

Bài Thực Hành: Đánh Giá Hệ Thống RAG với TruLens và LangChain

Mục Tiêu

- Xây dựng một hệ thống RAG đơn giản sử dụng LangChain.
- Tích hợp TruLens để theo dõi và đánh giá hiệu suất của hệ thống.
- Áp dụng các hàm phản hồi (feedback functions) để đo lường các chỉ số như độ liên quan, tính chính xác và sự phù hợp của ngữ cảnh.docs.pinecone.io+1TruEra+1

Bước 1: Cài Đặt Môi Trường

Cài đặt các thư viện cần thiết:

```
pip install trulens trulens-apps-langchain trulens-providers-openai openai langchain langchainhub langchain-openai langchain_community faiss-cpu bs4 tiktoken
```

Thiết lập khóa API cho OpenAI:

```
import os  
os.environ["OPENAI_API_KEY"] = "sk-..." # Thay thế bằng khóa API của bạn
```

Bước 2: Tạo Hệ Thống RAG Đơn Giản

Tạo một ứng dụng RAG đơn giản sử dụng LangChain:

```
from langchain.chains import RetrievalQA  
from langchain.vectorstores import FAISS  
from langchain.embeddings import OpenAIEmbeddings  
from langchain.chat_models import ChatOpenAI  
from langchain.document_loaders import TextLoader  
  
# Tải dữ liệu
```

```

loader = TextLoader("data.txt")
documents = loader.load()

# Tạo vector store
embeddings = OpenAIEmbeddings()
vectorstore = FAISS.from_documents(documents, embeddings)

# Tạo hệ thống RAG
qa_chain = RetrievalQA.from_chain_type(
    llm=ChatOpenAI(temperature=0),
    retriever=vectorstore.as_retriever()
)

```

Bước 3: Tích Hợp TruLens

Sử dụng TruLens để theo dõi và đánh giá hệ thống: [YouTube+5TruLens+5YouTube+5](#)

```

from trulens_eval import Tru
from trulens_eval.feedback import Feedback
from trulens_eval.feedback.provider.openai import OpenAI as OpenAIFeedbackProvider
from trulens_eval.apps.langchain import instrument_langchain

# Khởi tạo Tru
tru = Tru()

# Thiết lập các hàm phản hồi
openai_provider = OpenAIFeedbackProvider()
f_qa_relevance = Feedback(openai_provider.relevance).on_input_output()
f_context_relevance = Feedback(openai_provider.relevance).on_input().on_retrieved_context()
f_groundedness = Feedback(openai_provider.groundedness).on_input().on_retrieved_context().on_output()

# Tích hợp TruLens vào hệ thống
qa_chain_recorder = instrument_langchain(
    qa_chain,
    app_id="langchain_app",
    feedbacks=[f_qa_relevance, f_context_relevance, f_groundedness]
)

```

Bước 4: Gửi Truy Vấn và Đánh Giá

Gửi truy vấn và đánh giá phản hồi:

```
with qa_chain_recorder as recorder:  
    response = qa_chain_recorder.query("What is the capital of France?")  
    print(response)
```

Bước 5: Khám Phá Kết Quả trong Dashboard

Khởi chạy dashboard của TruLens để xem kết quả đánh giá: docs.pinecone.io

```
tru.run_dashboard()
```

Dashboard sẽ hiển thị các chỉ số như:

- **QA Relevance:** Độ liên quan giữa câu hỏi và câu trả lời.
 - **Context Relevance:** Độ liên quan giữa truy vấn và ngữ cảnh được truy xuất.
 - **Groundedness:** Mức độ mà câu trả lời dựa trên ngữ cảnh được cung cấp. docs.pinecone.io
- [Lab Lab+2Analytics Vidhya+2Zilliz+2](#)

Bước 6: Tối Ưu Hệ Thống

Dựa trên các phản hồi và chỉ số từ TruLens, bạn có thể:

- Điều chỉnh kích thước chunk hoặc chiến lược chunking.
- Thay đổi mô hình embedding hoặc vector store.
- Tối ưu prompt để cải thiện độ chính xác và tính phù hợp của câu trả lời. [Zilliz+1TruLens+1](#)

Phiên bản #1

Được tạo 8 tháng 5 2025 04:01:49 bởi Đỗ Ngọc Tú

Được cập nhật 8 tháng 5 2025 04:06:38 bởi Đỗ Ngọc Tú