

LegalEyes: Simplifying Legal Document with AI Insights

Madhura Badave^{1*}, Ankita Chavan², Payal Jadhav³, Sudeep Rathod⁴,
Dr. D. S. Hirolikar⁵

^{1,2,3,4,5}Department Of Information Technology PDEA's College Of Engineering, Savitribai Phule Pune University,
Pune, Maharashtra, India

ABSTRACT

LegalEyes represents a complex web-based platform that simplifies the involvement of a wide audience in the reading of complicated legal texts such as citizens, students, job seekers, and paralegal workers. Conventional legal prose is often full of specialized language and not clearly written sentences, putting the understanding at risk and creating the possibility of making wrong judgments. Artificial intelligence (AI), machine learning (ML), and natural language processing (NLP) are automated technologies that have been smoothly incorporated into the full-stack architecture of LegalEyes. The user can submit a manuscript in PDF, DOC, DOCX or plain-text form and the system will then create summaries that are concise, salient or precarious clauses highlighted, and plain and jargon-free explanations are provided. LegalEyes works on a Retrieval-Augmented Generation (RAG) model, a combination of clause-level semantic-vector recovery and limited inference with large language models. The platform uses large language models as a part of an AI orchestration layer to improve contextual context, whereas the semantic retrieval based on vectors ensures the accurate analysis. Besides document deconstruction, LegalEyes has a navigational portal where people navigate through government and legal steps, and this defines necessary documentation and directs the user to the verifiable resources. Focusing on explainable AI concepts, the system has placed all the data under stringent security measures and restricted access measures, thus protecting the information of users. Finally, through intelligent automation and a user friendly design, LegalEyes will lead to demystification of the law material, less bias towards error and equalization of the situation, making legal interpretation more open, understandable, and accessible.

Keywords: Legal Document Analysis, Artificial Intelligence, Natural Language Processing, Large Language Models

INTRODUCTION

In the electronic era, people are daily exposed to legal and official paperwork, be it in getting jobs, in acquiring property or in registering in government services. The terms used there are usually very technical and most of the times vague making it unreachable to non-legal experts. Many will therefore accept terms or give up documentations without proper knowledge, a situation that may end up in judicial courts or monetary loss. New developments in AI, ML, and NLP provide a chance to automatize the interpretation of such texts and enrich it. Smart AI software is able to scan large amounts of documents, identify trends and generate plain-language descriptions that fall in between legalese and common language. LegalEyes is a project envisioned as an AI-based application, which will autonomously process legal documents and provide guidance in the form of structured advice through a convenient web application interface. It is based on large language models and retrieval-augmented methods, and breaks down, outlines crucial or risky clauses, and provides coherent explanations. The architecture uses a microservice approach, with a hybrid Java backend and Python-based AI orchestration to support scalability and modularity. Ideally, LegalEyes will have a full-fledged guidance-portal where it will take the user through the complexities of the official paperwork breaking down the information into easily digestible bits and listing the necessary documents. LegalEyes can provide greater levels of transparency, accessibility, and trust in legal and administration due to its explainable AI structure and strong security controls.

LITERATURE SURVEY

The implementation of Artificial Intelligence in the legal field has in recent years evolved out of primitive, rule-based document systems into advanced transformer-based implementation and large language model (LLM) systems. First generation solutions brought about by knowledge were more focused on the extraction of structured information and

careful rule engineering. An example is Takale, which promoted a Ripple Down Rules (RDR) system of legal document summarization, which proved effective in identifying rhetorical positions in a knowledge acquisition process in small steps [1]. Such methods can be scaled but find it difficult to be applicable to the variety of legal corpora that contemporary practice requires. Summarization efficiency was also increased by the introduction of deep-learning methods. Wang and Wu introduced the SAC-VAE architecture, a beautiful combination of the variational auto-encoders and reinforcement learning, that can handle high dimensional pieces of law and improve the summarization performance significantly [2]. The systems are however still largely compression based, with little contextual clause-level reasoning. Transformer-based models have significantly improved semantic cognition in tasks of legal natural language processing. The review of the applications of LLM to the legal systems has been done extensively by Siino and colleagues, who claimed to achieve better results in the areas of contract analysis, legal search, and document summarization [3]. Similarly, Davane et al. presented a model of Legal-Lens, a system that combines NLP-based clauses detection and rule based validation to perform a contract risk analysis [4]. The importance of entity recognition and systematic exploration to legal analytics is highlighted by knowledge-based service platforms, represented by the IDJ system [5]. To supplement these efforts, realistic AI-based systems, such as law chatbots, document automation systems, aim to make legal services more accessible and interactive to users [6]. With these developments, however, available solutions tend to separate individual capabilities, including summarization, entity extraction, or chatbot automation. This has created an urgent need to integrate systems that can harmonize or unify a set of vectors, semantic retrieval, structured LLM reasoning and explainable risk detection in a scalable architecture, an aim that the proposed LegalEyes system aims to accomplish.

MOTIVATION OF THE PROJECT

Documentation in the law is largely written to a professional audience and is full of heavy-handed terminology, long sentences and contingencies which are difficult to understand even to a non-lawyer. People still put their signatures or produce such papers without proper understanding and put themselves in a disadvantaged position legally, financially, or procedurally. Reviewing manually is time-consuming and capital-intensive which makes it unfeasible when applied to large masses of the population. The LegalEyes idea arose due to the realization of this gap between the obscure character of legal texts and everyday knowledge. It was imperative to come up with a complex, computerized tool which would de-juridify legal text, highlight important provisions and shed a lot of light on the risks which a reader may be exposed to in a reader-friendly format. The solution to this requirement is LegalEyes, which uses AI, ML, and NLP to process documents automatically and give feasible recommendations. Risk indicators, abstract summaries, and stepwise instructions enable the user to manage legal documents with confidence and eliminate the need to pay an attorney, access legal information more easily, and increase openness and reliability in the system.

PROPOSED SYSTEM

LegalEyes basically is a cloud native web platform that helps regular people to get the hang of legal documents without having to become a lawyer. Legal papers are replete with jargons and vague clauses that perplex the average user. LegalEyes addresses this by combining AI, ML, and NLP in a secure and scalable stack. The architecture is divided into three layers of microservices, which are used to separate user frontends, business logic and AI processing. This makes this whole thing ever so much more stable, easier to scale, and easier to maintain. The Spring Boot app manages the sessions, coordinates the flow of documents and routes the requests between the front end and the AI services in a smooth way. Heavy lifting, such as the semantic analysis, summarization, clause spotting and contextual reasoning, is left to a python-based AI layer with large language models. Keeping the workload of AI separate means that the UI is kept snappy and backend reliable. LegalEyes is modular and extensible, meaning that while keeping user data private, it scales up. The idea is to empower everyone with a basic understanding of the law by making legalese easily accessible by translating it into understandable actionable insights.

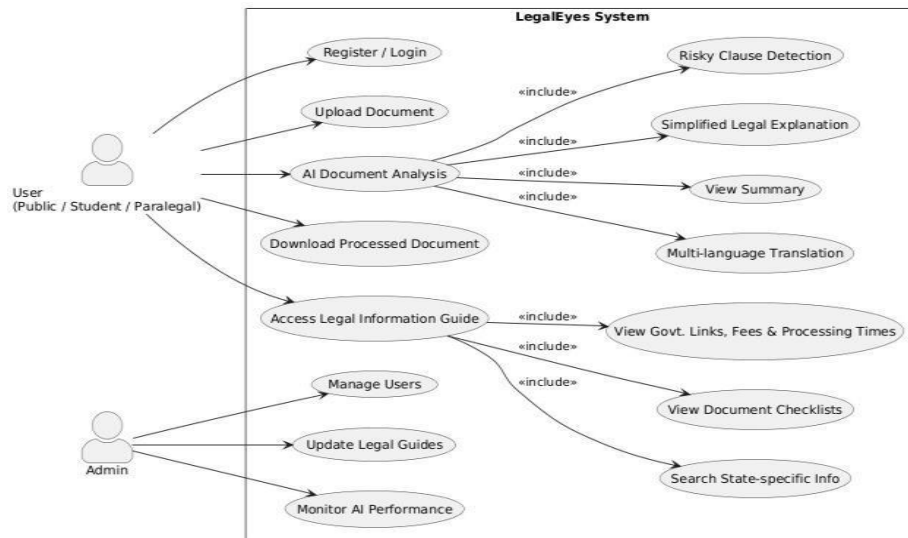


Fig. 1: Use-Case Diagram

METHODOLOGY

This part breaks down how the LegalEyes platform actually works from start to finish - like a flowchart to the whole process. It's all about the user experience and how each and every piece of parts talks to the next one. **STEP 1: User**

Authentication and Access Control Mechanism

First, users will sign up and log in using a secure, JWT-based system. The platform does not let anyone do anything unless it is the right person, keeping an eye on who has access to what. This is used to keep all that sensitive stuff locked up and prevent the unwanted people are denied from peeping.

STEP 2: Upload documents and Metadata handling

Once you're in, you can upload legal or official files - PDF, DOC, DOCX or TXT. The backend checks that the file is legit and pulls out some basic information like what type of file it is, how big it is and when you uploaded it. All that data goes straight into the database and then we get started with the analysis.

STEP 3: AI-Driven Document Processing and Semantic analysis

The verified doc continues to a Python-based AI layer that is built with FastAPI. The system takes the document and splits it into workable chunks, converts the chunks into vector embeddings and stores them in ChromaDB. Using a Retrieval-Augmented Generation (RAG) approach, we retrieve the most relevant context and pass it to big models such as GPT-4 Turbo and Mistral. They churn out summaries and pick out the key clauses and flag up any legal risks.

STEP 4: Response Generation and Explainability

The AI spits out clear and user-friendly summaries, risk markers and plain English explanations - all in a package of the form of a JSON. The point here is that you want to be able to make the AI's work transparent so that you can follow the reasoning in each insight.

STEP 5: Recording of data and Result Visualization

All the user info, file meta, and logs are stored in the PostgreSQL, and the vector embeddings are stored in the ChromaDB for smart search later. Once everything's processed, the results are displayed on the front end where you can see the summaries and highlighted clauses and the explanations laid out in real time on an interactive dashboard.

STEP 6: Document Procedure Guidance

There's a portal besides just analyzing documents provides you with step by step legal and administrative info. It includes a list of paperwork that you will need, the steps to the procedure that you should take, and links to reliable sources. That way, it is possible to make smarter choices without having to hire a lawyer.

ARCHITECTURAL DESIGN

LegalEyes uses a three-tier microservices setup that would keep things modular, scalable and fault tolerant for AI-driven analysis of legal documents. Its architecture consists of a refined front-end, a robust back-end, and a dedicated AI layer which have very clear responsibilities.

9.1 Frontend

The front-end is built with React 18 and TypeScript to provide a single-page application that is responsive. This enables the users to upload the documents, review AI generated summaries and evaluations, and can easily browse areas of legal guides using simple and interactive dashboard. The design is more usable, accessible, and consistent to cross platforms, guaranteeing effortless user experience on various devices.

9.2 Backend Core

Security Layer Spring Boot 3.3 is responsible for the logic part of the application, i.e., it is the orchestrator. The purpose of this security layer is to handle the management of APIs with the help of RESTful APIs, to implement the authentication and authorization of the JWT, for scheduling workflow of the documents. This layer is for all of the logic and metadata handling for business logic and then passes the AI heavy tasks to the AI layer.

9.3 External AI Models

Polyglot persistence strategy is used. PostgreSQL is a database that is used to store structured data such as user profiles, document metadata and audit logs; it is ACID compliant. At the same time, ChromaDB is a vector database, which contains embeddings allowing quick semantic search. This two-storage system guarantees that contextually appropriate information is obtained in an optimized manner.

9.4 AI Processing Engine

FastAPI along with Python performs all the processing for AI. It chunking documents, creating embeddings, running semantic queries and generating responses using LLMs such as GPT-4 Turbo and Mistral. Using Retrieval-Augmented Generation (RAG) allows the answers to have a better contextual touch. The results are returned as a JSON so that they can be fed back to the UI from the backend.

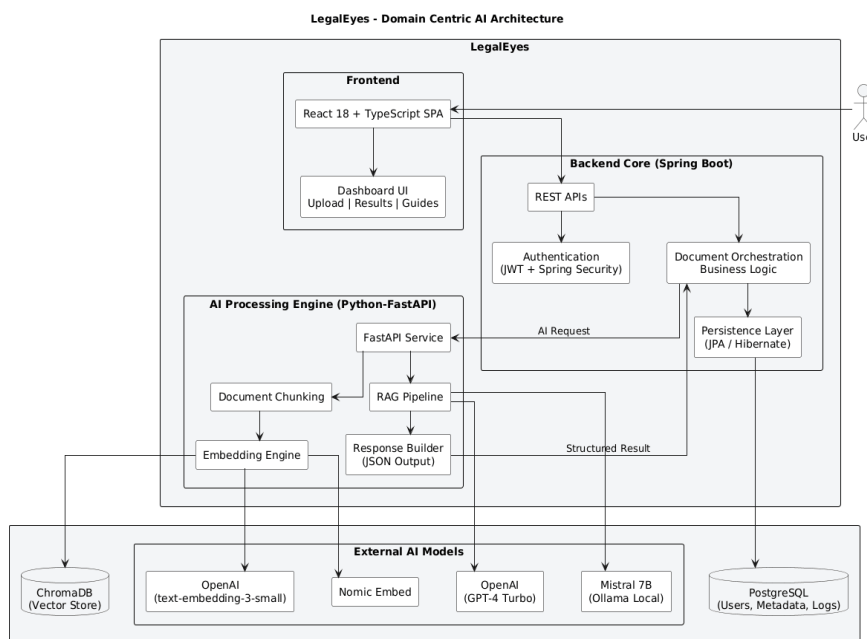


Fig. 2: Architecture Diagram

APPLICATION FEATURES

The platform provides a series of features that are aimed at replicating the work of a personal legal assistant. **10.1 General User Functionality Document Upload and Analysis.**

The user can drop PDF, DOC and TXT files to be automatically analyzed. Simple Comparisons - Severe language is converted into a simple language that is easy to understand. Clause and risk Highlights - Key clauses and risks are indicated by an annotation of the explanations. Step-by-Step Guides- Organized help is given in order to know and compose official documentation. Secure Authentication - Authentications are secured using secure authentication tokens and JWT.

10.2 Legal Professional and Institutional Features.

Explainable AI - The findings are understandable, and the professionals can examine AI derivations. Semantic Search- Vectorbased search helps to quickly find relevant parts of documents. Real time Processing: Optimized back-end and

artificial intelligence layers will provide the ability to analyze large amounts of documents within a short time. Interactive Dashboard The React interface presents summaries, highlight and AI feedback in a well structured way. Data Security & Compliance- There are strict controls that are placed to maintain data confidentiality and integrity.

RESULT

The LegalEyes framework was evaluated on a highly selective set of legal texts, which included legal texts of rental agreements, service contracts and policy documents. The evaluation focused on three main points, namely the strength of document processing, effectiveness of semantic retrieval and effectiveness of RAG-based reasoning pipeline. The system had a text extraction success rate of 98% in various types of documents during the document ingestion phase. The accuracy of the clauses level was 96%, which confirmed the presence of the adequate legal boundaries within the semantic indexing by the chunking strategy. Embedding generation had a consistency of 95%, which highlights a consistent semantic description of the content of the clauses. Using the semantic similarity retrieval mechanism, relevant accuracy was approximately 92% in a range of legal queries. RAG-based reasoning provided made 90% correct decisions, which were of a legally sound nature. Citation traceability was performed at 94%, which showed a strong foundation of generated responses on retrieved evidence of clause. Lastly, the confidence -scoring program was found to have a reliability of 88% which indicated a consistent semantic correspondence between retrieved clauses and generated interpretations. Overall, the findings of this analysis support the argument that the combination of the clause-level chunking, semantic retrieval, and limited language-model inference can help contextualize and improve clarity in AI-assisted legal document analysis.

CONCLUSION

The implementation of the LegalEyes proves that Artificial Intelligence and Natural Language Processing can be used to make complex documents in the legal field easier to understand by ordinary users. It is demonstrated here that the system incorporates semantic retrieval into a Retrieval-Augmented Generation (RAG) system to generate correct summaries, detect important or risky clauses, and create clear and plain-language descriptions. The experimental assessment on the basis of curated legal documents demonstrated a higher degree of contextual relevance, higher quality clause-level reasoning, and less ambiguity than the traditional methods of using keywords. The microservices structure, authentication, and hybrid data storage plan also guaranteed reliability, scalability, as well as confidentiality of data in the system. Altogether, LegalEyes fulfilled its goal of simplifying the legal material and making it easier to comprehend and use. To improve it further, in the future, the platform can be enhanced to multilingual summarization to accommodate people with different lingual backgrounds. Voice interaction and speech to text features can enhance accessibility to a user who has limited typing abilities. Also, it is possible to enhance decision-support functionality by incorporating experts recommendation functionalities and increasing the legal knowledge base. The further refinement of AI models and training in specific fields will increase accuracy, the quality of reasoning, and real-time operation, which will allow LegalEyes to become a full fledged digital legal assistant.

REFERENCES

- [1] Davane DR, Shirsath AP, Kalamkar GA, Itankar PY. Legal-Lens: AI-Driven Legal Document Review and Risk Analysis. *Int J Sci Res Eng Manag.* 2025;9(4). doi:10.55041/IJSREM45876.
- [2] Sheetal Ajaykumar Takale. Knowledge Representation for Legal Document Summarization. *Int J Innov Res Comput Sci Technol.* 2023;11(4):61–66. doi:10.55524/ijrcst.2023.11.4.11.
- [3] Xukang Wang, Ying Cheng Wu. Empowering legal justice with AI: A reinforcement learning SAC-VAE framework for advanced legal text summarization. *PLoS ONE.* 2024;19(10):e0312623. doi:10.1371/journal.pone.0312623.
- [4] Sumit Dahiya. Developing AI-Powered Java Applications in the Cloud: Harnessing Machine Learning for Innovative Solutions. *Innov Comput Sci J.* 2024;10(1).
- [5] Atharva-Avinash-Mahajan *, Sai-Bhor, Rushikesh-Patil, Swati-Rane and Prasad Reshim. Legal chatbot and document generator using ReFlaPsy. <https://doi.org/10.30574/wjarr.2025.26.1.1.370>
- [6] Marco Siino, Mariana Falco, Daniele Croce, And Paolo Rosso, Exploring LLMs Applications in Law: A Literature Review on Current Legal NLP Approaches. doi: 10.1109/ACCESS.2025.3533217