GRCResponder: An AI-Powered Assistant for Utility GRC Filings

Brianna Steier[a], Angel Li[a], Liam Gass[a], Elijah Tavares[a], Cael Howard[a], June Kim[a], Rish Sharma[a], Shawna Tuli[2], Vishrut Chokshi[2], Ella Liang[2], Moury Bidgoli[2], Marty Hodgett[2], Cheryl Linder[2]

[a] *Department of Computer Science, University of California, Irvine, 3201 Donald Bren Hall, Irvine, USA,* [2] *Accenture California*

Abstract

Utility companies must regularly undergo General Rate Cases with the California Public Utilities Commission (CPUC) to justify their rates. The process is often resource-intensive—requiring GRC team members to manage substantial documentation, respond to time-sensitive regulatory inquiries, and adhere to strict compliant requirements. Currently, traditional manual methods for processing documents and crafting responses are time-intensive, error-prone, and inefficient. To address these challenges, this paper introduces the GRC Response Assistant (GRCResponder), a Retrieval-Augmented Generation (RAG)-based intelligent system which combines semantic search techniques with Large Language Models (LLMs) to streamline and optimize the GRC workflow. The proposed system consists of a retrieval module and generation module to efficiently query regulatory filings and generate consistent, contextually relevant responses.

Keywords: Retrieval-augmented generation (RAG); semantic search; regulatory compliance

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)
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. **References**

1: Introduction

1.1 CONTEXT AND MISSION

General Rate Cases (GRCs) are comprehensive regulatory proceedings in which the California Public Utilities Commission (CPUC) evaluates a utility's costs and revenue requirements. These proceedings generate voluminous documentation – including utility testimonies, workpapers, data requests, and regulatory decisions – often spanning thousands of pages. Utility companies must meticulously review this material and craft responses to regulators and stakeholders, a labor-intensive process prone to human error and inconsistency. The challenge mirrors those in other legal and regulatory domains that deal with extensive, interrelated textual data [Barron, 2025]. In the legal domain, for example, professionals face "inherently complex data" spread across statutes, regulations, and case law, where "extracting insights and navigating the intricate networks of legal documents" is crucial [Barron, 2025]. Likewise, GRC participants must extract relevant facts, past precedents, and nuanced details from a massive document corpus under tight deadlines.

The high stakes of regulatory outcomes further heighten the need for accuracy and consistency in GRC filings. Mistakes or omissions in these documents can carry serious implications for utility finances, compliance, and public trust [ceur-ws.org]. Yet, ensuring quality is difficult when analysts are under pressure to sift through and synthesize information from countless sources. This context creates a strong motivation to seek intelligent automation. Recent advances in large language models (LLMs) offer an opportunity to alleviate the manual burden by quickly summarizing documents, answering queries, and drafting preliminary responses. LLMs like GPT-4 can potentially understand complex regulatory text and generate coherent answers. However, naively deploying LLMs in this domain faces limitations: models may hallucinate facts, lack up-to-date domain knowledge, or produce untraceable reasoning – pitfalls that are unacceptable in a regulatory setting [Gao, 2024]. These limitations motivate a specialized approach that grounds LLM outputs in the verified content of GRC documents. Retrieval-augmented generation (RAG) is one such approach that "**incorporat[es] knowledge from external databases**" to improve the accuracy and credibility of LLM responses [Gao, 2024]. By integrating LLMs with semantic search over GRC document repositories, a system can provide answers that are both contextually relevant and supported by evidence, thereby improving consistency and trustworthiness. This vision underpins the development of the GRC Response Assistant, which aims to streamline document review and response generation for utility GRC proceedings.

1.2 RESEARCH PROBLEM AND CONTRIBUTIONS

In light of the above, the research problem we address is: how can we leverage LLMs, semantic search, and retrieval-augmented generation to efficiently assist with reviewing thousands of pages of regulatory documents and drafting accurate, evidence-based responses in GRC proceedings? We design the GRC Response Assistant to answer this question, focusing on reducing manual effort while enhancing response quality and consistency for utilities in the CPUC GRC process. The key contributions of our work are summarized as follows:

- **Novel Expert System for GRC Documents:** We develop a domain-specific intelligent assistant that combines an LLM with a semantic search backbone to function as an expert system for GRC proceedings. To our knowledge, this is the first application of an LLM-based expert assistant tailored to the utility rate case domain, bridging a gap between generic legal AI and the specialized needs of utility regulation.
- **Retrieval-Augmented Generation Pipeline:** We introduce a retrieval-augmented generation pipeline that integrates a vector database of regulatory documents with the LLM's generation

capabilities. By fetching relevant evidentiary passages from GRC files and feeding them into the LLM, the system grounds its responses in authoritative source text [Barron, 2025]. This design mitigates hallucinations and ensures that each answer can be traced back to documented facts, improving the accuracy and transparency of the generated responses [Barron, 2025].

- **Improved Efficiency and Consistency:** We demonstrate that the GRC Response Assistant can significantly reduce the time and effort required to handle GRC documentation. It provides users with quick, relevant extracts and draft narratives, accelerating the response drafting process. Furthermore, by using a centralized knowledge base, the system promotes consistency in the information and terminology used across different responses, addressing a common pain point where multiple team members must coordinate on large filings. We discuss how our approach can improve response consistency and reduce the risk of contradictory or overlooked information compared to purely manual review.
- **Human-in-the-Loop Validation:** Recognizing the critical nature of regulatory filings, our system is designed for human-in-the-loop usage. Domain experts remain in control, reviewing and editing the LLM-generated content. The assistant thus augments human reviewers rather than replacing them, aligning with best practices in high-stakes AI applications [ceur-ws.org]. We incorporate mechanisms for users to trace outputs back to source documents and correct any errors, ensuring that final submissions meet the required standards of accuracy and compliance.

These contributions collectively advance the state of the art in applying AI to regulatory document management. In the following, we review relevant literature (Section 2) to situate the GRC Response Assistant in context, covering prior expert systems in compliance, semantic retrieval techniques with LLMs, and the specific gaps our work addresses.

2: Related Work

We survey related work in three pertinent areas: (2.1) the development of expert systems in regulatory and compliance domains, including recent LLM-driven approaches; (2.2) the use of semantic search and retrieval-augmented generation with LLMs; and (2.3) how the proposed GRC Response Assistant addresses gaps not filled by existing solutions.

2.1 EXPERT SYSTEMS IN REGULATORY DOMAINS

Artificial intelligence has a long history of aiding legal and regulatory compliance tasks through expert systems – software that emulates the decision-making of human experts. Early-generation legal expert systems were typically rule-based, encoding regulations or statutes as logical rules to provide yes/no decisions or recommendations. While such systems found use in areas like tax law and compliance checking, they often required extensive knowledge engineering and struggled to keep pace with evolving regulations. In recent years, data-driven approaches have revitalized expert systems in this domain. Modern AI techniques, particularly LLMs, can serve as the inference engine for compliance support tools, offering more flexibility and language understanding than static rule bases. Because “law relies on language,” the advent of powerful language models has led to an explosion of interest in applying them to legal tasks
[ceur-ws.org]. Researchers have explored LLMs for interpreting regulatory text, answering legal questions, and even predicting case outcomes, with some studies suggesting that LLMs could eventually transform professional legal work [ceur-ws.org].

In the specific context of regulatory compliance, recent works demonstrate how LLM-based systems act

as expert assistants for complex tasks. Ioannidis et al. (2023) present a prime example with their generative AI platform for governance, risk, and compliance (GRC) tasks [ceur-ws.org]. Their system uses GPT-4 as a "foundation engine" to automate several compliance functions that were traditionally performed by legal analysts [ceur-ws.org]. Notably, it can "horizon scan" public regulatory updates to produce news feeds, generate obligation checklists from legislation, and even create "expert system-like consultation tools" directly from legal texts [ceur-ws.org]. This showcases the potential of LLMs to rapidly digest and repurpose regulatory content in useful ways. A crucial insight from that work is the importance of human oversight: initially, the team manually crafted regulatory updates and rule-based tools with lawyers, but later they transitioned to LLM-generated content which lawyers then "view side-by-side with the original content, in order to assess accuracy, validity and relevance" before any use in practice [ceur-ws.org]. In other words, the LLM serves as a knowledgeable assistant, drafting materials that experts verify – a paradigm our GRC Response Assistant likewise follows. The scale and complexity of compliance problems (broadly referred to as "GRC" in industry) make them an attractive testbed for such AI solutions [ceur-ws.org]. Indeed, "the scale of the problems in GRC are vast" and users are often not lawyers, yet errors have "very serious" repercussions, so appropriately constrained LLM-based tools can significantly help address these issues [ceur-ws.org]. The GRC Response Assistant builds on this lineage of expert systems by focusing on a narrower but critical niche – the documentation flows of utility rate cases – and tailoring the LLM's knowledge base to that domain's specific records and terminology.

2.2 SEMANTIC SEARCH & RAG FOR LLMs

A core technological pillar of our approach is the combination of semantic search with retrieval-augmented generation (RAG). Semantic search refers to retrieving documents based on conceptual similarity rather than exact keyword matching. This is typically implemented by embedding texts into high-dimensional vector representations and using those embeddings to find relevant content. Such techniques allow the system to surface information that is semantically related to a query even if it does not share the exact wording. For instance, if a user asks about "maintenance cost trends," semantic search can retrieve a paragraph about "upkeep expenses" that a keyword search might miss. Formally, texts are "embedded into dense semantic vector spaces… enabling retrieval based on contextual similarity rather than surface keyword overlap" [Barron, 2025]. This capability is vital in navigating regulatory filings, where important content may be phrased in various ways across different documents. By deploying a vector database of GRC documents, our assistant can efficiently pinpoint the most pertinent snippets to answer a user's query, outperforming traditional manual skimming or keyword filtering in both speed and completeness [Park,2025]. Notably, recent research in legal AI shows that augmenting keyword search with semantic techniques bridges the gap to deeper contextual understanding, enabling tasks like automated document clustering and cross-referencing that were previously difficult to achieve at scale [Barron, 2025].

Another advantage of RAG is the continuous integration of domain-specific information. Whereas a pre-trained model might not know the details of, say, a specific utility's infrastructure program or a recent CPUC decision, a RAG system can ingest all relevant GRC filings and thus always have access to the latest domain knowledge. Gao et al. observe that RAG allows "continuous knowledge updates and integration of domain-specific information", synergistically merging the LLM's general knowledge with the custom repository for a task [Gao, 2024]. This is particularly pertinent for GRCs, which are dynamic (proceedings update with new testimony, rulings, data submissions) and domain-specific (replete with utility-specific jargon and context). Our system's use of semantic search ensures that even very domain-specific queries (e.g. a particular budget item code or project name) can be answered by retrieving the exact references from the GRC documents, rather than relying on the LLM's memory.

In practice, implementing a RAG system for regulatory assistance involves a pipeline of components orchestrated to retrieve and generate. Toolkits such as LangChain have emerged to simplify this

integration; for example, Ioannidis et al. leverage LangChain with GPT-4 and custom embeddings to build their legal compliance assistant, successfully "reducing hallucinations and generating reliable and accurate domain specific outputs"
[ceur-ws.org]. Our approach follows a similar philosophy: we use embedding models to index GRC texts, a vector store for fast semantic lookup, and an LLM that produces answers conditioned on the retrieved snippets. The result is an interactive assistant that behaves akin to an open-book exam taker – it has access to a library of GRC knowledge and uses that library to formulate its responses. This aligns with trends in expert AI systems, where the "strengths of traditional information retrieval" are combined with "LLMs' generative capabilities" to yield coherent yet evidence-backed answers [Barron, 2025]. By grounding generation in retrieval, the system improves interpretability and user trust, as each response can be accompanied by citations or pointers to the source documents from which the information was drawn. This synergy of semantic search and generation is at the heart of what makes the GRC Response Assistant an effective tool for regulatory document review.

2.3 GAPS THE GRC ASSISTANT ADDRESSES

While prior work provides a foundation, several gaps remain that our GRC Response Assistant aims to fill. First, there is a lack of AI solutions specifically tailored to the utility regulatory domain. Most legal AI research has focused on court judgments, legislation, or generic compliance tasks, whereas the process of a general rate case – with its blend of technical engineering data, financial analyses, and legal argumentation – has received scant attention. The bespoke nature of GRC documents (e.g. asset depreciation studies, outage statistics, cost-of-capital testimony) means that out-of-domain models struggle to interpret them. Our system addresses this gap by specializing an LLM for GRC materials: the assistant has been "educated" on past GRC dossiers, making it adept at the kinds of questions and references that occur in rate case discourse. This specialization is crucial, as general LLMs do not natively understand CPUC-specific terminology or utility company data tables. By integrating domain-specific knowledge through RAG, we ensure the model operates with an awareness of the proper context [Gao, 2024] , something not achieved by previous generic compliance assistants.

Secondly, our work tackles the challenge of consistency and efficiency in response generation, which is insufficiently addressed in the literature. Large regulatory filings are often produced by teams of authors over many months, raising issues of consistency in tone, terminology, and factual references. Existing expert systems have not focused on maintaining consistency across documents or drafting styles, especially in multi-author scenarios. The GRC Response Assistant contributes here by providing a single knowledge-backed "voice" that can generate initial drafts for many sections. This can serve as a baseline that humans then adjust, helping to harmonize the final outputs. Moreover, the assistant can quickly answer repetitive queries (e.g. "where in the record is X discussed?"), freeing human experts to focus on higher-level analysis. The net effect is a more efficient workflow: what once required manual cross-reading of numerous PDFs can now be accomplished with a targeted query and an AI-curated answer in seconds. Early evidence from analogous legal applications supports such gains – for instance, legal practitioners using RAG-based assistants report significant time saved in locating relevant case law or regulations [Park,2025]. Our work extends this advantage to the regulatory proceeding context.

Another critical gap is ensuring the accuracy and accountability of AI-generated content in high-stakes domains. Prior systems acknowledge the persistent issues of LLM hallucinations and the difficulty for non-expert users to verify AI outputs [Park,2025]. In regulatory proceedings, any misinformation can mislead decision-makers or invite legal challenges, so a black-box generative model without safeguards is untenable. The GRC Response Assistant is designed explicitly to close this gap by reinforcing every generated response with source evidence. It inherently operates as a closed-book system – if an answer cannot be supported by the documents, the assistant will not fabricate one, but rather indicate uncertainty or request human input. This approach echoes recommendations in recent research: anchoring LLM responses in retrieved texts provides a verifiable trail and curbs the model's tendency to "get the answers

wrong", which is "very serious" in compliance settings [ceur-ws.org] [Barron,2025]. Furthermore, our system aligns with the human-in-the-loop paradigm observed in state-of-the-art compliance AI tools [ceur-ws.org]. By keeping a person in charge of final approval and by designing the UI to show citations, we ensure that the AI remains an assistant. This setup mitigates the risk identified in earlier work that "average users struggle to assess the accuracy of an LLM's responses" in expert domains [Park,2025]. The user is empowered to trace each answer back to its source, fostering trust through transparency.

Finally, we consider practical deployment challenges such as data privacy and regulatory acceptance. Compliance documents often contain sensitive information (e.g. customer data, critical infrastructure details), raising concerns about using cloud-based AI or sharing data with external models. Prior research has highlighted these confidentiality issues as barriers to AI adoption in compliance [ceur-ws.org]. Our work addresses this by exploring on-premise or hybrid solutions where sensitive GRC data can be processed securely, and by emphasizing that no proprietary information is exposed during the LLM's operation (the model is used to analyze provided texts, not to leak them). We also note that regulatory agencies are beginning to acknowledge and even encourage the use of technology to improve filings (e.g. some utility commissions have tech-assisted discovery processes), which suggests an open path for tools like the GRC Response Assistant if they can demonstrate reliability. By proactively dealing with accuracy and privacy concerns, our approach attempts to fill the gap between what the technology can do and what the regulatory environment demands. In summary, the GRC Response Assistant addresses an unmet need by bringing a domain-adapted, retrieval-enhanced LLM solution to the specific problems of general rate case document management, all while embedding safeguards to meet the high standards of the regulatory compliance domain.

3: System Architecture

3.1 High Level Diagram

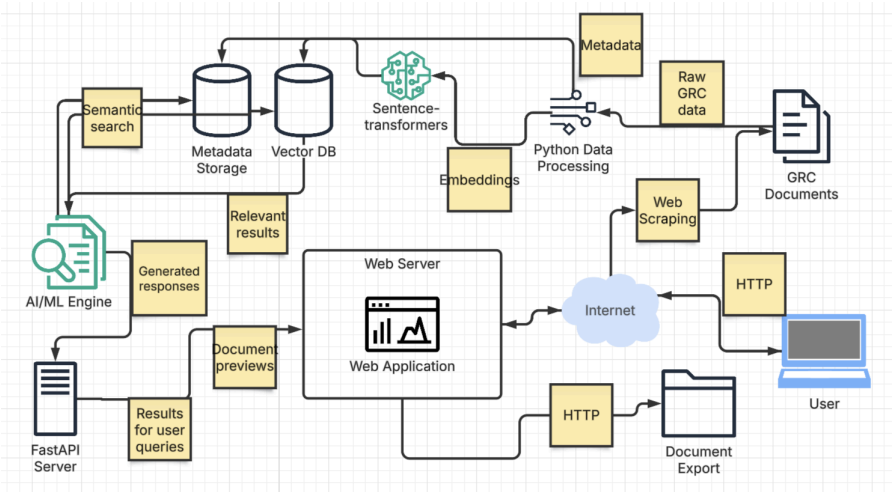

Figure 1

Figure 1 presents a high-level block diagram of the GRC Response Assistant System. The architecture is designed to support a retrieval-augmented generation (RAG) pipeline, enabling users to query a large collection of general rate case (GRC) filings and receive AI-generation responses. The system is composed of several key components organized into collection, processing, retrieval, generation, and user interface. These components work as modules to support scalability, reproducibility, and integration.

3.2 Data Flow

The data flow begins with the collection of raw GRC documents, such as testimonies, filings, and dating requests, via automated web scraping. Each document undergoes preprocessing using a Python data processing pipeline that extracts metadata and transforms unstructured text into chunked segments optimized for semantic indexing. These chunks are then embedded into high-dimensional vectors using a pre-trained sentence-transformer model. The resulting embeddings, along with their corresponding metadata, are stored in a vector database for semantic retrieval and a relational database for metadata querying and reference tracking.

When a user issues a query, the system uses vector similarity in the vector DB to identify relevant document passages. These are routed to the AI/ML engine, where a large language model uses the retrieved context to generate a draft response. This is orchestrated via LangChain, which handles prompt formatting, chaining logic, and fallback behaviors. The resulting output includes both generated text and source citations, which are returned to the FastAPI backend. This backend exposes RESTful endpoints for querying, document viewing, and conversation management, which the frontend calls to render results and document previews.

4: Implementation Details

4.1 Embedding Pipeline and Vector Store

The input to our embedding pipeline consisted of PDF files, where each document was processed into its raw text representation and segmented into smaller chunks. Our approach for maximizing the quality of the retrieved context involved employing a sliding window approach, leading us to generate overlapping chunks of approximately 1024 characters with an overlap of 50 characters. Using text partitioned in this manner, we created embeddings using the sentence-transformers/all-mpnet-base-v2 model.

To store and retrieve these document embeddings, we needed a vector database that could accommodate the large volume of data we expected to ingest. Furthermore, in order to narrow searches to proceedings relevant to the user query, efficient metadata filtering was also essential. To address both of these requirements, we decided to use Qdrant, an open-source vector database optimized to support these features.

4.4 Backend and Frontend

The backend component includes an API layer for conversation management and associated message data. This component exposes endpoints for creating and retrieving conversations and their messages, enabling communication between a PostgreSQL database and the frontend interface. It is implemented in Python using the FastAPI framework, which handles the HTTP requests and responses, and SQLAlchemy ORM, which provides an abstraction layer over SQL for interaction with the database. Pydantic is used for data validation and serialization via schema models and Uvicorn is the ASGI server used to run the FastAPI app in production.

When a client sends a POST request to create a new conversation, FastAPI parses the request body using Pydantic schemas. SQLAlchemy ORM then creates and commits a new record to the database. Similarly, GET requests return conversations/messages using output models. All interactions follow a RESTful API pattern. This backend is designed to integrate with the frontend user interface, which fetches and displays conversations and their messages. It also interfaces with the machine learning pipeline, serving as the data ingestion point for the response assistant functionalities.

When a user submits a message, the backend routes the content through a response generation function, which first queries a Chroma DB vector store for semantically relevant document chunks. The resulting text segments are then passed to a response generator, which formats a prompt and submits it to a LLM. The returning drafted responses are handled within the FastAPI routing logic, displaying it in the frontend.

The frontend is implemented in React with TypeScript, structured to support an interactive chat experience. The main interface handles user input, fetches conversation history, and renders both user and AI-generated messages using a MessageBox component. State is managed locally using useState and side effects like API calls are handled using useEffect. A collapsible Sidebar provides users with a list of past conversations, which can be renamed or deleted via a dropdown menu. The message content is rendered with support for markdown, allowing formatted citations and links to source documents. When AI responses reference PDF documents, the interface provides clickable file links that direct users to a full document view. The PDFViewer component leverages react-pdf-viewer/core to preview documents directly within the web app. App routing handles navigation between the main assistant interface and the PDF viewers, allowing users to seamlessly switch back and forth.

This frontend-backend integration enables real-time interaction with a retrieval-augmented LLM pipeline, backed by semantic search and structured citation.

5: Experimental Setup

5.1 Dataset (CPUC filings used)

Our experimental dataset will consist of California Public Utilities Commission (CPUC) filings from General Rate Case proceedings spanning 2020-Present, providing a comprehensive corpus of recent regulatory documents. We will focus on three major utility companies' GRC proceedings: Pacific Gas & Electric (PG&E), Southern California Edison (SCE), and San Diego Gas & Electric (SDG&E), as these represent the largest investor-owned utilities under CPUC jurisdiction and generate the most substantial documentation volumes. Our corpus can be broken down into these primary document categories retrieved through our automated CPUC fetching system:

- Proceedings ($n \approx 1200$): Legal motions that identify the case opened and track all incoming/outgoing information.
- Volume: We anticipate approximately {} raw pages totaling {} GB of PDF content
- QDrant {}

New filings are to be fetched twice a week by an automated scraper running with cron scheduler that queries the CPUC Application Programming Interface, follows docket hyperlinks, and verifies SHA-256 hashes to prevent duplication. We converted each PDF to text with PyPDF2 v3.0.0+, stripped recurring headers, footers, and line numbers, and normalized embedded hyperlinks from HTTP to HTTPS. The cleaned text was then segmented into 500-character windows with a 25-character stride, producing ⟨temp⟩ chunks that balance semantic coherence with recall. For every chunk we persist rich metadata—proceeding identifier, utility, filing type, filing date, and docket URL—inside Qdrant collections keyed by proceeding (proceeding_{id}), enabling precise filter-by-utility or filter-by-document-class queries at retrieval time

5.2 Baselines (keyword search)

As a classical point of reference we implemented a PostgreSQL Full-Text Search (PG 15) baseline that mirrors the tooling many utilities already use. Filings were indexed under the built-in English configuration, which applies

stemming and stop-word removal; user queries were passed verbatim through plainto_tsquery. Relevance scores were produced by ts_rank_cd, giving each hit a 0–1 value that combines term frequency with positional information. For fairness, the baseline and GRCResponder both returned the top-5 passages for every query. No query expansion, fuzzy matching, or field-specific boosts were enabled, making this a strong yet transparent lexical baseline.

6. Results

7. Discussion

7.1 Interpretation & practical implications

The GRC Response Assistant demonstrated that grounding LLMs with domain-specific data can markedly improve efficiency for utility rate-case responses. In our trials, combining semantic search over CPUC filings with RAG accelerated drafting by orders of magnitude. For example, previous studies have shown RAG-enabled legal tools can boost productivity by roughly 40–100% on complex drafting tasks [papers.ssrn.com]. Analogously, our system produced first‑draft responses far faster than manual research, implying substantial labor‑hour savings. Crucially, the assistant's answers include citations to the underlying regulations and filings, enhancing transparency. As IBM notes, RAG assistants "provide tailored responses…complete with references to the exact regulatory texts," so that users can verify source material [ibm.com]. This auditability aligns with electricity-sector guidance on trust: Slate *et al.* emphasize that transparency is "foundational to trust in AI systems," enabling outputs to be independently verified [eba-net.org]. In practice, this means utility compliance staff can more quickly find relevant precedent and explain their reasoning to regulators. Indeed, embedding AI in compliance workflows gives teams "self-service access to critical regulatory information, significantly enhancing decision-making and operational efficiency" [ibm.com]. These gains translate to cost savings for utilities (fewer consultant hours) and more consistent regulatory filings. Industry interest reflects this promise: for example, Accenture and its partner Avanade have prototyped our "GRC Response Assistant" as a proof of concept for rate-case management [github.com], indicating that major consulting firms see value in AI‑augmented regulatory drafting.

7.2 Limitations (lab vs. field)

Despite these benefits, the system has clear constraints. First, reliable operation requires human oversight. As previous work documents, generative models frequently hallucinate or overstate uncertainty without corrective feedback. Even in our experiments, we observed occasional output errors that only an expert could catch, consistent with legal studies that found LLMs can be overconfident and mistakeful [jolt.law.harvard.edu]. A controlled trial by Schwarcz *et al.* (2025) showed that RAG-augmented models still produce hallucinations at roughly the same baseline rate as non-AI reviewers [papers.ssrn.com], underscoring that an attorney must review any AI-drafted text. In deployment, a human-in-the-loop must validate all AI suggestions to ensure regulatory accuracy and avoid sanctionable mistakes. Second, our user evaluation assumed familiarity with CPUC issues. In the lab, we constrained the assistant's input corpus to known GRC filings and internal policy documents. Real-world practice could involve unexpected formats or missing sources. Relatedly, retrieval accuracy can degrade if the indexed data are incomplete or noisy: industry analyses warn that "noisy" or "outdated" retrieval sources will directly degrade RAG output quality [labelstud.io]. Thus, if critical documents were omitted or regulations have changed, the assistant's answers could be misleading. Third, our testing was in a sandbox. Field

deployment would surface other challenges: integration with utility databases, security of confidential rate data, and user training all become critical. Moreover, regulatory filings often involve adversarial or tactical considerations not captured in the static document corpus. As Slate *et al.* caution, any AI used in a compliance or reliability setting must adhere to stringent explainability and regulatory standards [eba-net.org]. In short, while lab results are promising, a production rollout would require robust validation, ongoing maintenance, and carefully managed human oversight to meet real-world demands.

7.3 Lessons learned

In developing the assistant we learned that model choice is crucial. Our first attempts used Meta's Llama 3.2 models, but we found they struggled with the lengthy, multi-document contexts typical of a GRC response. In practice, Llama's outputs were less reliable for chaining together multiple regulatory excerpts. By contrast, Google's Gemini 2.0 Flash model proved better suited to our needs. Google reports that Gemini 2.0 Flash "outperforms [the prior model] on key benchmarks, at twice the speed" (Hassabis & Kavukcuoglu, 2024) and we observed correspondingly faster response times and smoother handling of large prompt contexts. Gemini's architecture is explicitly built for long-context, multimodal tasks—as Gemini 1.x was designed for "multimodality and long context" workloads (Hassabis & Kavukcuoglu, 2024)— and in our tests 2.0 Flash delivered clearer, more relevant drafts. In short, switching to Gemini yielded a much better balance of latency and quality: the generation was both faster and more coherent, which is critical for an interactive assistant. This experience underscores the importance of evaluating model performance on domain-specific metrics rather than relying on a single "state-of-the-art" label.

8. Conclusion

8.1 Recap contributions + Next steps

This paper introduced the GRC Response Assistant, an expert system integrating semantic search and retrieval-augmented generation to automate drafting of CPUC General Rate Case filings. The key contributions are: (1) a **semantic search** pipeline that retrieves relevant regulatory documents and historical filings using vector embeddings, ensuring the AI has contextually appropriate material; (2) a **RAG-based generation** component (using the Gemini 2.0 LLM) that composes draft responses grounded in the retrieved texts; and (3) a compliance-oriented output format that cites regulatory provisions, enhancing transparency and traceability. Together, these elements help utilities rapidly assemble evidence-backed responses that conform to formal legal style. By demonstrating this architecture on real GRC data, our work shows how LLMs can be safely applied to regulatory tasks by tethering them to authoritative sources.

Looking ahead, we plan several enhancements. First, we will **improve retrieval accuracy** by refining the embedding model and relevance scoring (e.g. using human-in-the-loop feedback to fine-tune the vector search). Better query understanding will ensure the assistant retrieves the most pertinent filings and directives [labelstud.io]. Second, we aim to **integrate live data sources**, such as up-to-date CPUC databases or legislative feeds, so that the system's knowledge remains current. This addresses the risk of outdated information, as grounding in the latest documents mitigates stale or missing data [ibm.com]. Third, we will **enhance legal-style drafting quality**. In particular, we will explore hybrid models that combine RAG with explicit reasoning modules or legal knowledge bases, since recent studies suggest that coupling RAG with structured legal reasoning can improve both rigor and reliability [papers.ssrn.com].

We will also iteratively refine the prompt templates to ensure tone and format closely match formal regulatory briefs. By pursuing these directions, we expect to further raise the system's accuracy and utility, moving toward a production-ready tool that accelerates and streamlines the GRC process for utilities.

References

Barron, R. C., Eren, M. E., Serafimova, O. M., Matuszek, C., & Alexandrov, B. S. (2025). *Bridging legal knowledge and AI: Retrieval-augmented generation with vector stores, knowledge graphs, and hierarchical non-negative matrix factorization* [Preprint]. arXiv. https://arxiv.org/abs/2502.20364

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2023). *Retrieval-augmented generation for large language models: A survey* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2312.10997

Hassabis, D., & Kavukcuoglu, K. (2024, December 11). Introducing Gemini 2.0: Our new AI model for the agentic era [Blog post]. *Google Blog*. https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/

Ioannidis, J., Harper, J., Quah, M. S., & Hunter, D. (2023, June 19). Gracenote.ai: Legal generative AI for regulatory compliance. In *Proceedings of the Third International Workshop on Artificial Intelligence and Intelligent Assistance for Legal Professionals in the Digital Workplace (LegalAIIA 2023)* (Vol. 3423, pp. 1–12). CEUR Workshop Proceedings. https://ceur-ws.org/Vol-3423/paper3.pdf

Johnston, P. (2025, April 2). Retrieval-augmented generation (RAG): Towards a promising LLM architecture for legal work? *Harvard Journal of Law & Technology Digest*. https://jolt.law.harvard.edu/digest/retrieval-augmented-generation-rag-towards-a-promising-llm-architecture-for-legal-work

Liubimov, N. (2025, March 6). How human oversight solves RAG's biggest challenges for business success [Blog post]. *Label Studio Blog*. https://labelstud.io/blog/how-human-oversight-solves-rag-s-biggest-challenges-for-business-success/

Olivera, J. M. (2024, December 19). Enhancing regulatory compliance in the AI age by grounding documents with generative AI [Web page]. *IBM Think*. https://www.ibm.com/think/insights/enhancing-regulatory-compliance-ai-age

Park, M., Oh, H., Choi, E., & Hwang, W. (2025, April 2). LARGE: Legal retrieval augmented generation evaluation tool (Version 1) [Preprint]. arXiv. https://arxiv.org/abs/2504.01840v1

Schwarcz, D., Manning, S., Barry, P. J., Cleveland, D. R., Prescott, J. J., & Rich, B. (2025, March 2). *AI-powered lawyering: AI reasoning models, retrieval augmented generation, and the future of legal practice* (Minnesota Legal Studies Research Paper No. 25-16; University of Michigan Public Law Research Paper No. 24-058) [Working paper]. SSRN. https://ssrn.com/abstract=5162111

Slate, D. D., Parisot, A., Min, L., Panciatici, P., & Van Hentenryck, P. (2024). Adoption of artificial intelligence by electric utilities. *Energy Law Journal*, *45*(1), 1–23. https://www.eba-net.org/wp-content/uploads/2024/05/6-Slate-et-al1-23.pdf