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

By Lianwei Deng(19874)

Table of contents

01

Preparing the Data and Launching the Fine Tuning

Convert the XLSX data file into JSONL format for fine-tuning the model using Pandas and OpenAI tools.

Write a python file to convert the XLSX data:

It will have this format:

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

```
# 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)
```

02 Command to Prepare Data

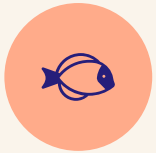
Analyze and prepare the data using the OpenAI tools fine_tunes.prepare_data command.

$openai tools fine_tunes.prepare_data -f drug_malady_data.jsonl

```
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_met
rics --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 com
pletions, 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 ap
proximately take half an hour per job ahead of you.
```

03 Command to Train the Model

export
OPENAI_API_KEY=xxxxxxxx



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"

Notice: classification_n_classes is getting from the xlsx file

04 Get the result

After running step3 command, you can check job process by using the following command:

openai api fine_tunes.follow -i <JOB ID>

```
[2023-11-20 18:58:34] Created fine-tune: ft-Ud7e3YAlwTO3IlSl7eWnHAz0
[2023-11-20 19:04:48] Fine-tune costs $0.05
[2023-11-20 19:04:48] Fine-tune enqueued. Queue number: 0
[2023-11-20 19:04:50] Fine-tune started
[2023-11-20 19:10:10] Completed epoch 1/4
[2023-11-20 19:20:30] Completed epoch 3/4
[2023-11-20 19:26:06] Uploaded model: ada:ft-personal:drug-malady-data-2023-11-21-03-26-06
[2023-11-20 19:26:07] Uploaded result file: file-QmWGslMVmDmrr5lbEa5OmLd2
[2023-11-20 19:26:07] Fine-tune succeeded

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

openai api completions.create -m ada:ft-personal:drug-malady-data-2023-11-21-03-26-06 -p <YOUR_PROMPT>
```

05 Conclusion

Structured Data Transformation:

- Conversion to JSONL format.
- Use of unique identifiers for maladies.

Python Script and Pandas:

- Efficient data preparation.
- Adaptable to various dataset sizes.

CLI Commands for Insight:

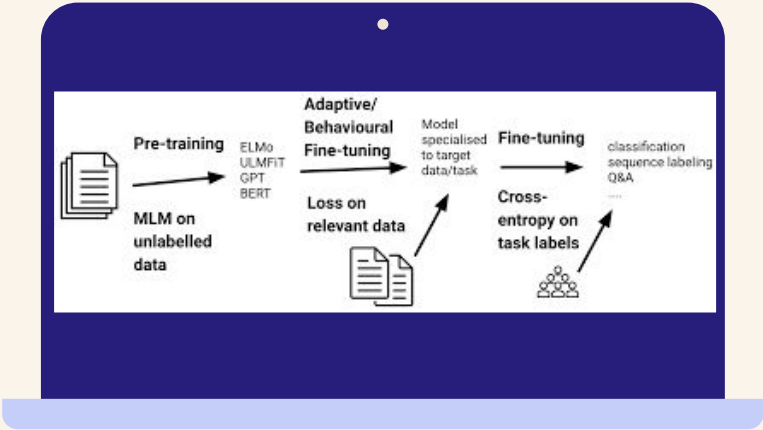
- Analyzing data
- Using fine_tunes.prepare_data.
- Choosing model and monitoring progress.

Empowering Customization:

- Fine-tuning results in a tailored, powerful model.
- Iterative process for targeted applications.

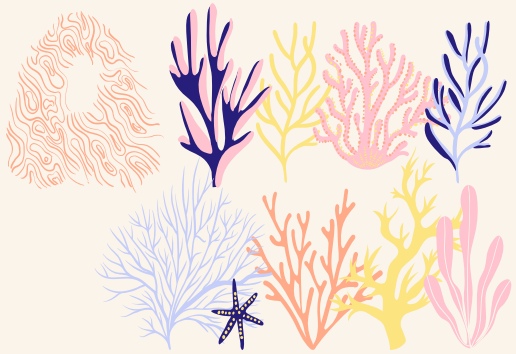
Enhancing Model Performance:

- Leveraging advanced language models.
- Optimizing for specific use cases.



References

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



Thanks!

Do you have any questions?

youremail@freepik.com
+34 654 321 432
yourwebsite.com