

Start coding or [generate](#) with AI.

FineTuning Using Google Gemma's Model

✓ Install All The Required Libraries

```
!pip3 install -q -U bitsandbytes==0.42.0
!pip3 install -q -U peft==0.8.2
!pip3 install -q -U trl==0.7.10
!pip3 install -q -U accelerate==0.27.1
!pip3 install -q -U datasets==2.17.0
!pip3 install -q -U transformers==4.38.0
!pip3 install pyarrow==14.0.1 requests==2.31.0
```



_____	105.0/105.0 MB 9.2 MB/s eta 0:00:00
_____	183.4/183.4 kB 5.5 MB/s eta 0:00:00
_____	21.3/21.3 MB 65.7 MB/s eta 0:00:00
_____	150.9/150.9 kB 3.4 MB/s eta 0:00:00
_____	547.8/547.8 kB 27.5 MB/s eta 0:00:00
_____	103.4/103.4 kB 14.4 MB/s eta 0:00:00
_____	39.9/39.9 MB 15.8 MB/s eta 0:00:00
_____	116.3/116.3 kB 17.0 MB/s eta 0:00:00
_____	64.9/64.9 kB 8.5 MB/s eta 0:00:00
_____	194.1/194.1 kB 19.3 MB/s eta 0:00:00
_____	134.8/134.8 kB 19.8 MB/s eta 0:00:00

ERROR: pip's dependency resolver does not currently take into account all the packages that might be required by your environment. Pip's dependency resolver does not currently take into account all the packages that might be required by your environment. google-colab 1.0.0 requires requests==2.31.0, but you have requests 2.32.3 which is incompatible. ibis-framework 8.0.0 requires pyarrow<16,>=2, but you have pyarrow 17.0.0 which is incompatible.

_____	279.7/279.7 kB 7.7 MB/s eta 0:00:00
_____	536.6/536.6 kB 14.1 MB/s eta 0:00:00
_____	8.5/8.5 MB 62.5 MB/s eta 0:00:00
_____	3.6/3.6 MB 93.4 MB/s eta 0:00:00

Collecting pyarrow==14.0.1

Downloading pyarrow-14.0.1-cp310-cp310-manylinux_2_28_x86_64.whl (38.0 MB)

_____	38.0/38.0 MB 17.5 MB/s eta 0:00:00
-------	------------------------------------

Collecting requests==2.31.0

Downloading requests-2.31.0-py3-none-any.whl (62 kB)

_____	62.6/62.6 kB 9.6 MB/s eta 0:00:00
-------	-----------------------------------

Requirement already satisfied: numpy>=1.16.6 in /usr/local/lib/python3.10/dist-packages

Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages

Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages

Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages

Installing collected packages: requests, pyarrow

Attempting uninstall: requests

Found existing installation: requests 2.32.3

```
Uninstalling requests-2.32.3:
  Successfully uninstalled requests-2.32.3
Attempting uninstall: pyarrow
  Found existing installation: pyarrow 17.0.0
  Uninstalling pyarrow-17.0.0:
    Successfully uninstalled pyarrow-17.0.0
Successfully installed pyarrow-14.0.1 requests-2.31.0
```

✓ Import some required Libraries

```
import os
import transformers
import torch
from google.colab import userdata
from datasets import load_dataset
from trl import SFTTrainer #Supervise FIne-Tuning
from peft import LoraConfig #Lora Technique
from transformers import AutoTokenizer, AutoModelForCausalLM
from transformers import BitsAndBytesConfig, GemmaTokenizer
```

```
os.environ["HF_TOKEN"] = userdata.get('HF_TOKEN')
```

```
model_id = "google/gemma-2b"
bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.bfloat16
)
```

```
tokenizer = AutoTokenizer.from_pretrained(model_id, token=os.environ["HF_TOKEN"])
model = AutoModelForCausalLM.from_pretrained(model_id,
                                              quantization_config=bnb_config,
                                              device_map={"":0},
                                              token=os.environ["HF_TOKEN"])
```

```

➔ /usr/local/lib/python3.10/dist-packages/huggingface_hub/file_download.py:1132: FutureWarning:
  warnings.warn(
tokenizer_config.json: 100% 33.6k/33.6k [00:00<00:00, 664kB/s]
tokenizer.model: 100% 4.24M/4.24M [00:00<00:00, 37.5MB/s]
tokenizer.json: 100% 17.5M/17.5M [00:00<00:00, 88.0MB/s]
special_tokens_map.json: 100% 636/636 [00:00<00:00, 15.4kB/s]
config.json: 100% 627/627 [00:00<00:00, 14.7kB/s]
model.safetensors.index.json: 100% 13.5k/13.5k [00:00<00:00, 294kB/s]
Downloading shards: 100% 2/2 [00:35<00:00, 14.70s/it]
model-00001-of- 4.95G/4.95G [00:34<00:00, 167MB/s]
00002.safetensors: 100%
model-00002-of- 67.1M/67.1M [00:00<00:00, 384MB/s]
00002.safetensors: 100%
Loading checkpoint shards: 100% 2/2 [00:21<00:00, 9.05s/it]

```

```

text = "Quote: Imagination is more,"
device = "cuda:0"
inputs = tokenizer(text, return_tensors="pt").to(device)

outputs = model.generate(**inputs, max_new_tokens=20)
print(tokenizer.decode(outputs[0], skip_special_tokens=True))

```

```

➔ Quote: Imagination is more, than knowledge.

I am a self-taught artist, born in 1985 in

```

```

text = "Quote: Imagination is more"
device = "cuda:0"
inputs = tokenizer(text, return_tensors="pt").to(device)

outputs = model.generate(**inputs, max_new_tokens=20)
print(tokenizer.decode(outputs[0], skip_special_tokens=True))

```

```

➔ Quote: Imagination is more important than knowledge. Knowledge is limited. Imagination e

- Albert Einstein

The

```

```

os.environ["WANDB_DISABLED"] = "false"

```

✓ Fine-Tuning With LoRA Configuration

```
lora_config = LoraConfig(
    r = 8,
    target_modules = ["q_proj", "o_proj", "k_proj", "v_proj",
                      "gate_proj", "up_proj", "down_proj"],
    task_type = "CAUSAL_LM",
)
```

```
from datasets import load_dataset
```

```
data = load_dataset("Abirate/english_quotes")
data = data.map(lambda samples: tokenizer(samples["quote"]), batched=True)
```



Downloading readme: 100%

5.55k/5.55k [00:00<00:00, 94.1kB/s]

Downloading data: 100%

647k/647k [00:00<00:00, 1.90MB/s]

Generating train split: 2508/0 [00:00<00:00, 23228.21 examples/s]

Map: 100%

2508/2508 [00:00<00:00, 6156.68 examples/s]

```
data["train"]["quote"]
```



```
[
    "Be yourself; everyone else is already taken.",
    "I'm selfish, impatient and a little insecure. I make mistakes, I am out of control and at times hard to handle. But if you can't handle me at my worst, then you sure as hell don't deserve me at my best.",
    "Two things are infinite: the universe and human stupidity; and I'm not sure about the universe.",
    "So many books, so little time.",
    "A room without books is like a body without a soul.",
    "Be who you are and say what you feel, because those who mind don't matter, and those who matter don't mind.",
    "You've gotta dance like there's nobody watching,Love like you'll never be hurt,Sing like there's nobody listening,And live like it's heaven on earth.",
    "You know you're in love when you can't fall asleep because reality is finally better than your dreams.",
    "You only live once, but if you do it right, once is enough.",
    "Be the change that you wish to see in the world.",
    "In three words I can sum up everything I've learned about life: it goes on.",
    "If you want to know what a man's like, take a good look at how he treats his inferiors, not his equals.",
    "Don't walk in front of meâ€¦ I may not followDon't walk behind meâ€¦ I may not leadWalk beside meâ€¦ just be my friend",
    "If you tell the truth, you don't have to remember anything.",
    "Friendship ... is born at the moment when one man says to another "What! You too? I thought that no one but myself . . .",
    "A friend is someone who knows all about you and still loves you.",
    "Always forgive your enemies; nothing annoys them so much.",
]
```

'"To live is the rarest thing in the world. Most people exist, that is all."',
 '"Live as if you were to die tomorrow. Learn as if you were to live forever."',
 '"Darkness cannot drive out darkness: only light can do that. Hate cannot drive out hate: only love can do that."',
 '"Without music, life would be a mistake."',
 '"We accept the love we think we deserve."',
 '"I am so clever that sometimes I don't understand a single word of what I am saying."',
 '"To be yourself in a world that is constantly trying to make you something else is the greatest accomplishment."',
 '"Here's to the crazy ones. The misfits. The rebels. The troublemakers. The round pegs in the square holes. The ones who see things differently. They're not fond of rules. And they have no respect for the status quo. You can quote them, disagree with them, glorify or vilify them. About the only thing you can't do is ignore them. Because they change things. They push the human race forward. And while some may see them as the crazy ones, we see genius. Because the people who are crazy enough to think they can change the world, are the ones who do."',
 '"Insanity is doing the same thing, over and over again, but expecting different results."',
 '"I believe that everything happens for a reason. People change so that you can learn to let go, things go wrong so that you appreciate them when they're right, you believe lies so you eventually learn to trust no one but yourself, and sometimes good things fall apart so better things can fall together."',
 '"Twenty years from now you will be more disappointed by the things that you didn't do than by the ones you did do. So throw off the bowlines. Sail away from the safe harbor. Catch the trade winds in your sails. Explore. Dream. Discover."',
 '"It is better to be hated for what you are than to be loved for what you are not."',
 '"Only once in your life, I truly believe, you find someone who can completely turn your world around. You tell them things that you've never shared with another soul and they absorb everything you say and actually want to hear more. You share hopes for the future. dreams that will never come true. goals that were never achieved and

```
def formatting_func(example):
    text = f"Quote: {example['quote']}\nAuthor: {example['author']}"
    return [text]
```

```
data['train']
```

```
Dataset({
  features: ['quote', 'author', 'tags', 'input_ids', 'attention_mask'],
  num_rows: 2508
})
```

```

trainer = SFTTrainer(
    model=model,
    train_dataset=data['train'],
    args=transformers.TrainingArguments(
        per_device_train_batch_size=1,
        gradient_accumulation_steps=4,
        warmup_steps=2,
        max_steps=100,
        learning_rate=2e-4,
        fp16=True,
        logging_steps=1,
        output_dir="outputs",
        optim="paged_adamw_8bit"
    ),
    peft_config=lora_config,
    formatting_func=formatting_func,
)

```

```

➡ /usr/local/lib/python3.10/dist-packages/huggingface_hub/file_download.py:1132: FutureWarning:
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/trl/trainer/sft_trainer.py:223: UserWarning: You
  warnings.warn(
Map: 100%                                2508/2508 [00:00<00:00, 15381.41 examples/s]
/usr/local/lib/python3.10/dist-packages/trl/trainer/sft_trainer.py:290: UserWarning: You
  warnings.warn(

```

```

trainer.train()

```



Step Training Loss

1	1.679700
2	0.629900
3	1.021600
4	1.029000
5	0.418700
6	1.222400
7	1.085300
8	0.329500
9	0.555300
10	0.467300
11	0.217800
12	0.574500
13	0.512700
14	0.130300
15	0.313300
16	0.249900
17	0.133500
18	0.330500
19	0.297500
20	0.070600
21	0.185600
22	0.166500
23	0.041300
24	0.138500
25	0.110200
26	0.025100
27	0.050700
28	0.037200
29	0.021500
30	0.014700

30	0.044 / 00
31	0.040800
32	0.013800
33	0.034400
34	0.020600
35	0.023500
36	0.040000
37	0.039800
38	0.017600
39	0.025100
40	0.026300
41	0.017600
42	0.040300
43	0.039200
44	0.013000
45	0.030700
46	0.031400
47	0.011900
48	0.040600
49	0.039500
50	0.014000
51	0.029400
52	0.026000
53	0.016000
54	0.041000
55	0.040300
56	0.015100
57	0.026600
58	0.027400
59	0.013000
60	0.039400
61	0.039500

62	0.010700
63	0.029600
64	0.026800
65	0.014200
66	0.039000
67	0.038900
68	0.013000
69	0.026100
70	0.025500
71	0.015000
72	0.039300
73	0.039300
74	0.012200
75	0.027700
76	0.027400
77	0.013100
78	0.038800
79	0.038800
80	0.011800
81	0.028000
82	0.025100
83	0.014700
84	0.039100
85	0.039000
86	0.014400
87	0.025200
88	0.027100
89	0.012300
90	0.038800
91	0.038800
92	0.012800

93	0.026200
94	0.026100
95	0.013000
96	0.038800
97	0.038800
98	0.013000
99	0.025900
100	0.025900

```
TrainOutput(global_step=100, training_loss=0.13974732809700072, metrics=
{'train_runtime': 74.0255, 'train_samples_per_second': 5.404,
 'train_steps_per_second': 1.351, 'total_flos': 55030101331200.0, 'train_loss':
```

```
text = "Quote: A women is like a tea bag,"
device = "cuda:0"
inputs = tokenizer(text, return_tensors="pt").to(device)

outputs = model.generate(**inputs, max_new_tokens=20)
print(tokenizer.decode(outputs[0], skip_special_tokens=True))
```



Quote: A women is like a tea bag, you can't tell how strong she is until you put her in
Author: Eleanor



```
text = "Quote: The opposite of love is not hate,"
device = "cuda:0"
inputs = tokenizer(text, return_tensors="pt").to(device)
```