Fine-tuning:
Drug Classification

Zhiyu Zhang

Project Description

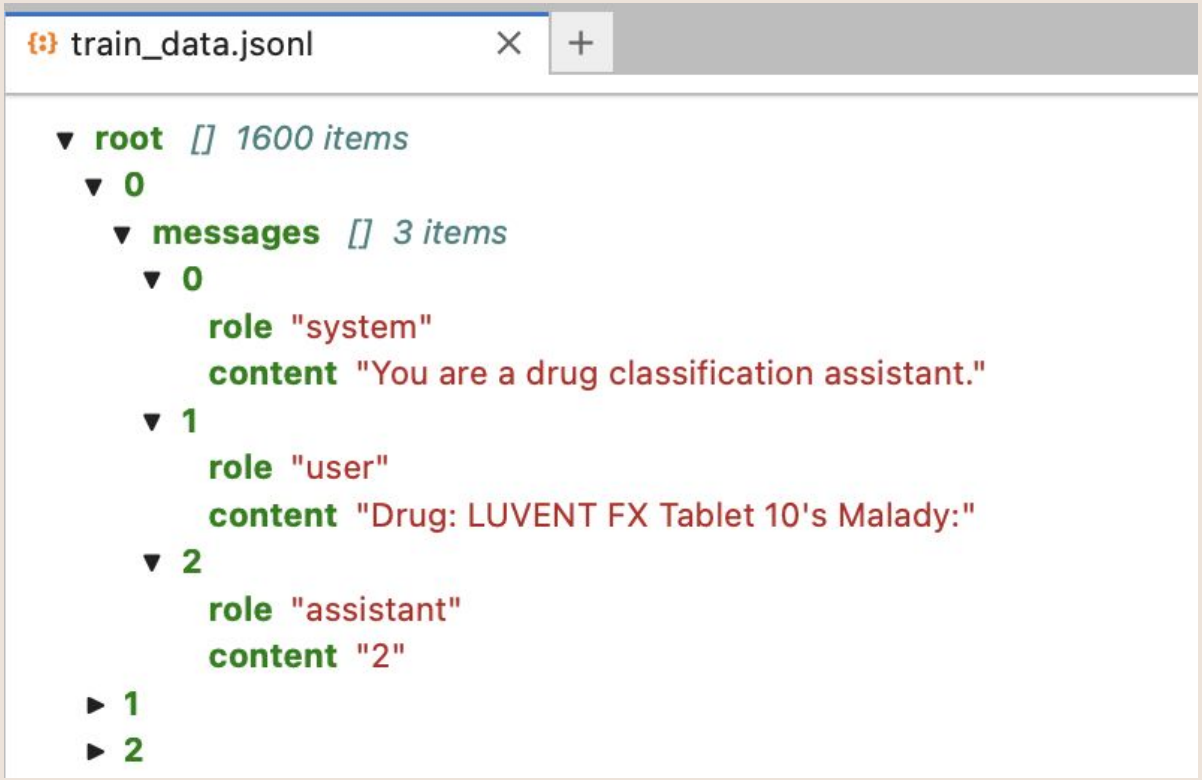
GPT-based drug classification model

- **Data Preparation**:
  - Convert data from an Excel file into JSONL format
  - Create training and validation files
- **Fine-Tuning**:
  - Use `gpt-4o-mini-2024-07-18` as the base model
- **Making Prediction**:
  - Classify drugs into predefined classes

Data Preparation

- **Load data** from `Medicine_description.xlsx`.
- **Split the data** into training and validation datasets (80% training, 20% validation).
- **Convert the data** into a JSONL format with the updated message-based structure.
- **Save the datasets** as `train_data.jsonl` and `val_data.jsonl`.

Data Preparation



Fine-tuning the Model

- Load OpenAI API Configuration.
- Upload the training and validation datasets.
- Fine-tune the model.
- Retrieve the fine-tuned model ID.
- With **2000 data entries**, the training takes **~43 minutes** and cost **$0.58**.

Fine-tuning the Model

```python
# Create a fine-tuned model
fine_tune_job = client.fine_tuning.jobs.create(
  training_file=train_data.id,
  model="gpt-4o-mini-2024-07-18",
  validation_file=val_data.id,
  suffix="drug-classifier"
)
```

```python
# Print the fine-tuning job details
print(f"Fine-tuning job created with ID: {fine_tune_job.id}")
updated_job = client.fine_tuning.jobs.retrieve(fine_tune_job.id)
print(f"Fine-tuned Model ID: {updated_job.fine_tuned_model}")
```

```
Fine-tuning job created with ID: ftjob-9w2w7l[redacted]
Fine-tuned Model ID: ft:gpt-4o-mini-2024-07-18:personal:drug-classifier:[redacted]
```

Testing the Model

```
# Use the fine-tuned model
drugs = [
    "A CN Gel(Topical) 20gmA CN Soap 75gm",  # Class 0
    "Addnok Tablet 20'S",                    # Class 1
    "ABICET M Tablet 10's",                  # Class 2
]

for drug_name in drugs:
    prompt = "Drug: {}\nMalady:".format(drug_name)
    completion = client.chat.completions.create(
      model=updated_job.fine_tuned_model,
      messages=[
        {"role": "user", "content": prompt}
      ])
    print(completion.choices[0].message.content)
```

```
0
2
2
```

Testing the Model

Map numerical classes to malady names for readability:

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

# Returns a drug class for each drug
for drug in drugs:
    drug_name = drug.split("'")[1]
    prompt = "Drug: {}\nMalady:".format(drug)
    completion = client.chat.completions.create(
      model=updated_job.fine_tuned_model,
      messages=[
        {"role": "user", "content": prompt}
      ])
    response = completion.choices[0].message.content

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

```
A CN Gel(Topical) 20gmA CN Soap 75gm is used for Acne.

Addnok Tablet 20 is used for Allergies.

ABICET M Tablet 10 is used for Allergies.
```

Limitation

- The model's accuracy depends on the quality and size of the training dataset.
- Unseen drugs or ambiguous inputs might lead to incorrect classifications or failures.

Thanks.

Zhiyu Zhang