Legal Insights Engine

Team members :

Mani Deepak Reddy Aila
Mitali Chouthai
Pranitha Poosa

Table of Contents

Introduction :

Project Overview :

The Legal Insights Engine is an innovative project designed to leverage advanced data engineering techniques and generative AI to enhance legal research, decision-making, and case management processes. By harnessing the power of knowledge graphs and similarity search, the project aims to provide valuable insights into legal cases and facilitate exploration of the relationships between various legal entities and their attributes.

Scope :

The scope of the Legal Insights Engine encompasses the development of a comprehensive platform for analysing and visualising legal proceedings data. The system will enable users to explore categories, outcomes of a case, and relationships within legal documents, facilitating informed decision-making and easier understanding. Target users include legal professionals, researchers, policymakers, and people with low or no knowledge about law seeking to gain deeper insights into legal proceedings and outcomes.

Objectives :

- To develop a scalable and user-friendly platform for analysing and understanding legal case data.
- To create a knowledge graph representing relationships between all the case attributes.
- To provide visualisations and reports for exploring legal cases distribution as per categories and courts.
- To integrate a chatbot with retrieval augmented generation (RAG) capabilities for answering questions about possible outcomes, case details.
- To automate the end-to-end workflow using Apache Airflow for enhanced efficiency and scalability.

Problem Statement :

Legal professionals often face challenges in efficiently accessing and navigating vast amounts of legal data scattered across various sources. Traditional methods of legal research and case analysis can be time-consuming and labour-intensive, leading to inefficiencies in decision-making and resource utilisation. Additionally, the complexity and interconnectedness of legal cases make it difficult for users to identify relevant information and understand the relationships or commonalities between different cases.

Solution :

The Legal Insights Engine aims to address these challenges by leveraging advanced data engineering techniques and Generative AI to provide streamlined access to legal information and insights. By integrating with currently popular graph database technologies, the project seeks to empower legal professionals with the tools and capabilities needed to efficiently explore, analyse, and interpret legal cases. Through similarity based recommendations, semantic & graph search functionalities, and a chatbot interface, the Legal Insights Engine aims to enhance the effectiveness and efficiency of legal research, decision-making, and case management processes.

Tools Used and Their Importance :

Snowflake : Snowflake is a cloud-based data warehousing platform that allows us to store and query large volumes of structured and semi-structured data. Its scalability and performance make it ideal for handling the diverse and complex datasets found in legal case data.

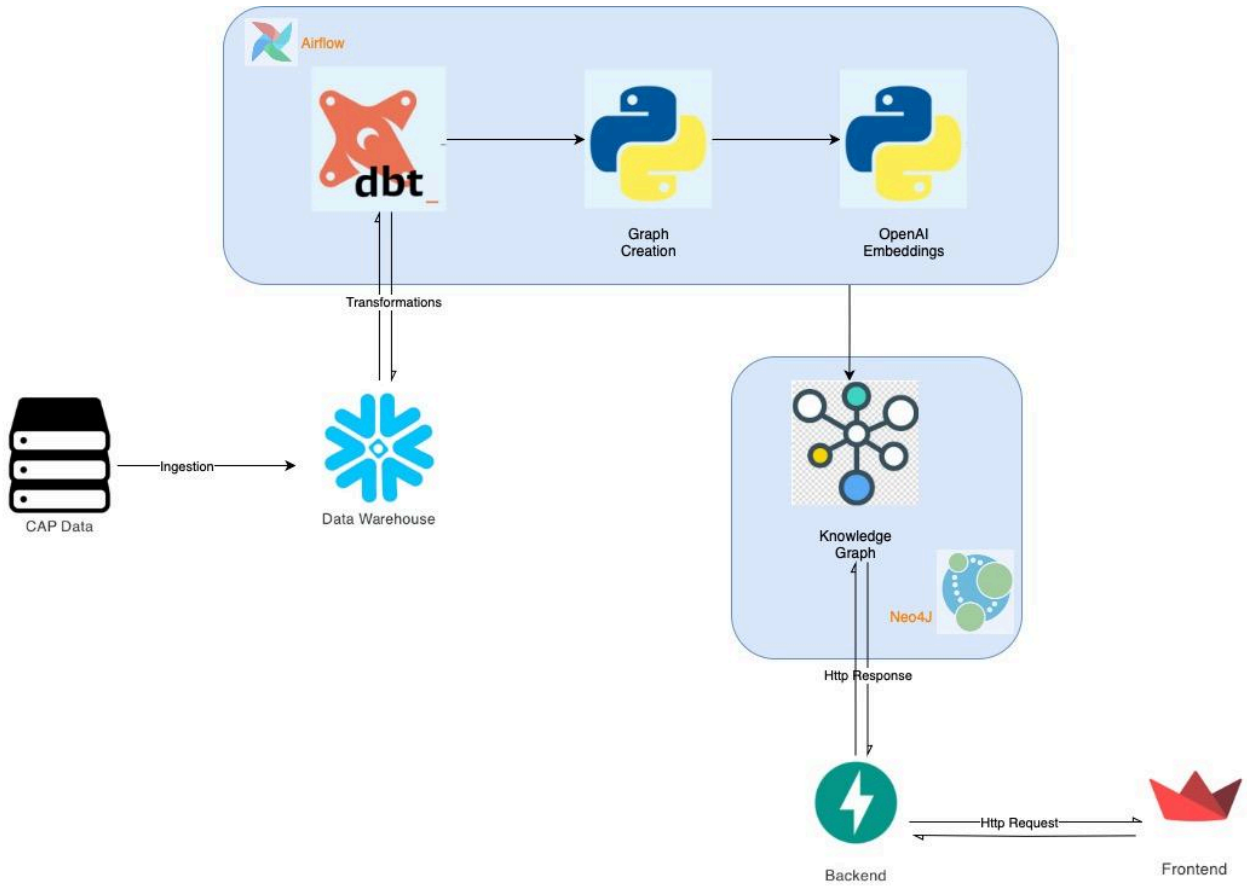
dbt (data build tool) : dbt is a transformation tool that enables us to clean, organise, and transform our data before further processing. Its SQL-based approach provides a structured and efficient way to define data transformations, ensuring the integrity and quality of our data.

Python : Python is a versatile programming language widely used for data manipulation, analysis, and visualisation. We use Python scripts for various tasks such as data processing, graph creation, and embeddings generation, leveraging its rich ecosystem of libraries and frameworks for efficient data handling.

Airflow : Airflow is an open-source platform for workflow automation and scheduling. It allows us to orchestrate and monitor data processing tasks, ensuring that our pipeline runs smoothly and efficiently. Its intuitive interface and robust scheduling capabilities make it an essential tool for managing complex data workflows.

Neo4j : Neo4j is a graph database management system that enables us to store and query the knowledge graph representing relationships between legal case elements. Its graph-based approach allows for flexible modeling of complex relationships, providing valuable insights and facilitating advanced data analysis.

Architecture Diagram



Approach :

Data Gathering :

Initially, we aimed to collect our dataset from the Oyez project, which provides audio recordings and transcripts for all Supreme Court cases. Our plan was to transcribe the audio recordings using Assembly AI and preprocess the data before applying it to a Large Language Model (LM) for question-answering, leveraging the Retrieve-and-Generate (RAG) technique. However, we encountered a significant challenge : the duration of each audio recording was typically one hour or longer, making this approach cost-prohibitive.

As a next step, we came across the Caselaw Access Project(CAP), an initiative by Harvard Innovation Library offering free, public access to over 6.5 million decisions published by state and federal courts throughout U.S. history.

Comprehensive Legal Corpus: CAP provides access to a vast collection of case law documents from various courts, including federal and state courts. That way it was easy to extract cases from one particular state - in this project, we have chosen North Carolina jurisdiction and there were about 250 cases in this state.

Structured Data: CAP data is typically available in structured formats such as XML or JSON, containing metadata and legal citations. This structured nature of the data facilitates easier extraction and organisation of information, which is crucial for building accurate knowledge graphs.

Granular Information: CAP dataset includes detailed information about cases, such as case citations, parties involved, judges, decisions, and majority opinions. This granular level of detail allows for the creation of rich and informative knowledge graphs that capture the relationships between different cases.

Cost-Effectiveness: Unlike Oyez, which primarily offers audio recordings and transcripts, CAP provides text-based case law documents, which are generally more cost-effective to process and analyse.

Data Transformation :

Data in our project is stored in Snowflake, a cloud-based data warehousing platform, and transformed using dbt (data build tool). Initially, our raw data was in JSON format, which was loaded into Snowflake as a JSON file in a single column. We used dbt to transform this data by flattening nested columns, renaming columns for clarity, and removing unwanted columns. Dbt allows us to define transformations in SQL, providing a structured and efficient way to clean and organise our data before further processing.

Synthetic data generation using LLM :

We generated synthetic data by creating a new column called "case_category" in our dataset. This column contained the category labels inferred by the Large Language Model(LLM) GPT - 3.5 for each case. By incorporating this synthetic data into our dataset, we enriched it with additional information about the categorization of legal cases.

Knowledge Graph Creation :

The knowledge graph is a key component of our project, representing relationships between different elements of the case attributes. We use Neo4j, a graph database management system, to store and query the knowledge graph. Neo4j allows us to model complex relationships between cases, categories, courts, and opinions, providing a flexible and scalable platform for exploring legal data. By leveraging the power of graph databases, we can uncover insights and patterns within our data that may not be apparent through traditional relational databases.

Data Orchestration :

Following the dbt pipeline, our data undergoes additional processing for knowledge graph creation and OpenAI embeddings generation. We developed Python scripts for these tasks, leveraging the flexibility and power of Python for data manipulation and analysis. These Python scripts are orchestrated using Airflow, an open-source platform for workflow automation and scheduling. Airflow allows us to schedule and monitor data processing tasks, ensuring that our pipeline runs smoothly and efficiently.

Features of the application :

Before we move on to the features and functionality of the application, we need to understand the following points -

The knowledge graph has the following :

Nodes and Properties :

Case - case_id, case_name, decision_date, citation

Category - category

Court - court_name

Opinion - author, opinion

And below are the relationships between the nodes :

has_opinion - between case and opinion node

is_a - between case and a category

belongs_to - between case and a court

cites_to - between case and other cases to which this case cites to

Home page -

The Home page has the information about the nodes i.e their definitions in the graph and the relationships with other nodes and the details of each instance of nodes.

Graph Exploration -

Here a case_id can be entered and a user can visually view the case with relations like a particular case cites to what other case/cases and also the normal relations like which case belongs to which category and below that there will be the majority opinion in that case.

Search -

The search feature within the Legal Insights Engine serves as a vital tool for users navigating the complexities of legal data. It provides rapid access to pertinent information, enabling users to swiftly retrieve specific case details through graph traversal and conduct in-depth analysis through advanced queries. By facilitating contextual understanding and supporting informed decision-making, the search feature empowers users to explore legal landscapes with confidence.

We have incorporated two search techniques -

Graph based search -

Text to cypher graph search is a search method of querying a graph database using natural language text as input, which is translated into cypher queries. Cypher is a query language designed for Neo4j graph databases. In this case the users can input a question and the system interprets this text to generate a cypher query. This approach simplifies the querying process.

Semantic Search -

- Vectorization of Opinion Texts: To enable semantic search, all opinion texts within the knowledge graph are first vectorized using OpenAI Embeddings. This involves representing each opinion text as a dense, high-dimensional vector, capturing its semantic meaning and context.
- Vectorization of Query: When a user submits a query, such as a natural language question, the query text is also vectorized using the same OpenAI Embeddings model. This converts the query into a vector representation, allowing for semantic similarity comparison with the vectorized opinion texts.
- Similarity Calculation: The vectorized query is then compared to the vector representations of all opinion texts in the knowledge graph using cosine similarity or other distance metrics. This calculates the similarity score between the query and each opinion text, identifying the most similar documents.
- Contextualized Querying: Once the most similar opinion text is identified, the chatbot performs contextualised querying and graph traversal.

- Graph Traversal: Through graph traversal, the chatbot navigates the relationships and connections within the knowledge graph to retrieve additional case details and relevant information. This ensures that the response is comprehensive and includes not only direct answers but also insights derived from the graph structure.

Analytics -

Unlocking the power of data analytics, the Legal Insights Engine offers a suite of analytical tools and visualisations. From plotting opinion embeddings in 2D space to exploring correlations between case categories and courts, these analytics provide valuable insights for research and decision-making. With interactive plots and intuitive visualisations, users can gain deeper insights into legal trends and patterns, driving informed decision-making processes.

Register a case -

This is a form and when submitted it adds a new case to the existing graph.

Similarity -

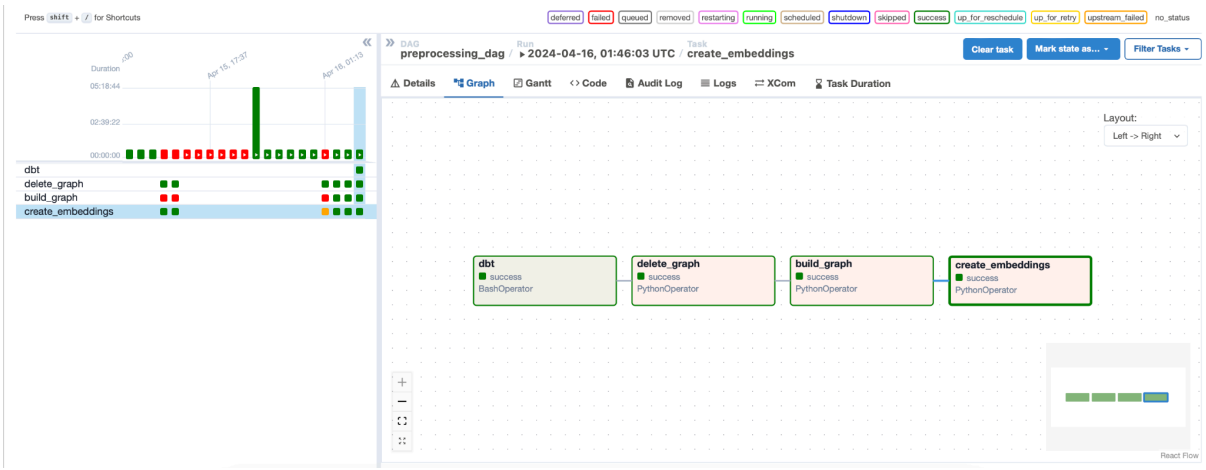
Here we have 3 factors to calculate similarity - cites_to, category and court. So we assigned weights 0.5, 0.4 and 0.1 respectively to each of the 3 and we calculate the final score as 0.5 times multiplied by similarity between the 2 cases based on cites_to + 0.4 times multiplied by similarity between 2 cases based on category + 0.1* similarity between 2 cases based on the court each case belongs to.

Now we ask the user to enter one case_id of his interest. So it finds the similarity score based on each of the other cases and returns the most similar cases. We can customise the weights of each of the 3 mentioned factors and it also gives an explanation on why these two cases are similar.

Jaccard similarity calculation is a heuristic strategy to adjust the importance of each category in measuring similarity between legal cases. It is chosen for its suitability in measuring similarity between sets, making it ideal for comparing relationships between legal cases. Its simplicity and effectiveness in capturing overlap between cases based on shared attributes make it a practical choice for analysing the connections within legal datasets.

For Legal Insights Engine, using jaccard similarity makes it simple and effective in capturing overlap between cases.

Airflow DAG -



Challenges :

- Manual Graph Creation : Creating the knowledge graph from heterogeneous legal data sources required extensive data cleaning, normalization, and schema mapping, posing challenges in ensuring consistency and accuracy.
- Airflow Orchestration : Orchestrating data pipelines with Apache Airflow presented challenges in dependency management, and error handling, and dbt local run.
- Model Interpretability: Interpreting and explaining the outputs of machine learning models, such as language models and embeddings, posed challenges in understanding the underlying decision-making processes and explaining model predictions to stakeholders.
- Working with a large dataset for implementing RAG technique : Due to extremely text heavy opinions of the cases, the data had undergone chunking which did not initially give correct answers. Knowledge graph solved this problem.
- Choosing the right LLM - Vertex AI couldn't answer the questions when tried on a subset of cases.

Learnings :

- Learned the importance of data standardization and automation in streamlining graph creation processes, enabling consistency and accuracy in the representation of legal data.
- Gained insights into workflow design and optimization through Airflow orchestration, including task concurrency, airflow local run and error handling strategies.
- Learned strategies for enhancing scalability and performance, including optimizing data processing pipelines, implementing graph databases and leveraging cloud computing resources.

Business Value -

The Legal Insights Engine delivers tangible business value by enhancing the efficiency and effectiveness of legal research and case analysis processes. By providing streamlined access to legal information, insights, and recommendations, the project enables legal professionals to make more informed decisions, mitigate risks, and optimise resource utilisation. Moreover, the project's automation capabilities reduce manual effort and improve workflow efficiency, leading to cost savings and increased productivity. Additionally, the Legal Insights Engine can help law firms and legal departments differentiate themselves in a competitive market by offering innovative and technology-driven solutions to clients.

Cross Domain Applications -

The same principles and technologies used in the Legal Insights Engine can be applied to other domains and datasets to derive similar benefits. For example, in the healthcare industry, a Healthcare Insights Engine could leverage medical records, clinical trials data, and research publications to provide healthcare professionals with personalised treatment recommendations, research insights, and decision support tools. By harnessing natural language processing, graph database technologies, and automation capabilities, the Healthcare Insights Engine could streamline medical research, improve patient outcomes, and drive innovation in healthcare delivery.

Similarly, in the finance sector, a Financial Insights Engine could analyse market data, investment portfolios, and economic indicators to provide investors, financial analysts, and decision-makers with actionable insights and recommendations for portfolio management, risk assessment, and strategic planning. By adapting the concepts and methodologies from the Legal Insights Engine, organisations across various industries can unlock the power of data-driven decision-making and drive business success.

Conclusion:

The Legal Insights Engine represents a paradigm shift in how we approach legal complexities. By harnessing the power of technology and data-driven insights, it empowers individuals, businesses, and legal professionals to navigate through the maze of legal intricacies with confidence and clarity. In an ever-evolving legal landscape, having access to reliable and comprehensive legal guidance is not just an advantage – it's a necessity. With the Legal Insights Engine by your side, you can embark on your legal journey with confidence, knowing that you have a trusted ally guiding you every step of the way. By harnessing the power of graph-based analysis, semantic search, and advanced analytics, it empowers users with actionable insights and fosters informed decision-making in legal proceedings. Whether it's exploring legal landscapes, searching for relevant case information, or analysing similarity between cases, the Legal Insights Engine serves as a trusted companion for navigating the complexities of law and unlocking hidden insights within legal data.