```
!pip install -q transformers[sentencepiece] accelerate --upgrade
```

Install libraries

```
from transformers import pipeline, AutoTokenizer, AutoModelForSeq2SeqLM
import textwrap
```

Choose model

```
MODEL = "google/flan-t5-base"
```

Load model & tokenizer

```
print(f"Loading model {MODEL} ...")
tokenizer = AutoTokenizer.from_pretrained(MODEL)
model = AutoModelForSeq2SeqLM.from_pretrained(MODEL)
pipe = pipeline("text2text-generation", model=model, tokenizer=tokenizer, device=DEVICE)
```

```
Loading model google/flan-t5-base ...
Device set to use cpu
```

Helper to generate and print neatly

```
def gen_and_print(prompt, max_length=256, num_return_sequences=1, description=None):
    if description:
        print("\n" + "="*60)
        print(description)
        print("-"*60)
    print("PROMPT:")
    print(textwrap.indent(prompt, "  "))
    out = pipe(prompt, max_length=max_length, do_sample=False, num_return_sequences=num_return_sequences)
    print("\nOUTPUT:")
    for i, o in enumerate(out, 1):
        print(f"[{i}] {o['generated_text']}\n")
```

1) Summarization : Zero-shot summarization (just give instruction + text)

```
summ_text = (
    "Article: India has announced a major initiative to expand renewable energy capacity over the next decade. "
    "The government plans to add large-scale solar and wind installations, invest in grid upgrades, and provide subsidies "
    "for domestic manufacturing of key components. The program aims to create jobs and reduce emissions dramatically."
)
zero_shot_summary_prompt = f"Summarize the following article in 2-3 lines:\n\n{summ_text}"
gen_and_print(zero_shot_summary_prompt, description="Zero-shot: Summarization (2-3 lines)")

```

```

============================================================
Zero-shot: Summarization (2-3 lines)
------------------------------------------------------------
PROMPT:
  Summarize the following article in 2-3 lines:

  Article: India has announced a major initiative to expand renewable energy capacity over the next decade. The government plans to add
Both `max_new_tokens` (=256) and `max_length`(=256) seem to have been set. `max_new_tokens` will take precedence. Please refer to the dc

OUTPUT:
[1] India's government has announced plans to increase the number of renewable energy installations in the country.
```

Few-shot summarization (provide examples then new input)

```
few_shot_examples = (
    "Example 1:\nText: The cat slept on the warm windowsill all afternoon.\nSummary: A cat relaxed in the sunlight.\n\n"
    "Example 2:\nText: Rain disrupted the city's traffic during the morning commute causing long delays.\nSummary: Heavy rain caused maj
)
few_shot_prompt = (
    f"{few_shot_examples}"
    f"Now do the same for the new text:\nText: {summ_text}\nSummary:"
)
gen_and_print(few_shot_prompt, description="Few-shot: Summarization (learn style from examples)")

```

```
============================================================
Few-shot: Summarization (learn style from examples)
------------------------------------------------------------
PROMPT:
  Example 1:
  Text: The cat slept on the warm windowsill all afternoon.
  Summary: A cat relaxed in the sunlight.

  Example 2:
  Text: Rain disrupted the city's traffic during the morning commute causing long delays.
  Summary: Heavy rain caused major traffic delays.

  Now do the same for the new text:
  Text: Article: India has announced a major initiative to expand renewable energy capacity over the next decade. The government plans t
  Summary:
Both `max_new_tokens` (=256) and `max_length`(=256) seem to have been set. `max_new_tokens` will take precedence. Please refer to the dc

OUTPUT:
[1] India's government has announced plans to expand its renewable energy capacity.
```

Coding

Zero-shot coding: ask for a function with specification

```
def run_prompt(prompt):
    out = pipe(prompt, max_length=100, do_sample=False)
    print(f"\nPROMPT:\n{prompt}\n")
    print(f"OUTPUT:\n{out[0]['generated_text']}\n{'-'*50}")
```

```
prompt1 = "Write a Python function to add two numbers a and b and return the result."
run_prompt(prompt1)
```

```
Both `max_new_tokens` (=256) and `max_length`(=100) seem to have been set. `max_new_tokens` will take precedence. Please refer to the dc

PROMPT:
Write a Python function to add two numbers a and b and return the result.

OUTPUT:
a, b = map(int, input().split()) print(a + b)
--------------------------------------------------
```

Few-shot coding: show example input-output pairs

```
from transformers import AutoTokenizer, AutoModelForCausalLM
import torch


model_name = "gpt2"


tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(model_name)

```

```

prompt = """Example:
Input: 4
Output: Even

Example:
Input: 9
Output: Odd

Now, write a Python function that checks if a number is odd or even, and demonstrate with input 7.
"""


inputs = tokenizer(prompt, return_tensors="pt")

outputs = model.generate(
    **inputs,
    max_new_tokens=150,
    do_sample=False
)


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

```

```
tokenizer_config.json:  100%  26.0/26.0  [00:00<00:00,  1.14kB/s]
config.json:  100%  665/665  [00:00<00:00,  28.4kB/s]
vocab.json:  100%  1.04M/1.04M  [00:00<00:00,  10.2MB/s]
merges.txt:  100%  456k/456k  [00:00<00:00,  8.58MB/s]
tokenizer.json:  100%  1.36M/1.36M  [00:00<00:00,  13.9MB/s]
model.safetensors:  100%  548M/548M  [00:17<00:00,  45.0MB/s]
generation_config.json:  100%  124/124  [00:00<00:00,  9.36kB/s]
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Example:
Input: 4
Output: Even

Example:
Input: 9
Output: Odd

Now, write a Python function that checks if a number is odd or even, and demonstrate with input 7.

>>> from math import random >>> from math import random2 >>> from math import random22 >>> from math import random22 >>> from math impor
```

Creative Writing

Role-based prompt: ask the model to act as a poet

```
from transformers import AutoTokenizer, AutoModelForCausalLM
import torch


model_name = "gpt2-medium"

tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(model_name)

# Role-based prompt
prompt = """You are a professional creative writer.
Write a story about a cricketer life."""

inputs = tokenizer(prompt, return_tensors="pt")

outputs = model.generate(
    **inputs,
    max_new_tokens=200,
    do_sample=True,
    temperature=0.9,
    top_p=0.95
)

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

```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
You are a professional creative writer.
Write a story about a cricketer life. Do this one.
What's wrong with that?
This isn't about cricketers. This isn't about India's Test cricket team. This is about you writing a story about cricketers life and wha
Now, what is so wrong with writing a story about people you don't know? A man named Murali Vijay, a cricketer born in 1950 who went to D
```

Chain-of-Thought (CoT) style for a reasoning task.

```
from transformers import AutoTokenizer, AutoModelForCausalLM
import torch

model_name = "gpt2-medium"

tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(model_name)

# Chain-of-thought style creative writing
prompt = """Let's write a creative short story step by step.

Step 1: Decide the main character.
Step 2: Decide the setting.
Step 3: Decide the conflict.
Step 4: Write the story.

Story:
"""

inputs = tokenizer(prompt, return_tensors="pt")

outputs = model.generate(
    **inputs,
    max_new_tokens=300,
    do_sample=True,
    temperature=0.9,
    top_p=0.9
)

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

```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Let's write a creative short story step by step.

Step 1: Decide the main character.
Step 2: Decide the setting.
Step 3: Decide the conflict.
Step 4: Write the story.

Story:

The protagonist is an intelligent, ambitious young man who has decided to pursue a career in science. He begins to discover a secret wor

Step 4: Write your story.

Step 5: Write a short story for your website.

Step 6: Go to a publisher to ask them to pay for a story.

Step 7: Write your story for sale.

Step 8: Your story will be made into a TV show.

Step 9: Your story will be turned into a movie.

Step 10: Your story will be published in newspapers around the world.

Step 11: Your story will be put on TV.
Step 12: Your story will be put on the big screen.

Step 13: Your story will be a movie.

Step 14: Your story will be made into a motion picture.

Step 15: Your story will be played in theaters.

Step 16: Your story will be broadcast on the radio.

Step 17: Your story will be on television.

Step 18: Your story will be a TV show.

Step 19: Your story will be a movie.

Step 20: Your story will be a movie
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```