perplexity

Как загружать файл json в ollama и langchain вместе с промтом

Чтобы загрузить JSON-файл и использовать его вместе с Ollama и LangChain, а также передать промпт (запрос) к модели, нужно выполнить несколько шагов, объединяющих загрузку данных и вызов LLM через LangChain с использованием Ollama. Вот подробное руководство:

1. Загрузка JSON-файла в LangChain

LangChain предоставляет класс `JSONLoader` для удобной загрузки JSON и JSONL файлов в объекты `Document`. Это позволяет извлекать нужные поля из JSON с помощью jq-схемы.

Пример загрузки JSON с использованием `JSONLoader`:

```
from langchain.document_loaders import JSONLoader

# Путь к JSON-файлу
file_path = "path/to/your/file.json"

# Создаем загрузчик с указанием jq-схемы для извлечения нужных данных
loader = JSONLoader(
    file_path=file_path,
    jq_schema=".data[]",  # пример пути к массиву в JSON
    content_key=".attributes.message",  # поле, которое будет содержимым документа
    is_content_key_jq_parsable=True
)

# Загружаем данные в список документов LangChain
documents = loader.load()

print(documents)
```

Таким образом можно загрузить и структурировать данные из JSON для последующей обработки[1] [2].

2. Использование Ollama с LangChain

Для работы с Ollama в LangChain есть интеграция через класс `Ollama` из `langchain_community.llms`. Пример создания LLM с Ollama:

```
from langchain_community.llms import Ollama

llm = Ollama(model="mistral")  # или другая модель, загруженная в Ollama
```

Для вызова модели с промптом через LangChain можно использовать цепочку с шаблоном промпта:

```
from langchain.prompts import ChatPromptTemplate
from langchain.chains import LLMChain

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

chain = LLMChain(llm=llm, prompt=prompt)

response = chain.run({"question": "Your question here"})
print(response)
```

3. Совместное использование JSON и Ollama с промптом

Чтобы использовать загруженный JSON и передать его содержимое в промпт Ollama через LangChain, можно сделать так:

- Загрузить JSON в документы LangChain (`documents`).
- Извлечь нужный текст (например, объединить содержимое всех документов).
- Передать этот текст в качестве контекста или части промпта для Ollama.

Пример:

```
# Объединяем содержимое всех документов в один текст
context = "\n".join([doc.page_content for doc in documents])

# Создаем промпт с контекстом и вопросом
prompt_text = f"Данные: {context}\n\nВопрос: {user_question}"

response = llm(prompt_text)
print(response)
```

Или с использованием цепочки и шаблона:

```
from langchain.prompts import PromptTemplate
from langchain.chains import LLMChain

template = """
Данные: {context}

Вопрос: {question}
"""

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

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

```
response = chain.run({"context": context, "question": "Ваш вопрос"})
print(response)
```

4. Особенности и советы

- Убедитесь, что модель Ollama загружена локально (`ollama pull <model>`).
- Для корректной работы с JSON в Ollama через LangChain иногда нужно указывать формат ответа как JSON (`format="json"`), особенно если ожидается структурированный ответ[3] [4].
- При больших JSON-файлах можно делать предварительную фильтрацию или суммирование данных, чтобы не перегружать промпт[5].
- Если используете Chainlit или другой UI, можно интегрировать загрузку JSON и вызов Ollama в обработчики сообщений[6].

Таким образом, общий процесс:

1. Загрузить JSON в LangChain через `JSONLoader`.
2. Подготовить текстовый контекст из JSON.
3. Создать объект Ollama LLM в LangChain.
4. Сформировать промпт с контекстом и вопросом.
5. Вызвать модель и получить ответ.

Это позволит эффективно работать с JSON-данными и Ollama в связке с LangChain.

⁂

1. https://python.langchain.com/docs/how_to/document_loader_json/
2. https://cheatsheet.md/langchain-tutorials/langchain-load-json
3. https://docs.litellm.ai/docs/providers/ollama
4. https://stackoverflow.forums.wiki/viewtopic.php?t=197308
5. https://www.reddit.com/r/ollama/comments/1cr1vit/how_to_get_output_from_the_big_json/?tl=ru
6. https://hackernoon.com/lang/ru/как-использовать-ollama-на-практике-с-местными-фильмами-и-создать-чат-бота