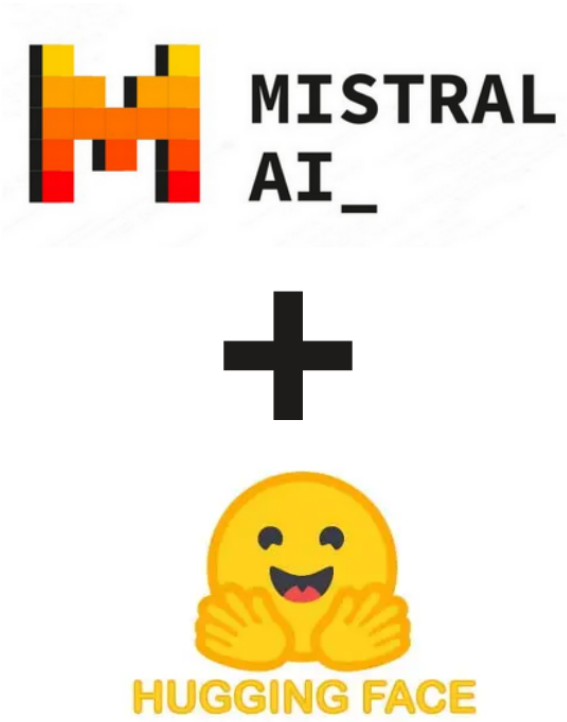


```
!pip install -q accelerate peft bitsandbytes pip install git+https://github.com/huggingface/transformers trl py7zr auto-gptq optimum
```

```
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 265.7/265.7 kB 2.4 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 168.3/168.3 kB 13.6 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 92.6/92.6 MB 10.4 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 139.1/139.1 kB 19.3 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.0/67.0 kB 9.4 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.8/4.8 MB 66.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 403.3/403.3 kB 48.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 507.1/507.1 kB 52.5 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 78.9/78.9 kB 11.3 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.1/2.1 MB 88.5 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 412.3/412.3 kB 48.4 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 138.9/138.9 kB 17.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.7/49.7 kB 5.6 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 93.1/93.1 kB 13.3 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.0/3.0 MB 103.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.3/1.3 MB 84.6 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.2/12.2 MB 45.6 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 46.0/46.0 kB 6.7 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 86.8/86.8 kB 11.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 115.3/115.3 kB 15.4 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.8/134.8 kB 19.2 MB/s eta 0:00:00
  Building wheel for transformers (pyproject.toml) ... done
```

```
from huggingface_hub import notebook_login
notebook_login()
```

```
Token is valid (permission: write).

n has been saved in your configured git credential helpe

ur token has been saved to /root/.cache/huggingface/tok

Login successful
```

```
import torch
from datasets import load_dataset, Dataset
from peft import LoraConfig, AutoPeftModelForCausalLM, prepare_model_for_kbit_training, get_peft_model
from transformers import AutoModelForCausalLM, AutoTokenizer, GPTQConfig, TrainingArguments
from trl import SFTTrainer
import os
```

```
data = load_dataset("samsum", split="train")
data_df = data.to_pandas()
data_df["text"] = data_df[["dialogue", "summary"]].apply(lambda x: "###Human: Summarize this following dialogue: " + x["dialogue"] + "\
print(data_df.iloc[0])
```

```
Downloading data: 100%                                                  6.06M/6.06M [00:01<00:00, 2.24MB/s]
Downloading data: 100%                                                  347k/347k [00:01<00:00, 264kB/s]
Downloading data: 100%                                                  335k/335k [00:01<00:00, 202kB/s]
Generating train split:       14732/0 [00:00<00:00, 88801.04 examples/s]
Generating test split:       819/0 [00:00<00:00, 29329.04 examples/s]
Generating validation split:       818/0 [00:00<00:00, 28853.50 examples/s]
id                                                          13818513
dialogue    Amanda: I baked  cookies. Do you want some?\r\...
summary     Amanda baked cookies and will bring Jerry some...
text        ###Human: Summarize this following dialogue: A...
Name: 0, dtype: object
```

```
data = Dataset.from_pandas(data_df)
tokenizer = AutoTokenizer.from_pretrained("TheBloke/Mistral-7B-Instruct-v0.1-GPTQ")
tokenizer.pad_token = tokenizer.eos_token
quantization_config_loading = GPTQConfig(bits=4, disable_exllama=True, tokenizer=tokenizer)

model = AutoModelForCausalLM.from_pretrained(
    "TheBloke/Mistral-7B-Instruct-v0.1-GPTQ",
    quantization_config=quantization_config_loading,
    device_map="auto",
)
print(model)
```

```
tokenizer_config.json:                                  1.46k/1.46k [00:00<00:00,
100%                                                    54.7kB/s]
tokenizer.model:                                        493k/493k [00:00<00:00,
100%                                                    21.7MB/s]
tokenizer.json:                                         1.80M/1.80M [00:00<00:00,
100%                                                    25.6MB/s]
special_tokens_map.json:                                    72.0/72.0 [00:00<00:00,
100%                                                        2.43kB/s]
Using `disable_exllama` is deprecated and will be removed in version 4.37. Use `use_e
config.json: 100%                                               963/963 [00:00<00:00, 55.5kB/s]
You passed `quantization_config` to `from_pretrained` but the model you're loading a
model.safetensors:                                      4.16G/4.16G [00:37<00:00,
100%                                                    189MB/s]
generation_config.json:                                     116/116 [00:00<00:00,
100%                                                        9.14kB/s]
MistralForCausalLM(
  (model): MistralModel(
    (embed_tokens): Embedding(32000, 4096, padding_idx=0)
    (layers): ModuleList(
      (0-31): 32 x MistralDecoderLayer(
        (self_attn): MistralAttention(
          (rotary_emb): MistralRotaryEmbedding()
          (k_proj): QuantLinear()
          (o_proj): QuantLinear()
          (q_proj): QuantLinear()
          (v_proj): QuantLinear()
        )
        (mlp): MistralMLP(
          (act_fn): SiLU()
          (down_proj): QuantLinear()
          (gate_proj): QuantLinear()
          (up_proj): QuantLinear()
```

```
model.config.use_cache=False
model.config.pretraining_tp=1
model.gradient_checkpointing_enable()
model = prepare_model_for_kbit_training(model)

peft_config = LoraConfig(
        r=16,
        lora_alpha=16,
        lora_dropout=0.05,
        bias="none",
        task_type="CAUSAL_LM",
        target_modules=["q_proj", "v_proj"],
)
model = get_peft_model(model, peft_config)
```

```
training_arguments = TrainingArguments(
    output_dir="mistral-finetuned-samsum",
    per_device_train_batch_size=8,
    gradient_accumulation_steps=1,
    optim="paged_adamw_32bit",
    learning_rate=2e-4,
    lr_scheduler_type="cosine",
    save_strategy="epoch",
    logging_steps=100,
    num_train_epochs=1,
    max_steps=250,
    fp16=True,
    push_to_hub=True
)
```

```
trainer = SFTTrainer(
    model=model,
    train_dataset=data,
    peft_config=peft_config,
    dataset_text_field="text",
    args=training_arguments,
    tokenizer=tokenizer,
    packing=False,
    max_seq_length=512
)
```

```
Map:                                                14732/14732 [00:06<00:00, 2458.66
100%                                                examples/s]
/usr/local/lib/python3.10/dist-packages/trl/trainer/sft_trainer.py:282: UserWarning:
```

```
trainer.train()
trainer.push_to_hub()
```

```
/usr/local/lib/python3.10/dist-packages/torch/utils/checkpoint.py:429: UserWarning: 1
  warnings.warn(
```


[250/250 42:22, Epoch 0/1]

Step	Training Loss
100	1.874100
200	1.763900

```
adapter_model.safetensors:                          27.3M/27.3M [00:01<00:00,
100%                                                34.3MB/s]
events.out.tfevents.1704598985.62e9329c87fc.325.0:          5.70k/5.70k
100%                                                [00:00<00:00,
                                                    10.9kB/s]
tokenizer.model: 100%                               493k/493k [00:00<00:00, 30.1kB/s]
training_args.bin:                                  4.73k/4.73k [00:00<00:00,
```

```
! cp -r /content/mistral-finetuned-samsum /content/drive/MyDrive/
```

```
cp: cannot create directory '/content/drive/MyDrive/': No such file or directory
```

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

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

```
from peft import AutoPeftModelForCausalLM
from transformers import GenerationConfig
from transformers import AutoTokenizer
import torch
tokenizer = AutoTokenizer.from_pretrained("/content/mistral-finetuned-samsum")

inputs = tokenizer("""
###Human: Summarize this following dialogue: Vasanth: I'm at the railway station in Chennai Karthik: No problems so far? Vasanth: no, e
###Assistant: """, return_tensors="pt").to("cuda")

model = AutoPeftModelForCausalLM.from_pretrained(
    "/content/mistral-finetuned-samsum",
    low_cpu_mem_usage=True,
    return_dict=True,
    torch_dtype=torch.float16,
    device_map="cuda")

generation_config = GenerationConfig(
    do_sample=True,
    top_k=1,
    temperature=0.1,
    max_new_tokens=25,
    pad_token_id=tokenizer.eos_token_id
)
```

```
import time
st_time = time.time()
outputs = model.generate(**inputs, generation_config=generation_config)
print(tokenizer.decode(outputs[0], skip_special_tokens=True))
print(time.time()-st_time)
```

```
###Human: Summarize this following dialogue: Vasanth: I'm at the railway station in Chennai Karthik: No problems so far? Vasanth: n
###Assistant:  Vasanth is at the railway station in Chennai. Everything is going smoothly. He will meet Karthik soon.
3.426438331604004
```