Step 1: Install + Imports

```
!pip install -q transformers datasets accelerate
```

```
import os
import json
import torch

from huggingface_hub import login
from google.colab import userdata
from datasets import Dataset, DatasetDict
from transformers import (
    AutoTokenizer,
    GPT2LMHeadModel,
    DataCollatorForLanguageModeling,
    Trainer,
    TrainingArguments,
)

login(userdata.get('HF'))
```

Step 2: Setup Google Drive

```
from google.colab import drive
drive.mount('/content/drive')

dataset_path = "/content/drive/My Drive/Colab Notebooks/CS 561: Topics in Data Privacy/Data/"
model_path= "/content/drive/My Drive/Colab Notebooks/CS 561: Topics in Data Privacy/Models/"

output_dir = os.path.join(model_path, "gpt2_baseline_poisoned")
```

```
Mounted at /content/drive
```

Step 3: Load Dataset

```
def load_jsonl_as_strings(path):
    texts = []
    with open(path, "r", encoding="utf-8") as f:
        for line in f:
            line = line.strip()
            if not line:
                continue
            obj = json.loads(line)     # each line is a JSON string, so obj is a Python str
            texts.append(str(obj))
    return texts
```

```
train_file = os.path.join(dataset_path, "train.jsonl")
train_texts = load_jsonl_as_strings(train_file)

print("Train dataset size:", len(train_texts))

train_dataset = Dataset.from_dict({"text": train_texts})
train_dataset
```

```
Train dataset size: 5132
Dataset({
    features: ['text'],
    num_rows: 5132
})
```

Step 4: Load Tokenizer

```
tokenizer = AutoTokenizer.from_pretrained("gpt2")

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

MAX_LEN = 128  # as we selected earlier
```

```
def tokenize_function(batch):
    return tokenizer(
        batch["text"],
        truncation=True,
        max_length=MAX_LEN,
        padding=False,
    )

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

tokenized_train
```

```
/usr/local/lib/python3.12/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
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(
tokenizer_config.json: 100%  26.0/26.0 [00:00<00:00, 532B/s]
config.json: 100%  665/665 [00:00<00:00, 22.2kB/s]
vocab.json: 100%  1.04M/1.04M [00:00<00:00, 1.62MB/s]
merges.txt: 100%  456k/456k [00:00<00:00, 8.78MB/s]
tokenizer.json: 100%  1.36M/1.36M [00:00<00:00, 1.56MB/s]
Map: 100%  5132/5132 [00:15<00:00, 333.39 examples/s]
Dataset({
    features: ['input_ids', 'attention_mask'],
    num_rows: 5132
})
```

Step 5: Load GPT-2 Model

```
model = GPT2LMHeadModel.from_pretrained("gpt2")

model.resize_token_embeddings(len(tokenizer))
model.config.pad_token_id = tokenizer.pad_token_id

print("Using device:", "cuda" if torch.cuda.is_available() else "cpu")
```

```
model.safetensors: 100%  548M/548M [00:07<00:00, 132MB/s]
generation_config.json: 100%  124/124 [00:00<00:00, 2.52kB/s]
Using device: cuda
```

Step 6: Setup Data Collator

```
data_collator = DataCollatorForLanguageModeling(
    tokenizer=tokenizer,
    mlm=False,
)
```

Step 7: Setup Training Arguments and Trainer

```
training_args = TrainingArguments(
    output_dir=output_dir,
    overwrite_output_dir=True,

    num_train_epochs=2,
    per_device_train_batch_size=2,
    gradient_accumulation_steps=4,
    learning_rate=5e-5,
    warmup_steps=100,

    logging_steps=20,
```

```
    logging_dir=os.path.join(output_dir, "logs"),

    save_strategy="epoch",   # save at end of each epoch
    save_total_limit=2,

    fp16=True,
    gradient_checkpointing=True,

    report_to="none",
)
```

```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_train,
    tokenizer=tokenizer,
    data_collator=data_collator,
)
```

```
/tmp/ipython-input-3956626528.py:1: FutureWarning: `tokenizer` is deprecated and will be removed in version 5.0.0 for `Trainer._
  trainer = Trainer(
```

Step 8: Train the Model

```
trainer.train()
```

```
The tokenizer has new PAD/BOS/EOS tokens that differ from the model config and generation config. The model config and generat
`use_cache=True` is incompatible with gradient checkpointing. Setting `use_cache=False`...
`loss_type=None` was set in the config but it is unrecognized. Using the default loss: `ForCausalLMLoss`.
```

[1284/1284 06:37, Epoch 2/2]

Step	Training Loss
20	4.211900
40	4.015800
120	3.630600
140	3.759300
160	3.754000
180	3.610000
200	3.667600
220	3.632100
240	3.698000
260	3.705700
280	3.638700
300	3.673000
320	3.609400
340	3.622100
360	3.620900
380	3.584800

Step 9: Save the Model

```
trainer.save_model(output_dir)
tokenizer.save_pretrained(output_dir)

print("✅ Saved final GPT-2 baseline model to:", output_dir)
```

```
✅ Saved final GPT-2 baseline model to: /content/drive/My Drive/Colab Notebooks/CS 561: Topics in Data Privacy/Models/gpt2_basel
```