LLM - Legal Layman's Mansplaininator by the Forensic Fiends

Ariadne Dimarogona
apdimaro@asu.edu
Arizona State University

Aditi Ganapathi
aganap12@asu.edu
Arizona State University

Easton Kelso
eakelso@asu.edu
Arizona State University

October 28, 2025

1 Problem Statement

The admissibility of digital evidence involves complex regulations and guidelines and chain of custody procedures, often hundreds of pages long([1, 2]), with additional guidelines per state ([3]). In cases, new and complex technologies are handled as evidence, which did not exist when these regulations were written. When the "Forensic Examination of Digital Evidence: A Guide for Law Enforcement" was first published in 2004, it could not predict the types of technologies that are handled by Law Enforcement in 2025.

It is a well known fact and issue that laws and regulations are always playing catch up with emerging technologies. This makes it extremely challenging for lawyers to understand how electronic evidence should be handled when technology is always advancing and how to infer admissibility upon forensic expert testimony on such technology. This testimony often leads to dealing with extremely technical details. For lawyers, the technical jargon, as well as the constant evolution of new technology, is incredibly hard to keep up with. Similarly, for Forensic Examiners, finding ways to explain highly complex technical evidence to juries, lawyers, and judges is increasingly a challenging aspect as technology gets more interconnected.

According to a guest lecturer in the Scottsdale Police Department, oftentimes they have to rely on analogies, which can be hard to come up with. This leads to the authors' project on creating a system that would be able to assist an expert witness and their legal council on how to present their findings in a court of law.

2 Main Objectives

To bridge this gap between technical expertise and legal expertise, we propose a tool that can parse through expert witness testimony, highlight potential legal issues, and offer translation in layman's terms to better explain technical details. It would include a clear, easy-to-use interface that uses a Retrieval Augmented Generation (RAG) that references the long guideline documents listed above to highlight potential admissibility issues and a large language model (LLM) to analyze the witness testimony, in order to flag potentially non-compliant portions of a testimony and generate analogies for sections that are not in layman's terms. Then users can look back at those specific portions and use their expertise to determine whether the testimony is compliant or not and, equipped with their new understanding of the testimony from the tool's translation, provide their arguments in court.

To summarize, our objectives are as follows:

1. Create a RAG model and pair it with an LLM to be able to highlight potentially non-compliant portions of a technical witness testimony
2. Create prompts for an LLM to generate translations from technical Jargon to easy-to-understand summaries with analogies of technical details.
3. Create a visual interface for a user to easily paste written witness testimony for analysis and feedback.

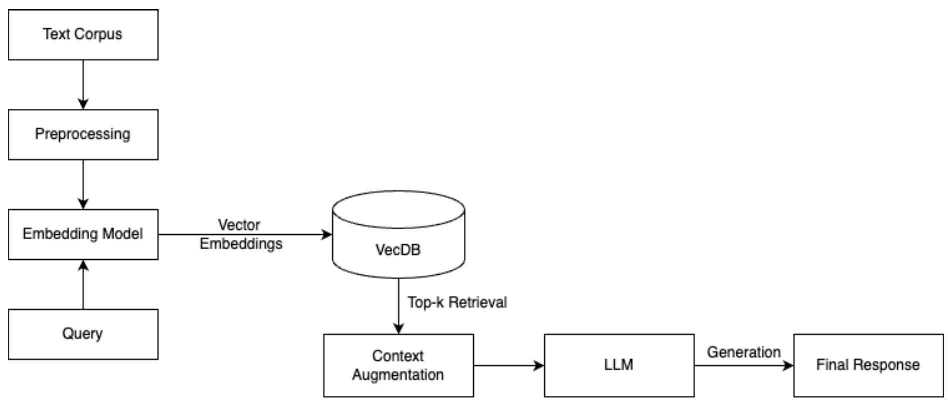

Figure 1: Proposed architecture for the RAG model

3 Methodology

This project proposes to build a RAG-based AI tool designed to automate compliance checks for digital forensic expert witnesses. This system aims to cross-reference testimonies, reports, and evidence-handling procedures against the Federal Rules of Civil Procedure, NIJ digital evidence manuals, and chain-of-custody guidelines, ensuring procedural integrity and admissibility of digital evidence. Figure 1 illustrates the overall system pipeline.

This section explains the process of data collection, tool development, interface setup, workflow design, and feedback collection for model evaluation and refinement.

3.1 Phase 1: Data Collection

Legal and forensic documents will be compiled as the foundation of the tool's knowledge base. Primary sources include the Federal Rules of Civil Procedure, National Institute of Justice (NIJ) digital evidence manuals, jurisdiction-specific guidelines, and legacy manuals such as the DOJ Search and Seizure Manual. Additionally, publicly accessible court case transcripts and opinions will be collected to serve as ground truth for validating the system's outputs. These documents will be pre-processed, segmented into meaningful sections, and embedded into semantic vector representations to facilitate retrieval.

3.2 Phase 2: Creating the compliance highlighter using RAG

We propose to use a RAG model consisting of an embedding model, vector database, and a large language model (LLM). Figure 1 shows the proposed architecture of our setup.

All collected materials will be processed using a Sentence-Transformer embedding model. The resulting vectors will be stored in a FAISS vector database to enable efficient similarity searches.

At query time, the retriever component will encode user inputs (for example, "Does this forensic report meet NIJ chain-of-custody standards?") and compare them to embeddings in the database. The top-k most semantically similar document chunks will be retrieved. The value of k will be determined after experimentation.

The generation component will consist of an LLM. Each user query will be combined with retrieved document excerpts via prompt augmentation. This will help produce more relevant, context-aware responses and suggestions. While the exact LLM will be finalized depending on performance, we intend to prioritize using LLMs specifically fine-tuned for forensic and cybersecurity tasks (for example, ForensicGPT).

3.3 Phase 3: Creating the Technical to Layman Translator

We will be creating the translations via a prompt to an LLM. In order to find a good prompt, we put forward a systematic plan. All three members of the group will iteratively come up with the best prompt each to put forward. We will then,

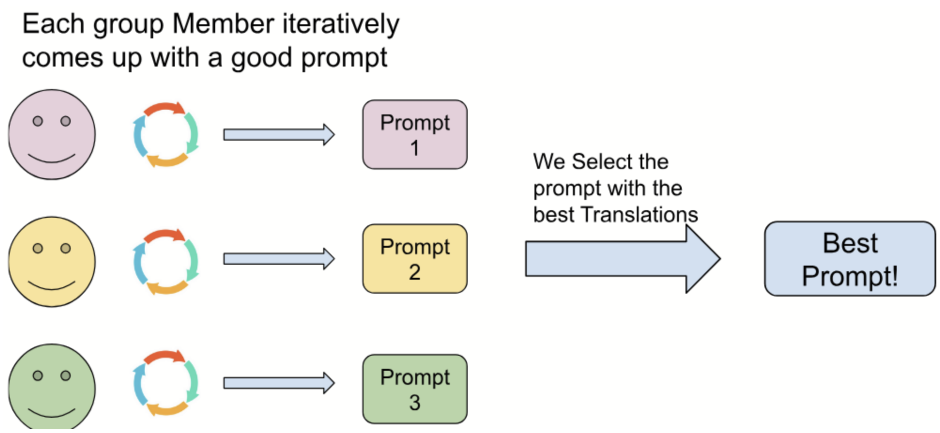

Figure 2: Creating the prompt

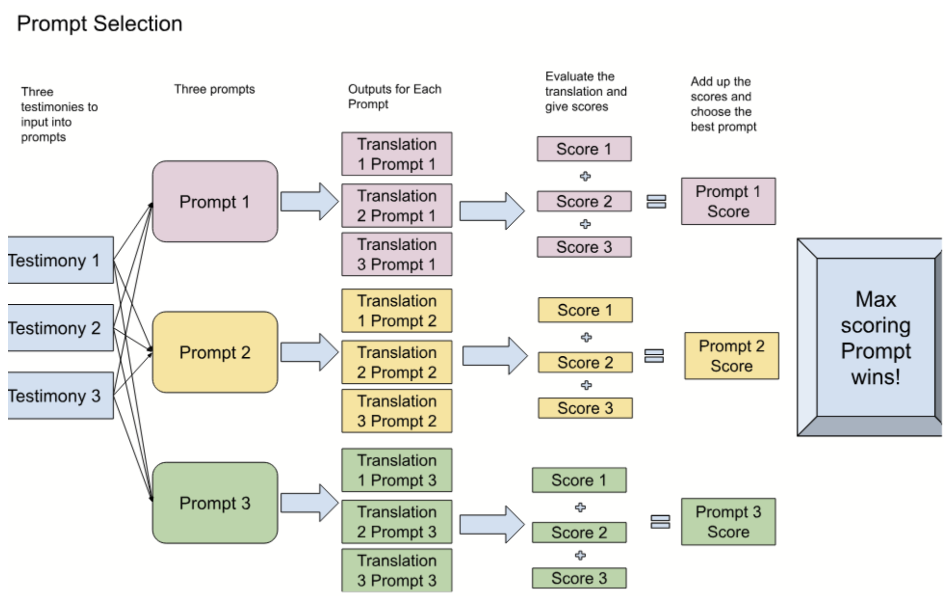

Figure 3: Prompt Selection

as a group, select the best prompt among our top three. We do this, instead of simply iterating to find a single prompt, because this strategy will allow us to explore a larger search space for an ideal prompt. We will select prompts based on the methods and criteria below in order to choose the best one to use in our final project demo.

To select the best prompts, we plan to test them against the same transcripts as the compliance highlighter from court cases pulled from PACER that include technical expert testimonies (see Figure 2 and Figure 3). We will test these transcripts against each prompt and self-evaluate the outputs of the prompts. We will test against 3 expert testimonies and evaluate those 3 for each prompt for a total of 9 translations. We will select the prompt that has the best translations for the largest number of testimonies.

The following criteria will be employed for our evaluation of each translation:

1. Accuracy: Translation must be accurate and true to the original technical statement.
2. Clarity: Translation must be clear and concise, and avoid technical jargon.
3. Accessibility: Translation must match the audience's assumed knowledge; In this case, lawyers. It MUST speak plainly and use analogies for easier understanding of the technical details.

3.4 Phase 4: Creating the Interface

The interface will be implemented using Python's Streamlit Library. This is a popular front-end library tool for AI tools and is easy to implement. We plan to host our interface for a live demo during presentations using one of our own machines as the server. We plan to also explore free hosting solutions through GitHub Pages or huggingface's free tier hosting. It is important that we are able to host the platform with a simple GUI so we can ask relevant stakeholders, who may not be technically proficient, to use our platform and give us feedback.

4 Timeline and Roles

We aim to finish this project before the first listed project presentation date of November 24th.

We aim to follow an agile-like development approach, ensuring that individual deliverable portions are done between sprint cycles. We will work in the following three sprint cycles:

- RAG Compliance Highlighter: 2 weeks 10/22/25 - 11/5/25
- Technical Layman Translator: 1 week 11/5/25 - 11/14/25
- Interface, Presentation, and asking for expert feedback. 1 week 11/14/25 - 11/23/25

Below we track our roles and responsibilities as well as their status. In Table 1, we provide soft deadlines for sub-portions within sprints to keep on track.

Table 1: **Roles and Deadlines**

Portion	Owner	Status	Deadline
Working on Proposal	Ariadne, Easton, Aditi	**Complete**	10/24/25
Gather Data for the RAG model and testing	Easton	**In progress**	10/27/25
Prepare and clean data for the RAG Pipeline and testing	Ariadne, Easton	Not started	10/29/25
Prepare and implement the RAG Pipeline with data	Aditi	Not started	11/03/25
Implement the testing plan for outputs	Ariadne, Easton	Not started	11/05/25
Come up with a translator prompt	Ariadne, Easton, Aditi	Not started	11/07/25
Test and grade the prompt outputs to choose the best prompt	Ariadne, Easton, Aditi	Not started	11/14/25
Create an Interface Using Streamlit	Ariadne	Not started	11/19/25
Gather legal expert opinion on the tool	Easton	Not started	11/21/25
Work on Presentation	Ariadne, Easton, Aditi	Not started	11/23/25

References

[1] National Institute of Justice. Forensic examination of digital evidence: A guide for law enforcement. 2004.

[2] National Institute of Justice. Digital evidence policies and procedures manual. 2020.

[3] State of California Department of Justice. Investigations prosecutions guidelines.