Module Project

RAG (Retrieval-Augmented Generation) sử dụng Streamlit

Đinh Nhật Thành

Ngày 21 tháng 6 năm 2025

Mục lục

Tóm tắt ý chính

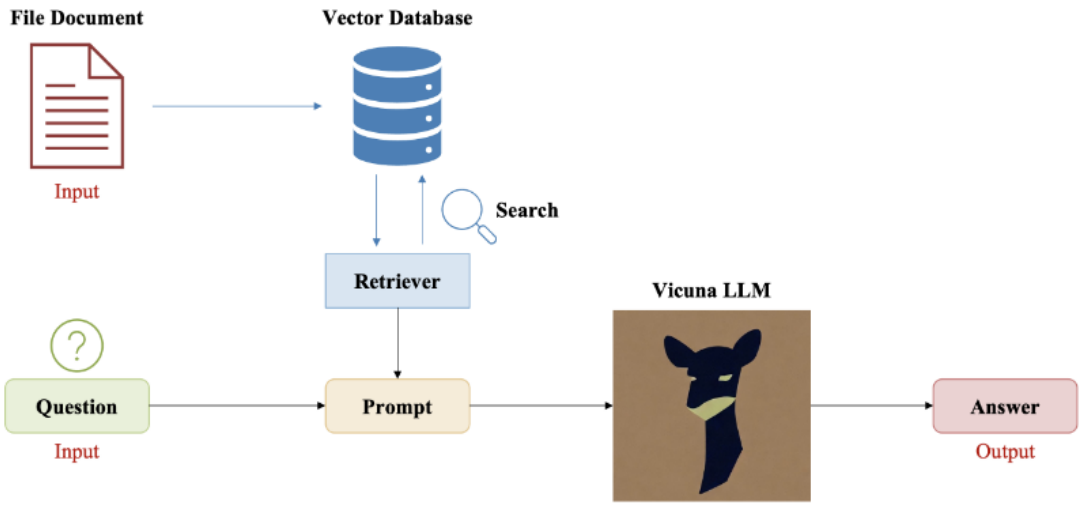
Hệ thống được xây dựng nhằm hỗ trợ người dùng truy vấn thông tin từ tài liệu PDF bằng cách sử dụng mô hình ngôn ngữ lớn (LLM) kết hợp với cơ sở dữ liệu vector. Người dùng upload tài liệu PDF lên giao diện web được xây dựng bằng Streamlit, hệ thống sẽ tự động chia nhỏ văn bản, mã hoá thành vector, và sử dụng RAG để sinh ra câu trả lời theo truy vấn.

Hệ thống RAG sử dụng Streamlit

Kiến trúc hệ thống

Hệ thống bao gồm các thành phần chính sau:

1. Giao diện người dùng bằng **Streamlit**.
2. Phân tích và trích xuất văn bản từ file PDF.
3. Chia nhỏ nội dung thành các đoạn (chunk) bằng `SemanticChunker`.
4. Mã hóa và lưu trữ các đoạn văn bản dưới dạng vector sử dụng `HuggingFaceEmbeddings` và cơ sở dữ liệu `Chroma`.
5. Truy hồi thông tin liên quan dựa trên câu hỏi người dùng.
6. Tạo prompt tùy chỉnh và gửi đến mô hình LLM để sinh câu trả lời.



Hình 1: Kiến trúc tổng quan của hệ thống RAG

```
embeddings = HuggingFaceEmbeddings(
    model_name='bkai-foundation-models/vietnamese-bi-encoder')

model = AutoModelForCausalLM.from_pretrained("lmsys/vicuna-7b-v1.5", ...)
tokenizer = AutoTokenizer.from_pretrained("lmsys/vicuna-7b-v1.5")
pipeline = pipeline('text-generation', model=model, tokenizer=tokenizer)
```

```
embeddings = HuggingFaceEmbeddings(model_name='bkai-foundation-models/vietnamese-bi-encoder')
loader = PyPDFLoader(tmp_file_path)
documents = loader.load()

splitter = SemanticChunker(embeddings=embeddings, ...)
docs = splitter.split_documents(documents)
```

```
loader = PyPDFLoader(tmp_file_path)
vector_db = Chroma.from_documents(docs, embedding=embeddings)
retriever = vector_db.as_retriever()
```

3. Truy hồi văn bản liên quan

```
st.set_page_config(page_title="PDF RAG Assistant")
uploaded_file = st.file_uploader("Tải PDF")
question = st.text_input("Đặt câu hỏi:")
```

4. Giao diện Streamlit

```
st.set_page_config(page_title="PDF RAG Assistant")
uploaded_file = st.file_uploader("Tải PDF")
question = st.text_input("Đặt câu hỏi:")
```

Prompting

Mục tiêu prompt

Prompt được thiết kế để đảm bảo mô hình trả về kết quả như mong đợi:

Prompt Thông Thường

Prompt có chỉ dẫn, ví dụ

```
Dựa vào nội dung sau, hãy:
1. Tóm tắt tối đa 3 ý chính, kèm theo số trang nếu có.
2. Trả lời câu hỏi bằng tiếng Việt ngắn gọn và chính xác.
3. Nếu không có thông tin liên quan, hãy để "answer" là "Không có dữ liệu liên quan

Trả kết quả ở định dạng JSON như sau:

{
    "main_ideas": [
    {"point": "Ý chính 1", "source": "Trang ..."},
    {"point": "Ý chính 2", "source": "Trang ..."}
    ],
    "answer": "Câu trả lời"
}

Nội dung tài liệu: {context}
Câu hỏi: {question}
```

```
Output:
 Nội
 Câu hỏi:
 Các mối đe dọa tầng cảm biến trong IoT

 Trả lời:
    * Khai thác thiết bị vật lý: Người tấn công có thể trích xuất dữ liệu từ bộ nhớ
    * Tấn công side-channel: Người tấn công có thể chiết xuất khóa bí mật dùng để m
    * Chỉnh sửa firmware: Người tấn công có thể lợi dụng việc thiếu cơ chế xác thực
```