Double-click (or enter) to edit

```
import json
from datetime import datetime
import pandas as pd
```

```
!pip install transformers
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.11/dist-packages (4.51.3)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from transformers) (3.18.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.30.0 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.30.2)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/dist-packages (from transformers) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from transformers) (2.32.3)
Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.21.1)
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.5.3)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.30.0->transform
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.30.0-
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (2025.1.31)
```

```
!huggingface-cli login
```

```
    _|    _|  _|    _|    _|_|_|    _|_|_|  _|_|_|  _|      _|    _|_|_|      _|_|_|_|    _|_|      _|_|_|  _|_|_|_|
    _|    _|  _|    _|  _|        _|          _|    _|_|    _|  _|            _|        _|    _|  _|        _|
    _|_|_|_|  _|    _|  _|  _|_|  _|  _|_|    _|    _|  _|  _|  _|  _|_|      _|_|_|    _|_|_|_|  _|        _|_|_|
    _|    _|  _|    _|  _|    _|  _|    _|    _|    _|    _|_|  _|    _|      _|        _|    _|  _|        _|
    _|    _|    _|_|      _|_|_|    _|_|_|  _|_|_|  _|      _|    _|_|_|      _|        _|    _|    _|_|_|  _|_|_|_|

    A token is already saved on your machine. Run `huggingface-cli whoami` to get more information or `huggingface-cli logout` if you wa
    Setting a new token will erase the existing one.
    To log in, `huggingface_hub` requires a token generated from https://huggingface.co/settings/tokens .
Enter your token (input will not be visible):
Add token as git credential? (Y/n) y
Token is valid (permission: read).
The token `AI_Agents_AV` has been saved to /root/.cache/huggingface/stored_tokens
Cannot authenticate through git-credential as no helper is defined on your machine.
You might have to re-authenticate when pushing to the Hugging Face Hub.
Run the following command in your terminal in case you want to set the 'store' credential helper as default.

git config --global credential.helper store

Read https://git-scm.com/book/en/v2/Git-Tools-Credential-Storage for more details.
Token has not been saved to git credential helper.
Your token has been saved to /root/.cache/huggingface/token
Login successful.
The current active token is: `AI_Agents_AV`
```

```
import json
import re
import pandas as pd

# Read the CSV file into a Pandas DataFrame
df = pd.read_csv("/content/drive/MyDrive/test.csv")

def clean_text(text):
    # Convert the input to a string if it's not already
    text = str(text)  # This will handle integers and other data types

    # Remove HTML tags
    text = re.sub(r"<.*?>", "", text)

    # Remove URLs
    text = re.sub(r"https?://\S+|www\.\S+", "", text)

    # Remove special characters (keep only alphanumeric and whitespace)
    text = re.sub(r"[^\w\s]", "", text)
```

```
    # Handle missing values (replace with empty string)
    text = text if text else ""

    return text

def preprocess_text(text):
  return clean_text(text)

# Convert to JSONL format with preprocessing:
with open("/content/drive/MyDrive/fine_tuning_data.jsonl", "w") as f:
    for index, row in df.iterrows():
        # Assuming your CSV has columns named "instruction" and "output"
        instruction = row["text"]
        output = row["label"]

        # Check for missing values
        if pd.isnull(instruction) or pd.isnull(output):
            continue  # Skip this row if either is missing

        json_object = {
            "messages": [
                {"role": "system", "content": "You are a helpful and harmless AI assistant."},
                {"role": "user", "content": preprocess_text(instruction)},
                {"role": "assistant", "content": preprocess_text(output)},
            ]
        }
        f.write(json.dumps(json_object) + "\n")
```

```
from transformers import AutoModelForCausalLM, AutoTokenizer

model_id = "openai-community/gpt2"  # Repo ID

tokenizer = AutoTokenizer.from_pretrained(model_id)
model = AutoModelForCausalLM.from_pretrained(model_id)

# ... (Use the model and tokenizer for inference or fine-tuning) ...
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secre
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
```

```
import json

def extract_text_from_jsonl(file_path):
  text_data = []
  with open("/content/drive/MyDrive/fine_tuning_data.jsonl", "r") as f:
    for line in f:
                data = json.loads(line)
                text_data.append(data["messages"][1]["content"] + " " + data["messages"][2]["content"])  # Concatenate user and assistant m
  return text_data

text_data = extract_text_from_jsonl("/content/drive/MyDrive/fine_tuning_data.jsonl")
```

```
!pip install -q bitsandbytes accelerate transformers peft
!pip install -U bitsandbytes
!pip install datasets transformers
```

Show hidden output

```
from transformers import TrainingArguments, Trainer, AutoModelForCausalLM
from transformers import AutoTokenizer
from datasets import load_dataset, Dataset
import torch
from peft import LoraConfig, get_peft_model

# Load the tokenizer
tokenizer = AutoTokenizer.from_pretrained("openai-community/gpt2")
tokenizer.pad_token = tokenizer.eos_token  # Set pad token
```

```
# Load the base model with bitsandbytes for quantization
model = AutoModelForCausalLM.from_pretrained(
    "openai-community/gpt2",
    load_in_8bit=True,  # Enable 8-bit quantization
    device_map="auto",  # Automatically assign to the appropriate device
)


# Define the LoRA configuration
lora_config = LoraConfig(
    r=8,  # Rank of the LoRA matrices
    lora_alpha=16,  # Scaling factor for the LoRA matrices
    lora_dropout=0.05,  # Dropout probability for the LoRA layers
    bias="none",  # Bias type for the LoRA layers
    task_type="CAUSAL_LM",  # Task type for fine-tuning
    fan_in_fan_out=True # Explicitly setting fan_in_fan_out to True to suppress the warning
)

# Apply LoRA to the base model
model = get_peft_model(model, lora_config)
model.print_trainable_parameters()  # Print the number of trainable parameters

# Define training arguments
training_args = TrainingArguments(
    output_dir="./results",
    per_device_train_batch_size=4,  # Reduce batch size if needed
    per_device_eval_batch_size=4,  # Reduce batch size if needed
    num_train_epochs=3,
    logging_dir="./logs",
    fp16=True,  # Enable mixed precision training if supported by your hardware
    gradient_accumulation_steps=4,  # Increase gradient accumulation steps if needed
)

# Load the text data you extracted
# Assuming 'text_data' is a list of strings from your 'extract_text_from_jsonl' function
# Create a Hugging Face Dataset from your text data
train_dataset = Dataset.from_dict({"text": text_data})

def tokenize_function(examples):
    return tokenizer(examples["text"], padding="max_length", truncation=True)

# Tokenize the dataset
tokenized_datasets = train_dataset.map(tokenize_function, batched=True)


# Access the 'train' split of the dataset
train_dataset = tokenized_datasets

# Add the labels to the inputs
def prepare_inputs_for_training(examples):
    examples["labels"] = [[-100] + x[:-1] for x in examples["input_ids"]]
    return examples

train_dataset = train_dataset.map(
    prepare_inputs_for_training,
    batched=True,
    remove_columns=["text"],
)

# Create the Trainer and start training
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
)

trainer.train()

# Save the model
trainer.save_model("./fine_tuned_gpt2")
```

```
The `load_in_4bit` and `load_in_8bit` arguments are deprecated and will be removed in the future versions. Please, pass a `BitsAndBytesC
trainable params: 294,912 || all params: 124,734,720 || trainable%: 0.2364
Map: 100%                                    2000/2000 [00:00<00:00, 2288.21 examples/s]
Map: 100%                                    2000/2000 [00:01<00:00, 1425.13 examples/s]
No label_names provided for model class `PeftModelForCausalLM`. Since `PeftModel` hides base models input arguments, if label_names is n
/usr/local/lib/python3.11/dist-packages/bitsandbytes/autograd/_functions.py:315: UserWarning: MatMul8bitLt: inputs will be cast from tor
  warnings.warn(f"MatMul8bitLt: inputs will be cast from {A.dtype} to float16 during quantization")
[353/375 14:39 < 00:55, 0.40 it/s, Epoch 2.82/3]
Step  Training Loss
[375/375 15:37, Epoch 3/3]
Step  Training Loss
```

```
import pandas as pd

df_validation = pd.read_csv("/content/drive/MyDrive/validation.csv")

# Assuming your CSV has columns named "text" (for prompts) and "label" (for expected outputs)
prompts = df_validation["text"].tolist()
expected_outputs = df_validation["text"].tolist()


from transformers import AutoModelForCausalLM, AutoTokenizer

model_id = "/content/fine_tuned_gpt2"
tokenizer = AutoTokenizer.from_pretrained("openai-community/gpt2")  # Load the tokenizer using the original model ID
model = AutoModelForCausalLM.from_pretrained(model_id)

# Set the padding token after loading the tokenizer
tokenizer.pad_token = tokenizer.eos_token

inputs = tokenizer(prompts, return_tensors="pt", padding=True, truncation=True)


from transformers import pipeline

generator = pipeline("text-generation", model=model, tokenizer=tokenizer)

generated_outputs = []
for prompt in prompts:
    generated_output = generator(prompt, max_length=100, num_return_sequences=1)[0]['generated_text']  # Adjust parameters as needed
    generated_outputs.append(generated_output)
```

```
Device set to use cuda:0
Truncation was not explicitly activated but `max_length` is provided a specific value, please use `truncation=True` to explicitly trunca
```

```
!pip install nltk  # Install NLTK if you haven't already
import nltk
from nltk.translate.bleu_score import sentence_bleu

bleu_scores = []
for generated, expected in zip(generated_outputs, expected_outputs):
    # Convert 'expected' to a string before splitting
    bleu_score = sentence_bleu([str(expected).split()], generated.split())  # Calculate BLEU score
    bleu_scores.append(bleu_score)

average_bleu = sum(bleu_scores) / len(bleu_scores)
print(f"Average BLEU score: {average_bleu}")
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.11/dist-packages (3.9.1)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from nltk) (8.1.8)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.11/dist-packages (from nltk) (2024.11.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from nltk) (4.67.1)
/usr/local/lib/python3.11/dist-packages/nltk/translate/bleu_score.py:577: UserWarning:
The hypothesis contains 0 counts of 4-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
/usr/local/lib/python3.11/dist-packages/nltk/translate/bleu_score.py:577: UserWarning:
The hypothesis contains 0 counts of 2-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
```

```
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
/usr/local/lib/python3.11/dist-packages/nltk/translate/bleu_score.py:577: UserWarning:
The hypothesis contains 0 counts of 3-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
Average BLEU score: 0.9691038014873151
```

```
!pip install rouge-score
from rouge_score import rouge_scorer

scorer = rouge_scorer.RougeScorer(['rouge1', 'rouge2', 'rougeL'], use_stemmer=True)

# Calculate and print ROUGE scores for all generated outputs
for generated_output, expected_output in zip(generated_outputs, expected_outputs):
  scores = scorer.score(generated_output, str(expected_output))  # Convert expected_output to string
  print(f"Generated Output: {generated_output}")
  print(f"Expected Output: {expected_output}")
  print(f"ROUGE Scores: {scores}")
  print("-" * 20)  # Print a separator between examples
```

```
--------------------
Generated Output: i had been feeling fabulous and full of energy but easter weekend wiped me out and i havent been able to recover
Expected Output: i had been feeling fabulous and full of energy but easter weekend wiped me out and i havent been able to recover
ROUGE Scores: {'rouge1': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'rouge2': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'r
--------------------
Generated Output: i feel im supposed to hate dams amp all the control of nature that they represent but sometimes they really are the
Expected Output: i feel im supposed to hate dams amp all the control of nature that they represent but sometimes they really are the
ROUGE Scores: {'rouge1': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'rouge2': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'r
--------------------
Generated Output: i feel like i got to know her a bit and what i did get to know i really liked
Expected Output: i feel like i got to know her a bit and what i did get to know i really liked
ROUGE Scores: {'rouge1': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'rouge2': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'r
--------------------
Generated Output: im okay but feeling a little apprehensive as my dad has a minor operation today
Expected Output: im okay but feeling a little apprehensive as my dad has a minor operation today
ROUGE Scores: {'rouge1': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'rouge2': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'r
--------------------
Generated Output: i just feel too overwhelmed i can t see the forest for the trees as the saying goes
Expected Output: i just feel too overwhelmed i can t see the forest for the trees as the saying goes
ROUGE Scores: {'rouge1': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'rouge2': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'r
--------------------
Generated Output: i cant help but feel sentimental about the fact that we were drawn here
Expected Output: i cant help but feel sentimental about the fact that we were drawn here
ROUGE Scores: {'rouge1': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'rouge2': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'r
--------------------
Generated Output: i feel i should make is how surprised but entertained i was by the inclusion of so many popular culture and gaming
Expected Output: i feel i should make is how surprised but entertained i was by the inclusion of so many popular culture and gaming r
ROUGE Scores: {'rouge1': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'rouge2': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'r
--------------------
Generated Output: i feel so tortured by it
Expected Output: i feel so tortured by it
ROUGE Scores: {'rouge1': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'rouge2': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'r
--------------------
Generated Output: i feel a bit rude leaving you hanging there from my last post with an almost done room and then radio silence
Expected Output: i feel a bit rude leaving you hanging there from my last post with an almost done room and then radio silence
ROUGE Scores: {'rouge1': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'rouge2': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'r
--------------------
Generated Output: im having ssa examination tomorrow in the morning im quite well prepared for the coming exam and somehow i feel num
Expected Output: im having ssa examination tomorrow in the morning im quite well prepared for the coming exam and somehow i feel numb
ROUGE Scores: {'rouge1': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'rouge2': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'r
--------------------
Generated Output: i constantly worry about their fight against nature as they push the limits of their inner bodies for the determina
Expected Output: i constantly worry about their fight against nature as they push the limits of their inner bodies for the determinat
ROUGE Scores: {'rouge1': Score(precision=1.0, recall=0.9666666666666667, fmeasure=0.983050847457627), 'rouge2': Score(precision=1.0,
--------------------
Generated Output: i feel its important to share this info for those that experience the same thing
Expected Output: i feel its important to share this info for those that experience the same thing
ROUGE Scores: {'rouge1': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'rouge2': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'r
--------------------
Generated Output: i truly feel that if you are passionate enough about something and stay true to yourself you will succeed
Expected Output: i truly feel that if you are passionate enough about something and stay true to yourself you will succeed
ROUGE Scores: {'rouge1': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'rouge2': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'r
--------------------
Generated Output: i feel like i just wanna buy any cute make up i see online or even the one
Expected Output: i feel like i just wanna buy any cute make up i see online or even the one
ROUGE Scores: {'rouge1': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'rouge2': Score(precision=1.0, recall=1.0, fmeasure=1.0), 'r
--------------------
```

```
!pip install nltk
import nltk
nltk.download('wordnet')  # Download WordNet for METEOR

from nltk.translate.meteor_score import meteor_score

meteor_score_value = meteor_score([expected_output.split()], generated_output.split())  # Replace with your actual outputs
print(meteor_score_value)
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.11/dist-packages (3.9.1)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from nltk) (8.1.8)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.11/dist-packages (from nltk) (2024.11.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from nltk) (4.67.1)
0.9999142661179699
[nltk_data] Downloading package wordnet to /root/nltk_data...
```