```
!pip install -U transformers -q
!pip install -U accelerate -q
!pip install keras_nlp -q
!pip install datasets -q
!pip install huggingface-hub -q
!pip install rouge-score -q
```

```
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.7/7.7 MB 57.0 MB/s eta 0:00:00
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 295.0/295.0 kB 27.9 MB/s eta 0:00:00
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.8/3.8 MB 71.4 MB/s eta 0:00:00
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.3/1.3 MB 62.7 MB/s eta 0:00:00
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 268.8/268.8 kB 21.9 MB/s eta 0:00:00
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 258.1/258.1 kB 3.6 MB/s eta 0:00:00
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 590.1/590.1 kB 4.7 MB/s eta 0:00:00
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 950.8/950.8 kB 17.7 MB/s eta 0:00:00
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.5/6.5 MB 23.9 MB/s eta 0:00:00
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 519.6/519.6 kB 7.8 MB/s eta 0:00:00
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 115.3/115.3 kB 11.6 MB/s eta 0:00:00
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 194.1/194.1 kB 16.6 MB/s eta 0:00:00
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.8/134.8 kB 9.4 MB/s eta 0:00:00
       Preparing metadata (setup.py) ... done
       Building wheel for rouge-score (setup.py) ... done
```

```
import nltk
nltk.download("all",quiet=True)
import torch
import numpy as np
import tensorflow as tf
from tensorflow import keras
```

```
from datasets import load_dataset
dataset = load_dataset("xsum", split="train")
print(dataset)
```

```
Downloading builder script: 100%                     5.76k/5.76k [00:00<00:00, 267kB/s]
Downloading readme: 100%                             6.24k/6.24k [00:00<00:00, 405kB/s]
Downloading data files: 100%                         2/2 [00:21<00:00, 9.07s/it]
Downloading data: 100%                               255M/255M [00:16<00:00, 18.0MB/s]
Downloading data:                                    2.72M/? [00:00<00:00, 9.22MB/s]
Generating train split: 100%                         204045/204045 [01:06<00:00, 4133.31 examples/s]
Generating validation split: 100%                    11332/11332 [00:25<00:00, 407.33 examples/s]
Generating test split: 100%                          11334/11334 [00:27<00:00, 501.37 examples/s]
Dataset({
    features: ['document', 'summary', 'id'],
    num_rows: 204045
})
```

```
print(dataset[0])
```

```
{'document': 'The full cost of damage in Newton Stewart, one of the areas worst affected, is still being assessed.\nRepair work is ongoing in Hawick and many roads in Peeblesshire remain badly affected by standing water.\nTrains on the w
```

```
datasets = dataset.train_test_split(train_size=0.05,test_size=0.02)
```

```
print(len(datasets['train']))
print(len(datasets['test']))
```

```
10202
4081
```

```
train = datasets['train']
test = datasets['test']
```

```
MAX_INPUT_LENGTH = 1024
MIN_TARGET_LENGTH = 5
MAX_TARGET_LENGTH = 128
BATCH_SIZE = 8
LEARNING_RATE = 0.002
MAX_EPOCHS = 20
MODEL_CHECKPOINT = "t5-small" # Name of Model
```

```
from transformers import AutoTokenizer
tokenizer = AutoTokenizer.from_pretrained(MODEL_CHECKPOINT)
```

```
Downloading (…)okenizer_config.json: 100%      2.32k/2.32k [00:00<00:00, 145kB/s]

Downloading (…)ve/main/spiece.model: 100%      792k/792k [00:00<00:00, 3.37MB/s]

Downloading (…)/main/tokenizer.json: 100%      1.39M/1.39M [00:00<00:00, 5.62MB/s]
```

```
if MODEL_CHECKPOINT in ["t5-small", "t5-base"]:
  prefix = "summarize: "
else:
  prefix = ""
```

```
#Preprocessing
def preprocess_function(examples):
  inputs = [prefix + doc for doc in examples["document"]]
  model_inputs = tokenizer(inputs, max_length=MAX_INPUT_LENGTH,truncation=True)
  # Setup the tokenizer for targets
  with tokenizer.as_target_tokenizer():
    labels = tokenizer(
    examples["summary"], max_length=MAX_TARGET_LENGTH, truncation=True
    )
  model_inputs["labels"] = labels["input_ids"]
  return model_inputs
```

```
tokenized_train = train.map(preprocess_function, batched=True)
tokenized_test = test.map(preprocess_function, batched=True)
```

```
Map: 100%      10202/10202 [00:21<00:00, 458.95 examples/s]
/usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils_base.py:3864: UserWarning: `as_target_tokenizer` is deprecated and will be removed in v5 of Transformers. You can tokenize your labels by using the argument `text_tar
  warnings.warn(
Map: 100%      4081/4081 [00:06<00:00, 602.14 examples/s]
```

```
import transformers
from transformers import TFAutoModelForSeq2SeqLM, AutoModelForSeq2SeqLM,DataCollatorForSeq2Seq, Seq2SeqTrainingArguments, Seq2SeqTrainer
model = AutoModelForSeq2SeqLM.from_pretrained(MODEL_CHECKPOINT)
data_collator = DataCollatorForSeq2Seq(tokenizer, model=model)
```

```
Downloading (…)lve/main/config.json: 100%      1.21k/1.21k [00:00<00:00, 84.6kB/s]

Downloading model.safetensors: 100%      242M/242M [00:00<00:00, 273MB/s]

Downloading (…)neration_config.json: 100%      147/147 [00:00<00:00, 6.47kB/s]
```

```
import nltk
import numpy as np
from datasets import load_metric
metric = load_metric("rouge")
```

```
    <ipython-input-14-6c8ec00f0fdc>:4: FutureWarning: load_metric is deprecated and will be removed in the next major version of datasets. Use 'evaluate.load' instead, from the new library 🤗 Evaluate: https://huggingface.co/docs/evaluate
      metric = load_metric("rouge")
    Downloading builder script:                                   5.65k/? [00:00<00:00, 237kB/s]
```

```
def compute_metrics(eval_pred):
  predictions, labels = eval_pred
  preds = np.where(predictions != -100, predictions, tokenizer.pad_token_id)
  decoded_preds = tokenizer.batch_decode(preds, skip_special_tokens=True)
  # Replace -100 in the labels as we can't decode them.
  labels = np.where(labels != -100, labels, tokenizer.pad_token_id)
  decoded_labels = tokenizer.batch_decode(labels, skip_special_tokens=True)
  # Rouge expects a newline after each sentence
  decoded_preds = ["\n".join(nltk.sent_tokenize(pred.strip())) for pred in decoded_preds]
  decoded_labels = ["\n".join(nltk.sent_tokenize(label.strip())) for label in decoded_labels]
  result = metric.compute(predictions=decoded_preds,references=decoded_labels, use_stemmer=True)
  # Extract a few results
  result = {key: value.mid.fmeasure * 100 for key, value in result.items()}
  # Add mean generated length
  prediction_lens = [np.count_nonzero(pred != tokenizer.pad_token_id) for pred in predictions]
  result["gen_len"] = np.mean(prediction_lens)
  return {k: round(v, 4) for k, v in result.items()}
```

```
if torch.cuda.is_available():
  device = torch.device("cuda")
  print("GPU is available and being used")
else:
  device = torch.device("cpu")
  print("GPU is not available, using CPU instead")
```

```
    GPU is available and being used
```

```
model_name = MODEL_CHECKPOINT.split("/")[-1]
#model = model_name.to(device)
args = Seq2SeqTrainingArguments(
  f"{model_name}-finetuned",
  evaluation_strategy = "epoch",
  learning_rate=LEARNING_RATE,
  per_device_train_batch_size=BATCH_SIZE,
  per_device_eval_batch_size=BATCH_SIZE,
  weight_decay=0.01,
  save_total_limit=3,
  num_train_epochs=MAX_EPOCHS,
  predict_with_generate=True,
  fp16=True
)
```

```
import accelerate
accelerate.__version__
```

```
    '0.23.0'
```

```
trainer = Seq2SeqTrainer(
  model.to(device),
  args,
  train_dataset=tokenized_train,
  eval_dataset=tokenized_test,
  data_collator=data_collator,
  tokenizer=tokenizer,
  compute_metrics=compute_metrics,
    compute_metrics=compute_metrics
)
```

```
trainer.train()
```

```
You're using a T5TokenizerFast tokenizer. Please note that with a fast tokenizer, using the `__call__` method is faster than using a method to encode the text followed by a call to the `pad` method to get a padded encoding.
```

[19141/25520 3:26:14 < 1:08:44, 1.55 it/s, Epoch 15/20]

Epoch	Training Loss	Validation Loss	Rouge1	Rouge2	Rougel	Rougelsum	Gen Len
1	3.094800	2.808268	26.058300	6.499300	20.776100	20.778800	18.663100
2	2.744000	2.761016	27.835800	7.706500	22.151700	22.162000	18.902200
3	2.467700	2.769797	27.480300	7.680800	21.972500	21.973500	18.699600
4	2.264600	2.775719	28.053000	8.100700	22.291300	22.300000	18.837500
5	2.047600	2.825001	28.401000	8.337400	22.691500	22.700300	18.773300
6	1.892900	2.870928	27.989600	8.097500	22.379500	22.371500	18.739500
7	1.682700	2.943558	28.572200	8.473400	22.702700	22.703700	18.821900
8	1.571400	3.034054	28.352000	8.217300	22.599400	22.596900	18.795100
9	1.379700	3.132841	27.614500	8.034900	22.075800	22.073000	18.876700
10	1.275600	3.250896	28.717800	8.568900	22.842900	22.848700	18.837300
11	1.158100	3.380930	28.288000	8.331000	22.528600	22.524000	18.844200
12	1.000400	3.517771	28.200800	8.383100	22.553400	22.547600	18.883100
13	0.904300	3.664958	28.296800	8.537700	22.538100	22.537800	18.836800
14	0.787200	3.797339	28.110100	8.304800	22.375000	22.380000	18.859300

[378/511 03:42 < 01:18, 1.70 it/s]

```
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1260: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length. We recommend setting `max_new_tokens` to control the maximum
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1260: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length. We recommend setting `max_new_tokens` to control the maximum
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1260: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length. We recommend setting `max_new_tokens` to control the maximum
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1260: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length. We recommend setting `max_new_tokens` to control the maximum
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1260: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length. We recommend setting `max_new_tokens` to control the maximum
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1260: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length. We recommend setting `max_new_tokens` to control the maximum
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1260: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length. We recommend setting `max_new_tokens` to control the maximum
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1260: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length. We recommend setting `max_new_tokens` to control the maximum
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1260: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length. We recommend setting `max_new_tokens` to control the maximum
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1260: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length. We recommend setting `max_new_tokens` to control the maximum
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1260: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length. We recommend setting `max_new_tokens` to control the maximum
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1260: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length. We recommend setting `max_new_tokens` to control the maximum
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1260: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length. We recommend setting `max_new_tokens` to control the maximum
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1260: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length. We recommend setting `max_new_tokens` to control the maximum
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1260: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length. We recommend setting `max_new_tokens` to control the maximum
  warnings.warn(
```

```
predict_results = trainer.predict(tokenized_test,max_length=128, num_beams=3)
```

In [24]:

```python
predict_results = trainer.predict(tokenized_test,max_length=128, num_beams=3)
```

```
/opt/conda/lib/python3.10/site-packages/torch/nn/parallel/_functions.py:68: UserWarning: Was asked to gather along dimension 0, but all input
tensors were scalars; will instead unsqueeze and return a vector.
  warnings.warn('Was asked to gather along dimension 0, but all '
```

In [25]:

```python
if args.predict_with_generate:
    # Replace -100 with pad_token_id in predictions
    preds = np.where(predict_results.predictions != -100, predict_results.predictions, tokenizer.pad_token_id)
    # Decode batched predictions into text, skipping special tokens and cleaning up spaces
    predictions = tokenizer.batch_decode(preds, skip_special_tokens=True, clean_up_tokenization_spaces=True)
    # Strip leading/trailing spaces from each prediction
    predictions = [pred.strip() for pred in predictions]
```

In [26]:

```python
test['summary'][:2]
```

Out[26]:

```
['Premiership club Saracens have re-signed Australia international lock Will Skelton on a two-year contract.',
 'A former World War Two German submariner was welcomed as one of the guests of honour at a club for British veterans.']
```

In [27]:

```python
predictions[:2]
```

Out[27]:

```
['Australia coach Mark Skelton has been unable to add to his Test caps while with Saracens. the 24-year-old has won 18 Test caps for the Wall
abies since joining on a short-term deal from Super Rugby side Warratahs.',
 'Horst Jackson, 90, was captured in Gibraltar during the conflict and settled in Lincolnshire.']
```