

✓ Title of Mini-Project : Text summarization using transformers and weights and biases

```
!pip install -q transformers datasets rouge_score wandb
```

```

Preparing metadata (setup.py) ... done
491.2/491.2 kB 8.6 MB/s eta 0:00:00
116.3/116.3 kB 6.0 MB/s eta 0:00:00
183.9/183.9 kB 9.6 MB/s eta 0:00:00
143.5/143.5 kB 9.3 MB/s eta 0:00:00
194.8/194.8 kB 11.0 MB/s eta 0:00:00
Building wheel for rouge_score (setup.py) ... done
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the sou
torch 2.6.0+cu124 requires nvidia-cublas-cu12==12.4.5.8; platform_system == "Linux" and platform_machine == "x86_64", but you have r
torch 2.6.0+cu124 requires nvidia-cuda-cupti-cu12==12.4.127; platform_system == "Linux" and platform_machine == "x86_64", but you ha
torch 2.6.0+cu124 requires nvidia-cuda-nvrtc-cu12==12.4.127; platform_system == "Linux" and platform_machine == "x86_64", but you ha
torch 2.6.0+cu124 requires nvidia-cuda-runtime-cu12==12.4.127; platform_system == "Linux" and platform_machine == "x86_64", but you
torch 2.6.0+cu124 requires nvidia-cudnn-cu12==9.1.0.70; platform_system == "Linux" and platform_machine == "x86_64", but you have nv
torch 2.6.0+cu124 requires nvidia-cufft-cu12==11.2.1.3; platform_system == "Linux" and platform_machine == "x86_64", but you have nv
torch 2.6.0+cu124 requires nvidia-curand-cu12==10.3.5.147; platform_system == "Linux" and platform_machine == "x86_64", but you have
torch 2.6.0+cu124 requires nvidia-cusolver-cu12==11.6.1.9; platform_system == "Linux" and platform_machine == "x86_64", but you have
torch 2.6.0+cu124 requires nvidia-cuspars-cu12==12.3.1.170; platform_system == "Linux" and platform_machine == "x86_64", but you ha
torch 2.6.0+cu124 requires nvidia-nvjitlink-cu12==12.4.127; platform_system == "Linux" and platform_machine == "x86_64", but you hav
gcsfs 2025.3.2 requires fsspec==2025.3.2, but you have fsspec 2024.12.0 which is incompatible.
```

```
import wandb
wandb.login()
```

```

wandb: Using wandb-core as the SDK backend. Please refer to https://wandb.me/wandb-core for more information.
wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally: https://wandb.me/wandb-server)
wandb: You can find your API key in your browser here: https://wandb.ai/authorize
wandb: Paste an API key from your profile and hit enter: .....
wandb: WARNING If you're specifying your api key in code, ensure this code is not shared publicly.
wandb: WARNING Consider setting the WANDB_API_KEY environment variable, or running `wandb login` from the command line.
wandb: No netrc file found, creating one.
wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc
wandb: Currently logged in as: rajpatil1405 (rajpatil1405-met-s-institute-of-engineering) to https://api.wandb.ai. Use `wandb login
True
```

```
!pip install evaluate
```


```

Collecting evaluate
  Downloading evaluate-0.4.3-py3-none-any.whl.metadata (9.2 kB)
Requirement already satisfied: datasets>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from evaluate) (3.5.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages (from evaluate) (2.0.2)
Requirement already satisfied: dill in /usr/local/lib/python3.11/dist-packages (from evaluate) (0.3.8)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (from evaluate) (2.2.2)
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.11/dist-packages (from evaluate) (2.32.3)
Requirement already satisfied: tqdm>=4.62.1 in /usr/local/lib/python3.11/dist-packages (from evaluate) (4.67.1)
Requirement already satisfied: xxhash in /usr/local/lib/python3.11/dist-packages (from evaluate) (3.5.0)
Requirement already satisfied: multiprocessing in /usr/local/lib/python3.11/dist-packages (from evaluate) (0.70.16)
Requirement already satisfied: fsspec>=2021.05.0 in /usr/local/lib/python3.11/dist-packages (from fsspec[http]>=2021.05.0->evaluate) (2025.3.2)
Requirement already satisfied: huggingface-hub>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from evaluate) (0.30.2)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from evaluate) (24.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from datasets>=2.0.0->evaluate) (3.18.0)
Requirement already satisfied: pyarrow>=15.0.0 in /usr/local/lib/python3.11/dist-packages (from datasets>=2.0.0->evaluate) (18.1.0)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.11/dist-packages (from datasets>=2.0.0->evaluate) (3.11.15)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from datasets>=2.0.0->evaluate) (6.0.2)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.7.0->evaluate) (4.12.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests>=2.19.0->evaluate) (3.3.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests>=2.19.0->evaluate) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests>=2.19.0->evaluate) (2.3.1)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests>=2.19.0->evaluate) (2025.8.3)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas->evaluate) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas->evaluate) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas->evaluate) (2025.2)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets>=2.0.0->evaluate) (2.4.4)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets>=2.0.0->evaluate) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets>=2.0.0->evaluate) (25.3.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets>=2.0.0->evaluate) (1.5.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets>=2.0.0->evaluate) (6.1.0)
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets>=2.0.0->evaluate) (0.2.0)
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets>=2.0.0->evaluate) (1.18.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas->evaluate) (1.17.0)
Downloading evaluate-0.4.3-py3-none-any.whl (84 kB)
84.0/84.0 kB 2.4 MB/s eta 0:00:00
Installing collected packages: evaluate
Successfully installed evaluate-0.4.3
```

```
import torch
from datasets import load_dataset
```


```
import evaluate
from transformers import (
    AutoTokenizer,
    AutoModelForSeq2SeqLM,
    DataCollatorForSeq2Seq,
    TrainingArguments,
    Trainer
)
```

```
# Load ROUGE metric
rouge = evaluate.load("rouge")
```

 /usr/local/lib/python3.11/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 it as :
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(
Downloading builder script: 100% 6.27k/6.27k [00:00<00:00, 294kB/s]

```
wandb.init(project="text-summarization-nlp", name="bart-colab-mini")
```

```
# Load Dataset
dataset = load_dataset("cnn_dailymail", "3.0.0")
train_data = dataset["train"].shuffle(seed=42).select(range(300))
val_data = dataset["validation"].select(range(100))
```

 Tracking run with wandb version 0.19.9
Run data is saved locally in /content/wandb/run-20250419_061214-oj35jk5u
Syncing run [bart-colab-mini](#) to [Weights & Biases \(docs\)](#)
View project at <https://wandb.ai/rajpatil1405-met-s-institute-of-engineering/text-summarization-nlp>
View run at <https://wandb.ai/rajpatil1405-met-s-institute-of-engineering/text-summarization-nlp/runs/oj35jk5u>

README.md: 100%	15.6k/15.6k [00:00<00:00, 212kB/s]
train-00000-of-00003.parquet: 100%	257M/257M [00:01<00:00, 178MB/s]
train-00001-of-00003.parquet: 100%	257M/257M [00:05<00:00, 21.7MB/s]
train-00002-of-00003.parquet: 100%	259M/259M [00:01<00:00, 203MB/s]
validation-00000-of-00001.parquet: 100%	34.7M/34.7M [00:00<00:00, 160MB/s]
test-00000-of-00001.parquet: 100%	30.0M/30.0M [00:00<00:00, 159MB/s]
Generating train split: 100%	287113/287113 [00:16<00:00, 8791.44 examples/s]
Generating validation split: 100%	13368/13368 [00:00<00:00, 32612.18 examples/s]
Generating test split: 100%	11490/11490 [00:00<00:00, 39611.52 examples/s]

```
model_checkpoint = "facebook/bart-base"
tokenizer = AutoTokenizer.from_pretrained(model_checkpoint)
model = AutoModelForSeq2SeqLM.from_pretrained(model_checkpoint)
```

 config.json: 100% 1.72k/1.72k [00:00<00:00, 23.4kB/s]
vocab.json: 100% 899k/899k [00:00<00:00, 4.87MB/s]
merges.txt: 100% 456k/456k [00:00<00:00, 10.6MB/s]
tokenizer.json: 100% 1.36M/1.36M [00:00<00:00, 15.0MB/s]
model.safetensors: 100% 558M/558M [00:06<00:00, 82.1MB/s]

```
max_input_length = 512
max_target_length = 128
```

```
def preprocess_data(examples):
    inputs = examples["article"]
    targets = examples["highlights"]
    model_inputs = tokenizer(inputs, max_length=1024, truncation=True, padding="max_length")

    # Tokenize targets for the summaries
    with tokenizer.as_target_tokenizer():
        labels = tokenizer(targets, max_length=256, truncation=True, padding="max_length")

    model_inputs["labels"] = labels["input_ids"] # Add labels to the tokenized inputs
    return model_inputs
```

```
data_collator = DataCollatorForSeq2Seq(tokenizer=tokenizer, model=model)
```

```

# 🇬🇧 Metric
rouge = evaluate.load("rouge")

def compute_metrics(eval_pred):
    predictions, labels = eval_pred
    decoded_preds = tokenizer.batch_decode(predictions, skip_special_tokens=True)
    decoded_labels = tokenizer.batch_decode(labels, skip_special_tokens=True)
    result = rouge.compute(predictions=decoded_preds, references=decoded_labels, use_stemmer=True)
    return {key: value.mid.fmeasure * 100 for key, value in result.items()}

print(train_data.column_names)

↔ ['article', 'highlights', 'id']

from transformers import TrainingArguments

# Define training arguments without evaluation_strategy
training_args = TrainingArguments(
    output_dir="./results",
    learning_rate=2e-5,
    per_device_train_batch_size=2,
    per_device_eval_batch_size=2,
    num_train_epochs=1,
    weight_decay=0.01,
    logging_dir="./logs",
    report_to="wandb", # Log everything to W&B
    save_total_limit=1,
    remove_unused_columns=False # Ensure unused columns aren't removed
)

from transformers import AutoTokenizer, AutoModelForSeq2SeqLM

# Load the model and tokenizer
model_name = "t5-small" # Or the model you are using
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSeq2SeqLM.from_pretrained(model_name)

# Preprocess the data
train_data = train_data.map(preprocess_data, batched=True)
val_data = val_data.map(preprocess_data, batched=True)

# Set the format
train_data.set_format(type='torch', columns=['input_ids', 'attention_mask', 'labels'])
val_data.set_format(type='torch', columns=['input_ids', 'attention_mask', 'labels'])

# Create the Trainer
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_data,
    eval_dataset=val_data,
    tokenizer=tokenizer
)

# Start training
trainer.train()

```

```

tokenizer_config.json: 100%                2.32k/2.32k [00:00<00:00, 58.3kB/s]

spiece.model: 100%                        792k/792k [00:00<00:00, 4.26MB/s]

tokenizer.json: 100%                      1.39M/1.39M [00:00<00:00, 14.8MB/s]

config.json: 100%                        1.21k/1.21k [00:00<00:00, 51.0kB/s]

trainer.evaluate()

model.safetensors: 100% [50/50 03:06]      242M/242M [00:03<00:00, 52.0MB/s]
{ 'eval_loss': 1.921242117881775,          147/147 [00:00<00:00, 3.76kB/s]
  'generation_config.json': 100%,
  'eval_runtime': 193.6095,
  'eval_samples_per_second': 0.517,         300/300 [00:03<00:00, 79.12 examples/s]
  'eval_steps_per_second': 0.258,
  '/usr/local/lib/python3.11/dist-packages/transformers/tokenization_utils_base.py:3980: UserWarning: `as_target_tokenizer` is deprecated
  ...
}

def summarize(text):
    inputs = tokenizer.encode(text, return_tensors="pt", max_length=512, truncation=True)
    summary_ids = model.generate(
        inputs,
        max_length=150,
        min_length=30,
        length_penalty=2.0,
        num_beams=4,
        early_stopping=True
    )
    return tokenizer.decode(summary_ids[0], skip_special_tokens=True)

custom_text = """
The global climate crisis continues to be a major concern for nations around the world.
Recent studies show rising sea levels, increased frequency of extreme weather events,
and record-breaking temperatures. World leaders are calling for urgent action,
highlighting the need for renewable energy, reforestation, and reduced carbon emissions.
"""

print("📄 Original Text:\n", custom_text)
print("\n📄 Summary:\n", summarize(custom_text))

🔄 📄 Original Text:

The global climate crisis continues to be a major concern for nations around the world.
Recent studies show rising sea levels, increased frequency of extreme weather events,
and record-breaking temperatures. World leaders are calling for urgent action,
highlighting the need for renewable energy, reforestation, and reduced carbon emissions.

📄 Summary:
Welt leaders are calling for urgent action, highlighting the need for renewable energy, reforestation, and reduced carbon emissions

```

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