🔍 LLM-Powered Semantic Dataset Search Engine

💁🏻‍♀️ Wenjing Lin
🗓️ Nov 28, 2023

Berkeley
UNIVERSITY OF CALIFORNIA

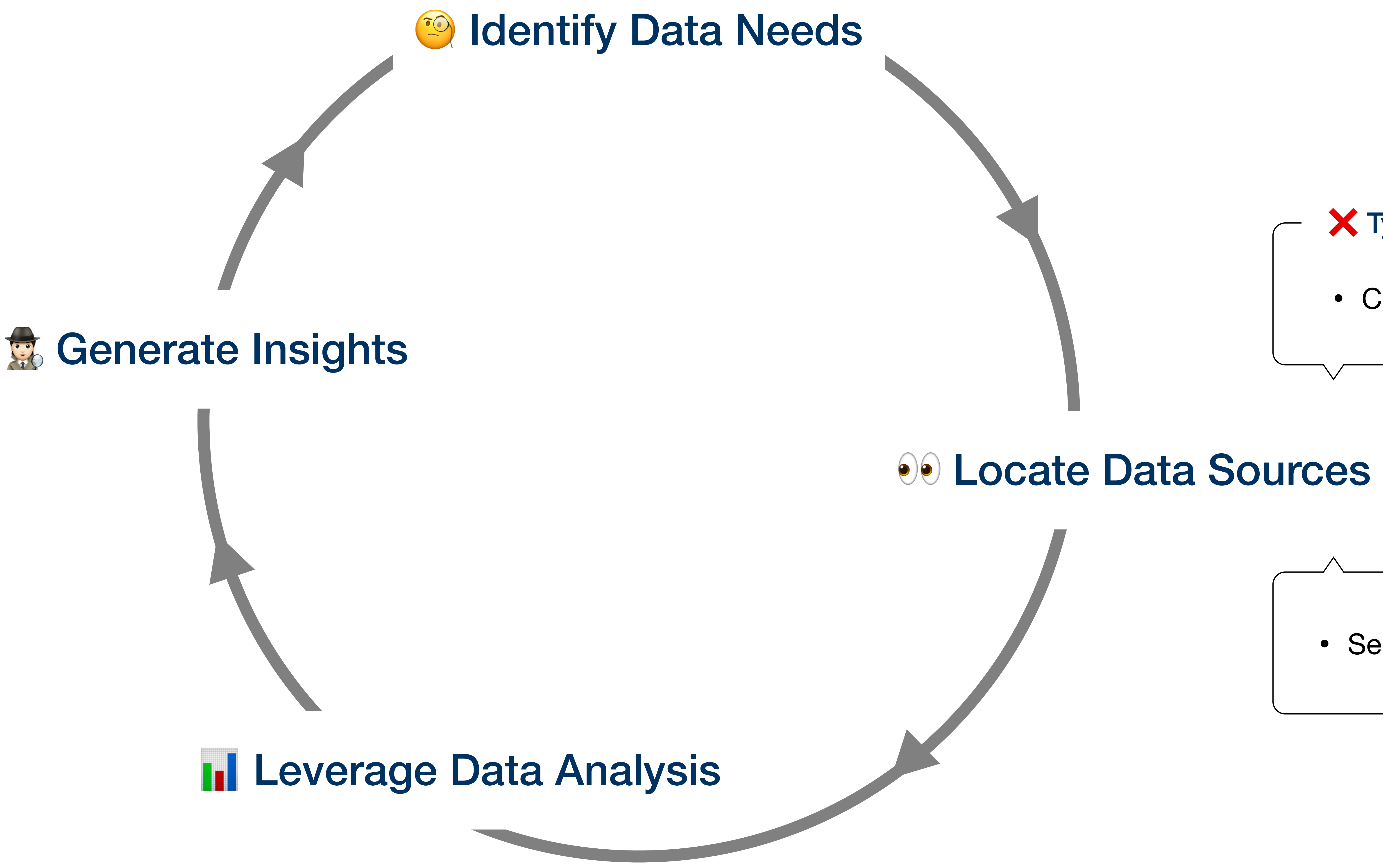
01

Introduction

01 Introduction

Problem Statement

⛔ Inefficient dataset retrieval process among DS/DAs



🧐 Identify Data Needs

👀 Locate Data Sources

📊 Leverage Data Analysis

🕵️ Generate Insights

❌ Typical approach

- Consult w/ data engineers

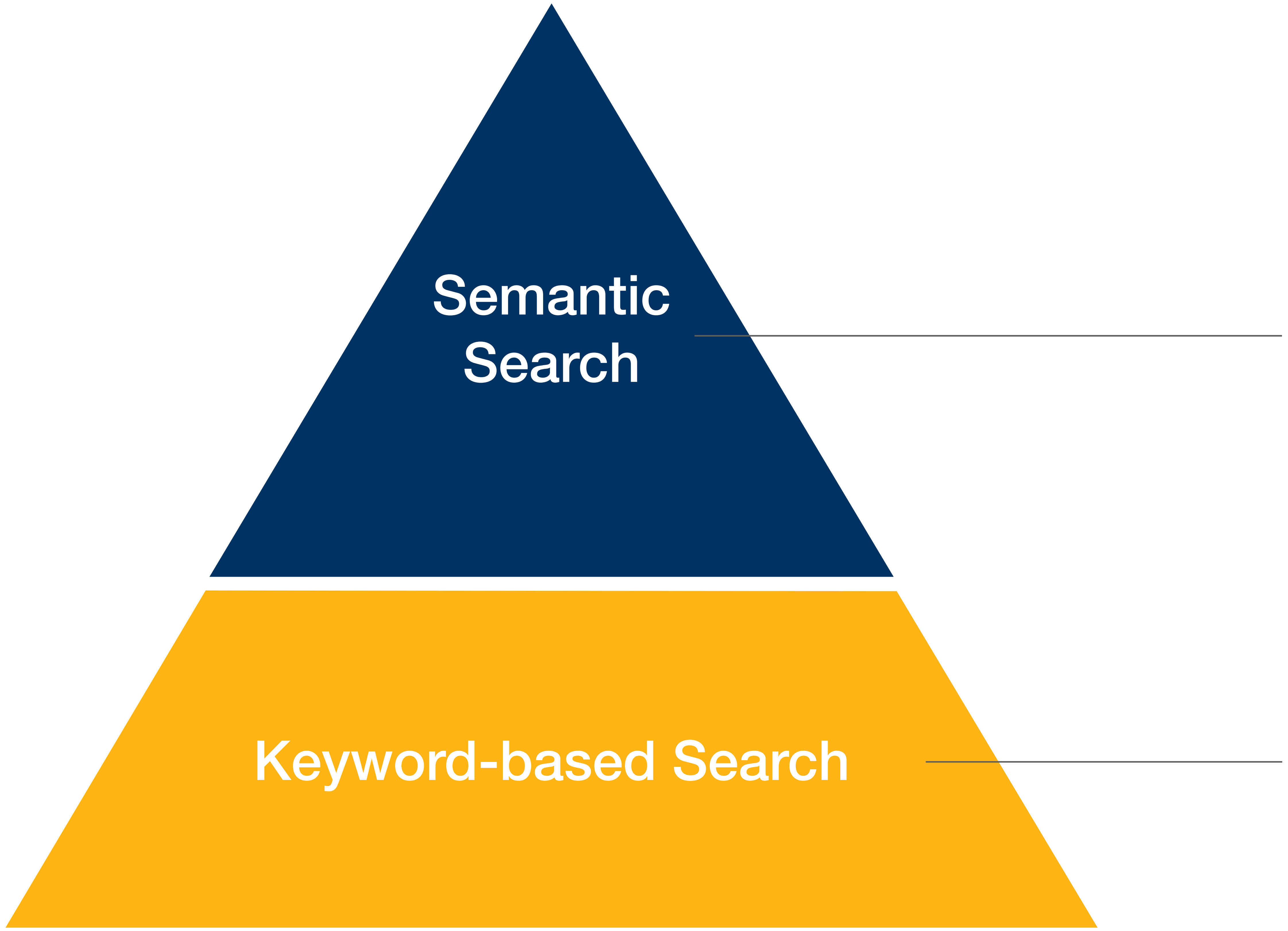
✅ Proposed approach

- Semantic dataset search engine

01 Introduction

Related Work

😾 Predominantly focus on keyword-based searches



Transcend the constraints of metadata-reliant searches

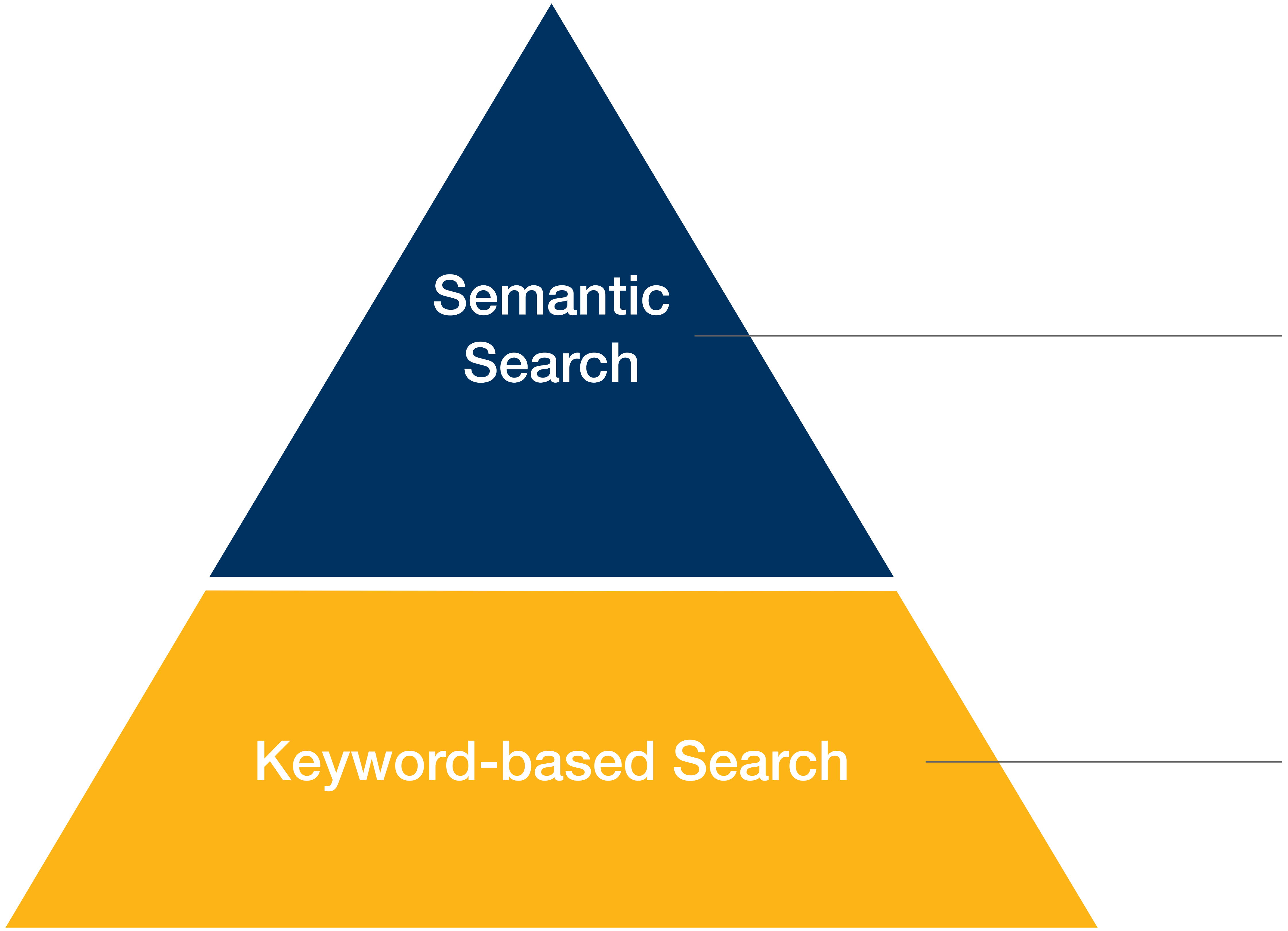
- Dataset profiling: rely on intrinsic info, e.g., statistical type annotation
- Pre-trained language models: static nature of the training data

Match metadata w/ user query

- Heavily rely on the quality and comprehensiveness of metadata
- Limited expressiveness

Proposed Solution

✨ LLM-Powered semantic dataset search engine



👍 **Information-needs-driven profiling**

- Incorporate contextual attributes beyond statistical type annotations, e.g., landing history, data retention, and clarification of ambiguous table attributes

👍 **Flexible data embedding updates**

- Adopt a Retrieval Augmented Generation (RAG) strategy, facilitating the convenient updating of embeddings in a vector store

👍 **Enhanced query expressivity**

- Enable users to employ intuitive natural language queries to articulate their information needs

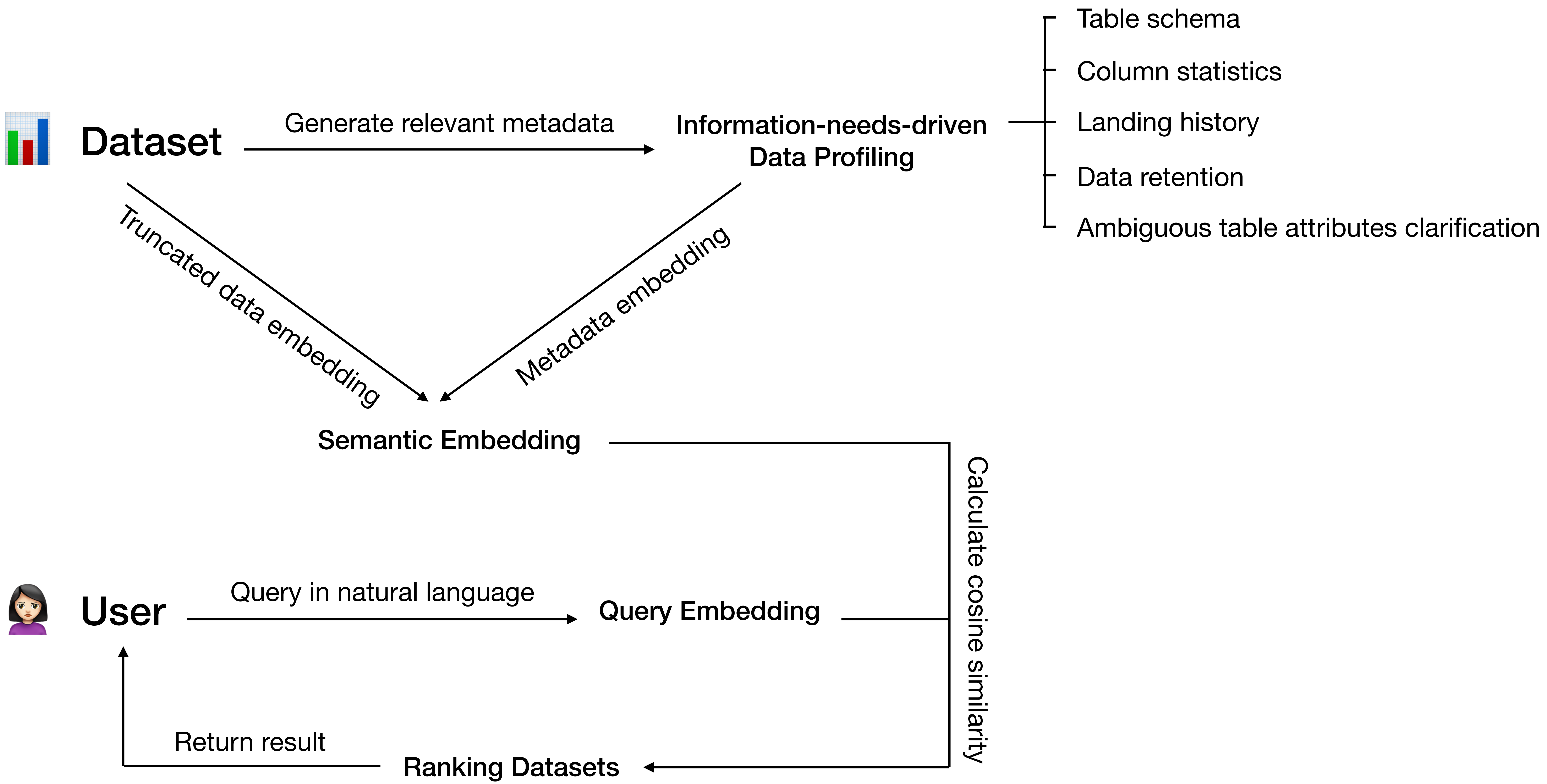
02

System Overview

02 System Overview

System Pipeline

⚙️ Propose “featurized-truncated-embedding” technique

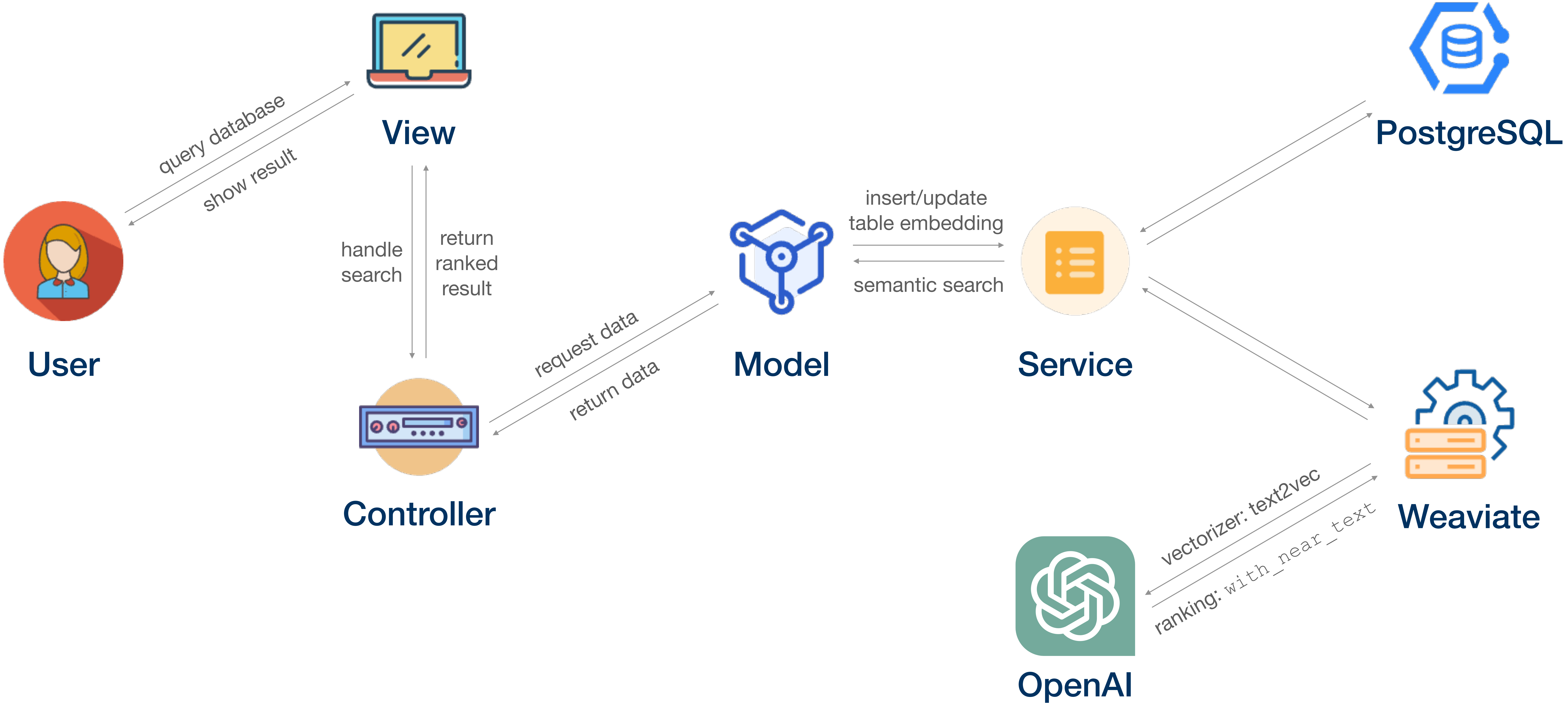


Berkeley
UNIVERSITY OF CALIFORNIA

02 System Overview

System Architecture

👩🏻‍💻 Leverage vector database along with OpenAI

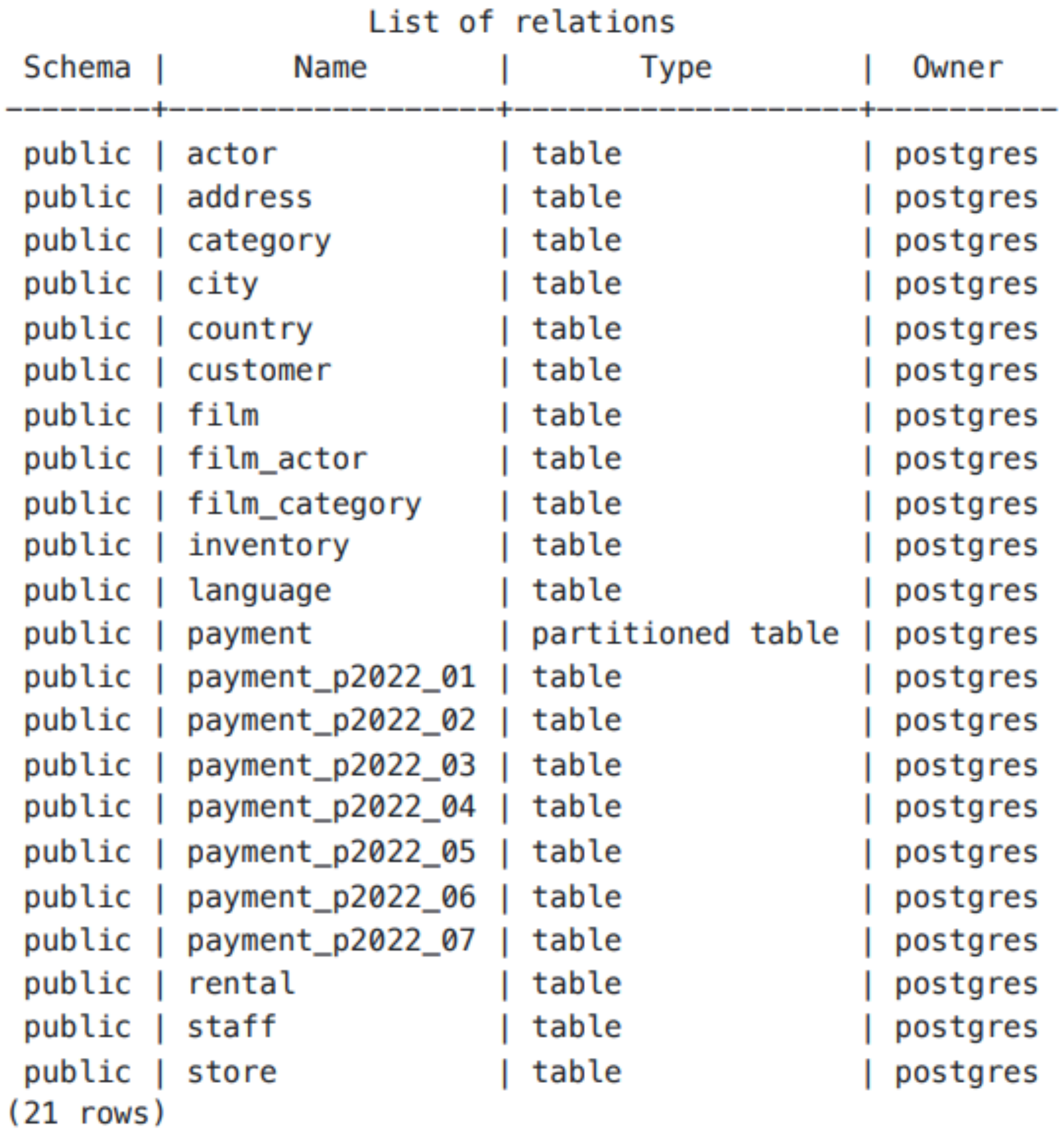


Berkeley
UNIVERSITY OF CALIFORNIA

02 System Overview

Sample Database

🗃️ Use `Pagila` as the PostgreSQL sample database



```
                       List of relations
 Schema |        Name        |        Type       |  Owner
--------+--------------------+-------------------+----------
 public | actor              | table             | postgres
 public | address            | table             | postgres
 public | category           | table             | postgres
 public | city               | table             | postgres
 public | country            | table             | postgres
 public | customer           | table             | postgres
 public | film               | table             | postgres
 public | film_actor         | table             | postgres
 public | film_category      | table             | postgres
 public | inventory          | table             | postgres
 public | language           | table             | postgres
 public | payment            | partitioned table | postgres
 public | payment_p2022_01   | table             | postgres
 public | payment_p2022_02   | table             | postgres
 public | payment_p2022_03   | table             | postgres
 public | payment_p2022_04   | table             | postgres
 public | payment_p2022_05   | table             | postgres
 public | payment_p2022_06   | table             | postgres
 public | payment_p2022_07   | table             | postgres
 public | rental             | table             | postgres
 public | staff              | table             | postgres
 public | store              | table             | postgres
(21 rows)
```

03

Use Cases

04

Future Work

05 Future Work

Improvements

🔛 Todos: detailed evaluation & system enhancement

More Comprehensive Evaluation

- **Objective evaluation**: Generate more queries to evaluate the precision
- **Subjective evaluation**: Invite industry data professionals to provide feedback for the system

Current System Enhancement

- Increase system **scalability**
- More **functionality** support: e.g., table embedding update, suggest recommended query in returned output