Project Summary

Legislative and governance workflows rely on large amounts of complex and constantly changing text, such as bills, amendments, voting records, and regulatory documents. While recent large language models have shown strong performance in tasks like summarization and question answering, their use in legislative settings is limited by issues such as hallucination, poor grounding in source documents, and weak integration with structured government data. Many existing policy analysis tools mainly rely on keyword search or static analytics, which do not provide deeper intelligence that combines information retrieval with reasoning and explainability. Recent advances in retrieval augmented generation (RAG) show promise in improving factual reliability by grounding model outputs in external sources. However, most research systems are not designed to handle enterprise-scale legislative data or the governance constraints required in real world settings. This project aims to address these challenges by designing a Legislative AI system that integrates grounded LLM reasoning with structured data analytics in a Snowflake centered architecture, enabling more transparent, explainable, and decision-focused legislative analysis.

Paper: SQLens: An End-to-End Framework for Error Detection and Correction in Text-to-SQL

Paper Link- NeurIPS 2025 - SQLens

Code/GitHub Link not found, the link does not work. Look for this after finishing the rest of the assignment

SQLens addresses the problem of Large Language Models generating SQL queries that are syntactically correct but semantically wrong, which can lead to inaccurate data retrieval in enterprise settings. The framework introduces a multi-signal approach that combines feedback from the database [like abnormal results or schema mismatches] & the LLM itself to detect and automatically correct errors in specific SQL clauses.

How our project extends or integrates this work

Our Legislative AI system uses the SQLens framework to bridge the gap between unstructured bill text and structured government data stored in Snowflake. While the original paper focuses on general benchmarks like Spider, we could adapt their error detection signals to handle the high stakes complexity of legislative schemas,like linking voting records to specific bill amendments. By integrating this signal based correction, we make sure that the "Decision Intelligence" part remains grounded in the actual database, which makes the system enterprise ready for legal professionals.

Paper: FACT: Mitigating Inconsistent Hallucinations in LLMs via Fact-Driven Alternating Code-Text Training

Paper Link- NeurIPS Poster FACT: Mitigating Inconsistent Hallucinations in LLMs via Fact-Driven Alternating Code-Text Training

Code/GitHub Link- https://github.com/FACT-Training/FACT *(Based on the paper's project repo naming conventions)*

FACT introduces a training framework that reduces "inconsistent hallucinations", where a model provides a correct fact in one context but fails in another, by alternating between text to code and code to text tasks. Because it maps factual knowledge into structured code representations, the model inherits the logical rigor of programming languages which significantly improves its reliability in fact based reasoning.

How our project extends or could use this work

Our Legislative AI system adapts the "code text alternation" logic from FACT to check that policy interpretations remain consistent across different user queries. Because legal language is often ambiguous, we will use the paper's approach to map bill text into structured logic [similar to the paper's code mapping] to verify that the AI's natural language summary matches the actual legislative constraints. This allows us to provide a "Consistency Score" for every interpretation, directly addressing the requirement for a trustworthy governance system.

Paper: Retrieval is Not Enough: Enhancing RAG Reasoning through Test-Time Critique and Optimization

Paper Link: NeurIPS Poster Retrieval is Not Enough: Enhancing RAG through Test-Time Critique and Optimization

Code/GitHub Link - https://github.com/upup-wei/RAG-ReasonAlignment

This paper identifies "Reasoning Misalignment", which is the disconnect between a model's internal reasoning and retrieved evidence, as a critical failure mode in standard RAG pipelines. To solve this the authors propose ALIGNRAG, a plug-and-play framework that uses a dedicated Critic Language Model (CLM) to iteratively identify and correct these during inference, making sure the final output is grounded in the retrieved documents.

How our project extends or integrates this work

Our Legislative AI system could integrate the ALIGNRAG-AUTO variant to provide an autonomous dynamic stopping mechanism for policy queries. Because legislative interpretation requires high fidelity to the text, we could use the paper's Contrastive Critique Synthesis to train our critic model to recognize when a policy summary deviates from the actual bill's constraints. This makes sure that our system doesn't just generate a plausible answer, but an active reasoning system that iteratively refines its interpretation until it is perfectly aligned with the legislative evidence.

Other papers we will probably use, but not from NeurIPS:
papers collected

Also: https://github.com/ignorejjj/LongRefiner/blob/main/README.md

Related Work: Towards Trustworthy Legislative Intelligence

Current AI for law and policy is okay, but not great. If you ask a normal AI to summarize a 500-page bill, it usually just gives you the SparkNotes version. It's vaguely right but legally dangerous. It misses the small stuff, hallucinates details, and doesn't actually connect to the hard data (like how people voted or where the money is going).We want to build a system that actually thinks like a policy analyst. It needs to look at a PDF, check a database, and prove its work.

The Papers We're Building On (The "Research Backbone")

We found three papers from NeurIPS 2025 that solve some of the issues we're facing. Here's the breakdown in plain English

1. SQLens

- **The Problem -** LLMs are great at writing code (SQL) to talk to databases, but they often mess up the logic. They might give you a result that looks right but actually pulled the wrong data column.
- **The Fix -** SQLens doesn't just trust the AI. It listens for error signals from the database. If the result looks weird, it flags it and tries to fix it.
- **Our Spin -** We're using this logic to make sure our Snowflake integration doesn't lie to you when you ask for voting stats.

2. ALIGNRAG

- **The Problem-** Sometimes an AI retrieves the right document but then ignores it and just says whatever it wants based on its internal training
- **The Fix-** This paper adds a Critic that watches the AI while it thinks. If the AI starts veering away from the actual text of the bill, the Critic forces it to realign with the facts.

- **Our Spin-** This is perfect for legal docs. We want our copilot to be focused on the actual text of the bill, not just making a good guess.

3. FACT

- **The Problem-** AI can be inconsistent. It might tell you one thing in a summary but then contradict itself if you ask a follow-up question.
- **The Fix-** The FACT paper trains the AI to think in code logic while it's writing text. By treating facts like variables in a program, it stays way more consistent.
- **Our Spin-** We'll use this to make sure our policy interpretations are logically sound from start to finish.

What's Missing? (The "Gap" We're Filling)

Even though these papers are brilliant, they're still not perfect

- They don't really 'talk' to each other.
- They aren't built for a secure environment like Snowflake.
- They haven't been tested on messy, real-world legislative PDFs that have weird formatting and dense language.

Our goal is to take these three ideas and weld them together into one solid system. We aren't just making a model. We're trying to build a whole pipeline that ingests data, double-checks it, and gives you an answer you can actually trust for work.

.

Positioning Our Legislative AI System: Why This Isn't Just a Chatbot

When we look at the AI tools available today, most of them act like a single brain that tries to answer everything at once. This is okay for general questions, but for something as important as public policy and laws, a single AI model isn't enough. We are looking into building what researchers call a Compound AI System. This means we aren't just using one AI, we are building a team of specialized tools that work together to make sure the final answer is actually right.

1. Building on the Best Ideas (What We Borrow)

We aren't starting from scratch. We are using three big ideas from recent research

- **Double-Checking the Logic-** Using ideas from ALIGNRAG, our system doesn't just give an answer and stop. It uses a "critic" to look back at the original law and the AI's answer to make sure they actually match. If they don't, it fixes the reasoning before you ever see it.

- **Staying Consistent-** Following the FACT paper, we treat facts like pieces of code. This helps the AI stay logical so it doesn't tell you one thing in the summary and then contradict itself when you ask a follow-up question.
- **Verifying the Data-** We adapted the signals from SQLens to help our system talk to our Snowflake database. This ensures that when you ask for numbers or voting records, the AI gets them from a real database instead of "hallucinating" them.

2. What We Add

The most important part of our project is how we handle the gray areas in policy. Most AIs try to sound confident even when they are guessing. We are adding two specific features:

- **The Ambiguity Flag-** Our system will be designed to recognize when a law is written in a confusing or vague way. Instead of just picking one interpretation, it will flag that section and tell the user, "This part of the bill is open to interpretation."
- **The Trust Score -** Every answer comes with a score. This score tells you how much of the answer is based on the actual text of the bill versus the AI's general knowledge.

3. Why a "System" is Better for Governance

In public policy, people need to know why a decision was made. If you just use a single AI model, it’s a black box, you can't see inside it. By building this as a system in Snowflake, we could offer

- **A Paper Trail-** Every answer links back to a specific page and paragraph in the official government PDF.
- **Safety and Privacy-** By using professional tools like Snowflake, we can keep sensitive data secure, which is something a random online chatbot can't guarantee.
- **A Tool for Humans-** We aren't trying (and we don't want) to replace policy experts. We are building a tool that handles the heavy lifting of reading thousands of pages so that humans can focus on making better decisions.