Uruchamianie lokalnego modelu Ollama z Langchain

Ten przewodnik krok po kroku pokazuje, jak skonfigurować i uruchomić lokalny model językowy Ollama, integrując go z biblioteką Langchain w Pythonie. Dzięki temu możesz wykorzystać moc dużych modeli językowych bezpośrednio na swoim komputerze, korzystając z elastyczności i funkcjonalności Langchain.

Wymagania

Przed rozpoczęciem upewnij się, że masz zainstalowane następujące elementy:

- **Ollama:** Zainstalowany i uruchomiony na Twoim systemie. Pobierz Ollama ze strony https://ollama.com/ i postępuj zgodnie z instrukcjami instalacji dla Twojego systemu operacyjnego.
- **Python:** Wersja 3.7 lub nowsza.
- **Biblioteki Python:** `langchain` i `langchain-community`. Zainstalujemy je w kolejnych krokach.

Kroki instalacji i uruchomienia

1. **Uruchom Ollama i pobierz model:**
    - **Uruchom serwer Ollama:** Otwórz terminal i uruchom serwer Ollama w tle:

      ```
      ollama serve
      ```

    - **Pobierz model:** Wybierz model językowy, którego chcesz użyć (np. `llama2`, `mistral`, `codellama`). Pobierz go za pomocą polecenia:

      ```
      ollama pull llama2
      ```

      Zastąp `llama2` nazwą wybranego modelu. Pobieranie może zająć trochę czasu w zależności od rozmiaru modelu i szybkości Twojego łącza internetowego.

2. **Utwórz środowisko wirtualne (zalecane):**

   Zaleca się korzystanie ze środowisk wirtualnych w Pythonie, aby uniknąć konfliktów zależności.

   ```
   python -m venv venv
   source venv/bin/activate  # Linux/macOS
   # venv\Scripts\activate  # Windows
   ```

3. **Zainstaluj Langchain i langchain-community:**

   W aktywowanym środowisku wirtualnym zainstaluj niezbędne biblioteki:

   ```
   pip install langchain langchain-community
   ```

4. **Utwórz plik Python `ollama_langchain.py`:**

Stwórz nowy plik o nazwie `ollama_langchain.py` i wklej do niego poniższy kod:

```
from langchain_community.llms import Ollama
from langchain.prompts import PromptTemplate
from langchain.chains import LLMChain

# 1. Inicjalizacja modelu Ollama
ollama_llm = Ollama(model="llama2")  # Użyj nazwy modelu, który pobrałeś

# 2. Definicja szablonu promptu (opcjonalne, ale dobre praktyka)
prompt_template = """
Odpowiedz na poniższe pytanie w zwięzły i pomocny sposób.

Pytanie: {question}
"""
prompt = PromptTemplate(template=prompt_template, input_variables=["question"])

# 3. Tworzenie łańcucha LLM (opcjonalne, ale przydatne do bardziej złożonych
zadań)
llm_chain = LLMChain(prompt=prompt, llm=ollama_llm)

# 4. Zadawanie pytania i uzyskanie odpowiedzi
question = "Jaka jest stolica Polski?"
response = llm_chain.run(question)  # Użyj llm_chain.run(question) dla łańcucha
# response = ollama_llm.invoke(question) # Alternatywnie, użyj
ollama_llm.invoke(question) bezpośrednio

print(f"Pytanie: {question}")
print(f"Odpowiedź: {response}")
```

5. **Uruchom kod Python:**

   W terminalu, w katalogu gdzie znajduje się plik `ollama_langchain.py`, uruchom skrypt:

   ```
   python ollama_langchain.py
   ```

   Powinieneś zobaczyć pytanie i odpowiedź wygenerowaną przez lokalny model Ollama.

Wyjaśnienie kodu

- **`from langchain_community.llms import Ollama`**: Importuje klasę `Ollama` z pakietu `langchain-community.llms`, która umożliwia integrację z Ollama w Langchain.
- **`ollama_llm = Ollama(model="llama2")`**: Tworzy instancję modelu Ollama, gdzie `model="llama2"` określa nazwę modelu do użycia. Upewnij się, że nazwa modelu odpowiada modelowi, który został pobrany w kroku 1.
- **`PromptTemplate` i `LLMChain` (opcjonalne, ale zalecane):** Użycie tych klas pozwala na zdefiniowanie szablonu promptu (`PromptTemplate`) i stworzenie łańcucha LLM (`LLMChain`), który łączy prompt z modelem. To ułatwia organizację i zarządzanie promptami, szczególnie w bardziej złożonych aplikacjach.
- **`llm_chain.run(question)` / `ollama_llm.invoke(question)`**: Wykonuje zapytanie do modelu. `llm_chain.run(question)` używa łańcucha LLM, natomiast

`ollama_llm.invoke(question)` (lub `predict()` w starszych wersjach Langchain) wykonuje zapytanie bezpośrednio do modelu Ollama.
- **`print(f"Pytanie: {question}")` i `print(f"Odpowiedź: {response}")`**: Wyświetla zadane pytanie i odpowiedź otrzymaną od modelu.

Co dalej?

Po pomyślnym uruchomieniu podstawowej integracji Ollama z Langchain, możesz eksplorować dalsze możliwości:

- **Eksperymentuj z różnymi modelami Ollama:** Wypróbuj inne modele dostępne w Ollama, zmieniając parametr `model` w `Ollama(model="nazwa_modelu")`.
- **Zmieniaj i ulepszaj prompty:** Modyfikuj szablon promptu w `PromptTemplate`, aby dostosować sposób, w jaki model odpowiada na pytania.
- **Wykorzystaj zaawansowane funkcje Langchain:** Poznaj bardziej zaawansowane łańcuchy, takie jak łańcuchy sekwencyjne, routingowe, czy dodawanie pamięci do konwersacji.
- **Zajrzyj do dokumentacji:** Szczegółowe informacje i przykłady znajdziesz w oficjalnej dokumentacji Langchain: https://python.langchain.com/docs/ oraz dokumentacji integracji z Ollama w `langchain-community`: https://python.langchain.com/docs/integrations/llms/ollama.

Rozwiązywanie problemów

- **Problem z połączeniem z Ollama:** Upewnij się, że serwer Ollama jest uruchomiony poleceniem `ollama serve` i działa na porcie 11434. Sprawdź ustawienia firewalla, czy nie blokuje połączeń.
- **Błąd "Model not found":** Zweryfikuj, czy model został poprawnie pobrany za pomocą `ollama pull nazwa_modelu` i czy nazwa modelu w kodzie Python jest identyczna.
- **Błędy importu Langchain:** Upewnij się, że biblioteki `langchain` i `langchain-community` zostały poprawnie zainstalowane w aktywnym środowisku wirtualnym.

Mam nadzieję, że ten przewodnik okaże się pomocny! Powodzenia w eksperymentowaniu z Ollama i Langchain!