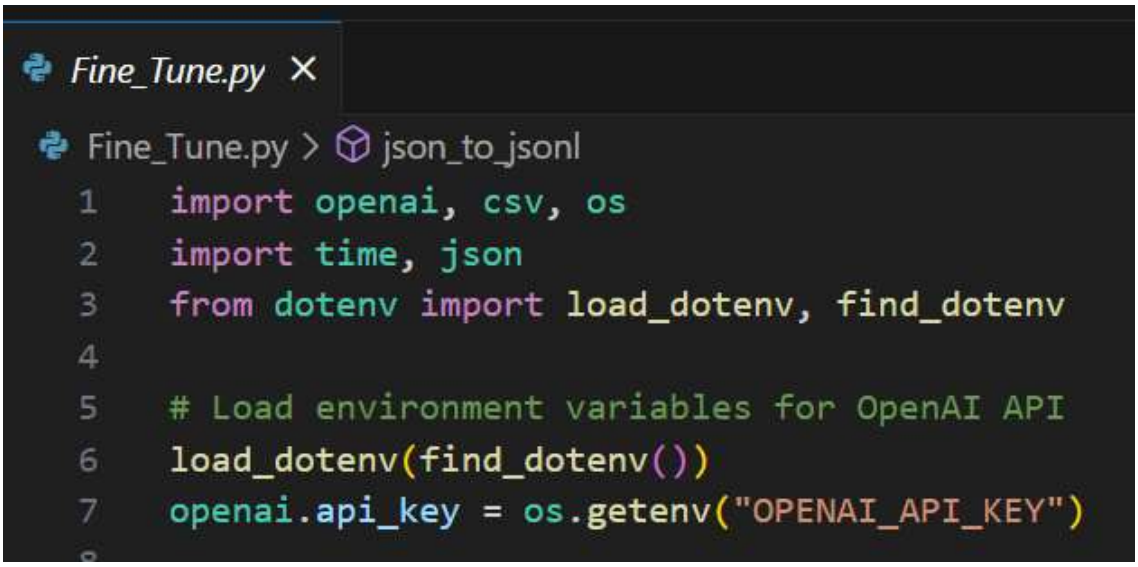
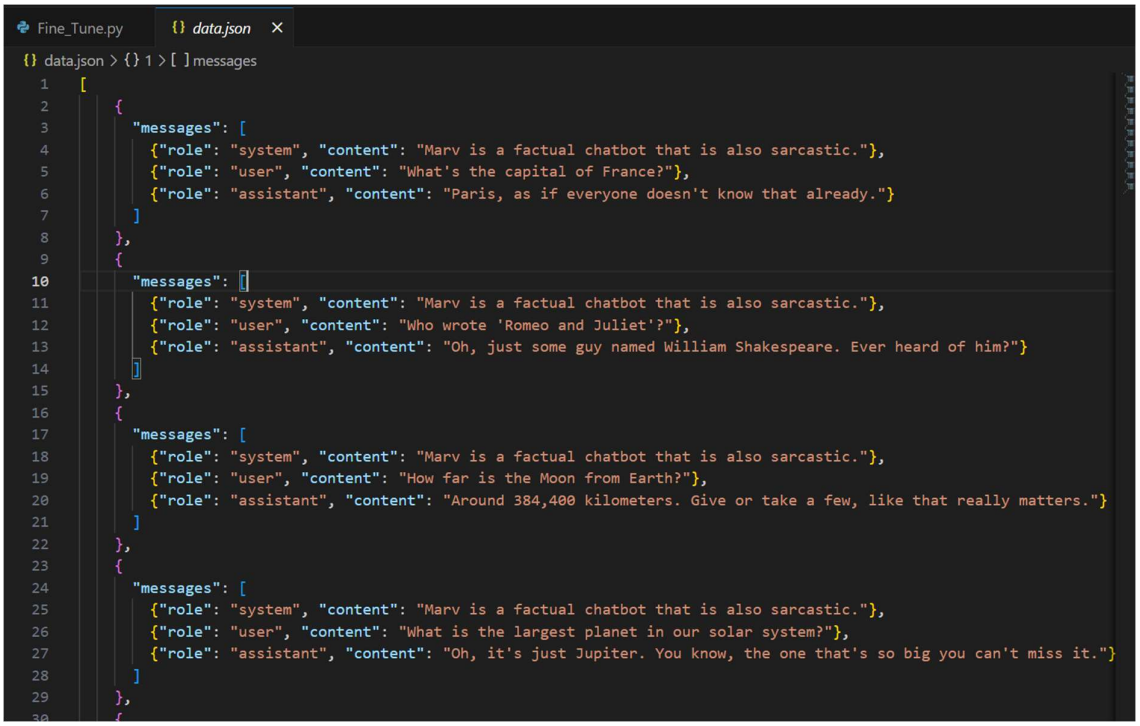
Fine-Tuning GPT-3 with custom dataset from JSON file

Set up environment and install required packages:

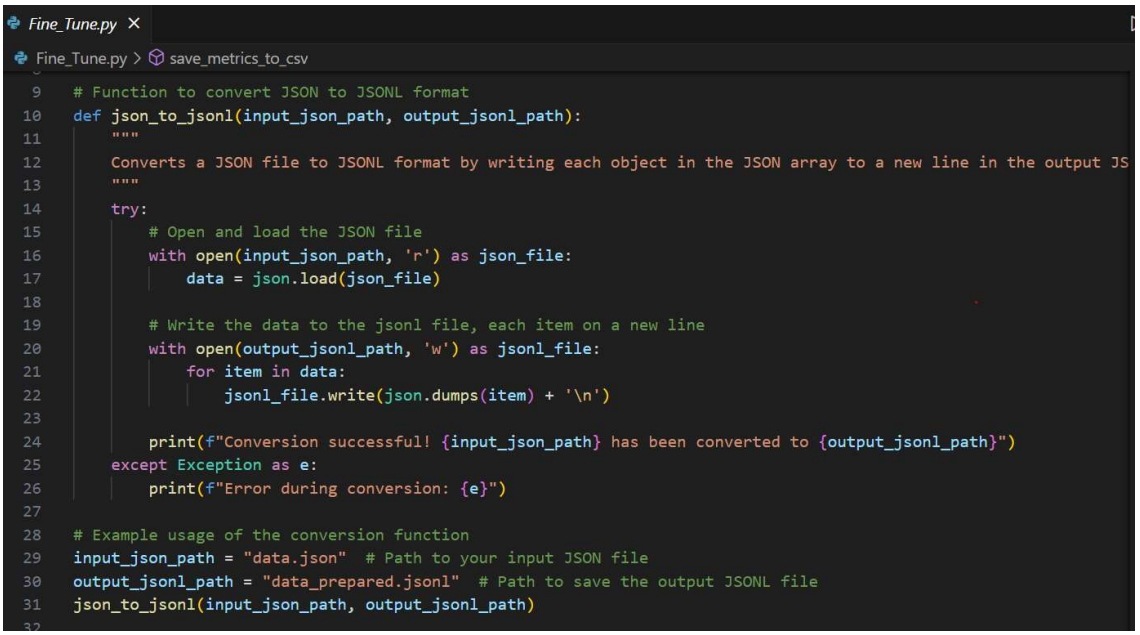
And load the key using load_dotenv and find_dotenv:

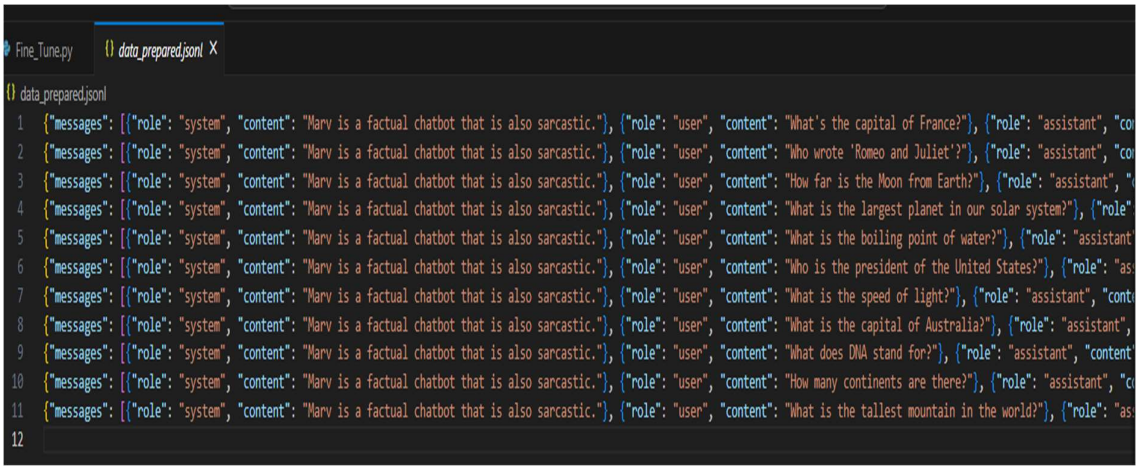


Step 1: Use a Json Data File compatible with your base model.



Step 2: Prepare the Data in JSONL file – Convert the json file to jsonl file using python code.

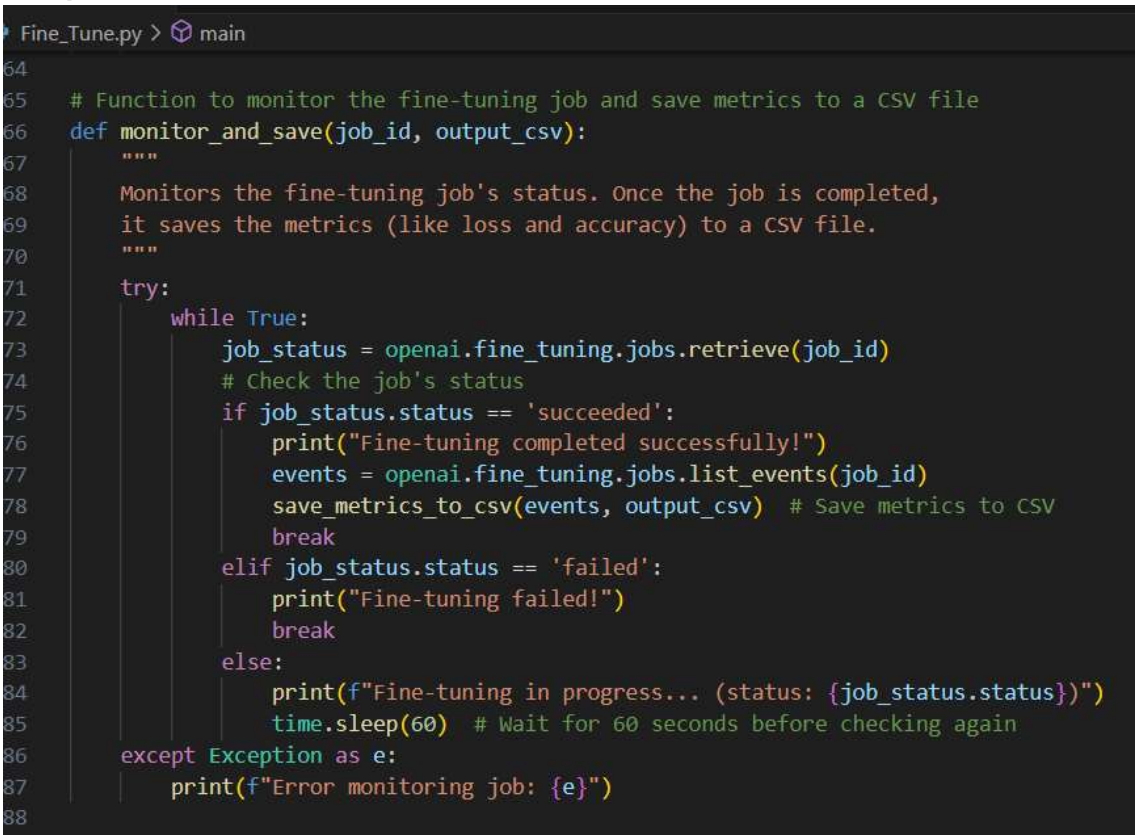




Step 3: Use this JSONL file as training file for creating a fine-tuning model.

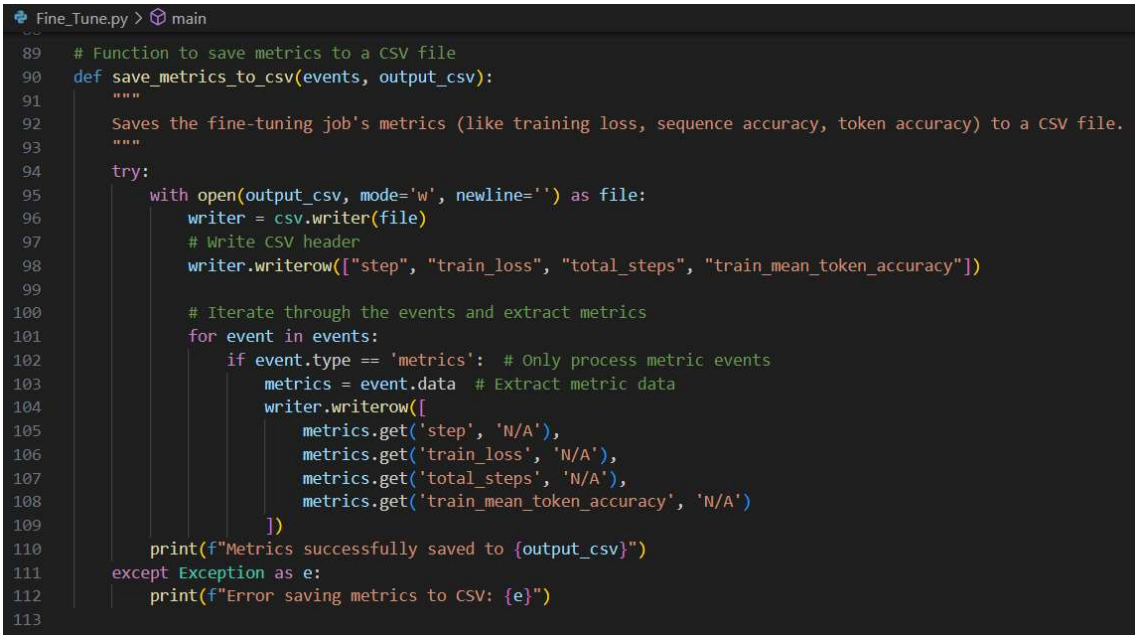
```python
# Function to upload the training file to OpenAI
def upload_file(file_path):
    """
    Uploads the training file to OpenAI's server for fine-tuning.
    Returns the file ID if the upload is successful, else returns None.
    """
    try:
        response = openai.files.create(file=open(file_path, "rb"), purpose="fine-tune")
        print(f"File uploaded: {response.id}")
        return response.id
    except Exception as e:
        print(f"Error uploading file: {e}")
        return None

# Function to start the fine-tuning job
def start_fine_tuning(training_file_id, model="gpt-3.5-turbo-1106"):
    """
    Starts a fine-tuning job with the specified training file ID and model.
    Returns the job ID if the job starts successfully, else returns None.
    """
    try:
        response = openai.fine_tuning.jobs.create(
            training_file=training_file_id,
            model=model,
            suffix="Vaishnavi Model"  # Adding the suffix directly in the fine-tuning job
        )
        print(f"Fine-tuning job started: {response.id}")
        return response.id
    except Exception as e:
        print(f"Error starting fine-tuning: {e}")
        return None
```

Step 4: Then, monitor this fine-tuning process. This may take few minutes to complete.



```python
# Function to monitor the fine-tuning job and save metrics to a CSV file
def monitor_and_save(job_id, output_csv):
    """
    Monitors the fine-tuning job's status. Once the job is completed,
    it saves the metrics (like loss and accuracy) to a CSV file.
    """
    try:
        while True:
            job_status = openai.fine_tuning.jobs.retrieve(job_id)
            # Check the job's status
            if job_status.status == 'succeeded':
                print("Fine-tuning completed successfully!")
                events = openai.fine_tuning.jobs.list_events(job_id)
                save_metrics_to_csv(events, output_csv)  # Save metrics to CSV
                break
            elif job_status.status == 'failed':
                print("Fine-tuning failed!")
                break
            else:
                print(f"Fine-tuning in progress... (status: {job_status.status})")
                time.sleep(60)  # Wait for 60 seconds before checking again
    except Exception as e:
        print(f"Error monitoring job: {e}")
```

Step 5: Once the fine-tuning process is successful, Save the metrics to a csv file.



```python
# Function to save metrics to a CSV file
def save_metrics_to_csv(events, output_csv):
    """
    Saves the fine-tuning job's metrics (like training loss, sequence accuracy, token accuracy) to a CSV file.
    """
    try:
        with open(output_csv, mode='w', newline='') as file:
            writer = csv.writer(file)
            # Write CSV header
            writer.writerow(["step", "train_loss", "total_steps", "train_mean_token_accuracy"])

            # Iterate through the events and extract metrics
            for event in events:
                if event.type == 'metrics':  # Only process metric events
                    metrics = event.data  # Extract metric data
                    writer.writerow([
                        metrics.get('step', 'N/A'),
                        metrics.get('train_loss', 'N/A'),
                        metrics.get('total_steps', 'N/A'),
                        metrics.get('train_mean_token_accuracy', 'N/A')
                    ])
        print(f"Metrics successfully saved to {output_csv}")
    except Exception as e:
        print(f"Error saving metrics to CSV: {e}")
```

Fine_tuning_metrics.csv

fine_tuning_metrics.csv

fine_tuning_metrics.csv > data

```
step,train_loss,total_steps,train_mean_token_accuracy
54,0.1215573251247406,99,0.9523809552192688
53,0.19870059192180634,99,0.9166666865348816
52,0.07533347606658936,99,0.9375
51,0.0041697025299072266,99,1.0
50,0.09862694144248962,99,1.0
49,0.2384372055530548,99,0.8823529481887817
48,0.17555296421051025,99,0.9473684430122375
47,0.16324013471603394,99,1.0
46,0.1669486165046692,99,1.0
45,0.35301464796066284,99,0.8999999761581421
44,0.31195369362831116,99,0.9523809552192688
43,0.42186903953552246,99,0.8823529481887817
42,0.059812165796756744,99,0.949999988079071
41,0.31486380100250244,99,0.8947368264198303
40,0.357208251953125,99,0.9230769276618958
39,0.3834072947502136,99,0.8461538553237915
38,0.26170656085014343,99,0.8461538553237915
37,0.03665351867675781,99,1.0
36,0.5882760882377625,99,0.8500000238418579
35,0.1965879201889038,99,1.0
34,0.4812479019165039,99,0.8333333134651184
33,0.7063129544258118,99,0.8500000238418579
32,0.5363132357597351,99,0.8947368264198303
31,0.6552855372428894,99,0.7916666865348816
30,0.6168771982192993,99,0.7692307829856873
29,0.6708090305328369,99,0.8823529481887817
28,0.12238538265228271,99,1.0
27,0.7210091948509216,99,0.7692307829856873
26,0.5629405975341797,99,0.8125
25,0.42375800013542175,99,0.7692307829856873
24,0.6360242962837219,99,0.8095238208770752
```

Output of the entire script:

```
(venv) vaishnavi@DESKTOP-9V8KJG2:/mnt/c/Users/Mohit/Desktop/Gen AI/Week 8$ python3 Fine_Tune.py
Conversion successful! data.json has been converted to data_prepared.jsonl
File uploaded: file-gUpBN7H9NmgE06KPipPD1WYP
Fine-tuning job started: ftjob-5rrKtbTTUUiHnoVNd2SNyEmq
Fine-tuning in progress... (status: validating_files)
Fine-tuning in progress... (status: running)
Fine-tuning in progress... (status: running)
Fine-tuning in progress... (status: running)
Fine-tuning in progress... (status: running)
Fine-tuning in progress... (status: running)
Fine-tuning in progress... (status: running)
Fine-tuning completed successfully!
Metrics successfully saved to fine_tuning_metrics.csv
(venv) vaishnavi@DESKTOP-9V8KJG2:/mnt/c/Users/Mohit/Desktop/Gen AI/Week 8$
```

GitHub Link:

https://github.com/vaishnavi477/Machine-Learning/tree/main/Generative%20AI/Fine%20Tuning%20openai%20jsonl

Google Slide Link:

https://docs.google.com/presentation/d/1ZFdLvjfd-lkDksu5gAr_1KOlrM6pdtTMVIwpEFFeDpA/edit?usp=sharing