CASE CONNECT

Overview

The Case Connect project is designed to streamline legal research and decision-making by leveraging advanced technologies for data processing, semantic search, and data analysis. This platform integrates multiple components to allow users to interact with comprehensive legal case data, perform in-depth analysis, and derive insights that inform decision-making.
A key component of the project is TiDB, a distributed SQL database that provides strong consistency, high availability, and horizontal scalability. TiDB is instrumental in managing and scaling the vast amounts of legal data processed by the platform. It also enhances the application's search capabilities through TiDB Vector Search, which utilizes vector embeddings and similarity search. This enables more advanced, context-aware search functionalities, allowing users to perform deeper and more nuanced searches across the legal dataset.
The platform also incorporates OpenAI embeddings for semantic search, along with a robust frontend developed in Streamlit and a FastAPI backend, all containerized and deployed on Google Cloud Platform (GCP) for ease of deployment and scalability.

Dataset Description

The dataset used in the **Case Connect** project is sourced from the **CaseLaw Access Project**, an initiative by the Harvard Innovation Lab. This comprehensive dataset includes a vast collection of legal proceedings from across the United States, encompassing all published U.S. court decisions from the founding of the country to the present day. The dataset is meticulously curated and organized, providing detailed records of cases, including metadata such as case names, citations, court names, decision dates, and jurisdiction.

In the **Case Connect** project, the dataset is preprocessed to clean and structure the data for analysis. After preprocessing, the data is ingested into **TiDB**, a distributed SQL database that provides the necessary scalability and performance to handle large volumes of legal data. Additionally, vector embeddings are created from the text data using OpenAI's embedding models, which are also stored in TiDB. These embeddings enable advanced semantic search and similarity analysis, allowing the platform to deliver more accurate and context-aware search results.

Each case in the dataset includes detailed attributes such as:

Case ID: A unique identifier for each legal case.
Case Name: The official title of the case, often indicating the involved parties.

Decision Date: The date on which a decision was rendered.
Citation: Legal citations that reference the case in legal documents.
Category: The classification of the case, indicating its legal domain (e.g., "Criminal Case", "Medical Malpractice Case").
Opinion: The written judgment or opinion provided by the court, detailing the legal reasoning behind the decision.
Embedding: A vector representation of the case's content, enabling advanced semantic search capabilities.

In addition to the individual case details, the dataset also includes a rich set of relationships between cases, captured in a separate table. These relationships go beyond simple connections and include detailed descriptions of how cases are related, providing users with a nuanced understanding of legal precedents and their implications.

Application Use Cases

Case Connect is designed to serve a variety of use cases, including:

Legal Research: Legal professionals can use the application to quickly find relevant case law, explore how cases are connected, and understand the judicial reasoning behind decisions.
Case Analysis: Researchers can delve into the relationships between cases to uncover patterns, trends, and legal principles that influence case outcomes.
Legal Education: The application can be a valuable tool for law students and educators, providing a practical way to study legal precedents and their interconnections.
Advanced Legal Analytics: By leveraging vector search and relationship descriptions, users can perform advanced analytics, such as identifying cases with similar legal arguments or predicting case outcomes based on precedent.

The application provides a seamless interface for exploring this rich dataset, allowing users to search for cases by ID, perform semantic searches using vector embeddings, and visualize the complex web of case relationships through an interactive graph.

Key Features

Case Exploration: Retrieve and display detailed information about legal cases by entering a `case_id`.
Graph Visualization: Interactive graph that visualizes relationships between legal cases, allowing users to explore connections.
Advanced Search Functionalities:
Vector Search: Semantic search powered by TiDB's vector capabilities, enabling the discovery of similar cases based on legal arguments and opinions.
Text-to-SQL Capabilities: Users can input natural language queries, which are then translated into SQL to retrieve relevant data from the database.

Why TiDB?

TiDB Serverless was chosen for this project due to its capabilities that perfectly align with the needs of our application. Here's how TiDB was particularly useful:

1. **Seamless SQL Integration**:
   TiDB's MySQL compatibility with built-in vector search allowed us to combine traditional SQL queries with advanced semantic searches within the same system. This was crucial for efficiently querying our complex legal datasets without needing separate systems for relational data and vector searches.
2. **Built-in Vector Search**:
   With TiDB's native vector search capabilities, we were able to store embeddings and perform similarity searches directly in the database. This eliminated the need for additional vector databases and streamlined our workflow, making it easier to implement features like context-aware legal case retrieval.
3. **Scalable and Serverless**:
   TiDB's serverless architecture automatically scaled with our workload demands, ensuring that our application could handle varying loads without manual intervention. This allowed us to focus on developing features rather than managing infrastructure, which was particularly beneficial during development and testing phases.
4. **Detailed Relationship Descriptions**:
   TiDB's ability to handle complex, text-based relationship descriptions was essential for capturing the nuanced connections between legal cases. This feature enabled us to store and query detailed relationships, such as "Case A overturned by Case B" or "Case C referenced by Case D on medical malpractice," providing users with a richer understanding of how cases are interconnected. This level of detail is invaluable for legal professionals who need to grasp not just the existence of a relationship, but its specific nature and implications.
5. **GraphRAG and Knowledge Graphs**:
   TiDB supports the creation of knowledge graphs, enabling us to represent complex legal relationships and enhance our data's contextual understanding. By integrating GraphRAG (Graph Retrieval-Augmented Generation), we were able to combine traditional retrieval-augmented generation techniques with graph-based data structures, providing a deeper and more connected legal research experience.

Accessing the Application

Web Browser Access:

- Open your preferred web browser.
- Navigate to the application at: [Case Connect](http://34.135.129.25:80)

Using the Application

1. Home Page (Home.py):

- The landing page of the application serves as the central hub for users, featuring both vector search and graph search functionalities.

a. Vector Search

- Here, users can ask their legal questions directly in the vector search, and the system will provide answers along with relevant case references. Additionally, users can explore the relationships between cases using the graph search, making it easy to navigate through complex legal data and find the most pertinent information.

b. Graph Search

- Users can ask their questions in natural language and get SQL queries along with their results. This is powered by Chat2Query feature of TiDb serverless.

2. Case Explorer :

- This page allows users to explore the relationships between legal cases through an interactive graph.

- How to Use:

- Enter a `case_id` in the input field.
- Submit to visualize the main case and its related cases.

3. Case Dictionary:

- Here, we provide detailed insights into the tables, entities, and relationships that form the backbone of our application. This page is designed to help you navigate the data with ease by offering clear definitions of each column and how they contribute to the overall analysis. The **Case Dictionary** is an essential tool for anyone looking to delve deeper into the structure of legal data within **Case Connect**. By understanding these entities and relationships, users can better navigate the platform and extract meaningful insights from the legal cases

Backend API Access

1. Get Case Details and Related Cases (`main.py`):

- Endpoint: `/case_details/{case_id}`
- Method: GET
- Description: Retrieves detailed information about the specified case and related cases.
- Parameters:
  - `case_id` (int): The ID of the case.
- Response: JSON object with the case details and a list of related cases.

2. Text-to-SQL Query (`main.py`):

- Endpoint: `/query`
- Method: POST

- Description: Processes a natural language query, converts it into SQL, and retrieves the relevant data.
- Body:
  - `question` (str): The natural language query.
- Response: JSON object containing the SQL query and the results.

Deployment and Architecture

1. Docker Container:
- The application is containerized using Docker. The Dockerfile is located at the root level and defines the environment setup for running the application.
- The image is built and pushed to Google Cloud Artifact Registry and then deployed on Google Compute Engine (GCE).

2. GCE Deployment:
- The application is deployed on Google Compute Engine, which provides scalability and reliability.
- The deployment is configured for automatic restarts and high availability.

Technical Details

Backend: FastAPI
- Handles API requests, interacts with the TiDB database, and processes queries.

Frontend: Streamlit
- Provides an interactive UI for exploring legal cases and performing searches.

Database: TiDB
- Stores all legal case data and relationships, supports advanced search functionalities using vector embeddings.

Graph Visualization: `streamlit-agraph`
- Used in the graph exploration page to visualize legal case relationships.