```
!pip install -q transformers langchain langchain-community accelerate sentencepiece
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.5/2.5 MB 25.4 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 45.2/45.2 kB 2.0 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 363.4/363.4 MB 1.5 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.8/13.8 MB 123.7 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.6/24.6 MB 87.7 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 883.7/883.7 kB 56.4 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 664.8/664.8 MB 2.1 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 211.5/211.5 MB 5.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.3/56.3 MB 14.2 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 127.9/127.9 MB 8.1 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.5/207.5 MB 5.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 188.7/188.7 MB 6.3 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.1/21.1 MB 102.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 50.9/50.9 kB 5.2 MB/s eta 0:00:00
```

```
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM, pipeline
from langchain_community.llms import HuggingFacePipeline

# Load free open model
model_id = "google/flan-t5-xl"
tokenizer = AutoTokenizer.from_pretrained(model_id)
model = AutoModelForSeq2SeqLM.from_pretrained(model_id, device_map="auto")


pipe = pipeline("text2text-generation", model=model, tokenizer=tokenizer)
llm = HuggingFacePipeline(pipeline=pipe)
```

```
/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 secret in your
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:  2.54k/?  [00:00<00:00,  47.0kB/s]
spiece.model:  100%  792k/792k  [00:00<00:00,  125kB/s]
tokenizer.json:  2.42M/?  [00:00<00:00,  15.6MB/s]
special_tokens_map.json:  2.20k/?  [00:00<00:00,  83.8kB/s]
config.json:  1.44k/?  [00:00<00:00,  43.0kB/s]
model.safetensors.index.json:  53.0k/?  [00:00<00:00,  735kB/s]
Fetching 2 files:  100%  2/2  [13:19<00:00,  799.66s/it]
model-00002-of-00002.safetensors:  100%  1.95G/1.95G  [12:00<00:00,  4.33MB/s]
model-00001-of-00002.safetensors:  100%  9.45G/9.45G  [13:19<00:00,  45.2MB/s]
Loading checkpoint shards:  100%  2/2  [00:46<00:00,  20.52s/it]
generation_config.json:  100%  147/147  [00:00<00:00,  13.9kB/s]
Device set to use cuda:0
/tmp/ipython-input-3212210358.py:11: LangChainDeprecationWarning: The class `HuggingFacePipeline` was deprecated in LangChain 0.0.37 and will be r
  llm = HuggingFacePipeline(pipeline=pipe)
```

```
from langchain_core.prompts import ChatPromptTemplate

prompt = ChatPromptTemplate.from_messages([
    ("system", "You are a helpful assistant."),
    ("human", "{input}")
])

chain = prompt | llm

response = chain.invoke({"input": "Who is the founder of Google?"})
print("Answer:", response)
```

```
Answer: System: Google was founded by Larry Page and Sergey Brin.
```

```
response = chain.invoke({"input": "Who was the first president of the United States?"})
print("Answer:", response)
```

```
Answer: System: George Washington was the first president of the United States.
```

```
response = chain.invoke({"input": "What is the capital of France?"})
print("Answer:", response)
```

```
Answer: System: The capital of France is Paris.
```

```
response = chain.invoke({"input": "Translate 'Good morning, how are you?' into Spanish."})
print("Answer:", response)
```

```
Answer: Buena maana, cómo estás?
```

```
response = chain.invoke({"input": "How does photosynthesis work?"})
print("Answer:", response)
```

```
Answer: System: Photosynthesis is the process by which plants use light energy to make food for themselves.
```

```
response = chain.invoke({"input": "What is the capital of India?"})
print("Answer:", response)
```

```
You seem to be using the pipelines sequentially on GPU. In order to maximize efficiency please use a dataset
Answer: System: The capital of India is New Delhi.
```

```
response = chain.invoke({"input": "What is the capital of Russia?"})
print("Answer:", response)
```

```
Answer: System: The capital of Russia is Moscow.
```

```
response = chain.invoke({"input": "What is the internet?"})
print("Answer:", response)
```

```
Answer: System: The internet is a network of computers connected to the internet.
```

Start coding or generate with AI.