**Cấu trúc và Chức năng của Dự án HR Assistant**

**Tổng quan**

Dự án HR Assistant là một hệ thống trợ lý ảo chuyên về nhân sự, sử dụng cơ sở dữ liệu vector PostgreSQL với pgvector để lưu trữ và truy xuất dữ liệu từ tài liệu. Hệ thống này có khả năng trả lời các câu hỏi liên quan đến chính sách nhân sự, quy trình tuyển dụng, đào tạo và các vấn đề nhân sự khác dựa trên tài liệu nội bộ.

**Các thành phần chính**

1. Cấu trúc thư mục

* data/: Chứa tài liệu nguồn (PDF, TXT) để huấn luyện hệ thống
* src/: Mã nguồn chính của dự án
* logs/: Lưu trữ nhật ký hệ thống
* static/: Tài nguyên tĩnh cho giao diện người dùng
* requirements.txt: Danh sách các thư viện cần thiết

1. Các tệp mã nguồn chính

* db\_utils.py: Tiện ích làm việc với PostgreSQL và pgvector
* ingest.py: Xử lý và nhập tài liệu vào cơ sở dữ liệu vector
* main.py: Chứa logic chính của trợ lý HR
* app.py: Giao diện người dùng (Streamlit)
* setup\_postgres.py: Cài đặt và cấu hình PostgreSQL với pgvector

**Luồng dữ liệu và quy trình hoạt động**

1. Nhập dữ liệu (Ingestion)

* Đọc tài liệu từ thư mục data/ (hỗ trợ PDF, TXT)
* Chia nhỏ tài liệu thành các đoạn văn bản
* Tạo vector nhúng (embeddings) từ các đoạn văn bản bằng mô hình all-MiniLM-L6-v2
* Lưu trữ các vector này vào PostgreSQL với pgvector

1. Truy vấn và Phản hồi

* Người dùng đặt câu hỏi
* Hệ thống chuyển đổi câu hỏi thành vector nhúng
* Tìm kiếm các đoạn văn bản tương tự trong cơ sở dữ liệu
* Sử dụng mô hình ngôn ngữ (LLM) để tạo câu trả lời dựa trên ngữ cảnh tìm được

**Công nghệ chính**

1. Cơ sở dữ liệu Vector:

* PostgreSQL với pgvector extension
* Kết nối thông qua chuỗi: **postgresql://anhnguyen:password@localhost:5432/hr\_assistant**
* Tên bộ sưu tập mặc định: **hr\_**documents

1. Xử lý ngôn ngữ tự nhiên:

* Mô hình nhúng: HuggingFace all-MiniLM-L6-v2
* Mô hình LLM: Claude (Anthropic)/Gemini
* Chain LangChain với RetrievalQA

1. Công cụ và thư viện:

* LangChain: Framework xử lý chuỗi AI
* Psycopg2: Kết nối PostgreSQL
* DirectoryLoader: Đọc tài liệu
* RecursiveCharacterTextSplitter: Chia nhỏ tài liệu

**Đặc điểm nổi bật**

1. Cơ sở dữ liệu bền vững: Sử dụng PostgreSQL thay vì database vector nhẹ hơn như ChromaDB
2. Hiệu suất cao: pgvector cung cấp khả năng tìm kiếm tương tự hiệu quả
3. Tích hợp SQL: Kết hợp tìm kiếm vector với các truy vấn SQL
4. Tuân thủ ACID: Đảm bảo độ tin cậy và nhất quán dữ liệu

Dự án được thiết kế với kiến trúc module, giúp dễ dàng mở rộng và bảo trì trong tương la

**Chi tiết về Công nghệ Xử lý Ngôn ngữ Tự nhiên trong Dự án HR Assistant**

**1. Mô hình nhúng: HuggingFace all-MiniLM-L6-v2**

**Tổng quan**

* **all-MiniLM-L6-v2** là một mô hình nhúng (embedding model) được phát triển bởi Microsoft và được cung cấp qua HuggingFace
* Mô hình này là phiên bản nhỏ gọn (distilled) của mô hình BERT lớn hơn, được tối ưu hóa để tạo ra các vector nhúng chất lượng cao với hiệu suất tính toán tốt

**Đặc điểm kỹ thuật**

* **Kích thước**: L6 (6 lớp) - nhỏ gọn hơn so với các mô hình BERT tiêu chuẩn (thường có 12-24 lớp)
* **Không gian vector**: Tạo ra vector 384 chiều cho mỗi đoạn văn bản
* **Tối ưu hóa**: Sử dụng kỹ thuật chưng cất (distillation) để giảm kích thước mô hình trong khi vẫn giữ được hiệu suất tốt
* **Hiệu năng**: Cân bằng tốt giữa chất lượng và tốc độ xử lý

**Cách sử dụng trong dự án**

* Trong tệp **ingest.py**, mô hình được sử dụng để tạo ra vector nhúng cho các đoạn văn bản:

*embeddings = HuggingFaceEmbeddings(model\_name="all-MiniLM-L6-v2")*

* Các đoạn văn bản (chunks) từ tài liệu HR được chuyển đổi thành vector bằng mô hình này
* Những vector này sau đó được lưu trữ trong PostgreSQL với pgvector extension để hỗ trợ tìm kiếm tương tự

**Lợi ích**

* Hiệu quả trong việc nắm bắt ngữ nghĩa của văn bản
* Cho phép tìm kiếm ngữ nghĩa (thay vì chỉ tìm kiếm từ khóa)
* Tối ưu cho các ứng dụng thực tế nhờ kích thước nhỏ gọn và yêu cầu tài nguyên thấp

**2. Mô hình LLM: Claude (Anthropic)/Gemini**

**Tổng quan**

* **Claude** là một mô hình ngôn ngữ lớn (LLM) do Anthropic phát triển
* Được thiết kế với trọng tâm vào sự hữu ích, trung thực và an toàn

**Đặc điểm kỹ thuật**

* **Kiến trúc**: Dựa trên kiến trúc Transformer, tương tự như các mô hình LLM hiện đại khác
* **Training**: Được huấn luyện với phương pháp Constitutional AI (CAI) - tập trung vào việc điều chỉnh mô hình theo các nguyên tắc giá trị cụ thể
* **Cửa sổ ngữ cảnh**: Hỗ trợ cửa sổ ngữ cảnh lớn (100K tokens trong phiên bản mới nhất)
* **Khả năng đa phương thức**: Phiên bản mới hỗ trợ xử lý cả văn bản và hình ảnh

**Cách sử dụng trong dự án**

* Trong dự án HR Assistant, Claude được sử dụng để tạo ra câu trả lời dựa trên ngữ cảnh tìm được từ tài liệu
* Dựa trên thông tin từ memory, hệ thống đã được cấu hình để sử dụng Claude (Anthropic's LLM) cho việc trả lời câu hỏi

**Quy trình hoạt động**

1. Câu hỏi của người dùng được chuyển đổi thành vector bằng mô hình all-MiniLM-L6-v2
2. Hệ thống tìm kiếm các đoạn văn bản tương tự trong PostgreSQL dựa trên khoảng cách vector
3. Các đoạn văn bản có liên quan được truyền vào Claude cùng với câu hỏi
4. Claude phân tích ngữ cảnh và tạo ra câu trả lời phù hợp, chỉ dựa trên thông tin có trong tài liệu

**Lợi ích**

* Khả năng hiểu ngữ cảnh sâu và tạo câu trả lời chất lượng cao
* Có thể dựa chặt chẽ vào thông tin trong tài liệu tham khảo, tránh "hallucination"
* Phản hồi tự nhiên và giống con người, phù hợp cho vai trò trợ lý HR
* Khả năng xử lý câu hỏi phức tạp về chính sách nhân sự, quy trình và quy định

**Tích hợp hai mô hình trong dự án**

Trong dự án HR Assistant, việc kết hợp mô hình nhúng all-MiniLM-L6-v2 và mô hình LLM Claude tạo ra một hệ thống truy xuất thông tin mạnh mẽ:

1. **Giai đoạn nhúng và lưu trữ**:
   * Tài liệu được chia thành các đoạn nhỏ (500 ký tự với độ chồng lấp 50 ký tự)
   * all-MiniLM-L6-v2 chuyển đổi các đoạn này thành vector
   * Các vector được lưu trữ trong PostgreSQL với pgvector extension
2. **Giai đoạn truy vấn và trả lời**:
   * Câu hỏi được chuyển đổi thành vector bằng cùng một mô hình nhúng
   * Hệ thống tìm k=3 đoạn văn bản tương tự nhất
   * Claude sử dụng context này để tạo ra câu trả lời phù hợp

Sự kết hợp này tận dụng điểm mạnh của cả hai công nghệ: khả năng tìm kiếm tương tự hiệu quả của mô hình nhúng và khả năng tạo văn bản tự nhiên, chính xác của LLM.

**Mô hình all-MiniLM-L6-v2: Phí và Khả năng Xử lý Tài liệu Lớn**

**Về phí sử dụng**

**all-MiniLM-L6-v2 hoàn toàn miễn phí** để sử dụng với các lý do sau:

* Đây là mô hình mã nguồn mở được cung cấp thông qua HuggingFace Hub
* Được phát hành bởi Microsoft dưới giấy phép MIT License, cho phép sử dụng tự do trong cả dự án thương mại và phi thương mại
* Mô hình có thể được tải về và chạy cục bộ trên máy tính của bạn, không cần phí API hay phí đăng ký

**Khả năng xử lý tài liệu số lượng lớn**

**all-MiniLM-L6-v2 có thể xử lý số lượng tài liệu lớn** với một số cân nhắc:

**Ưu điểm cho tài liệu lớn:**

1. **Kích thước nhỏ gọn**: Đây là mô hình được chưng cất (distilled) từ các mô hình lớn hơn, nên nhẹ hơn và nhanh hơn đáng kể
2. **Hiệu quả tính toán**: Chỉ có 6 lớp (so với 12-24 lớp trong BERT), giúp tạo vector nhanh hơn
3. **Chạy offline**: Có thể chạy cục bộ trên CPU thông thường (không bắt buộc cần GPU)
4. **Vector kích thước vừa phải**: Tạo ra vector 384 chiều, cân bằng giữa chất lượng và hiệu quả lưu trữ

**Giới hạn và cân nhắc:**

1. **Tài nguyên phần cứng**:
   * Khi xử lý hàng ngàn tài liệu, cần đủ RAM (ít nhất 8-16GB để xử lý hiệu quả)
   * GPU có thể đẩy nhanh quá trình nhưng không bắt buộc
2. **Thời gian xử lý**:
   * Mô hình có thể xử lý khoảng 30-100 đoạn văn bản mỗi giây trên CPU thông thường
   * Với 10.000 đoạn văn bản, thời gian xử lý có thể từ vài phút đến hơn một giờ tùy vào phần cứng
3. **Lưu trữ vector**:
   * PostgreSQL với pgvector (như trong dự án của bạn) là lựa chọn tốt cho việc lưu trữ và truy vấn số lượng lớn vector
   * pgvector đã được chứng minh khả năng xử lý hàng triệu vector

**Tùy chỉnh cho tối ưu hóa**

Để xử lý hiệu quả số lượng lớn tài liệu, bạn có thể tùy chỉnh quy trình trong dự án hiện tại:

1. **Xử lý song song**: Thêm xử lý đa luồng khi tạo embeddings
2. **Điều chỉnh kích thước chunk**: Giảm **chunk\_size** xuống (hiện đang là 500) nếu bạn có quá nhiều tài liệu
3. **Batch processing**: Thêm xử lý theo batch khi thêm tài liệu vào cơ sở dữ liệu:

Với những tối ưu hóa này, all-MiniLM-L6-v2 hoàn toàn khả thi để xử lý từ hàng chục nghìn đến hàng trăm nghìn tài liệu trong một cài đặt doanh nghiệp thông thường.

Mô hình all-MiniLM-L6-v2 trong dự án HR Assistant là một mô hình nhúng chạy hoàn toàn offline. Nó sẽ cần kết nối internet trong lần chạy đầu tiên để tải mô hình về, nhưng sau đó sẽ hoạt động mà không cần kết nối mạng. Điều này đảm bảo tính riêng tư, hiệu suất và độ tin cậy cao cho ứng dụng HR Assistant.

**Cách Xóa Mô Hình HuggingFace Đã Tải Về**

Các mô hình HuggingFace (bao gồm all-MiniLM-L6-v2) sau khi tải về sẽ được lưu trữ trong thư mục cache. Dưới đây là các phương pháp để xóa mô hình đã tải:

**1. Xóa Thủ công qua Thư mục Cache**

Thư mục cache mặc định của HuggingFace thường nằm ở:

~/.cache/huggingface/

# Xem nội dung thư mục cache

ls -la ~/.cache/huggingface/hub/

# Xóa một mô hình cụ thể (ví dụ: all-MiniLM-L6-v2)

rm -rf ~/.cache/huggingface/hub/models--sentence-transformers--all-MiniLM-L6-v2/

**Cách Xử Lý Chia Đoạn Tiếng Việt trong RecursiveCharacterTextSplitter**

Trong dự án HR Assistant, tài liệu được chia thành các đoạn nhỏ với cấu hình:

text\_splitter = RecursiveCharacterTextSplitter(

chunk\_size=500,

chunk\_overlap=50,

)

**Cách tính ký tự đối với tiếng Việt**

Khi xử lý tiếng Việt bằng RecursiveCharacterTextSplitter, ký tự được tính theo cách sau:

1. **Unicode codepoint**: Mỗi ký tự Unicode được xem là một đơn vị riêng biệt
2. **Dấu thanh và dấu phụ**: Trong tiếng Việt (UTF-8), các chữ có dấu như "ă", "ê", "ơ" hoặc "ả", "ạ", "à" đều được tính là MỘT ký tự riêng biệt
3. **Khoảng trắng và dấu câu**: Cũng được tính là một ký tự riêng

**Ví dụ cụ thể**

Giả sử chúng ta có đoạn văn bản tiếng Việt sau:

*"Công ty ABC có chính sách nghỉ phép hàng năm như sau: Mỗi nhân viên được hưởng 12 ngày nghỉ phép có lương mỗi năm. Nhân viên có thể tích lũy tối đa 24 ngày nghỉ phép. Việc xin nghỉ phép cần được thực hiện trước ít nhất 3 ngày làm việc thông qua hệ thống quản lý nhân sự. Trong trường hợp khẩn cấp, nhân viên có thể liên hệ trực tiếp với quản lý để xin phép. Phòng nhân sự sẽ xem xét và phê duyệt đơn nghỉ phép trong vòng 24 giờ kể từ khi nhận được."*

**Cách chia đoạn với chunk\_size=500, chunk\_overlap=50:**

1. **Đoạn 1** (500 ký tự):

*"Công ty ABC có chính sách nghỉ phép hàng năm như sau: Mỗi nhân viên được hưởng 12 ngày nghỉ phép có lương mỗi năm. Nhân viên có thể tích lũy tối đa 24 ngày nghỉ phép. Việc xin nghỉ phép cần được thực hiện trước ít nhất 3 ngày làm việc thông qua hệ thống quản lý nhân sự. Trong trường hợp khẩn cấp, nhân viên có thể liên hệ trực tiếp với quản lý để xin phép. Phòng nhân sự sẽ xem xét và phê duyệt đơn nghỉ phép trong vòng"*

1. **Đoạn 2** (bắt đầu từ ký tự thứ 450, bao gồm 50 ký tự chồng lấp):

*"nhân viên có thể liên hệ trực tiếp với quản lý để xin phép. Phòng nhân sự sẽ xem xét và phê duyệt đơn nghỉ phép trong vòng 24 giờ kể từ khi nhận được."*

**Đặc điểm kỹ thuật**

1. **Điểm cắt thông minh**: RecursiveCharacterTextSplitter cố gắng cắt tại ranh giới tự nhiên như cuối câu (dấu chấm), dấu chấm phẩy, dấu xuống dòng, hoặc khoảng trắng - đây là điểm mạnh khi xử lý tiếng Việt
2. **Cách đếm ký tự với tiếng Việt**:
   * Chữ không dấu (a, b, c...): 1 ký tự
   * Chữ có dấu (ă, â, ê, ô...): 1 ký tự
   * Chữ có dấu thanh (à, á, ả, ã, ạ...): 1 ký tự
   * Dấu câu (. , ? !): 1 ký tự
   * Khoảng trắng: 1 ký tự
3. **Ví dụ đếm ký tự**:
   * "Xin chào" = 7 ký tự (X + i + n + [space] + c + h + à + o)
   * "Việt Nam" = 8 ký tự (V + i + ệ + t + [space] + N + a + m)

**Phân Tích Việc Sử Dụng Gemini 1.5 Pro Trong Ứng Dụng Trợ Lý HR**

Việc sử dụng **model="gemini-1.5-pro"** trong ứng dụng Trợ lý HR của bạn là một lựa chọn tốt vì những lý do sau:

**Khả Năng Của Mô Hình**

1. **Hiểu Đa Phương Tiện**: Gemini 1.5 Pro có thể xử lý cả đầu vào văn bản và hình ảnh (mặc dù hiện tại ứng dụng của bạn chỉ sử dụng văn bản).
2. **Cửa Sổ Ngữ Cảnh Lớn**: Gemini 1.5 Pro có cửa sổ ngữ cảnh rất lớn (lên đến 1 triệu token), lớn hơn đáng kể so với nhiều mô hình khác. Điều này cho phép Trợ lý HR của bạn xử lý và hiểu nhiều ngữ cảnh hơn từ tài liệu HR của bạn cùng một lúc.
3. **Hỗ Trợ Đa Ngôn Ngữ**: Mô hình có khả năng tiếng Việt mạnh mẽ, điều này rất cần thiết cho Trợ lý HR của bạn khi cần trả lời câu hỏi bằng tiếng Việt.
4. **Khả Năng Suy Luận**: Mô hình có khả năng suy luận tốt, giúp hiểu rõ hơn các chính sách HR phức tạp và cung cấp câu trả lời chính xác dựa trên tài liệu của bạn.

**Cân Nhắc Về Hiệu Suất**

1. **Độ Trễ & Thời Gian Phản Hồi**: Gemini 1.5 Pro cân bằng giữa sức mạnh và tốc độ - nó có khả năng mạnh hơn các mô hình nhỏ trong khi vẫn cung cấp phản hồi đủ nhanh cho ứng dụng chat tương tác.
2. **Hiệu Quả Chi Phí**: Mặc dù không phải là mô hình rẻ nhất, nó cung cấp sự cân bằng tốt giữa hiệu suất và chi phí so với các mô hình lớn nhất hiện có.

**Gemini 1.5 Pro - Không Miễn Phí**

Không, **gemini-1.5-pro** không miễn phí. Đây là mô hình cao cấp của Google với chi phí như sau:

* **Đầu vào**: $7.00 cho 1 triệu token
* **Đầu ra**: $21.00 cho 1 triệu token

Tuy nhiên, Google thường cung cấp **hạn ngạch miễn phí ban đầu** khi bạn đăng ký mới (thường là $300 tín dụng Google Cloud trong 90 ngày), có thể dùng để thử nghiệm Gemini.

**Các Lựa Chọn Miễn Phí của Google**

1. **Gemini 1.0 Pro** - **CÓ MIỄN PHÍ** (giới hạn):
   * Google API cung cấp hạn ngạch miễn phí 60 yêu cầu/phút
   * Giới hạn 2 triệu ký tự/ngày
   * Lý tưởng cho các ứng dụng phát triển và thử nghiệm
2. **Gemini 1.5 Flash** - **CÓ MIỄN PHÍ** (có giới hạn):
   * Một phiên bản nhẹ hơn, nhanh hơn của Gemini 1.5
   * Có hạn ngạch miễn phí tương tự Gemini 1.0 Pro
   * Hiệu suất tốt hơn Gemini 1.0 Pro nhưng kém hơn Gemini 1.5 Pro
3. **Vertex AI Free Tier**:
   * Google Cloud cung cấp gói miễn phí có giới hạn cho Vertex AI
   * Cho phép sử dụng một số mô hình Gemini với giới hạn hàng tháng

**Phân Tích Sử Dụng Mistral 7B Instruct chạy local Offline cho HR Assistant**

**Tổng Quan Về Mistral 7B Instruct**

Mistral 7B Instruct là một mô hình ngôn ngữ lớn (LLM) mã nguồn mở, được thiết kế để chạy cục bộ (locally) trên máy tính cá nhân hoặc máy chủ của bạn, thay vì phụ thuộc vào API của bên thứ ba như Google Gemini hoặc Claude.

**Ưu Điểm**

1. **Hoàn Toàn Miễn Phí**:
   * Không có chi phí API, bạn có thể sử dụng không giới hạn
   * Tiết kiệm chi phí đáng kể so với Gemini 1.5 Pro ($7/1M token đầu vào, $21/1M token đầu ra)
2. **Chạy Offline**:
   * Không cần kết nối internet để hoạt động
   * Bảo mật dữ liệu tốt hơn vì dữ liệu không rời khỏi máy chủ của bạn
3. **Kiểm Soát Hoàn Toàn**:
   * Tùy chỉnh và tinh chỉnh mô hình cho miền cụ thể (HR)
   * Không phụ thuộc vào các thay đổi chính sách API từ Google
4. **Hiệu Suất Tốt Cho Mô Hình Nhẹ**:
   * Khả năng hiểu và trả lời khá tốt cho một mô hình chỉ 7B tham số
   * Tối ưu hóa tốt cho các nhiệm vụ như hỏi đáp về tài liệu (RAG)

**Nhược Điểm**

1. **Yêu Cầu Phần Cứng**:
   * Cần GPU với ít nhất 8GB VRAM để chạy hiệu quả
   * Máy tính không có GPU mạnh sẽ chạy rất chậm
2. **Hiệu Suất Tiếng Việt**:
   * Khả năng tiếng Việt không bằng Gemini 1.5 Pro
   * Cần tinh chỉnh (fine-tuning) thêm để cải thiện hiệu suất cho ngôn ngữ Việt
3. **Độ Trễ Cao Hơn**:
   * Thời gian phản hồi có thể chậm hơn so với các API đám mây
   * Tốc độ phụ thuộc vào phần cứng máy chủ của bạn
4. **Giới Hạn Ngữ Cảnh**:
   * Cửa sổ ngữ cảnh nhỏ hơn (khoảng 8K token) so với Gemini 1.5 Pro (1M token)
   * Có thể gặp khó khăn khi xử lý tài liệu dài hoặc phức tạp