AI Enabled Drug Classification system

Using Fine Tune, OpenAI API and data.xls

Table of content

Introduction

- Drug Classification through Machine Learning. In this project, we harness the capabilities of OpenAI's GPT-3.5 to fine-tune a model, utilizing a curated dataset of 2000 drug examples from an Excel file. Our objective is to elevate drug classification, enabling precise categorization based on associated maladies.
- This project is all about teaching a smart computer system to figure out what kind of illness a medicine is meant to treat. First, we collect a bunch of information about different medicines and the illnesses they are supposed to help with. Then, we organize this information in a way that the computer can learn from. Finally, we let the computer use this knowledge to tell us which illness a specific medicine is designed to deal with

Design : Preparing the data and launching the FineTune

Step 1: Preparing the Data

Convert the XLSX data file into JSONL format for fine-tuning the model using Pandas and OpenAI tools. The data is formatted with prompts and completions for drug names and corresponding maladies. The script uses Pandas to transform the data and ensures each completion starts with a whitespace.

Step 2: Command to Prepare Data

Analyze and prepare the data using the OpenAI tools fine_tunes.prepare_data command. This command prompts for splitting the data into training and validation sets.

Step 3: Command to Train the Model

Use the provided command to train the model using fine_tunes.create. Specify parameters such as the training and validation data files, model type (ada), and classification metrics.

Design : Preparing the data and launching the FineTune cont..

Step 4: Checking Job Progress

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

Step 5: Completion of Fine-Tuning

When the fine-tuning job is completed, you'll receive an output confirming the completion, cost, and other details. Use the fine-tuned model for generating completions.

Implementation

1. **Create virtual env or activate the virtual environment**

   Command: “workon <your virtual env name> For example workon chatgpt”

   Command:python3 -m venv venv

   . venv/bin/activate

2. **Install packages :** pip install pandas openpyxl openai==0.28

```
(chatgpt) macs-MacBook-Pro: Fine Tuning-Drug Classification mac$ pip install openpyxl
Collecting openpyxl
  Downloading openpyxl-3.1.2-py2.py3-none-any.whl (249 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 250.0/250.0 kB 5.0 MB/s eta 0:00:00
Collecting et-xmlfile (from openpyxl)
  Using cached 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
```

3. **Process the Dataset:**

```
(chatgpt) macs-MacBook-Pro: Fine Tuning-Drug Classification mac$ python3 app.py
```

Implementation cont..

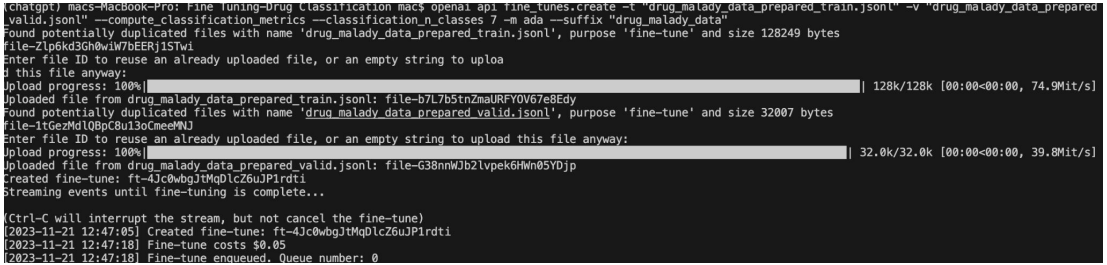
4. Prepare Data for Fine-Tuning:openai tools fine_tunes.prepare_data -f drug_malady_data.jsonl

```
(chatgpt) macs-MacBook-Pro: Fine Tuning-Drug Classification mac$ 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_classification_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 generating 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`. Queue will approximately take half
 an hour per job ahead of you.
```

5. Set Up OpenAI API Key:export OPENAI_API_KEY="your_api_key_here"

Implementation cont..

6. Fine-Tune the Model: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 7 \
-m ada \
--suffix "drug_malady_data"



7. Follow the Fine-Tuning Job:openai api fine_tunes.follow -i <JOB ID>

Here job ID is your fine tune: ft-<unique key> will display on the last step

```
(chatgpt) macs-MacBook-Pro: Fine Tuning-Drug Classification mac$ openai api fine_tunes.follow -i ft-4Jc0wbgJtMqDlcZ6uJP1rdti
[2023-11-21 12:47:05] Created fine-tune: ft-4Jc0wbgJtMqDlcZ6uJP1rdti
[2023-11-21 12:47:18] Fine-tune costs $0.05
[2023-11-21 12:47:18] Fine-tune enqueued. Queue number: 0
[2023-11-21 12:57:24] Fine-tune started
[2023-11-21 13:02:42] Completed epoch 1/4
[2023-11-21 13:07:52] Completed epoch 2/4

Job complete! Status: succeeded 🎉
Try out your fine-tuned model:

openai api completions.create -m ada:ft-personal:drug-malady-data-2023-11-21-21-18-39 -p <YOUR_PROMPT>
(chatgpt) macs-MacBook-Pro: Fine Tuning-Drug Classification mac$
```

8.Run the Test Script:python3 test.py

Test

```
(chatgpt) macs-MacBook-Pro: Fine Tuning-Drug Classification mac$ python3 test.py
What is 'A CN Gel(Topical) 20gmA CN Soap 75gm' used for? is used for Acne

What is 'Addnok Tablet 20'S' used for? is used for Adhd

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

(chatgpt) macs-MacBook-Pro: Fine Tuning-Drug Classification mac$
```

Enhancement

- Advanced Feature Integration:Explore the incorporation of additional features, such as dosage information, patient demographics, or side effects, to enhance the model's classification accuracy and broaden its scope.
- Real-Time Classification:Develop mechanisms for real-time drug classification, enabling immediate insights and responses in dynamic healthcare scenarios.
- Continuous Model Refinement:Implement a system for continuous model refinement through periodic updates, incorporating new drug information and maladies to ensure the model stays current and effective.

Conclusion

The project utilizes OpenAI's GPT-3.5 to fine-tune a model for precise drug classification based on a curated dataset of 2000 drug examples. The goal is to teach the system to accurately categorize medicines according to the illnesses they address, showcasing the potential for advanced machine learning applications in healthcare. This project lays the foundation for more efficient and refined drug classification systems.

Github link

https://github.com/DKruti/Machine-Learning/tree/master/Generative%20AI/FINE%20TUNE/2000%20Drug%20Examples

References

https://hc.labnet.sfbu.edu/~henry/sfbu/course/generative_ai/Advanced_Fine_Tuning_Drug_Classification/slide/exercise_Advanced_Fine_Tuning_Drug_Classification.html