**Quy trình xây dựng kho vector phục vụ cho RAG (Retrieval-Augmented Generation) bằng ChromaDB**

1. Chuyển PDF sang văn bản (text)
2. Chia nhỏ văn bản thành các đoạn (chunk)
3. Tạo embedding từ các đoạn
4. Lưu vào ChromaDB để có thể truy vấn sau này

**1. CÀI ĐẶT THƯ VIỆN hoặc dùng requirement**

pip install chromadb langchain pypdf sentence-transformers

**requirement.txt**

langchain

chromadb

sentence-transformers

pypdf

transformers

torch

accelerate

langchain-community

**2. Python**

from langchain.document\_loaders import PyPDFLoader

from langchain.text\_splitter import RecursiveCharacterTextSplitter

from langchain.embeddings import SentenceTransformerEmbeddings

import chromadb

from langchain.vectorstores import Chroma

# 1. Load PDF

loader = PyPDFLoader("BigData.pdf")

pages = loader.load() # Mỗi page là một Document object

# 2. Chia nhỏ văn bản

text\_splitter = RecursiveCharacterTextSplitter(

chunk\_size=500, # độ dài đoạn văn bản tối đa

chunk\_overlap=50 # trùng lặp giữa các đoạn

)

docs = text\_splitter.split\_documents(pages)

# 3. Khởi tạo embedding model

embedding\_function = SentenceTransformerEmbeddings(model\_name="all-MiniLM-L6-v2")

# 4. Tạo và lưu kho vector với Chroma

persist\_directory = "chroma\_bigdata"

vectordb = Chroma.from\_documents(

documents=docs,

embedding=embedding\_function,

persist\_directory=persist\_directory

)

vectordb.persist() # lưu database ra đĩa

print(" Kho vector đã được tạo và lưu thành công!")

**3. TRUY VẤN (RAG) :**

# Load lại ChromaDB đã lưu để truy vấn

vectordb = Chroma(

persist\_directory="chroma\_bigdata",

embedding\_function=embedding\_function

)

# Truy vấn

query = "Ứng dụng của dữ liệu lớn trong thống kê nhà nước là gì?"

docs = vectordb.similarity\_search(query, k=3)

for i, doc in enumerate(docs):

print(f"--- Kết quả {i+1} ---\n{doc.page\_content}\n")

**Mini project**

my\_rag\_project/

│

├── BigData.pdf

├── build\_vectorstore.py

├── rag\_query.py

├── requirements.txt

└── chroma\_bigdata/ ← thư mục chứa kho vector

build\_vectorstore.py: Chuyển PDF → chia đoạn → tạo embedding → lưu vào ChromaDB

rag\_query.py: Load vector DB → cho phép nhập câu hỏi → truy vấn vector → in ra câu trả lời gần đúng nhất

**build\_vectorstore.py:**

from langchain.document\_loaders import PyPDFLoader

from langchain.text\_splitter import RecursiveCharacterTextSplitter

from langchain.embeddings import SentenceTransformerEmbeddings

from langchain.vectorstores import Chroma

# Đường dẫn tới file PDF

pdf\_path = "BigData.pdf"

# Load file PDF

loader = PyPDFLoader(pdf\_path)

pages = loader.load()

# Chia nhỏ thành các đoạn văn bản

text\_splitter = RecursiveCharacterTextSplitter(

    chunk\_size=500,

    chunk\_overlap=50

)

documents = text\_splitter.split\_documents(pages)

# Tạo embedding từ mô hình Sentence Transformer

embedding\_function = SentenceTransformerEmbeddings(model\_name="all-MiniLM-L6-v2")

# Lưu vector store

vectorstore = Chroma.from\_documents(

    documents=documents,

    embedding=embedding\_function,

    persist\_directory="chroma\_bigdata"

)

vectorstore.persist()

print("✅ Vector store đã được tạo và lưu tại 'chroma\_bigdata'")

**📄 rag\_query.py**

from langchain.vectorstores import Chroma

from langchain.embeddings import SentenceTransformerEmbeddings

# Load embedding và vector store

embedding\_function = SentenceTransformerEmbeddings(model\_name="all-MiniLM-L6-v2")

vectorstore = Chroma(

    persist\_directory="chroma\_bigdata",

    embedding\_function=embedding\_function

)

# Nhập truy vấn từ người dùng

query = input("🔍 Câu hỏi của bạn: ")

# Truy vấn

results = vectorstore.similarity\_search(query, k=3)

# Hiển thị kết quả

print("\n📄 Các đoạn văn bản gần nhất:")

for i, doc in enumerate(results, 1):

    print(f"\n--- Kết quả {i} ---\n{doc.page\_content}")

**🧪 Cách chạy**

**#1 Tạo môi trường chạy có 2 cách ( ko được cách 1 thì làm cách 2)**

**C1:**

python -m venv rag\_env

source rag\_env/bin/activate # hoặc .\rag\_env\Scripts\activate trên Windows

**C2:**

pip install virtualenv

virtualenv rag\_env

.\rag\_env\Scripts\Activate.ps1

pip install langchain-community

**# 2. Cài thư viện**

pip install -r requirements.txt

**# 3. Tạo kho vector**

python build\_vectorstore.py

**# 4. Truy vấn**

python rag\_query.py