GRCResponder

Brianna Steier, Liam Gass, Elijah Tavares, Cael Howard,
June Kim, Rish Sharma, Angel Li

accenture

UCI Donald Bren School of Information & Computer Sciences

Table of Contents

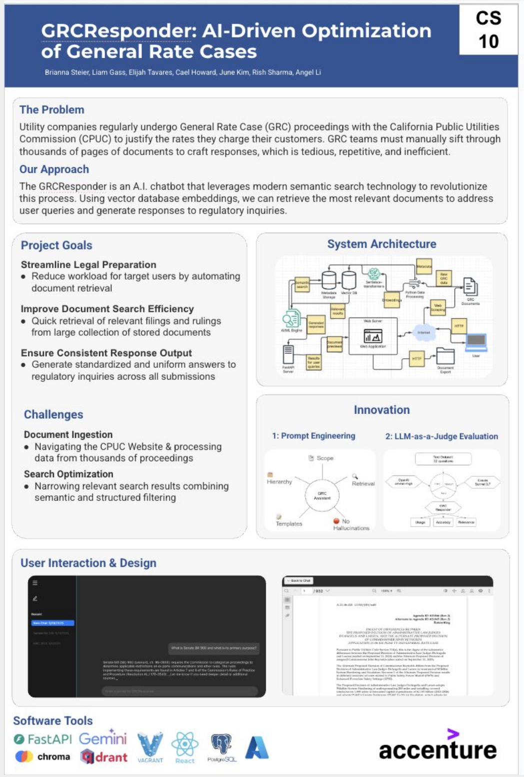
1 | CS 180 Course Updates

2 | Research Updates

3 | Implementation Updates

ICS Expo Recap

- Presented yesterday at ICS expo
- Pending assignments for the class
  - report and final demo



GRCResponder: AI-Driven Optimization of General Rate Cases

CS 10

Brianna Steier, Liam Gass, Elijah Tavares, Cael Howard, June Kim, Rish Sharma, Angel Li

The Problem

Utility companies regularly undergo General Rate Case (GRC) proceedings with the California Public Utilities Commission (CPUC) to justify the rates they charge their customers. GRC teams must manually sift through thousands of pages of documents to craft responses, which is tedious, repetitive, and inefficient.

Our Approach

The GRCResponder is an A.I. chatbot that leverages modern semantic search technology to revolutionize this process. Using vector database embeddings, we can retrieve the most relevant documents to address user queries and generate responses to regulatory inquiries.

Project Goals

Streamline Legal Preparation
- Reduce workload for target users by automating document retrieval

Improve Document Search Efficiency
- Quick retrieval of relevant filings and rulings from large collection of stored documents

Ensure Consistent Response Output
- Generate standardized and uniform answers to regulatory inquiries across all submissions

System Architecture

Challenges

Document Ingestion
- Navigating the CPUC Website & processing data from thousands of proceedings

Search Optimization
- Narrowing relevant search results combining semantic and structured filtering

Innovation

1: Prompt Engineering

2: LLM-as-a-Judge Evaluation

User Interaction & Design

What is Senate Bill 900 and what is its primary purpose?

Software Tools

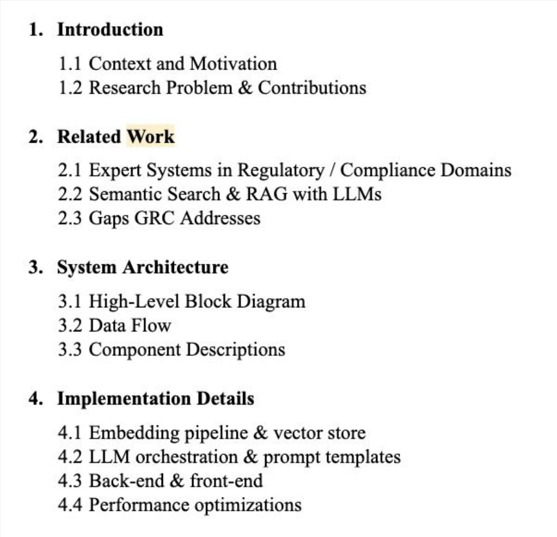
FastAPI Gemini chroma qdrant VAGRANT React PostgreSQL

accenture

Research Updates

Current Outline

Part 4, 7, and 8 done

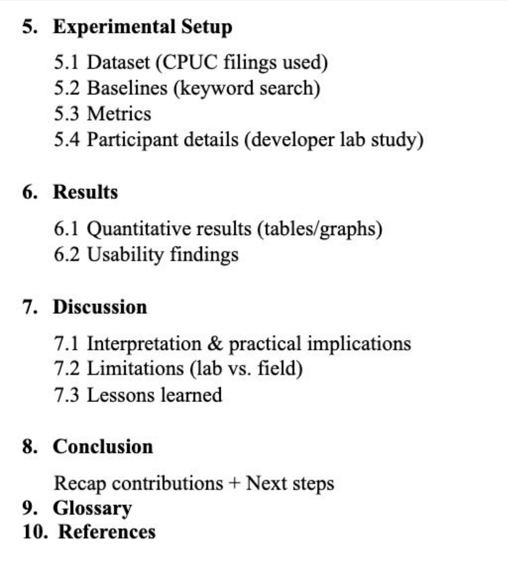


1. **Introduction**
   1.1 Context and Motivation
   1.2 Research Problem & Contributions

2. **Related Work**
   2.1 Expert Systems in Regulatory / Compliance Domains
   2.2 Semantic Search & RAG with LLMs
   2.3 Gaps GRC Addresses

3. **System Architecture**
   3.1 High-Level Block Diagram
   3.2 Data Flow
   3.3 Component Descriptions

4. **Implementation Details**
   4.1 Embedding pipeline & vector store
   4.2 LLM orchestration & prompt templates
   4.3 Back-end & front-end
   4.4 Performance optimizations



5. **Experimental Setup**
   5.1 Dataset (CPUC filings used)
   5.2 Baselines (keyword search)
   5.3 Metrics
   5.4 Participant details (developer lab study)

6. **Results**
   6.1 Quantitative results (tables/graphs)
   6.2 Usability findings

7. **Discussion**
   7.1 Interpretation & practical implications
   7.2 Limitations (lab vs. field)
   7.3 Lessons learned

8. **Conclusion**
   Recap contributions + Next steps
9. **Glossary**
10. **References**

Vector Database

Finalized Database

- Stores vectors of all documents from proceedings **filed** prior to June 25, 2025 since 2020
- Final set of documents is approximately 50 GB and 1,200 proceedings
  - Approximately 2.3 million vectors along with their text stored in the Qdrant database

Web Hosting

SSL Certificate

- unable to generate through letsencrypt due to firewall issues
- asking UCI for a new one

In the meantime…

- Switching to Docker + AWS

Questions?