Replicate

Contents

This example goes over how to use LangChain to interact with Replicate models

```
import os
from langchain.llms import Replicate
from langchain import PromptTemplate, LLMChain

os.environ["REPLICATE_API_TOKEN"] = "YOUR REPLICATE API TOKEN"
```

To run this notebook, you'll need to create a replicate account and install the replicate python client.

Find a model on the replicate explore page, and then paste in the model name and version in this format: model_name/version

For example, for this flan-t5 model, click on the API tab. The model name/version would be: `daanelson/flan-t5:04e422a9b85baed86a4f24981d7f9953e20c5fd82f6103b74ebc431588e1cec8`

Only the `model` param is required, but we can add other model params when initializing.

For example, if we were running stable diffusion and wanted to change the image dimensions:

```
Replicate(model="stability-ai/stable-
diffusion:db21e45d3f7023abc2a46ee38a23973f6dce16bb082a930b0c49861f96d1e5bf", input=
{'image_dimensions': '512x512'})
```

Skip to main content

```
llm = Replicate(model="daanelson/flan-
t5:04e422a9b85baed86a4f24981d7f9953e20c5fd82f6103b74ebc431588e1cec8")
```

```
prompt = """
Answer the following yes/no question by reasoning step by step.
Can a dog drive a car?
"""
llm(prompt)
```

```
'The legal driving age of dogs is 2. Cars are designed for humans to drive.
Therefore, the final answer is yes.'
```

We can call any replicate model using this syntax. For example, we can call stable diffusion.

```
text2image = Replicate(model="stability-ai/stable-
diffusion:db21e45d3f7023abc2a46ee38a23973f6dce16bb082a930b0c49861f96d1e5bf",
                       input={'image_dimensions': '512x512'})
```

```
image_output = text2image("A cat riding a motorcycle by Picasso")
image_output
```

```
'https://replicate.delivery/pbxt/Cf07B1zqzFQLOSBQcKG7m9beE74wf7kuip5W9VxHJFembefKE/
out-0.png'
```

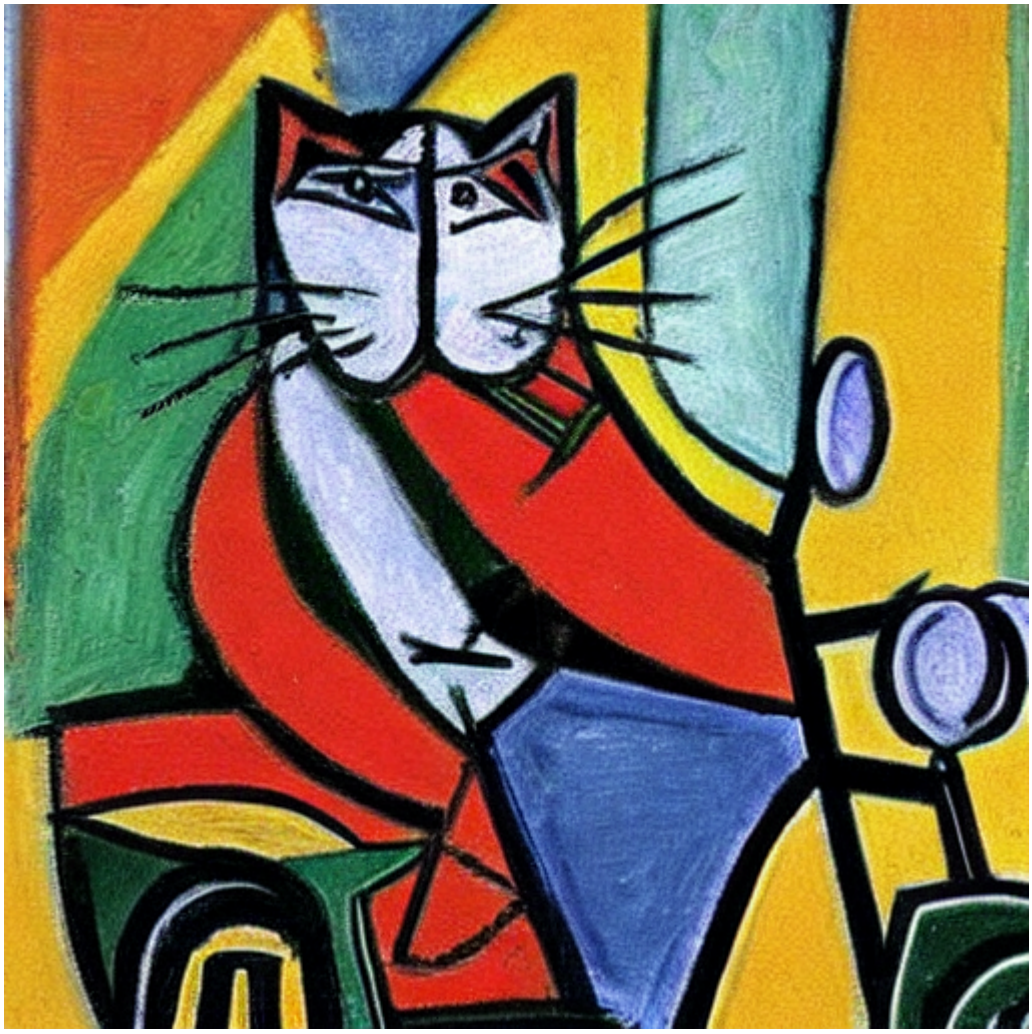
The model spits out a URL. Let's render it.

```
from PIL import Image
import requests
from io import BytesIO

response = requests.get(image_output)
img = Image.open(BytesIO(response.content))

img
```

Skip to main content



The whole point of langchain is to... chain! Here's an example of how do that.

```
from langchain.chains import SimpleSequentialChain
```

First, let's define the LLM for this model as a flan-5, and text2image as a stable diffusion model.

```
llm = Replicate(model="daanelson/flan-
t5:04e422a9b85baed86a4f24981d7f9953e20c5fd82f6103b74ebc431588e1cec8")
text2image = Replicate(model="stability-ai/stable-
diffusion:db21e45d3f7023abc2a46ee38a23973f6dce16bb082a930b0c49861f96d1e5bf")
```

First prompt in the chain

```
prompt = PromptTemplate(
    input_variables=["product"],
    template="What is a good name for a company that makes {product}?",
)
```

Skip to main content

Second prompt to get the logo for company description

```
second_prompt = PromptTemplate(
    input_variables=["company_name"],
    template="Write a description of a logo for this company: {company_name}",
)
chain_two = LLMChain(llm=llm, prompt=second_prompt)
```

Third prompt, let's create the image based on the description output from prompt 2

```
third_prompt = PromptTemplate(
    input_variables=["company_logo_description"],
    template="{company_logo_description}",
)
chain_three = LLMChain(llm=text2image, prompt=third_prompt)
```

Now let's run it!

```
# Run the chain specifying only the input variable for the first chain.
overall_chain = SimpleSequentialChain(chains=[chain, chain_two, chain_three],
verbose=True)
catchphrase = overall_chain.run("colorful socks")
print(catchphrase)
```

```
> Entering new SimpleSequentialChain chain...
novelty socks
todd & co.
https://replicate.delivery/pbxt/BedAP1PPBwXFfkmeD7xDygXO4BcvApp1uvWOwUdHM4tcQfvCB/out-0.png

> Finished chain.
https://replicate.delivery/pbxt/BedAP1PPBwXFfkmeD7xDygXO4BcvApp1uvWOwUdHM4tcQfvCB/out-0.png
```

```
response =
requests.get("https://replicate.delivery/pbxt/eq6foRJngThCAEBqse3nL3Km2MBfLnWQNd0Hy2SQRo2LuprCB/out-0.png")
img = Image.open(BytesIO(response.content))
img
```

Skip to main content