GPT4All

GitHub:nomic-ai/gpt4all an ecosystem of open-source chatbots trained on a massive collections of clean assistant data including code, stories and dialogue.

This example goes over how to use LangChain to interact with `GPT4All` models.

```
%pip install gpt4all > /dev/null
```

```
    Note: you may need to restart the kernel to use updated packages.
```

```
from langchain import PromptTemplate, LLMChain
from langchain.llms import GPT4All
from langchain.callbacks.streaming_stdout import StreamingStdOutCallbackHandler
```

```
template = """Question: {question}

Answer: Let's think step by step."""

prompt = PromptTemplate(template=template, input_variables=["question"])
```

Specify Model

To run locally, download a compatible ggml-formatted model.

Download option 1: The gpt4all page has a useful `Model Explorer` section:

- Select a model of interest
- Download using the UI and move the `.bin` to the `local_path` (noted below)

For more info, visit https://github.com/nomic-ai/gpt4all.

Download option 2: Uncomment the below block to download a model.

- You may want to update `url` to a new version, whih can be browsed using the gpt4all page.

```
local_path = (
    "./models/ggml-gpt4all-l13b-snoozy.bin"  # replace with your desired local file path
)

# import requests

# from pathlib import Path
# from tqdm import tqdm

# Path(local_path).parent.mkdir(parents=True, exist_ok=True)

# # Example model. Check https://github.com/nomic-ai/gpt4all for the latest models.
# url = 'http://gpt4all.io/models/ggml-gpt4all-l13b-snoozy.bin'

# # send a GET request to the URL to download the file. Stream since it's large
# response = requests.get(url, stream=True)
```

```
# # open the file in binary mode and write the contents of the response to it in chunks
# # This is a large file, so be prepared to wait.
# with open(local_path, 'wb') as f:
#     for chunk in tqdm(response.iter_content(chunk_size=8192)):
#         if chunk:
#             f.write(chunk)
```

```
# Callbacks support token-wise streaming
callbacks = [StreamingStdOutCallbackHandler()]

# Verbose is required to pass to the callback manager
llm = GPT4All(model=local_path, callbacks=callbacks, verbose=True)

# If you want to use a custom model add the backend parameter
# Check https://docs.gpt4all.io/gpt4all_python.html for supported backends
llm = GPT4All(model=local_path, backend="gptj", callbacks=callbacks, verbose=True)
```

```
llm_chain = LLMChain(prompt=prompt, llm=llm)
```

```
question = "What NFL team won the Super Bowl in the year Justin Bieber was born?"

llm_chain.run(question)
```