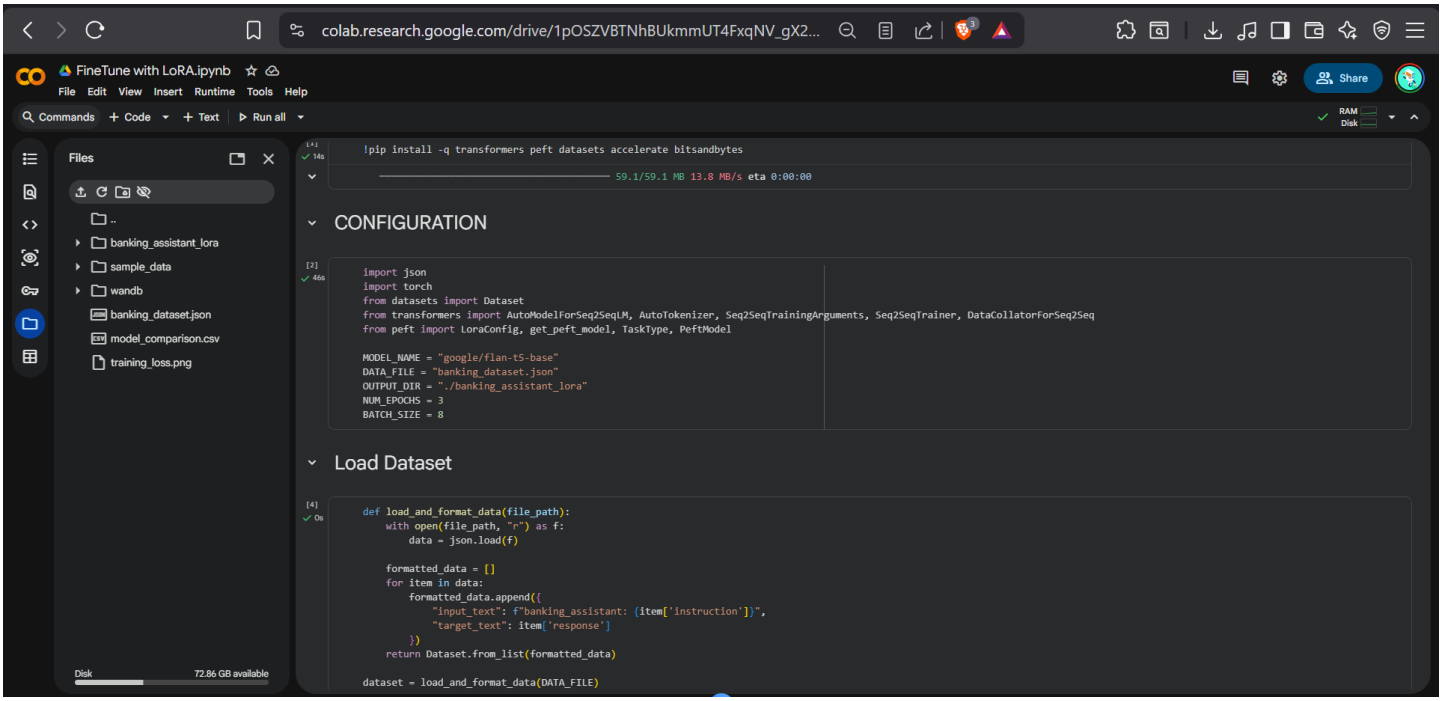
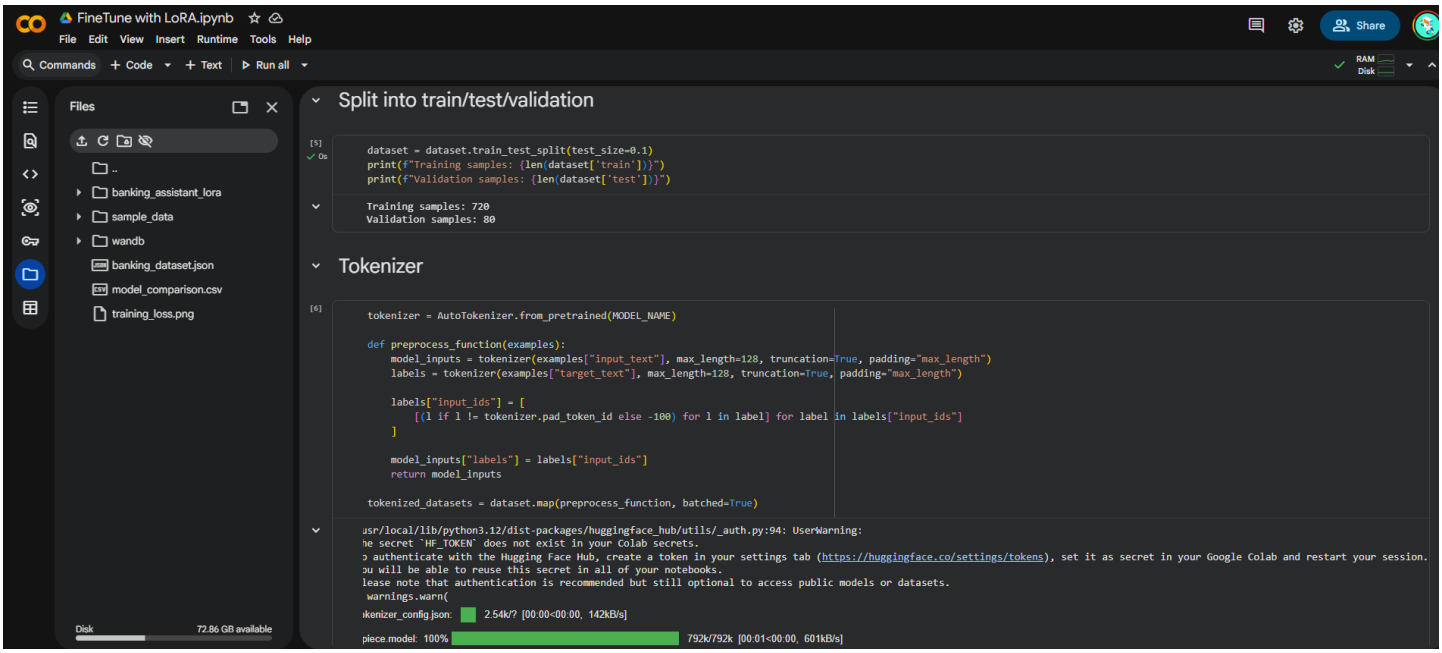
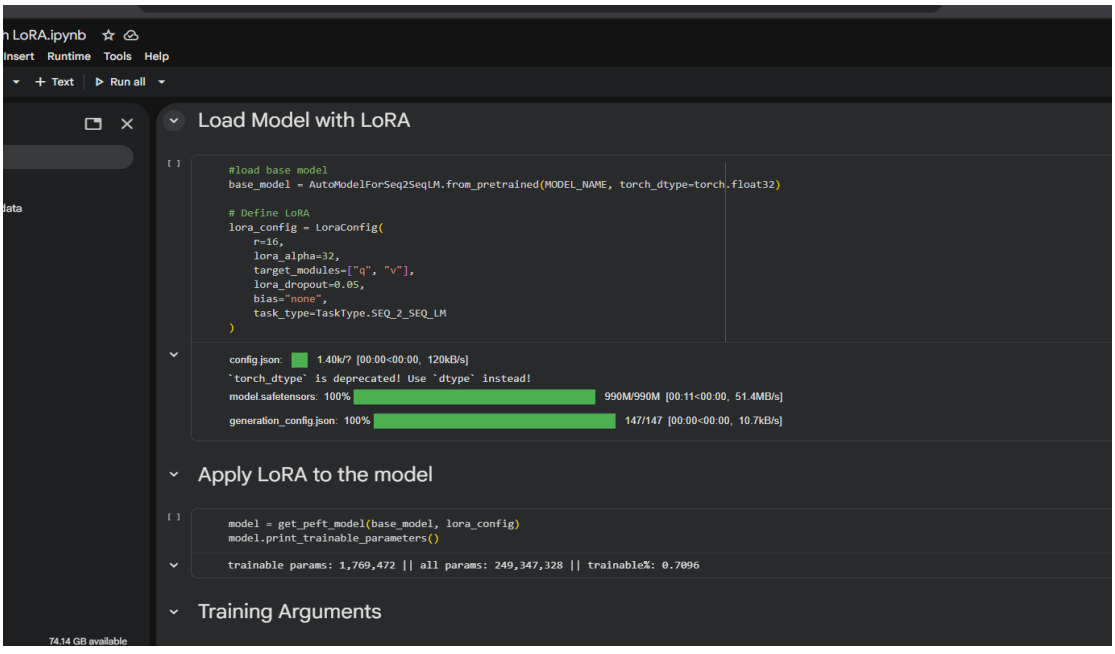

colab.research.google.com/drive/1pOSZVBTNhBUkmmUT4FxqNV_gX2...
FineTune with LoRA.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM Disk
Files
banking_assistant_lora
sample_data
wandb
banking_dataset.json
model_comparison.csv
training_loss.png
Disk 72.86 GB available
!pip install -q transformers peft datasets accelerate bitsandbytes
59.1/59.1 MB 13.8 MB/s eta 0:00:00
CONFIGURATION
import json
import torch
from datasets import Dataset
from transformers import AutoModelForSeq2SeqLM, AutoTokenizer, Seq2SeqTrainingArguments, Seq2SeqTrainer, DataCollatorForSeq2Seq
from peft import LoraConfig, get_peft_model, TaskType, PeftModel
MODEL_NAME = "google/flan-t5-base"
DATA_FILE = "banking_dataset.json"
OUTPUT_DIR = "./banking_assistant_lora"
NUM_EPOCHS = 3
BATCH_SIZE = 8
Load Dataset
def load_and_format_data(file_path):
with open(file_path, "r") as f:
data = json.load(f)
formatted_data = []
for item in data:
formatted_data.append({
"input_text": f"banking_assistant: {item['instruction']}",
"target_text": item['response']
})
return Dataset.from_list(formatted_data)
dataset = load_and_format_data(DATA_FILE)


FineTune with LoRA.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM Disk
Files
banking_assistant_lora
sample_data
wandb
banking_dataset.json
model_comparison.csv
training_loss.png
Disk 72.86 GB available
Split into train/test/validation
dataset = dataset.train_test_split(test_size=0.1)
print(f"Training samples: {len(dataset['train'])}")
print(f"Validation samples: {len(dataset['test'])}")
Training samples: 720
Validation samples: 80
Tokenizer
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
def preprocess_function(examples):
model_inputs = tokenizer(examples["input_text"], max_length=128, truncation=True, padding="max_length")
labels = tokenizer(examples["target_text"], max_length=128, truncation=True, padding="max_length")
labels["input_ids"] = [
[(l if l != tokenizer.pad_token_id else -100) for l in label] for label in labels["input_ids"]
]
model_inputs["labels"] = labels["input_ids"]
return model_inputs
tokenized_datasets = dataset.map(preprocess_function, batched=True)
usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
he secret `HF_TOKEN` does not exist in your Colab secrets.
o authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your session.
ou will be able to reuse this secret in all of your notebooks.
lease note that authentication is recommended but still optional to access public models or datasets.
warnings.warn(
kenizer_config.json: 2.54k/? [00:00<00:00, 142kB/s]
piece.model: 100% 792k/792k [00:01<00:00, 601kB/s]


LoRA.ipynb
Insert Runtime Tools Help
Text Run all
74.14 GB available
Load Model with LoRA
#load base model
base_model = AutoModelForSeq2SeqLM.from_pretrained(MODEL_NAME, torch_dtype=torch.float32)
# Define LoRA
lora_config = LoraConfig(
r=16,
lora_alpha=32,
target_modules=["q", "v"],
lora_dropout=0.05,
bias="none",
task_type=TaskType.SEQ_2_SEQ_LM
)
config.json: 1.40k/? [00:00<00:00, 120kB/s]
`torch_dtype` is deprecated! Use `dtype` instead!
model.safetensors: 100% 990M/990M [00:11<00:00, 51.4MB/s]
generation_config.json: 100% 147/147 [00:00<00:00, 10.7kB/s]
Apply LoRA to the model
model = get_peft_model(base_model, lora_config)
model.print_trainable_parameters()
trainable params: 1,769,472 || all params: 249,347,328 || trainable%: 0.7096
Training Arguments

LoRA.ipynb

sert Runtime Tools Help

+ Text | Run all

Training Arguments

```
training_args = Seq2SeqTrainingArguments(
    output_dir=OUTPUT_DIR,
    per_device_train_batch_size=BATCH_SIZE,
    per_device_eval_batch_size=BATCH_SIZE,
    learning_rate=3e-4,
    num_train_epochs=5,
    logging_dir=f"{OUTPUT_DIR}/logs",
    logging_strategy="steps",
    logging_steps=20,
    eval_strategy="epoch",
    save_strategy="epoch",
    save_total_limit=2,
    predict_with_generate=True,
    fp16=False,
    push_to_hub=False,
)
```

Trainer

```
trainer = Seq2SeqTrainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_datasets["train"],
    eval_dataset=tokenized_datasets["test"],
    tokenizer=tokenizer,
    data_collator=DataCollatorForSeq2Seq(tokenizer, model=model)
)

print("Starting training...")
trainer.train()
```

74.14 GB available

LoRA.ipynb

sert Runtime Tools Help

+ Text | Run all

Syncing run **treasured-aardvark-2** to Weights & Biases (docs)
View project at https://wandb.ai/anujananu812-university-of-moratuwa/huggingface
View run at https://wandb.ai/anujananu812-university-of-moratuwa/huggingface/runs/us8abugs
[361/450 02:37 < 00:39, 2.28 it/s, Epoch 4/5]

Epoch	Training Loss	Validation Loss
1	2.046800	1.496319
2	1.090500	0.661210
3	0.794400	0.411531

[9/10 00:01 < 00:00, 4.94 it/s]
[450/450 03:20, Epoch 5/5]

Epoch	Training Loss	Validation Loss
1	2.046800	1.496319
2	1.090500	0.661210
3	0.794400	0.411531
4	0.627800	0.346995
5	0.578300	0.324698

```
TrainOutput(global_step=450, training_loss=1.207777435514662, metrics={'train_runtime': 565.9482, 'train_samples_per_second': 6.361, 'train_steps_per_second': 0.795,
621173853388800.0, 'train_loss': 1.207777435514662, 'epoch': 5.0})
```

```
model.save_pretrained(OUTPUT_DIR)
tokenizer.save_pretrained(OUTPUT_DIR)
```

```
('./banking_assistant_lora/tokenizer_config.json',
 './banking_assistant_lora/special_tokens_map.json',
 './banking_assistant_lora/spiece.model',
 './banking_assistant_lora/added_tokens.json',
 './banking_assistant_lora/tokenizer.json')
```

LoRA.ipynb

ert Runtime Tools Help

+ Text | Run all

Compare Base vs Fine-Tuned

```
import pandas as pd
from transformers import pipeline

print("\n--- Generating Comparisons (Base vs Fine-Tuned) ---")

test_prompts = [
    "I lost my credit card, help!",
    "What is the balance of my Savings account?",
    "Transfer 500 to Mom.",
    "When does the bank close?",
    "Can I get a home loan?",
    "My platinum card was stolen.",
    "Send 200.50 to Alice.",
    "Do you are open on Sundays?",
    "What is the interest rate for a car loan?",
    "I need to block my debit card."
]

def generate_response(model, tokenizer, prompt):
    # Standard T5 format
    input_text = f"Instruction: {prompt}"
    inputs = tokenizer(input_text, return_tensors="pt").to(model.device)

    # Generate
    outputs = model.generate(**inputs, max_new_tokens=64)
    return tokenizer.decode(outputs[0], skip_special_tokens=True)
```

RA.ipynb

rt Runtime Tools Help

+ Text | ▷ Run all

Evaluation & Visualization

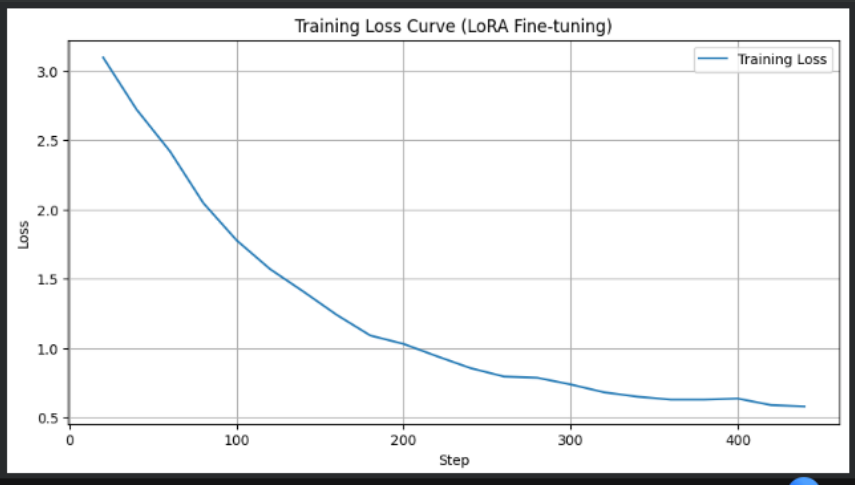
```
import matplotlib.pyplot as plt
import pandas as pd
```

Plot training loss

```
history = trainer.state.log_history
loss_data = {'step': [], 'loss': []}

for entry in history:
    if 'loss' in entry:
        loss_data['step'].append(entry['step'])
        loss_data['loss'].append(entry['loss'])

plt.figure(figsize=(10, 5))
plt.plot(loss_data['step'], loss_data['loss'], label='Training Loss')
plt.xlabel('Step')
plt.ylabel('Loss')
plt.title('Training Loss Curve (LoRA Fine-tuning)')
plt.legend()
plt.grid(True)
plt.savefig("training_loss.png")
plt.show()
```



74.14 GB available

RA.ipynb

rt Runtime Tools Help

Share

+ Text | ▷ Run all

RAM
Disk

```
results = []

print(f"{'Instruction':<40} | {'BASE Model (Before)':<30} | {'FINE-TUNED (After)'}")
print("-" * 110)

for prompt in test_prompts:
    # 1. Get Fine-Tuned Response
    ft_response = generate_response(model, tokenizer, prompt)

    # 2. Get Base Model Response (Temporarily disable LoRA)
    with model.disable_adapter():
        base_response = generate_response(model, tokenizer, prompt)

    print(f"{prompt:<40} | {base_response:<30} | {ft_response}")

    results.append({
        "instruction": prompt,
        "base_model": base_response,
        "fine_tuned_model": ft_response
    })

# Save comparisons to CSV for your submission
df = pd.DataFrame(results)
df.to_csv("model_comparison.csv", index=False)
print("\nSuccess! Saved comparison results to 'model_comparison.csv'")
```

```
--- Generating Comparisons (Base vs Fine-Tuned) ---
Instruction                              | BASE Model (Before)           | FINE-TUNED (After)
--------------------------------------------------------------------------------------------------------------
I lost my credit card, help!             | I lost my credit card, help!  | I have locked your credit card temporarily. Please check your mobile app to permanently block it or reques
What is the balance of my Savings account? | 0                           |  The balance of your Savings is $2682.00.
Transfer 500 to Mom.                     | 500 to Mom.                   | Transfer request for 500 received. Please verify the recipient's details before I confirm.
When does the bank close?                | late at night                 | We are currently closed. Our hours are 9 AM to 5 PM, Monday through Friday.
Can I get a home loan?                   | Yes                           | You can apply for a Home Loan directly through the app under the 'Loans' tab.
My platinum card was stolen.             | I'm sorry.                    | Please log in to your Apple app and select 'Pepper Card' immediately. I can also connect you to a secure l
Send 200.50 to Alice.                    | Alice: 200.50.                | Please send 200.50 to Alice. Please send it to your phone.
Do you are open on Sundays?              | no                            | We are closed on Sundays and public holidays.
What is the interest rate for a car loan? | 0                            |  You can apply for a car loan with a fixed interest rate. However, you can apply for a car loan with a fix
I need to block my debit card            | Is there a way to do this?    | To block your debit card, you need to enter your PIN. Shall we proceed?
```

74.14 GB available