```
!nvidia-smi
```

```
Thu Apr 25 13:09:20 2024
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| 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   71C    P8              11W /  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 transformers[sentencepiece] datasets sacrebleu rouge_score py7zr -q
```

```
!pip install --upgrade accelerate
!pip uninstall -y transformers accelerate
!pip install transformers accelerate
```

```
Requirement already satisfied: accelerate in /usr/local/lib/python3.10/dist-packages (0.29.3)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from accelerate) (1.25.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from accelerate) (24.0)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from accelerate) (5.9.5)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.10/dist-packages (from accelerate) (6.0.1)
Requirement already satisfied: torch>=1.10.0 in /usr/local/lib/python3.10/dist-packages (from accelerate) (2.2.1+cu121)
Requirement already satisfied: huggingface-hub in /usr/local/lib/python3.10/dist-packages (from accelerate) (0.22.2)
Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.10/dist-packages (from accelerate) (0.4.3)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (3.13.4)
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerat
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (1.12)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (3.3)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (3.1.3)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (2023.6.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->a
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0-
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->a
Requirement already satisfied: nvidia-cudnn-cu12==8.9.2.26 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accele
Requirement already satisfied: nvidia-cublas-cu12==12.1.3.1 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accel
Requirement already satisfied: nvidia-cufft-cu12==11.0.2.54 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accel
Requirement already satisfied: nvidia-curand-cu12==10.3.2.106 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->acc
Requirement already satisfied: nvidia-cusolver-cu12==11.4.5.107 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->a
Requirement already satisfied: nvidia-cusparse-cu12==12.1.0.106 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->a
Requirement already satisfied: nvidia-nccl-cu12==2.19.3 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->acceleratt
Requirement already satisfied: nvidia-nvtx-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->acceler
Requirement already satisfied: triton==2.2.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate) (2.2.0)
Requirement already satisfied: nvidia-nvjitlink-cu12 in /usr/local/lib/python3.10/dist-packages (from nvidia-cusolver-cu12==11.4.
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from huggingface-hub->accelerate) (2.31.0)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub->accelerate) (4.66.2
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.10.0->accelerate
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hu
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub->accelerat
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub->acc
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub->acc
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.10.0->accelerate) (1
Found existing installation: transformers 4.40.1
Uninstalling transformers-4.40.1:
  Successfully uninstalled transformers-4.40.1
Found existing installation: accelerate 0.29.3
Uninstalling accelerate-0.29.3:
  Successfully uninstalled accelerate-0.29.3
Collecting transformers
  Using cached transformers-4.40.1-py3-none-any.whl (9.0 MB)
Collecting accelerate
  Using cached accelerate-0.29.3-py3-none-any.whl (297 kB)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers) (3.13.4)
Requirement already satisfied: huggingface-hub<1.0,>=0.19.3 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.22.
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (1.25.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (24.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (2023.12.25)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers) (2.31.0)
Requirement already satisfied: tokenizers<0.20,>=0.19 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.19.1)
Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.4.3)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.66.2)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from accelerate) (5.9.5)
```

```
from transformers import pipeline
from datasets import load_dataset
import matplotlib.pyplot as plt
import pandas as pd
import torch.nn as nn

from transformers import AutoModelForSeq2SeqLM, AutoTokenizer,BartConfig

import torch
```

```
device = "cuda" if torch.cuda.is_available() else "cpu"
```

```
device
```

```
'cuda'
```

```
tokenizer = AutoTokenizer.from_pretrained('google/pegasus-cnn_dailymail')
```

```
spiece.model: 100%  1.91M/1.91M [00:01<00:00, 1.92MB/s]
```

```
import accelerate

# Create an instance of DataLoaderConfiguration with the desired parameters
dataloader_config = accelerate.DataLoaderConfiguration(
    dispatch_batches=None,
    split_batches=False,
    even_batches=True,
    use_seedable_sampler=True
)

# Pass the dataloader_config instance to the Accelerator constructor
accelerator = accelerate.Accelerator(dataloader_config=dataloader_config)
```

```
model_peg = AutoModelForSeq2SeqLM.from_pretrained('google/pegasus-cnn_dailymail').to(accelerator.device)
```

```
pytorch_model.bin: 100%  2.28G/2.28G [00:37<00:00, 61.7MB/s]
Some weights of PegasusForConditionalGeneration were not initialized from the model c
You should probably TRAIN this model on a down-stream task to be able to use it for p
generation_config.json: 100%  280/280 [00:00<00:00, 21.1kB/s]
```

```
dataset = load_dataset('cnn_dailymail','3.0.0')
```

```
dataset
```

```
DatasetDict({
    train: Dataset({
        features: ['article', 'highlights', 'id'],
        num_rows: 287113
    })
    validation: Dataset({
        features: ['article', 'highlights', 'id'],
        num_rows: 13368
    })
    test: Dataset({
        features: ['article', 'highlights', 'id'],
        num_rows: 11490
    })
})
```

```
from datasets import DatasetDict
import random

# Assuming
original_train_data = dataset['train']


total_rows = len(original_train_data)


desired_size = 2000

indices_to_keep = random.sample(range(total_rows), desired_size)

dataset['train'] = original_train_data.select(indices_to_keep)
```

```
from datasets import DatasetDict
import random

# Assuming
original_train_data = dataset['test']


total_rows = len(original_train_data)


desired_size = 2000

indices_to_keep = random.sample(range(total_rows), desired_size)

dataset['test'] = original_train_data.select(indices_to_keep)
```

```
from datasets import DatasetDict
import random

# Assuming
original_train_data = dataset['validation']


total_rows = len(original_train_data)


desired_size = 2000

indices_to_keep = random.sample(range(total_rows), desired_size)

dataset['validation'] = original_train_data.select(indices_to_keep)
```

```
dataset
```

```
DatasetDict({
        train: Dataset({
            features: ['article', 'highlights', 'id'],
            num_rows: 2000
        })
        validation: Dataset({
            features: ['article', 'highlights', 'id'],
            num_rows: 2000
        })
        test: Dataset({
            features: ['article', 'highlights', 'id'],
            num_rows: 2000
        })
    })
```

```
def convert_data(batch):
  input_encoding = tokenizer(batch['article'], max_length=512, truncation = True )

  with tokenizer.as_target_tokenizer():
    target_encoding = tokenizer(batch['highlights'], max_length=200, truncation = True )


  return {
      'input_ids' : input_encoding['input_ids'],
      'attention_mask' : input_encoding['attention_mask'],
      'labels' : target_encoding['input_ids']
  }
```

```
data_samsum_pt = dataset.map(convert_data, batched= True )
```

```
Map: 100%                                                2000/2000 [00:10<00:00, 191.27 examples/s]
/usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils_base.py:3921: UserWarning: `as_target_tokenizer` is deprecat
  warnings.warn(
Map: 100%                                                2000/2000 [00:06<00:00, 334.54 examples/s]
Map: 100%                                                2000/2000 [00:07<00:00, 287.88 examples/s]
```

```
data_samsum_pt['test']
```

```
Dataset({
    features: ['article', 'highlights', 'id', 'input_ids', 'attention_mask', 'labels'],
    num_rows: 2000
})
```

```
from transformers import DataCollatorForSeq2Seq, TrainingArguments, Trainer

seq_data_collater = DataCollatorForSeq2Seq(tokenizer, model =model_peg )
```

```
training_arg = TrainingArguments(
    output_dir = 'Pegasus_model',
    num_train_epochs=1,
    warmup_steps=500,
    per_device_train_batch_size=1,
    per_device_eval_batch_size=1,
    weight_decay = 0.01 ,
    logging_steps=10,
    evaluation_strategy='steps',
    eval_steps = 500,
    save_steps = 1e6 ,
    gradient_accumulation_steps=16
)
```

```
trainer = Trainer(model=model_peg,
                  args=training_arg,
                  tokenizer=tokenizer,
                  data_collator=seq_data_collater,
                  train_dataset=data_samsum_pt['test'],
                  eval_dataset=data_samsum_pt['validation'])
```

```
trainer.train()
```

```
[125/125 11:38, Epoch 1/1]
```

Step	Training Loss	Validation Loss

```
TrainOutput(global_step=125, training_loss=2.2831634368896485, metrics={'train_runtime': 705.5031, 'train_samples_per_second':
2.835, 'train_steps_per_second': 0.177, 'total_flos': 2707747013025792.0, 'train_loss': 2.2831634368896485, 'epoch': 1.0})
```

```
trainer.evaluate(data_samsum_pt['test'])
```

```
[2000/2000 04:13]
{'eval_loss': 1.693496584892273,
 'eval_runtime': 228.8487,
 'eval_samples_per_second': 8.739,
 'eval_steps_per_second': 8.739,
 'epoch': 1.0}
```

```
# Input text for summarization
input_text = "Your input text goes here. This could be a longer piece of text that you want to summarize."

# Tokenize input text
input_ids = tokenizer.encode(input_text, return_tensors='pt')

# Generate summary using the model
summary_ids = trainer.generate(input_ids, max_length=50, num_beams=4, early_stopping=True)

# Decode the generated summary
summary_text = tokenizer.decode(summary_ids[0], skip_special_tokens=True)

# Print the generated summary
print("Generated Summary:", summary_text)
```

```
from google.colab import drive
from transformers import PegasusForConditionalGeneration

# Mount Google Drive
drive.mount('/content/drive')

# Assuming your model is stored in a variable called model_peg
model_peg.save_pretrained('/content/drive/MyDrive/pretrained_data')
```

Start coding or generate with AI.