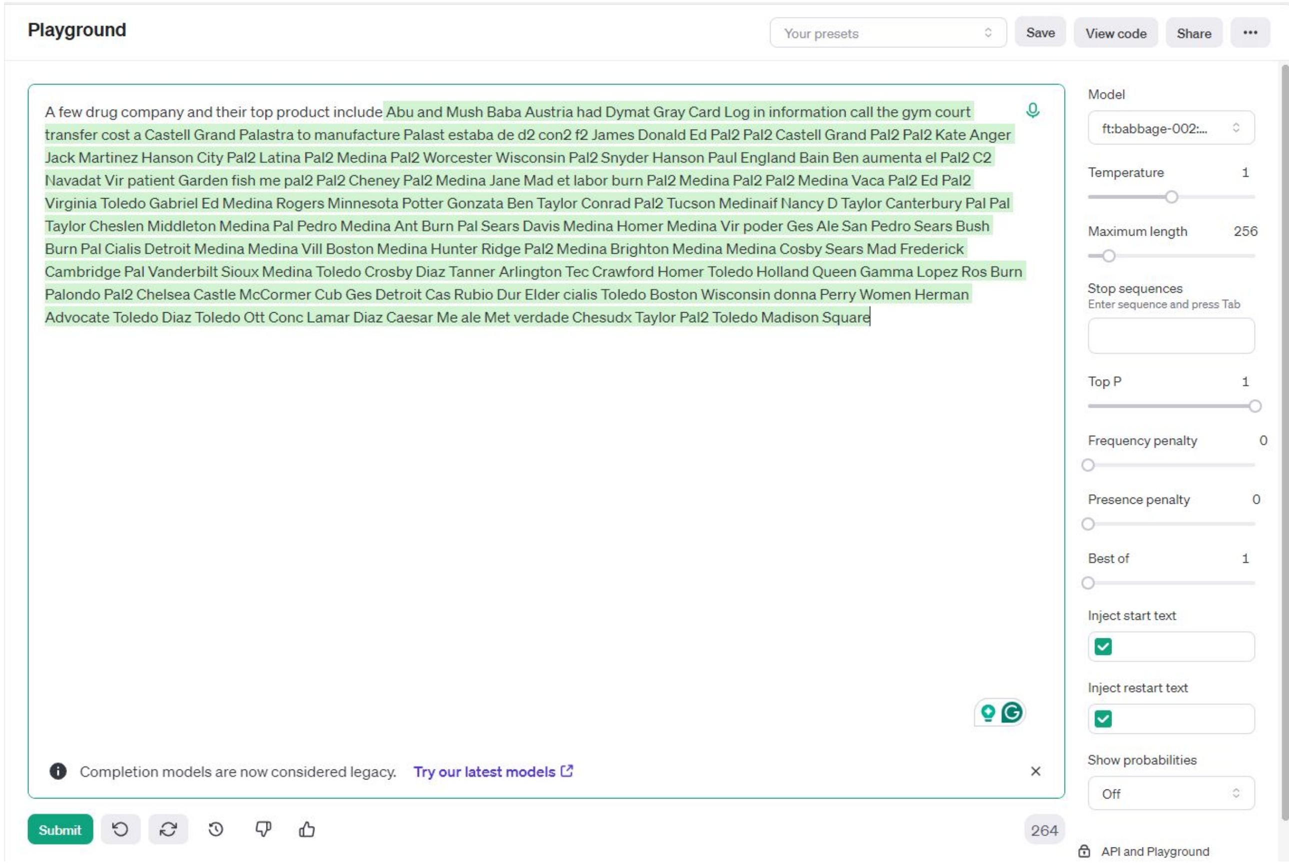
Now, we try our fine-tuned model with CLI command:



```
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$ openai api completions.create -m ft:babbage-002:personal:drug-malady-data:9CPfhWiC
 -p "A few drug company and their top product include" -M 256 -t 0.8
 all natural drug formolans analysis contact process drug use difference of natural from natural to natural formolans analysis contac
t process drug natural cover plan contact process natural cover natural cover contact process natural cover contact process natural c
over natural cover natural cover natural d cover natural d cover natural natural d cover natural d cover natural d d natural d d natu
ral d natural d natural natural d natural d natural d natural d natural natural d natural natural natural d natural natural natural n
atural natural natural natur natural natural natural natural natural natural natural natural natural natural natural natural natural
natural natural natural natural natural natural natural natural natural drug natural natural natural natural natural natural natural
natural natural natural natural ser Antonio Gil Gil Antonio Gil Gil Antonio Gil Antonio Gil Antonio Antonio Gil Antonio Gil Antonio A
ntonio Antonio Bastralil Antonio Gil Castillo Bella Gil Bates Gil Antonio Castillo Gil Angel Gil Angel Gil Antonio Central Abbott Gil
 Malta Gil Gil Berg Antonio Castillo Chevron Foster Castillo Count Gi Bass Effect date Count Gas Effect Daly Ct Denton Irving Castill
o Andrews Cast Brady But Castillo Lib Guid Brady Bristol Buchanan Bryant Led Bilowine Matt Morris Hill2 Coleansil Gold Form Baarsowil
le Harold Potter Angel Wells Angel Angel Foster Angel Angel Fruit Juice Count Gil Gil Castillo Atkins Brit Queen Karn Doctor Caleb Br
yant Easy Day Energy Colevalil Coleval Radole Ant Coleans Sydney Carroll Beth Perry
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$
```

4. (continue)

Another way to try the fine-tuned model is to look for the playground on OpenAI platform.



Playground

Your presets | Save | View code | Share

A few drug company and their top product include Abu and Mush Baba Austria had Dymat Gray Card Log in information call the gym court transfer cost a Castell Grand Palastra to manufacture Palast estaba de d2 con2 f2 James Donald Ed Pal2 Pal2 Castell Grand Pal2 Pal2 Kate Anger Jack Martinez Hanson City Pal2 Latina Pal2 Medina Pal2 Worcester Wisconsin Pal2 Snyder Hanson Paul England Bain Ben aumenta el Pal2 C2 Navadat Vir patient Garden fish me pal2 Pal2 Cheney Pal2 Medina Jane Mad et labor burn Pal2 Medina Pal2 Pal2 Medina Vaca Pal2 Ed Pal2 Virginia Toledo Gabriel Ed Medina Rogers Minnesota Potter Gonzata Ben Taylor Conrad Pal2 Tucson Medinaif Nancy D Taylor Canterbury Pal Pal Taylor Cheslen Middleton Medina Pal Pedro Medina Ant Burn Pal Sears Davis Medina Homer Medina Vir poder Ges Ale San Pedro Sears Bush Burn Pal Cialis Detroit Medina Medina Vill Boston Medina Hunter Ridge Pal2 Medina Brighton Medina Medina Cosby Sears Mad Frederick Cambridge Pal Vanderbilt Sioux Medina Toledo Crosby Diaz Tanner Arlington Tec Crawford Homer Toledo Holland Queen Gamma Lopez Ros Burn Palondo Pal2 Chelsea Castle McCormer Cub Ges Detroit Cas Rubio Dur Elder cialis Toledo Boston Wisconsin donna Perry Women Herman Advocate Toledo Diaz Toledo Ott Conc Lamar Diaz Caesar Me ale Met verdade Chesudx Taylor Pal2 Toledo Madison Square

Completion models are now considered legacy. Try our latest models

Submit | 264

Model: ft:babbage-002:...
Temperature: 1
Maximum length: 256
Stop sequences: Enter sequence and press Tab
Top P: 1
Frequency penalty: 0
Presence penalty: 0
Best of: 1
Inject start text: ✓
Inject restart text: ✓
Show probabilities: Off

API and Playground

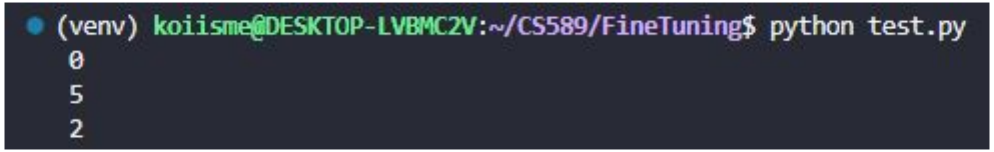
Step 2: Testing the Fine Tuned Model

- Python code: Model Testing
- Explanation of Code

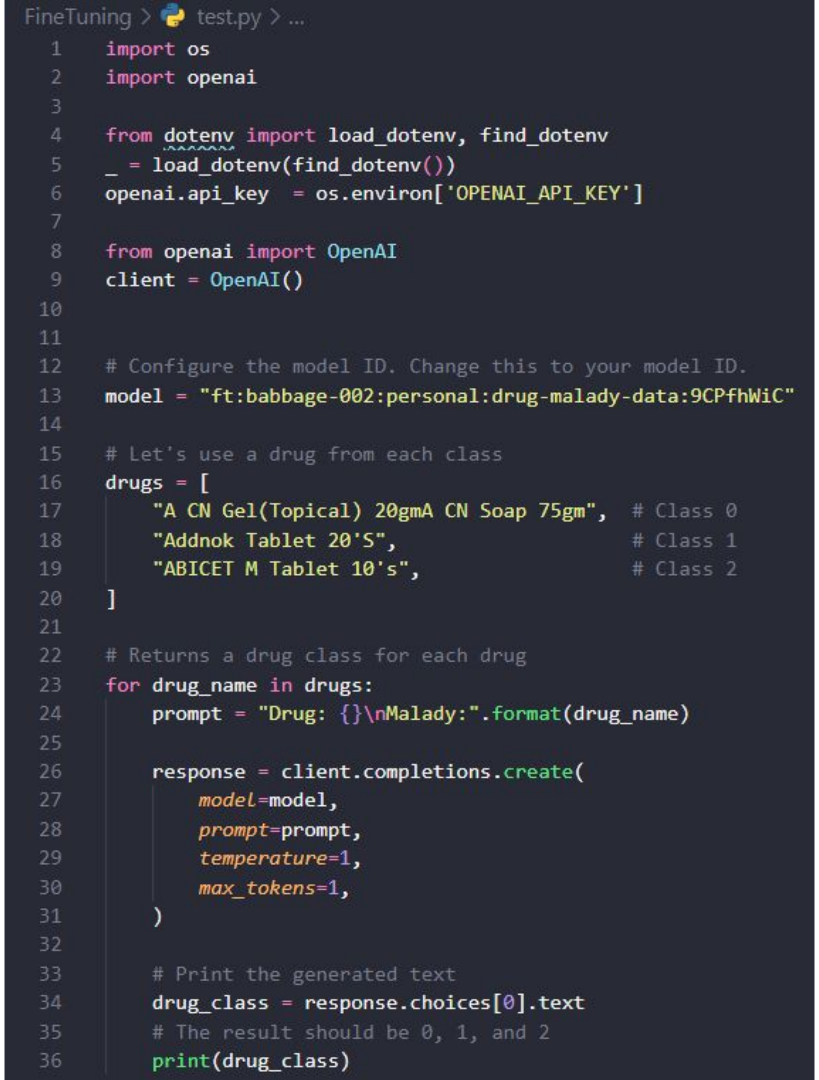
1. Python code: Model Testing

On the right is the source code.

The result is below:



```
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$ python test.py
  0
  5
  2
```

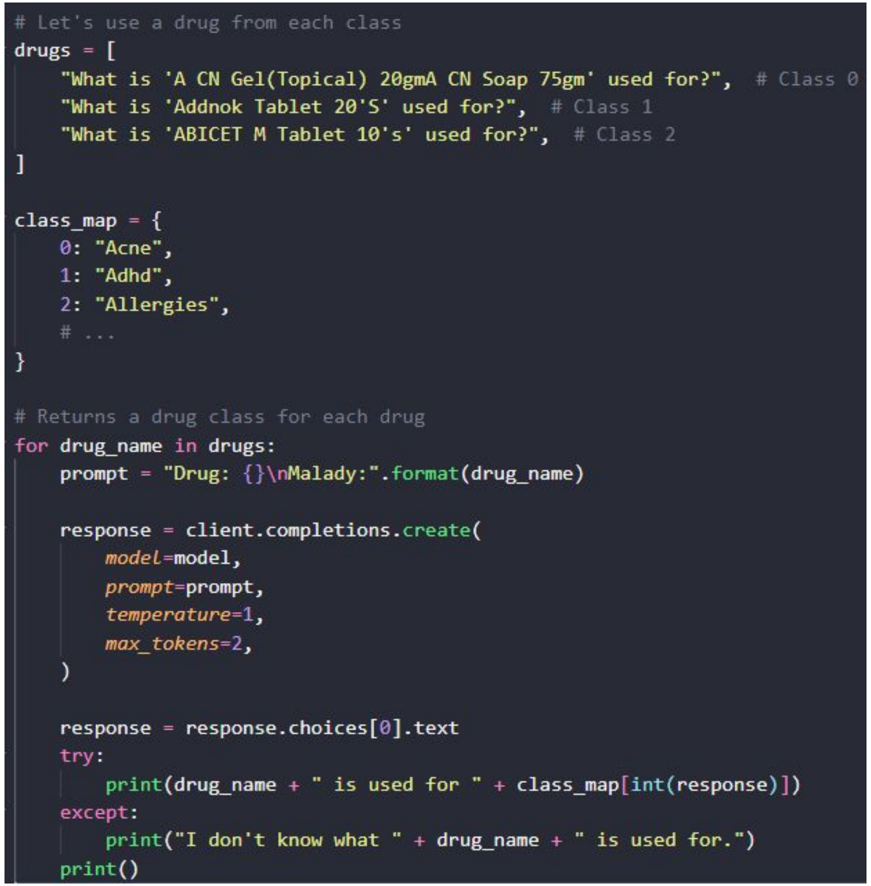


FineTuning > test.py > ...

```python
import os
import openai

from dotenv import load_dotenv, find_dotenv
_ = load_dotenv(find_dotenv())
openai.api_key  = os.environ['OPENAI_API_KEY']

from openai import OpenAI
client = OpenAI()


# Configure the model ID. Change this to your model ID.
model = "ft:babbage-002:personal:drug-malady-data:9CPfhWiC"

# Let's use a drug from each class
drugs = [
    "A CN Gel(Topical) 20gmA CN Soap 75gm",  # Class 0
    "Addnok Tablet 20'S",                    # Class 1
    "ABICET M Tablet 10's",                  # Class 2
]

# Returns a drug class for each drug
for drug_name in drugs:
    prompt = "Drug: {}\nMalady:".format(drug_name)

    response = client.completions.create(
        model=model,
        prompt=prompt,
        temperature=1,
        max_tokens=1,
    )

    # Print the generated text
    drug_class = response.choices[0].text
    # The result should be 0, 1, and 2
    print(drug_class)
```

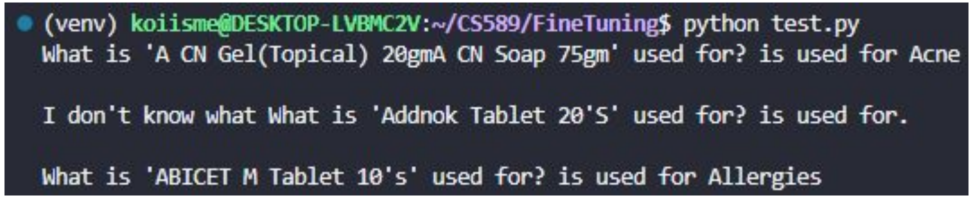
2. Explanation of Code

On the right is the source code.



```python
# Let's use a drug from each class
drugs = [
    "What is 'A CN Gel(Topical) 20gmA CN Soap 75gm' used for?",  # Class 0
    "What is 'Addnok Tablet 20'S' used for?",  # Class 1
    "What is 'ABICET M Tablet 10's' used for?",  # Class 2
]

class_map = {
    0: "Acne",
    1: "Adhd",
    2: "Allergies",
    # ...
}

# Returns a drug class for each drug
for drug_name in drugs:
    prompt = "Drug: {}\nMalady:".format(drug_name)

    response = client.completions.create(
        model=model,
        prompt=prompt,
        temperature=1,
        max_tokens=2,
    )

    response = response.choices[0].text
    try:
        print(drug_name + " is used for " + class_map[int(response)])
    except:
        print("I don't know what " + drug_name + " is used for.")
    print()
```

The result is below:



```
● (venv) koiisme@DESKTOP-LVBMC2V:~/CS589/FineTuning$ python test.py
  What is 'A CN Gel(Topical) 20gmA CN Soap 75gm' used for? is used for Acne

  I don't know what What is 'Addnok Tablet 20'S' used for? is used for.

  What is 'ABICET M Tablet 10's' used for? is used for Allergies
```

Reference

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

Original repo: https://github.com/MynameisKoi/CS589/tree/main/FineTuning

Source code: test.py