```
!pip install datasets
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (from datasets) (2.2.2)
Requirement already satisfied: requests>=2.32.2 in /usr/local/lib/python3.11/dist-packages (from datasets) (2.32.3)
Requirement already satisfied: tqdm>=4.66.3 in /usr/local/lib/python3.11/dist-packages (from datasets) (4.67.1)
Collecting xxhash (from datasets)
  Downloading xxhash-3.5.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (12 kB)
Collecting multiprocess<0.70.17 (from datasets)
  Downloading multiprocess-0.70.16-py311-none-any.whl.metadata (7.2 kB)
Collecting fsspec<=2024.12.0,>=2023.1.0 (from fsspec[http]<=2024.12.0,>=2023.1.0->datasets)
  Downloading fsspec-2024.12.0-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.11/dist-packages (from datasets) (3.11.15)
Requirement already satisfied: huggingface-hub>=0.24.0 in /usr/local/lib/python3.11/dist-packages (from datasets) (0.30.2)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from datasets) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from datasets) (6.0.2)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (2.6.1)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (1.3.2)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (25.3.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (1.5.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (6.4.3)
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (0.3.1)
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (1.19.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.24.0->da
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests>=2.32.2->datasets)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests>=2.32.2->datasets) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests>=2.32.2->datasets) (2.3.0
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests>=2.32.2->datasets) (2025.
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas->datasets) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas->datasets) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas->datasets) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas->datasets) (1
Downloading datasets-3.5.0-py3-none-any.whl (491 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 491.2/491.2 kB 13.8 MB/s eta 0:00:00
Downloading dill-0.3.8-py3-none-any.whl (116 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 116.3/116.3 kB 13.1 MB/s eta 0:00:00
Downloading fsspec-2024.12.0-py3-none-any.whl (183 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 183.9/183.9 kB 18.6 MB/s eta 0:00:00
Downloading multiprocess-0.70.16-py311-none-any.whl (143 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 143.5/143.5 kB 15.1 MB/s eta 0:00:00
Downloading xxhash-3.5.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (194 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 194.8/194.8 kB 19.6 MB/s eta 0:00:00
Installing collected packages: xxhash, fsspec, dill, multiprocess, datasets
  Attempting uninstall: fsspec
    Found existing installation: fsspec 2025.3.2
    Uninstalling fsspec-2025.3.2:
      Successfully uninstalled fsspec-2025.3.2
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the sour
gcsfs 2025.3.2 requires fsspec==2025.3.2, but you have fsspec 2024.12.0 which is incompatible.
torch 2.6.0+cu124 requires nvidia-cublas-cu12==12.4.5.8; platform_system == "Linux" and platform_machine == "x86_64", but you have nv
torch 2.6.0+cu124 requires nvidia-cuda-cupti-cu12==12.4.127; platform_system == "Linux" and platform_machine == "x86_64", but you have
torch 2.6.0+cu124 requires nvidia-cuda-nvrtc-cu12==12.4.127; platform_system == "Linux" and platform_machine == "x86_64", but you have
torch 2.6.0+cu124 requires nvidia-cuda-runtime-cu12==12.4.127; platform_system == "Linux" and platform_machine == "x86_64", but you h
torch 2.6.0+cu124 requires nvidia-cudnn-cu12==9.1.0.70; platform_system == "Linux" and platform_machine == "x86_64", but you have nvi
torch 2.6.0+cu124 requires nvidia-cufft-cu12==11.2.1.3; platform_system == "Linux" and platform_machine == "x86_64", but you have nvi
torch 2.6.0+cu124 requires nvidia-curand-cu12==10.3.5.147; platform_system == "Linux" and platform_machine == "x86_64", but you have
torch 2.6.0+cu124 requires nvidia-cusolver-cu12==11.6.1.9; platform_system == "Linux" and platform_machine == "x86_64", but you have
torch 2.6.0+cu124 requires nvidia-cusparse-cu12==12.3.1.170; platform_system == "Linux" and platform_machine == "x86_64", but you have
torch 2.6.0+cu124 requires nvidia-nvjitlink-cu12==12.4.127; platform_system == "Linux" and platform_machine == "x86_64", but you have
Successfully installed datasets-3.5.0 dill-0.3.8 fsspec-2024.12.0 multiprocess-0.70.16 xxhash-3.5.0
```

```
%pip install bitsandbytes
```

```
      Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
```

```
        Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
    Attempting uninstall: nvidia-curand-cu12
      Found existing installation: nvidia-curand-cu12 10.3.6.82
      Uninstalling nvidia-curand-cu12-10.3.6.82:
        Successfully uninstalled nvidia-curand-cu12-10.3.6.82
    Attempting uninstall: nvidia-cufft-cu12
      Found existing installation: nvidia-cufft-cu12 11.2.3.61
      Uninstalling nvidia-cufft-cu12-11.2.3.61:
        Successfully uninstalled nvidia-cufft-cu12-11.2.3.61
    Attempting uninstall: nvidia-cuda-runtime-cu12
      Found existing installation: nvidia-cuda-runtime-cu12 12.5.82
      Uninstalling nvidia-cuda-runtime-cu12-12.5.82:
        Successfully uninstalled nvidia-cuda-runtime-cu12-12.5.82
    Attempting uninstall: nvidia-cuda-nvrtc-cu12
      Found existing installation: nvidia-cuda-nvrtc-cu12 12.5.82
      Uninstalling nvidia-cuda-nvrtc-cu12-12.5.82:
        Successfully uninstalled nvidia-cuda-nvrtc-cu12-12.5.82
    Attempting uninstall: nvidia-cuda-cupti-cu12
      Found existing installation: nvidia-cuda-cupti-cu12 12.5.82
      Uninstalling nvidia-cuda-cupti-cu12-12.5.82:
        Successfully uninstalled nvidia-cuda-cupti-cu12-12.5.82
    Attempting uninstall: nvidia-cublas-cu12
      Found existing installation: nvidia-cublas-cu12 12.5.3.2
      Uninstalling nvidia-cublas-cu12-12.5.3.2:
        Successfully uninstalled nvidia-cublas-cu12-12.5.3.2
    Attempting uninstall: nvidia-cusparse-cu12
      Found existing installation: nvidia-cusparse-cu12 12.5.1.3
      Uninstalling nvidia-cusparse-cu12-12.5.1.3:
        Successfully uninstalled nvidia-cusparse-cu12-12.5.1.3
    Attempting uninstall: nvidia-cudnn-cu12
      Found existing installation: nvidia-cudnn-cu12 9.3.0.75
      Uninstalling nvidia-cudnn-cu12-9.3.0.75:
        Successfully uninstalled nvidia-cudnn-cu12-9.3.0.75
    Attempting uninstall: nvidia-cusolver-cu12
      Found existing installation: nvidia-cusolver-cu12 11.6.3.83
      Uninstalling nvidia-cusolver-cu12-11.6.3.83:
        Successfully uninstalled nvidia-cusolver-cu12-11.6.3.83
  Successfully installed bitsandbytes-0.45.5 nvidia-cublas-cu12-12.4.5.8 nvidia-cuda-cupti-cu12-12.4.127 nvidia-cuda-nvrtc-cu12-12.4.12
```

```python
from google.colab import userdata
userdata.get('HF_TOKEN')
```

```
'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'
```

```
!pip install sacrebleu rouge-score tqdm
```

```
Collecting sacrebleu
  Downloading sacrebleu-2.5.1-py3-none-any.whl.metadata (51 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 51.8/51.8 kB 2.8 MB/s eta 0:00:00
Collecting rouge-score
  Downloading rouge_score-0.1.2.tar.gz (17 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (4.67.1)
Collecting portalocker (from sacrebleu)
  Downloading portalocker-3.1.1-py3-none-any.whl.metadata (8.6 kB)
Requirement already satisfied: regex in /usr/local/lib/python3.11/dist-packages (from sacrebleu) (2024.11.6)
Requirement already satisfied: tabulate>=0.8.9 in /usr/local/lib/python3.11/dist-packages (from sacrebleu) (0.9.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages (from sacrebleu) (2.0.2)
Collecting colorama (from sacrebleu)
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Requirement already satisfied: lxml in /usr/local/lib/python3.11/dist-packages (from sacrebleu) (5.3.2)
Requirement already satisfied: absl-py in /usr/local/lib/python3.11/dist-packages (from rouge-score) (1.4.0)
Requirement already satisfied: nltk in /usr/local/lib/python3.11/dist-packages (from rouge-score) (3.9.1)
Requirement already satisfied: six>=1.14.0 in /usr/local/lib/python3.11/dist-packages (from rouge-score) (1.17.0)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from nltk->rouge-score) (8.1.8)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages (from nltk->rouge-score) (1.4.2)
Downloading sacrebleu-2.5.1-py3-none-any.whl (104 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 104.1/104.1 kB 7.1 MB/s eta 0:00:00
Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Downloading portalocker-3.1.1-py3-none-any.whl (19 kB)
Building wheels for collected packages: rouge-score
  Building wheel for rouge-score (setup.py) ... done
  Created wheel for rouge-score: filename=rouge_score-0.1.2-py3-none-any.whl size=24934 sha256=a7ddb64bd296796aeb6fccd37b836385839c85577
  Stored in directory: /root/.cache/pip/wheels/1e/19/43/8a442dc83660ca25e163e1bd1f89919284ab0d0c1475475148
Successfully built rouge-score
Installing collected packages: portalocker, colorama, sacrebleu, rouge-score
Successfully installed colorama-0.4.6 portalocker-3.1.1 rouge-score-0.1.2 sacrebleu-2.5.1
```

```python
import pickle
import torch
import numpy as np
from transformers import (
    AutoTokenizer,
    AutoModelForCausalLM,
    BitsAndBytesConfig,
    TrainingArguments,
    Trainer,
    DataCollatorForSeq2Seq,
    TrainerCallback,
    DataCollatorForLanguageModeling
)
from datasets import Dataset
from peft import LoraConfig, get_peft_model, prepare_model_for_kbit_training
import sacrebleu
from rouge_score import rouge_scorer
from tqdm import tqdm
import pandas as pd
import math


#Load validation data
# with open('/content/drive/MyDrive/data_postprocessed/valid_data_postprocessed.pkl', 'rb') as f:
#     valid_data = pickle.load(f)

# Load test data
with open('/content/drive/MyDrive/data_postprocessed/test_data_postprocessed.pkl', 'rb') as f:
    test_data = pickle.load(f)



#creating a prompt for the model

def create_prompt(record):
    # Start with the buggy code
    prompt = f"### Buggy Code:\n{record['before_merge']}\n\n"

    # Include traceback information if available
    if record.get('traceback_type') or record.get('full_traceback'):
        prompt += f"### Traceback:\n{record.get('traceback_type', '')}: {record.get('full_traceback', '')}\n\n"

    # Instruction for the model to output the fix
    prompt += "### Provide the corrected code below:\n"
    return prompt


def prepare_record(record):
    return {
        "input": create_prompt(record),  # the prompt that includes the context
        "output": record["after_merge"]    # the target corrected code
    }

# prepared_valid = [prepare_record(r) for r in valid_data]
prepared_test  = [prepare_record(r) for r in test_data]


# Load the tokenizer from your saved fine-tuned model directory
tokenizer = AutoTokenizer.from_pretrained("/content/drive/MyDrive/codellama_qlora_finetuned/fine-tuned/codellama-finetuned1/checkpoint-1323"

if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token


def tokenize_record(record, max_length=512):
    tokenized_input = tokenizer(
        record["input"], truncation=True, padding="max_length", max_length=max_length, return_tensors="pt"
    )
    tokenized_output = tokenizer(
        record["output"], truncation=True, padding="max_length", max_length=max_length, return_tensors="pt"
    )
    return {
        "input_ids": tokenized_input["input_ids"].squeeze(),
        "attention_mask": tokenized_input["attention_mask"].squeeze(),
        "labels": tokenized_output["input_ids"].squeeze()
    }

# tokenized_train = [tokenize_record(r) for r in prepared_train]
# tokenized_valid = [tokenize_record(r) for r in prepared_valid]
tokenized_test  = [tokenize_record(r) for r in prepared_test]
```

```
# valid_dataset = Dataset.from_dict({
#     "input_ids": [x["input_ids"].tolist() for x in tokenized_valid],
#     "attention_mask": [x["attention_mask"].tolist() for x in tokenized_valid],
#     "labels": [x["labels"].tolist() for x in tokenized_valid],
# })

test_dataset = Dataset.from_dict({
    "input_ids": [x["input_ids"].tolist() for x in tokenized_test],
    "attention_mask": [x["attention_mask"].tolist() for x in tokenized_test],
    "labels": [x["labels"].tolist() for x in tokenized_test],
})


quantization_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_compute_dtype=torch.float16,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_quant_type="nf4"
)


# Load fine-tuned model and tokenizer
model_path = "/content/drive/MyDrive/codellama_qlora_finetuned/fine-tuned/codellama-finetuned1/checkpoint-1323"
hf_token = "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"

model = AutoModelForCausalLM.from_pretrained(
    model_path,
    token = hf_token,
    quantization_config=quantization_config,
    device_map="auto")
```

```
config.json: 100%                                637/637 [00:00<00:00, 20.1kB/s]
model.safetensors.index.json: 100%               25.1k/25.1k [00:00<00:00, 1.27MB/s]
Fetching 2 files: 100%                           2/2 [02:00<00:00, 120.68s/it]
model-00001-of-00002.safetensors: 100%           9.98G/9.98G [02:00<00:00, 198MB/s]
model-00002-of-00002.safetensors: 100%           3.50G/3.50G [01:02<00:00, 86.2MB/s]
Loading checkpoint shards: 100%                  2/2 [01:09<00:00, 31.50s/it]
generation_config.json: 100%                     116/116 [00:00<00:00, 8.75kB/s]
```

```
# Evaluation arguments (no saving/logging)
eval_args = TrainingArguments(
    output_dir="./tmp_eval",  # Dummy folder
    per_device_eval_batch_size=1,  # Keep it small for GPU memory
    do_eval=True,
    logging_strategy="no",
    report_to=[],
    save_strategy="no"
)


# Initialize and run evaluation
trainer = Trainer(
    model=model,
    args=eval_args,
    eval_dataset=valid_dataset,
    tokenizer=tokenizer,
)
```

```
<ipython-input-34-06fac87f4cd3>:2: FutureWarning: `tokenizer` is deprecated and will be removed in version 5.0.0 for `Trainer.__init__`.
  trainer = Trainer(
```

```
results = trainer.evaluate()
```

```
[6651/9457 1:00:23 < 25:29, 1.84 it/s]
[9457/9457 1:25:53]
```

```
# Print evaluation metrics
print("Evaluation Results on Subset:")
for key, value in results.items():
    print(f"{key}: {value:.4f}")
```

```
Evaluation Results on Subset:
eval_loss: 16.3244
eval_model_preparation_time: 0.0030
eval_runtime: 5154.0504
eval_samples_per_second: 1.8350
eval_steps_per_second: 1.8350
```

```
import math
def preprocess_batch(record, max_length=512):
    # Tokenize both input and output as a single prompt-target string
    full_prompt = f"{record['input']}\n{record['output']}"
    tokenized = tokenizer(
        full_prompt,
        return_tensors="pt",
        padding="max_length",
        truncation=True,
        max_length=max_length
    )
    input_ids = tokenized["input_ids"].to(device)
    attention_mask = tokenized["attention_mask"].to(device)
    return input_ids, attention_mask


def compute_perplexity(input_ids, attention_mask):
    with torch.no_grad():
        outputs = model(input_ids=input_ids, attention_mask=attention_mask, labels=input_ids)
        loss = outputs.loss
        return math.exp(loss.item())


import random

# Reduce memory usage
subset_valid = random.sample(prepared_valid, 3000)

perplexities = []

for record in subset_valid:
    input_ids, attention_mask = preprocess_batch(record)
    perplexity = compute_perplexity(input_ids, attention_mask)
    perplexities.append(perplexity)

print(f"\n✅ Average Perplexity: {sum(perplexities)/len(perplexities):.2f}")
```

```

✅ Average Perplexity: 18.97
```

Testing the Fine tuned model against base model- codebased

```
#setting up base model
model_name = "meta-llama/CodeLlama-7b-hf"
hf_token = "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"

tokenizer = AutoTokenizer.from_pretrained(model_name, token=hf_token)
if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token
```

```
tokenizer_config.json: 100%          749/749 [00:00<00:00, 90.2kB/s]

tokenizer.model: 100%          500k/500k [00:00<00:00, 13.9MB/s]

tokenizer.json: 100%          1.84M/1.84M [00:00<00:00, 6.71MB/s]

special_tokens_map.json: 100%          411/411 [00:00<00:00, 55.0kB/s]
```

```
# Load Base Model
base_model = AutoModelForCausalLM.from_pretrained(
    model_name,
    device_map="auto",
    token=hf_token,
    torch_dtype=torch.float16
)
```

```
config.json: 100%                                    637/637 [00:00<00:00, 82.6kB/s]
model.safetensors.index.json: 100%                   25.1k/25.1k [00:00<00:00, 40.4kB/s]
Fetching 2 files: 100%                               2/2 [00:56<00:00, 56.02s/it]
model-00002-of-00002.safetensors: 100%               3.50G/3.50G [00:20<00:00, 195MB/s]
model-00001-of-00002.safetensors: 100%               9.98G/9.98G [00:55<00:00, 202MB/s]
Loading checkpoint shards: 100%                      2/2 [00:03<00:00, 1.78s/it]
generation_config.json: 100%                         116/116 [00:00<00:00, 14.2kB/s]
```

```
quantization_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.float16,
    llm_int8_enable_fp32_cpu_offload=True
)


# Load fine-tuned model
finetuned_model_path = "/content/drive/MyDrive/codellama_qlora_finetuned/fine-tuned/codellama-finetuned1/checkpoint-1323"
hf_token = "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"

fine_tune_model_1 = AutoModelForCausalLM.from_pretrained(
    finetuned_model_path,
    token = hf_token,
    quantization_config=quantization_config,
    device_map="auto",
    )
```

```
Loading checkpoint shards: 100%                      2/2 [00:16<00:00, 7.59s/it]
```

```
def generate_prediction(model, prompt, max_new_tokens=128):
    inputs = tokenizer(
        prompt,
        return_tensors="pt",
        truncation=True,
        padding=True,
        max_length=512
    ).to("cuda")

    outputs = model.generate(
        **inputs,
        max_new_tokens=max_new_tokens,
        do_sample=False,
        pad_token_id=tokenizer.eos_token_id
    )
    return tokenizer.decode(outputs[0], skip_special_tokens=True)



# Extract prompts and targets
prompts = [item["input"] for item in prepared_test]
targets = [item["output"] for item in prepared_test]


# Generate predictions
base_preds = []
fine_preds = []


print("Generating predictions... this will take a few minutes.")
for prompt in tqdm(prompts):
    with torch.no_grad():
        base_out = generate_prediction(base_model, prompt)
        torch.cuda.empty_cache()
        fine_out = generate_prediction(fine_tune_model, prompt)
        torch.cuda.empty_cache()
    base_preds.append(base_out.strip())
    fine_preds.append(fine_out.strip())
```

```
Generating predictions... this will take a few minutes.
100%|██████████| 161/161 [37:25<00:00, 13.95s/it]
```

```
# Save predictions to CSV
df = pd.DataFrame({
    "Prompt": prompts,
    "Target": targets,
    "Base Model Output": base_preds,
    "Fine-Tuned Model Output": fine_preds
})
df.to_csv("model_comparison_outputs.csv", index=False)
print("✅ Saved: model_comparison_outputs.csv")
```

```
✅ Saved: model_comparison_outputs.csv
```

```
# BLEU Scores
bleu_base = sacrebleu.corpus_bleu(base_preds, [targets]).score
bleu_fine = sacrebleu.corpus_bleu(fine_preds, [targets]).score


# ROUGE-L Scores
scorer = rouge_scorer.RougeScorer(['rougeL'], use_stemmer=True)
rouge_base = sum([scorer.score(t, p)['rougeL'].fmeasure for t, p in zip(targets, base_preds)]) / len(targets)
rouge_fine = sum([scorer.score(t, p)['rougeL'].fmeasure for t, p in zip(targets, fine_preds)]) / len(targets)


# Final Results
print("Evaluation Results:")
print(f"Base Model     => BLEU: {bleu_base:.2f}, ROUGE-L: {rouge_base:.3f}")
print(f"Fine-Tuned     => BLEU: {bleu_fine:.2f}, ROUGE-L: {rouge_fine:.3f}")
```

```
Evaluation Results:
Base Model     => BLEU: 55.24, ROUGE-L: 0.560
Fine-Tuned     => BLEU: 55.22, ROUGE-L: 0.559
```

```
# Set up data collator
data_collator = DataCollatorForLanguageModeling(tokenizer=tokenizer, mlm=False)


# Common training args for both evaluations
eval_args = TrainingArguments(
    output_dir="./eval_output",
    per_device_eval_batch_size=1,
    dataloader_drop_last=False,
    report_to="none"
)


# Base Model Perplexity
trainer_base = Trainer(
    model=base_model,
    args=eval_args,
    eval_dataset=tokenized_test,
    data_collator=data_collator,
    tokenizer=tokenizer
)
loss_base = trainer_base.evaluate()["eval_loss"]
ppl_base = math.exp(loss_base)
```

```
<ipython-input-15-31730ff9a37f>:2: FutureWarning: `tokenizer` is deprecated and will be removed in version 5.0.0 for `Trainer.__init__`.
  trainer_base = Trainer(
                                        [161/161 00:27]
```

```
# Fine-Tuned Model Perplexity
trainer_fine = Trainer(
    model=fine_tune_model_1,
    args=eval_args,
    eval_dataset=tokenized_test,
    data_collator=data_collator,
    tokenizer=tokenizer
)
loss_fine = trainer_fine.evaluate()["eval_loss"]
ppl_fine = math.exp(loss_fine)
```

```
<ipython-input-16-f934c03e065f>:2: FutureWarning: `tokenizer` is deprecated and will be removed in version 5.0.0 for `Trainer.__init__`.
  trainer_fine = Trainer(
                                                  [161/161 00:43]
```

```
# Print comparison
print("\n Perplexity Comparison:")
print(f"Base Model     => Loss: {loss_base:.4f}, Perplexity: {ppl_base:.2f}")
print(f"Fine-Tuned     => Loss: {loss_fine:.4f}, Perplexity: {ppl_fine:.2f}")
```

```
 Perplexity Comparison:
Base Model     => Loss: 0.9878, Perplexity: 2.69
Fine-Tuned     => Loss: 0.8636, Perplexity: 2.37
```

Testing the Fine tuned model against base model- astbased

```
def create_prompt(record):
    # Start with the buggy code
    prompt = f"### Buggy Code AST:\n{record['old_ast_json']}\n\n"

    # Include traceback information if available
    if record.get('traceback_type') or record.get('full_traceback'):
        prompt += f"### Traceback:\n{record.get('traceback_type', '')}: {record.get('full_traceback', '')}\n\n"

    # Instruction for the model to output the fix
    prompt += "### Provide the corrected code AST below:\n"
    return prompt


def prepare_record(record):
    return {
        "input": create_prompt(record),
        "output": record["after_merge"]
    }

prepared_test  = [prepare_record(r) for r in test_data]
```

```
#setting up base model
model_name = "meta-llama/CodeLlama-7b-hf"
hf_token = "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"

tokenizer = AutoTokenizer.from_pretrained(model_name, token=hf_token)
if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token
```

```
# Load Base Model
base_model = AutoModelForCausalLM.from_pretrained(
    model_name,
    device_map="auto",
    token=hf_token,
    torch_dtype=torch.float16
)
```

```
Loading checkpoint shards: 100%                                    2/2 [00:03<00:00,  1.77s/it]
```

```
# Load fine-tuned model
finetuned_model_path = "/content/drive/MyDrive/codellama_qlora_finetuned/fine-tuned/codellama-finetuned-ast/checkpoint-1323"
hf_token = "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"

fine_tune_model = AutoModelForCausalLM.from_pretrained(
    finetuned_model_path,
    token = hf_token,
    quantization_config=quantization_config,
    device_map="auto",
    )
```

```
Loading checkpoint shards: 100%                                    2/2 [00:16<00:00,  7.39s/it]
```

```
# Extract prompts and targets
prompts = [item["input"] for item in prepared_test]
targets = [item["output"] for item in prepared_test]
```

```
print(prompts)
```

```
['### Buggy Code AST:\n[\n    "Module(body=[FunctionDef(name=\'remove_lb_backend_address_pool_address\', args=arguments(posonlyargs=[],
```

```
# Generate predictions
base_preds = []
fine_preds = []


print("Generating predictions... this will take a few minutes.")
for prompt in tqdm(prompts):
    with torch.no_grad():
        base_out = generate_prediction(base_model, prompt)
        torch.cuda.empty_cache()
        fine_out = generate_prediction(fine_tune_model, prompt)
        torch.cuda.empty_cache()

    base_preds.append(base_out.strip())
    fine_preds.append(fine_out.strip())
```

```
Generating predictions... this will take a few minutes.
100%|██████████| 161/161 [37:39<00:00, 14.04s/it]
```

```
# Save predictions to CSV
df = pd.DataFrame({
    "Prompt": prompts,
    "Target": targets,
    "Base Model Output": base_preds,
    "Fine-Tuned Model Output": fine_preds
})
df.to_csv("model_comparison_outputs.csv", index=False)
print("✅ Saved: model_comparison_outputs.csv")
```

```
✅ Saved: model_comparison_outputs.csv
```

```
# BLEU Scores
bleu_base = sacrebleu.corpus_bleu(base_preds, [targets]).score
bleu_fine = sacrebleu.corpus_bleu(fine_preds, [targets]).score


# ROUGE-L Scores
scorer = rouge_scorer.RougeScorer(['rougeL'], use_stemmer=True)
rouge_base = sum([scorer.score(t, p)['rougeL'].fmeasure for t, p in zip(targets, base_preds)]) / len(targets)
rouge_fine = sum([scorer.score(t, p)['rougeL'].fmeasure for t, p in zip(targets, fine_preds)]) / len(targets)


# Final Results
print("Evaluation Results:")
print(f"Base Model     => BLEU: {bleu_base:.2f}, ROUGE-L: {rouge_base:.3f}")
print(f"Fine-Tuned     => BLEU: {bleu_fine:.2f}, ROUGE-L: {rouge_fine:.3f}")
```

```
Evaluation Results:
Base Model     => BLEU: 8.28, ROUGE-L: 0.255
Fine-Tuned     => BLEU: 7.75, ROUGE-L: 0.255
```

```
# Load the tokenizer from your saved fine-tuned model directory
tokenizer = AutoTokenizer.from_pretrained("/content/drive/MyDrive/codellama_qlora_finetuned/fine-tuned/codellama-finetuned-ast/checkpoint-13

if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token


def tokenize_record(record, max_length=512):
    tokenized_input = tokenizer(
        record["input"], truncation=True, padding="max_length", max_length=max_length, return_tensors="pt"
    )
    tokenized_output = tokenizer(
        record["output"], truncation=True, padding="max_length", max_length=max_length, return_tensors="pt"
    )
    return {
        "input_ids": tokenized_input["input_ids"].squeeze(),
        "attention_mask": tokenized_input["attention_mask"].squeeze(),
        "labels": tokenized_output["input_ids"].squeeze()
    }
```

```
tokenized_test  = [tokenize_record(r) for r in prepared_test]


test_dataset = Dataset.from_dict({
    "input_ids": [x["input_ids"].tolist() for x in tokenized_test],
    "attention_mask": [x["attention_mask"].tolist() for x in tokenized_test],
    "labels": [x["labels"].tolist() for x in tokenized_test],
})


# Set up data collator
data_collator = DataCollatorForLanguageModeling(tokenizer=tokenizer, mlm=False)


# Common training args for both evaluations
eval_args = TrainingArguments(
    output_dir="./eval_output",
    per_device_eval_batch_size=1,
    dataloader_drop_last=False,
    report_to="none"
)


# Base Model Perplexity
trainer_base = Trainer(
    model=base_model,
    args=eval_args,
    eval_dataset=test_dataset,
    data_collator=data_collator,
    tokenizer=tokenizer
)
loss_base = trainer_base.evaluate()["eval_loss"]
ppl_base = math.exp(loss_base)
```

```
<ipython-input-54-9ece493d0a2e>:2: FutureWarning: `tokenizer` is deprecated and will be removed in version 5.0.0 for `Trainer.__init__`.
  trainer_base = Trainer(
[161/161 00:07]
```

```
# Fine-Tuned Model Perplexity
trainer_fine = Trainer(
    model=fine_tune_model,
    args=eval_args,
    eval_dataset=tokenized_test,
    data_collator=data_collator,
    tokenizer=tokenizer
)
loss_fine = trainer_fine.evaluate()["eval_loss"]
ppl_fine = math.exp(loss_fine)
```

```
<ipython-input-55-1f00fe7c9950>:2: FutureWarning: `tokenizer` is deprecated and will be removed in version 5.0.0 for `Trainer.__init__`.
  trainer_fine = Trainer(
[161/161 00:16]
```

```
# Print comparison
print("\n Perplexity Comparison:")
print(f"Base Model     => Loss: {loss_base:.4f}, Perplexity: {ppl_base:.2f}")
print(f"Fine-Tuned     => Loss: {loss_fine:.4f}, Perplexity: {ppl_fine:.2f}")
```

```

 Perplexity Comparison:
Base Model     => Loss: 0.7002, Perplexity: 2.01
Fine-Tuned     => Loss: 0.3938, Perplexity: 1.48
```

Start coding or generate with AI.