

Xây dựng hệ thống RAG với LangChain và OpenAI

Mục tiêu

- Hiểu cách xử lý dữ liệu từ hệ thống tìm kiếm thông tin (retrieval).
- Hợp nhất dữ liệu đầu ra từ retrieval để sử dụng trong bước sinh văn bản (generation).
- Tạo prompt đơn giản để đưa vào mô hình sinh của OpenAI.
- Tích hợp LangChain để gọi API OpenAI một cách tiện lợi.
- Đánh giá hiệu suất và giới hạn của mô hình.

1. Tổng quan về bước retrieval

Sau khi hệ thống retrieval trả về kết quả, bạn sẽ nhận được một danh sách các tài liệu (docs_files) tương ứng với truy vấn của bạn.

Cấu trúc dữ liệu

```
len(docs_files) # số lượng kết quả
docs_files[0]   # (page_content, score)
```

- docs_files là danh sách các tuple.
- Mỗi tuple gồm:
 - page_content: nội dung văn bản.
 - score: điểm số đánh giá mức độ liên quan.

2. Hợp nhất văn bản để dùng trong bước generation

Trước khi sinh câu trả lời, chúng ta cần ghép các page_content thành một khối văn bản lớn:

```
context_text = "\n\n".join([doc.page_content for doc, score in docs_files])
```

Mục đích:

- Tạo bối cảnh thống nhất cho mô hình sinh văn bản.
- Dễ dàng truyền vào prompt.

3. Tạo prompt đơn giản cho mô hình sinh

```
prompt = f""Based on this context:
```

```
{context_text}
```

```
Please answer this question:
```

```
{query}
```

```
If you don't know the answer, just say you don't know."""
```

Lưu ý:

- Luôn khuyến khích mô hình trả lời "không biết" nếu không có thông tin.
- Tránh hiện tượng "hallucination" (mô hình bịa ra thông tin).

4. Gọi OpenAI API thông qua LangChain

```
from langchain.chat_models import ChatOpenAI
```

```
model = ChatOpenAI(  
    openai_api_key=API_KEY,  
    model_name="gpt-4o",  
    temperature=0  
)
```

- Sử dụng GPT-4o cho hiệu suất tốt hơn.
- `temperature = 0` để đảm bảo tính chính xác, không sáng tạo.

5. Thực thi truy vấn và hiển thị kết quả

```
response_text = model.invoke(prompt).content  
display(Markdown(response_text))
```

- Gọi mô hình để sinh câu trả lời dựa trên `prompt`.
- Hiển thị dưới dạng Markdown để dễ đọc.

6. Phân tích và đánh giá kết quả

- Kết quả chưa hoàn hảo: trả về "no comment" hoặc thông tin không đầy đủ.
- Lý do:
 - Prompt chưa tối ưu.
 - Dữ liệu quá nhiều (2000 tokens x 5 docs).
 - Truy vấn chưa rõ ràng ("Give me my worst reviews with comments").

7. Hướng phát triển tiếp theo

- Tối ưu prompt để rõ ràng hơn (VD: yêu cầu kèm comment cụ thể).
- Giới hạn context hoặc lọc dữ liệu trước khi truyền vào mô hình.
- Tạo các **hàm tái sử dụng** cho quá trình merge, prompt, gọi API.

8. Kết luận

Dù chưa tối ưu, nhưng bạn đã hoàn tất một pipeline RAG đơn giản:

- Tìm kiếm thông tin (Retrieval):** trả về các đoạn văn bản liên quan.
- Ghép context:** tạo ngữ cảnh thống nhất.
- Sinh văn bản (Generation):** mô hình trả lời dựa vào context và truy vấn.

Tác giả: Đỗ Ngọc Tú
Công Ty Phần Mềm VHTSoft

Phiên bản #1

Được tạo 6 tháng 5 2025 16:08:33 bởi Đỗ Ngọc Tú

Được cập nhật 6 tháng 5 2025 16:14:39 bởi Đỗ Ngọc Tú