```
!nvidia-smi
```

```
Thu Dec 18 16:10:34 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.54.15              Driver Version: 550.54.15      CUDA Version: 12.4      |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  Tesla T4                       Off | 00000000:00:04.0 Off   |                    0 |
| N/A   35C    P8                9W /  70W |      0MiB /  15360MiB   |     0%       Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI         PID   Type   Process name                          GPU Memory    |
|        ID   ID                                                            Usage         |
|=========================================================================================|
|  No running processes found                                                            |
+-----------------------------------------------------------------------------------------+
```

```
!pip install -U \
  transformers \
  datasets \
  accelerate \
  peft \
  bitsandbytes \
  trl
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.12/dist-packages (4.57.3)
Requirement already satisfied: datasets in /usr/local/lib/python3.12/dist-packages (4.4.1)
Requirement already satisfied: accelerate in /usr/local/lib/python3.12/dist-packages (1.12.0)
Requirement already satisfied: peft in /usr/local/lib/python3.12/dist-packages (0.18.0)
Requirement already satisfied: bitsandbytes in /usr/local/lib/python3.12/dist-packages (0.49.0)
Requirement already satisfied: trl in /usr/local/lib/python3.12/dist-packages (0.26.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from transformers) (3.20.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.34.0 in /usr/local/lib/python3.12/dist-packages (from transformers)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (25.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-packages (from transformers) (6.0.3)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2025.11.3
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from transformers) (2.32.4)
Requirement already satisfied: tokenizers<=0.23.0,>=0.22.0 in /usr/local/lib/python3.12/dist-packages (from transformers)
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.7.0)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.12/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: pyarrow>=21.0.0 in /usr/local/lib/python3.12/dist-packages (from datasets) (22.0.0)
Requirement already satisfied: dill<0.4.1,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from datasets) (0.3.8)
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (from datasets) (2.2.2)
Requirement already satisfied: httpx<1.0.0 in /usr/local/lib/python3.12/dist-packages (from datasets) (0.28.1)
Requirement already satisfied: xxhash in /usr/local/lib/python3.12/dist-packages (from datasets) (3.6.0)
Requirement already satisfied: multiprocess<0.70.19 in /usr/local/lib/python3.12/dist-packages (from datasets) (0.70.16)
Requirement already satisfied: fsspec<=2025.10.0,>=2023.1.0 in /usr/local/lib/python3.12/dist-packages (from fsspec[http]<
Requirement already satisfied: psutil in /usr/local/lib/python3.12/dist-packages (from accelerate) (5.9.5)
Requirement already satisfied: torch>=2.0.0 in /usr/local/lib/python3.12/dist-packages (from accelerate) (2.9.0+cu126)
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in /usr/local/lib/python3.12/dist-packages (from fsspec[http]<=2
Requirement already satisfied: anyio in /usr/local/lib/python3.12/dist-packages (from httpx<1.0.0->datasets) (4.12.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-packages (from httpx<1.0.0->datasets) (2025.11.12
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.12/dist-packages (from httpx<1.0.0->datasets) (1.0.
Requirement already satisfied: idna in /usr/local/lib/python3.12/dist-packages (from httpx<1.0.0->datasets) (3.11)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.12/dist-packages (from httpcore==1.*->httpx<1.0.0->data
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<1.0,>
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->transfo
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->transformers)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->accelerate) (75.2
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->accelerate) (1
Requirement already satisfied: networkx>=2.5.1 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->accelerate)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->accelerate) (3.1.6)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=2
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0-
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0-
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0
Requirement already satisfied: nvidia-nccl-cu12==2.27.5 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->acc
Requirement already satisfied: nvidia-nvshmem-cu12==3.3.20 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->ac
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0-
Requirement already satisfied: triton==3.5.0 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->accelerate) (3
```

Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas->datasets) (
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas->datasets) (2025.2)

```
!pip install evaluate rouge_score sacrebleu bert_score
```

Requirement already satisfied: evaluate in /usr/local/lib/python3.12/dist-packages (0.4.6)
Requirement already satisfied: rouge_score in /usr/local/lib/python3.12/dist-packages (0.1.2)
Requirement already satisfied: sacrebleu in /usr/local/lib/python3.12/dist-packages (2.5.1)
Requirement already satisfied: bert_score in /usr/local/lib/python3.12/dist-packages (0.3.13)
Requirement already satisfied: datasets>=2.0.0 in /usr/local/lib/python3.12/dist-packages (from evaluate) (4.4.1)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.12/dist-packages (from evaluate) (2.0.2)
Requirement already satisfied: dill in /usr/local/lib/python3.12/dist-packages (from evaluate) (0.3.8)
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (from evaluate) (2.2.2)
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.12/dist-packages (from evaluate) (2.32.4)
Requirement already satisfied: tqdm>=4.62.1 in /usr/local/lib/python3.12/dist-packages (from evaluate) (4.67.1)
Requirement already satisfied: xxhash in /usr/local/lib/python3.12/dist-packages (from evaluate) (3.6.0)
Requirement already satisfied: multiprocess in /usr/local/lib/python3.12/dist-packages (from evaluate) (0.70.16)
Requirement already satisfied: fsspec>=2021.05.0 in /usr/local/lib/python3.12/dist-packages (from fsspec[http]>=2021.05.0-
Requirement already satisfied: huggingface-hub>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from evaluate) (0.36.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.12/dist-packages (from evaluate) (25.0)
Requirement already satisfied: absl-py in /usr/local/lib/python3.12/dist-packages (from rouge_score) (1.4.0)
Requirement already satisfied: nltk in /usr/local/lib/python3.12/dist-packages (from rouge_score) (3.9.1)
Requirement already satisfied: six>=1.14.0 in /usr/local/lib/python3.12/dist-packages (from rouge_score) (1.17.0)
Requirement already satisfied: portalocker in /usr/local/lib/python3.12/dist-packages (from sacrebleu) (3.2.0)
Requirement already satisfied: regex in /usr/local/lib/python3.12/dist-packages (from sacrebleu) (2025.11.3)
Requirement already satisfied: tabulate>=0.8.9 in /usr/local/lib/python3.12/dist-packages (from sacrebleu) (0.9.0)
Requirement already satisfied: colorama in /usr/local/lib/python3.12/dist-packages (from sacrebleu) (0.4.6)
Requirement already satisfied: lxml in /usr/local/lib/python3.12/dist-packages (from sacrebleu) (6.0.2)
Requirement already satisfied: torch>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from bert_score) (2.9.0+cu126)
Requirement already satisfied: transformers>=3.0.0 in /usr/local/lib/python3.12/dist-packages (from bert_score) (4.57.3)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (from bert_score) (3.10.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from datasets>=2.0.0->evaluate) (3.20.
Requirement already satisfied: pyarrow>=21.0.0 in /usr/local/lib/python3.12/dist-packages (from datasets>=2.0.0->evaluate)
Requirement already satisfied: httpx<1.0.0 in /usr/local/lib/python3.12/dist-packages (from datasets>=2.0.0->evaluate) (0.
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-packages (from datasets>=2.0.0->evaluate) (6.
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in /usr/local/lib/python3.12/dist-packages (from fsspec[http]>=2
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub>=0.7.
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas->evaluate) (
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas->evaluate) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas->evaluate) (2025.2)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests>=2.19.0-
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests>=2.19.0->evaluate) (
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests>=2.19.0->evalu
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests>=2.19.0->evalu
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch>=1.0.0->bert_score) (75.2
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch>=1.0.0->bert_score) (1
Requirement already satisfied: networkx>=2.5.1 in /usr/local/lib/python3.12/dist-packages (from torch>=1.0.0->bert_score)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch>=1.0.0->bert_score) (3.1.6)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.0
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch>=1.0
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch>=1.0.0-
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch>=1.0.0-
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/python3.12/dist-packages (from torch>=1.0.0->
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.0.0
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib/python3.12/dist-packages (from torch>=1.0.
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib/python3.12/dist-packages (from torch>=1.0.
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in /usr/local/lib/python3.12/dist-packages (from torch>=1.0.0
Requirement already satisfied: nvidia-nccl-cu12==2.27.5 in /usr/local/lib/python3.12/dist-packages (from torch>=1.0.0->ber
Requirement already satisfied: nvidia-nvshmem-cu12==3.3.20 in /usr/local/lib/python3.12/dist-packages (from torch>=1.0.0->
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.0.0->be

```
from datasets import load_dataset

dataset = load_dataset(
    "json",
    data_files="/content/drive/MyDrive/LLM_Final_Project/afet_llm_final_dataset.jsonl",
    split="train"
)

print(dataset)
print("Toplam örnek:", len(dataset))
print("İlk örnek:\n", dataset[0])
```

```
Dataset({
    features: ['instruction', 'input', 'output'],
    num_rows: 400
})
Toplam örnek: 400
İlk örnek:
 {'instruction': 'Deprem sonrası iletişim tamamen kesildiğinde enkaz altında kalan yaralıya ilk yaklaşım nasıl olmalıdır?',
```

```
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM, BitsAndBytesConfig
```

```python
model_name = "Qwen/Qwen2-1.5B-Instruct"

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

tokenizer = AutoTokenizer.from_pretrained(
    model_name,
    trust_remote_code=True
)
tokenizer.pad_token = tokenizer.eos_token

model = AutoModelForCausalLM.from_pretrained(
    model_name,
    quantization_config=bnb_config,
    device_map="auto",
    trust_remote_code=True
)
```

```
/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
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: 1.29k/? [00:00<00:00, 42.0kB/s]

vocab.json: 2.78M/? [00:00<00:00, 34.5MB/s]

merges.txt: 1.67M/? [00:00<00:00, 9.90MB/s]

tokenizer.json: 7.03M/? [00:00<00:00, 48.0MB/s]

config.json: 100%   660/660 [00:00<00:00, 13.1kB/s]

model.safetensors: 100%   3.09G/3.09G [00:38<00:00, 84.3MB/s]

generation_config.json: 100%   242/242 [00:00<00:00, 25.5kB/s]

```python
from peft import LoraConfig, get_peft_model
import torch

lora_config = LoraConfig(
    r=16,
    lora_alpha=32,
    target_modules=["q_proj", "v_proj"],
    lora_dropout=0.05,
    bias="none",
    task_type="CAUSAL_LM"
)

model = get_peft_model(model, lora_config)
model.print_trainable_parameters()
```

```
trainable params: 2,179,072 || all params: 1,545,893,376 || trainable%: 0.1410
```

```python
from transformers import Trainer, TrainingArguments

def format_example(example):
    return f"""### Talimat:
{example['instruction']}

### Girdi:
{example['input']}

### Yanıt:
{example['output']}"""

def tokenize_function(example):
    text = format_example(example)

    tokenized = tokenizer(
        text,
        truncation=True,
        max_length=1024,
        padding="max_length"
```

```
        )

        tokenized["labels"] = tokenized["input_ids"].copy()
        return tokenized

    tokenized_dataset = dataset.map(tokenize_function, remove_columns=dataset.column_names)
```

Map: 100%   400/400 [00:00<00:00, 1619.85 examples/s]

```
from transformers import TrainingArgments

training_args = TrainingArguments(
    output_dir="/content/drive/MyDrive/LLM_Final_Project/afetllm_lora",
    per_device_train_batch_size=1,
    gradient_accumulation_steps=4,
    learning_rate=2e-4,
    num_train_epochs=3,
    fp16=True,
    logging_steps=10,
    save_steps=100,
    save_total_limit=2,
    report_to="none"
)
```

```
import torch
torch.cuda.empty_cache()
```

```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_dataset
)

trainer.train()
```

[300/300 10:42, Epoch 3/3]

Step	Training Loss
10	8.083000
20	0.721600
30	0.264500
40	0.219800
50	0.207500
60	0.187300
70	0.182100
80	0.177100
90	0.163300
100	0.172800
110	0.177500
120	0.181700
130	0.158300
140	0.159600
150	0.156000
160	0.152200
170	0.144800
180	0.154700
190	0.147700
200	0.125800
210	0.133000
220	0.142200
230	0.136900
240	0.132900
250	0.137400
260	0.161500
270	0.131100
280	0.153400
290	0.139900
300	0.152000

```python
from peft import PeftModel
from transformers import AutoModelForCausalLM

base_model = AutoModelForCausalLM.from_pretrained(
    "Qwen/Qwen2-1.5B-Instruct",
    quantization_config=bnb_config,
    device_map="auto",
    trust_remote_code=True
)

model = PeftModel.from_pretrained(
    base_model,
    "/content/drive/MyDrive/LLM_Final_Project/afetllm_lora/checkpoint-300"
)

model.eval()
```

```
PeftModelForCausalLM(
  (base_model): LoraModel(
    (model): Qwen2ForCausalLM(
      (model): Qwen2Model(
        (embed_tokens): Embedding(151936, 1536)
        (layers): ModuleList(
          (0-27): 28 x Qwen2DecoderLayer(
            (self_attn): Qwen2Attention(
              (q_proj): lora.Linear4bit(
                (base_layer): Linear4bit(in_features=1536, out_features=1536, bias=True)
                (lora_dropout): ModuleDict(
```

```
                        (default): Dropout(p=0.05, inplace=False)
                      )
                      (lora_A): ModuleDict(
                        (default): Linear(in_features=1536, out_features=16, bias=False)
                      )
                      (lora_B): ModuleDict(
                        (default): Linear(in_features=16, out_features=1536, bias=False)
                      )
                      (lora_embedding_A): ParameterDict()
                      (lora_embedding_B): ParameterDict()
                      (lora_magnitude_vector): ModuleDict()
                    )
                    (k_proj): Linear4bit(in_features=1536, out_features=256, bias=True)
                    (v_proj): lora.Linear4bit(
                      (base_layer): Linear4bit(in_features=1536, out_features=256, bias=True)
                      (lora_dropout): ModuleDict(
                        (default): Dropout(p=0.05, inplace=False)
                      )
                      (lora_A): ModuleDict(
                        (default): Linear(in_features=1536, out_features=16, bias=False)
                      )
                      (lora_B): ModuleDict(
                        (default): Linear(in_features=16, out_features=256, bias=False)
                      )
                      (lora_embedding_A): ParameterDict()
                      (lora_embedding_B): ParameterDict()
                      (lora_magnitude_vector): ModuleDict()
                    )
                    (o_proj): Linear4bit(in_features=1536, out_features=1536, bias=False)
                  )
                  (mlp): Qwen2MLP(
                    (gate_proj): Linear4bit(in_features=1536, out_features=8960, bias=False)
                    (up_proj): Linear4bit(in_features=1536, out_features=8960, bias=False)
                    (down_proj): Linear4bit(in_features=8960, out_features=1536, bias=False)
                    (act_fn): SiLUActivation()
                  )
                  (input_layernorm): Qwen2RMSNorm((1536,), eps=1e-06)
                  (post_attention_layernorm): Qwen2RMSNorm((1536,), eps=1e-06)
                )
              )
              (norm): Qwen2RMSNorm((1536,), eps=1e-06)
              (rotary_emb): Qwen2RotaryEmbedding()
            )
            (lm_head): Linear(in_features=1536, out_features=151936, bias=False)
          )
        )
```

```python
from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained(
    "Qwen/Qwen2-1.5B-Instruct",
    trust_remote_code=True
)
tokenizer.pad_token = tokenizer.eos_token
```

```python
prompt = """### Talimat:
Deprem sonrası iletişimin tamamen kesildiği bir ortamda yaralı tahliyesi nasıl planlanmalıdır?

### Girdi:
GSM ve telsiz yok, artçı sarsıntılar devam ediyor.

### Yanıt:
"""
```

```python
inputs = tokenizer(
    prompt,
    return_tensors="pt"
).to("cuda")
```

```python
import torch

with torch.no_grad():
    outputs = model.generate(
        **inputs,
        max_new_tokens=200,
        do_sample=True,
        temperature=0.7,
        top_p=0.9
    )
```

```
    print(tokenizer.decode(outputs[0], skip_special_tokens=True))
```

Talimat:
Deprem sonrası iletişimin tamamen kesildiği bir ortamda yaralı tahliyesi nasıl planlanmalıdır?

Girdi:
GSM ve telsiz yok, artçı sarsıntılar devam ediyor.

Yanıt:
Tahliye planı genel bir yasallaştırma ile sağlanır. İletişim olmayan bölgelerde herhangi bir temsili yapmaz.

```
with torch.no_grad():
    outputs = model.generate(
        **inputs,
        max_new_tokens=200,
        do_sample=False,       # ← önemli
        temperature=0.0        # ← önemli
    )

print(tokenizer.decode(outputs[0], skip_special_tokens=True))
```

The following generation flags are not valid and may be ignored: ['temperature', 'top_p', 'top_k']. Set `TRANSFORMERS_VERBOS
Talimat:
Deprem sonrası iletişimin tamamen kesildiği bir ortamda yaralı tahliyesi nasıl planlanmalıdır?

Girdi:
GSM ve telsiz yok, artçı sarsıntılar devam ediyor.

Yanıt:
Tahliye planı genel kitleye açık olarak sunulmalı. Yaralıların güvenliği ve bilgi sağlamlığı önemli.

```
base_model.eval()

with torch.no_grad():
    outputs = base_model.generate(
        **inputs,
        max_new_tokens=200
    )

print(tokenizer.decode(outputs[0], skip_special_tokens=True))
```

Talimat:
Deprem sonrası iletişimin tamamen kesildiği bir ortamda yaralı tahliyesi nasıl planlanmalıdır?

Girdi:
GSM ve telsiz yok, artçı sarsıntılar devam ediyor.

Yanıt:
Yaralıların temsilci olarak belirli etkinlikleri takip etmesini önerilir. Yaralıların hareket planını oluşturmada bilgiye er

```
sop_prompt = """### Talimat:
Deprem sonrası iletişimin tamamen kesildiği bir ortamda, AFAD benzeri resmi
arama-kurtarma prosedürlerine göre yaralı tahliyesi nasıl planlanmalıdır?

### Girdi:
GSM, telsiz ve internet yok.
Artçı sarsıntılar devam ediyor.
Binada gaz ve yapısal hasar riski mevcut.
Birden fazla yaralı var.

### Yanıt:
"""
```

```
base_model.eval()

inputs = tokenizer(
    sop_prompt,
    return_tensors="pt"
).to(base_model.device)

with torch.no_grad():
    output_base = base_model.generate(
        **inputs,
        max_new_tokens=200
    )

print("=== BASE MODEL ÇIKTISI ===")
```

```
print(tokenizer.decode(output_base[0], skip_special_tokens=True))
```

```
=== BASE MODEL ÇIKTISI ===
### Talimat:
Deprem sonrası iletişimin tamamen kesildiği bir ortamda, AFAD benzeri resmi
arama-kurtarma prosedürlerine göre yaralı tahliyesi nasıl planlanmalıdır?

### Girdi:
GSM, telsiz ve internet yok.
Artçı sarsıntılar devam ediyor.
Binada gaz ve yapısal hasar riski mevcut.
Birden fazla yaralı var.

### Yanıt:
Yaralılar 30 dakika sürenin başlangıcını belirlemektedir. Yerel kendi koruyucular tarafından belirlenen temel alanlar, korku
```

```
model.eval()

inputs = tokenizer(
    sop_prompt,
    return_tensors="pt"
).to(model.device)

with torch.no_grad():
    output_ft = model.generate(
        **inputs,
        max_new_tokens=200
    )

print("=== FINE-TUNED MODEL ÇIKTISI ===")
print(tokenizer.decode(output_ft[0], skip_special_tokens=True))
```

```
=== FINE-TUNED MODEL ÇIKTISI ===
### Talimat:
Deprem sonrası iletişimin tamamen kesildiği bir ortamda, AFAD benzeri resmi
arama-kurtarma prosedürlerine göre yaralı tahliyesi nasıl planlanmalıdır?

### Girdi:
GSM, telsiz ve internet yok.
Artçı sarsıntılar devam ediyor.
Binada gaz ve yapısal hasar riski mevcut.
Birden fazla yaralı var.

### Yanıt:
Yaralıların durumları ve çevresel durumu bilinirken, yaralıların güvenliği korunmalı.
```

```python
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM, BitsAndBytesConfig
from peft import PeftModel

# 1. Yapılandırmayı Yeniden Tanımla (Eksik olan buydu)
bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.float16
)

model_name = "Qwen/Qwen2-1.5B-Instruct"

# 2. Base Modeli Yükle
base_model = AutoModelForCausalLM.from_pretrained(
    model_name,
    quantization_config=bnb_config,
    device_map="auto",
    trust_remote_code=True
)

# 3. Kendi Eğittiğin LoRA Ağırlıklarını Yükle
# Klasör yolunun doğru olduğundan emin ol (checkpoint-300 veya en sonuncusu)
model = PeftModel.from_pretrained(
    base_model,
    "/content/drive/MyDrive/LLM_Final_Project/afetllm_lora/checkpoint-300"
)

# 4. Tokenizer'ı Yükle
tokenizer = AutoTokenizer.from_pretrained(model_name, trust_remote_code=True)
tokenizer.pad_token = tokenizer.eos_token

model.eval()
print("Model başarıyla yüklendi, değerlendirmeye hazır!")
```

```
/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
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(
```

config.json: 100%   660/660 [00:00<00:00, 21.6kB/s]

model.safetensors: 100%   3.09G/3.09G [00:30<00:00, 229MB/s]

generation_config.json: 100%   242/242 [00:00<00:00, 27.8kB/s]

tokenizer_config.json: 1.29k/? [00:00<00:00, 98.4kB/s]

vocab.json: 2.78M/? [00:00<00:00, 46.7MB/s]

merges.txt: 1.67M/? [00:00<00:00, 49.7MB/s]

tokenizer.json: 7.03M/? [00:00<00:00, 101MB/s]

Model başarıyla yüklendi, değerlendirmeye hazır!

```python
from datasets import load_dataset

# Veriyi yükle
dataset = load_dataset(
    "json",
    data_files="/content/drive/MyDrive/LLM_Final_Project/afet_llm_final_dataset.jsonl",
    split="train"
)

# Veriyi %10 test olacak şekilde böl (seed=42 ile her zaman aynı örnekleri seçeriz)
dataset_split = dataset.train_test_split(test_size=0.1, seed=42)
test_dataset = dataset_split["test"]

print(f"Test edilecek örnek sayısı: {len(test_dataset)}")
```

Generating train split: 400/0 [00:00<00:00, 3456.65 examples/s]

Test edilecek örnek sayısı: 40

```python
import evaluate
import torch
from tqdm import tqdm
import numpy as np

# Metrikleri yükle
rouge = evaluate.load("rouge")
```

```
bleu = evaluate.load("sacrebleu")

predictions = []
references = []

print("Model yanıt üretiyor, lütfen bekleyin...")

# Test setindeki her örnek için tahmin üret
for example in tqdm(test_dataset):
    # Eğitim formatına uygun prompt oluştur
    prompt = f"### Talimat:\n{example['instruction']}\n\n### Girdi:\n{example['input']}\n\n### Yanıt:\n"

    inputs = tokenizer(prompt, return_tensors="pt").to("cuda")

    with torch.no_grad():
        outputs = model.generate(
            **inputs,
            max_new_tokens=150,
            do_sample=False, # Skor ölçerken rastgeleliği kapatıyoruz
```