

Unit test cho hệ thống RAG

Viết **unit test cho hệ thống RAG (Retrieval-Augmented Generation)** giúp đảm bảo rằng các thành phần chính như Retriever, Generator, và Data Pipeline hoạt động chính xác, độc lập và có thể kiểm soát được. Dưới đây là hướng dẫn thực hành cách viết **unit test** cho hệ thống RAG sử dụng Python (với `pytest`) và thư viện phổ biến như LangChain hoặc custom code.

1. Các thành phần cần kiểm thử

Hệ thống RAG thường gồm:

- Retriever** – Tìm kiếm các đoạn văn bản phù hợp từ kho dữ liệu.
- Generator** – Sinh câu trả lời dựa trên ngữ cảnh và câu hỏi.
- RAG Pipeline** – Tổng thể pipeline kết hợp cả retriever và generator.
- Post-processing** (tùy chọn) – Xử lý đầu ra của LLM.

2. Cấu trúc ví dụ RAG

Giả sử bạn có pipeline như sau:

```
class RAGPipeline:
    def __init__(self, retriever, generator):
        self.retriever = retriever
        self.generator = generator

    def run(self, query):
        documents = self.retriever.retrieve(query)
        return self.generator.generate(query, documents)
```

3. Cách viết unit test

3.1. Tạo file `test_rag.py`

```
import pytest
from unittest.mock import MagicMock
from rag_pipeline import RAGPipeline

def test_rag_pipeline_returns_expected_output():
    # Mock retriever
```

```

mock_retriever = MagicMock()
mock_retriever.retrieve.return_value = ["This is a test document."]

# Mock generator
mock_generator = MagicMock()
mock_generator.generate.return_value = "This is a generated answer."

# Create pipeline
pipeline = RAGPipeline(mock_retriever, mock_generator)
result = pipeline.run("What is this?")

# Assertions
mock_retriever.retrieve.assert_called_once_with("What is this?")
mock_generator.generate.assert_called_once_with("What is this?", ["This is a test document."])
assert result == "This is a generated answer."

```

3.2. Test retriever riêng biệt

```

def test_retriever_returns_relevant_docs():
    from my_retriever import SimpleRetriever
    retriever = SimpleRetriever(["Paris is the capital of France."])
    docs = retriever.retrieve("What is the capital of France?")
    assert any("Paris" in doc for doc in docs)

```

4. Công cụ và kỹ thuật nâng cao

- **pytest fixtures** để khởi tạo dữ liệu.
- **mocking LLM API calls** để tránh chi phí gọi thực tế.
- **test coverage** để kiểm tra phần nào chưa được test.
- **snapshot testing** để so sánh kết quả sinh tự động với mẫu.

5. Chạy test

```

pytest test_rag.py -v

```

Gợi ý mở rộng

- Test tích hợp: Chạy full pipeline với vectordb thật (FAISS, Chroma).
- So sánh kết quả RAG vs non-RAG (benchmark chất lượng sinh).
- Kết hợp với **Promptfoo**, **LangSmith**, hoặc **TruLens** để test LLM đầu ra tự động.

Phiên bản #1

Được tạo 7 tháng 5 2025 16:06:28 bởi Đỗ Ngọc Tú

Được cập nhật 7 tháng 5 2025 16:10:46 bởi Đỗ Ngọc Tú