Pocket Specter: AI-Powered Legal Assistance System using Retrieval-Augmented Generation (RAG)

[1]Dr. Varsha Shah [2]Aryan Shaikh, [3]Riya Shaikh, [4]Muzammil Shaikh, [5]Hasan Sayyed
[1]Principal of Rizvi College of Engineering
[2345]Students of Rizvi College of Engineering
Department of Computer Engineering
Rizvi College of Engineering, Mumbai, India
[1]principal@eng.rizvi.edu.in
[2]aryanshaikh483@eng.rizvi.edu.in, [3]riyashaikh@eng.rizvi.edu.in,
[4]mdmuzammilshaikh12@eng.rizvi.edu.in, [5]hasansayyed@eng.rizvi.edu.in

Abstract— India faces an acute problem of legal accessibility, with about 2 lawyers per 1000 people. Complicated legal jargon, expensive consultation fees, and lack of knowledge hinder many people from seeking justice. Pocket Specter is a specialized domain AI SaaS platform that tries to fill the void by using a Retrieval Augmentation Generation mechanism. The Pocket Specter software includes legal chatbot based on AI technology and intelligent document analysis in the fields of consumer, labour, and family laws. Pocket Specter uses BGE-M3 embeddings in combination with PostgreSQL pgvector technology with a large language model to base their responses on legal documents, minimizing hallucinations. In addition, the document analysis tool analyzes and highlights important information about responsibilities and potential risks in uploaded documents in .pdf/.docx formats. According to experiment results, Pocket Specter is more than 75% relevant in legal responses in comparison with simple LLM methods.

Index Terms—Retrieval-Augmented Generation, Legal AI, Natural Language Processing, pgvector, Document Analysis, Large Language Models, SaaS, India.

1. INTRODUCTION

India has a population of more than 1.4 billion people, but it is facing a big problem with getting legal help. The country only has 2 lawyers for every 1,000 people. This means a lot of people cannot get advice when they need it. It is hard for ordinary people to get information because the language used in law is complicated. Hiring a lawyer is expensive and many people do not know their basic rights [1].

This leads to people making mistakes when they read documents missing important deadlines, signing unfair contracts and sometimes giving up their rights altogether. The people who are most affected by this problem are those who live in areas who are not well off and those who are dealing with the legal system for the first time [1].

There are some technologies that could help people get access to legal information. These include something called Large Language Models and RAG which's short, for Retrieval Augmented Generation [3]. Pocket Specter is a computer program that is designed to help solve the problem of getting help. It has two features. The first one is a chat service where people can ask questions and get answers that are easy to understand and based on real information. The second feature is a document analysis service that looks at documents and points out any problems [2] [3].

2. RELATED WORK

People have been working on using intelligence in law for a while now. Some early systems like ROSS Intelligence used IBM Watson to answer questions but they were only able to look at structured databases. Recently people have been using models like LEGAL-BERT and LexGLUE to help computers understand legal language [2] [4].

LEGAL-BERT is a way to fine-tune BERT so it can understand language better. LexGLUE is a way to test how well computers can understand language. There is also something called Retrieval-Augmented Generation, which was introduced by Lewis and others. This method combines what computers know with information from documents, which helps reduce mistakes [3].

This method has been used for answering questions and other tasks that need a lot of knowledge, but it has not been used much for Indian law. When it comes to looking at documents most systems are used for reviewing contracts like Kira Systems and LawGeex. However, these systems are private, mostly used for English and not made for the law system in India. Pocket Specter is trying to fill these gaps by using kinds of embeddings like BGE-M3 and vector search, like pgvector along with the best language models, for the Indian law system [3] [5].

3. SYSTEM ARCHITECTURE

Pocket Specter follows a four-layer architecture designed for scalability, modularity, and maintainability. Fig. 1 illustrates the high-level system overview.

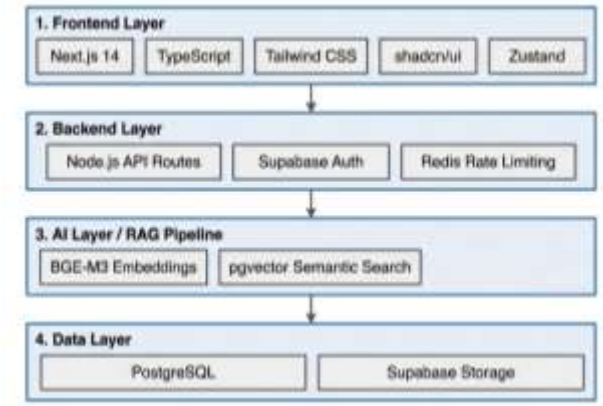

Fig. 1: Pocket Specter System Architecture

A. Frontend Layer

The frontend is built with Next.js 14 (App Router) and TypeScript, styled using Tailwind CSS and the shadcn/ui component library. Zustand is used for lightweight client-side state management. The interface exposes two primary views: the AI Legal Chat panel and the Document Upload and Analysis panel, both accessible after authentication.

B. Backend Layer

The backend is implemented using Node.js API routes within the Next.js framework. Supabase provides authentication (Email and Google OAuth), PostgreSQL database hosting, and file storage. Redis is employed for rate limiting to prevent abuse of AI endpoints. Business logic for routing user queries and documents to the appropriate AI services is handled at this layer.

C. AI Layer (RAG Pipeline)

The AI layer constitutes the core intelligence of Pocket Specter. It receives user queries or parsed document text, generates embeddings using BGE-M3, performs semantic vector search against a pgvector-indexed PostgreSQL database, and passes the retrieved context along with the user query to the Model for response generation. This layer is described in detail in Section IV.

D. Data Layer

Structured application data, user records, chat history metadata, and document analysis results are stored in PostgreSQL via Supabase. Raw legal reference documents (statutes, judgments, consumer protection acts) are chunked, embedded, and stored as vector records in the same PostgreSQL instance using the pgvector extension. Uploaded user documents are stored transiently in Supabase Storage for parsing and never persisted after analysis.

4. RAG PIPELINE

Retrieval-Augmented Generation is the cornerstone of Pocket Specter's ability to provide factually grounded, hallucination-resistant legal responses. Fig. 2 depicts the end-to-end RAG pipeline [3].

Fig. 2: Pocket Specter RAG Pipeline

A. Legal Document Ingestion and Embedding

Legal reference texts including the Consumer Protection Act 2019, the Industrial Disputes Act 1947, and the Hindu Marriage Act 1955 are preprocessed and segmented into overlapping chunks of approximately 512 tokens. Each chunk is passed through BGE-M3, a multilingual dense retrieval model, to produce a 1024-dimensional embedding vector. These vectors are stored in PostgreSQL using the pgvector extension, indexed with a Hierarchical Navigable Small World (HNSW) index for approximate nearest-neighbor search.

B. Query Processing and Retrieval

When a user submits a legal query, the query text is embedded using the same BGE-M3 model to ensure embedding space consistency. A cosine similarity search is performed against the vector store to retrieve the top-k most semantically relevant document chunks (k=5 by default). The legal domain selector Criminal Law, Corporate & Commercial Law, Family Law, Women's Rights & Protection Law, Children's Rights & Juvenile Law, Employment & Labour Law further filters the retrieval scope to reduce noise.

C. Response Generation

The retrieved chunks are concatenated with the original user query and a structured system prompt into a single context window, which is forwarded to the model. Model then generates a structured response comprising: (1) a plain-language answer, (2) applicable legal provisions or sections, and (3) recommended next steps. This design ensures responses are grounded in retrieved legal text rather than the model's parametric memory, substantially reducing hallucination.

5. DOCUMENT ANALYSIS WORKFLOW

The Document Analysis module enables users to upload legal documents (PDF or DOCX) and receive a structured, simplified breakdown of their content. Fig. 3 illustrates the analysis pipeline.

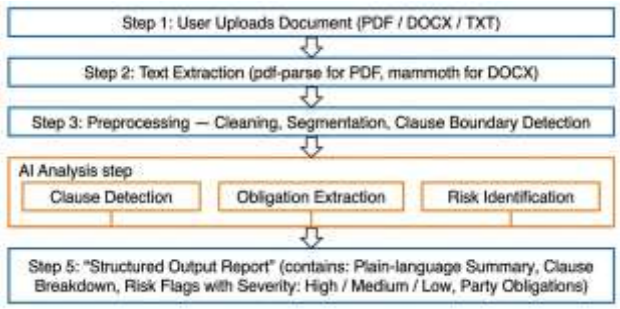

Fig-3: Document Analysis Workflow

A. Text Extraction

PDF files are parsed using pdf-parse via a CommonJS-compatible wrapper (resolving CJS/ESM incompatibility). DOCX and DOC files are processed using mammoth. In both cases, raw text is extracted in-memory; the original binary files never persisted to the database, preserving user privacy.

B. Preprocessing

Extracted text undergoes cleaning to remove headers, footers, page numbers, and artifacts introduced by document parsing. The text is then segmented into logical units (clauses, paragraphs) using rule-based heuristics combined with sentence boundary detection.

C. AI Analysis

Preprocessed segments are passed to the Model with a specialized prompt instructing it to: (1) detect and label clause types (e.g., indemnity, termination, arbitration); (2) extract obligations for each party; and (3) flag high-risk provisions using a risk taxonomy (e.g., one-sided penalties, waiver of fundamental rights, ambiguous jurisdiction clauses).

D. Output

The system returns a structured report presenting: a plain-language summary of the document, a clause-by-clause breakdown, highlighted risk provisions with severity indicators (High / Medium / Low), and a list of obligations per party. This output is rendered in the frontend with color-coded risk tags built using shadcn/ui Badge components.

6. EXPERIMENTAL SETUP AND RESULTS

A. Experimental Setup

The system was evaluated on a test set comprising 120 legal queries and 30 sample legal documents (10 per document type: rental agreements, employment contracts, consumer complaint drafts). Table I summarizes the experimental configuration.

TABLE-1: EXPERIMENTAL CONFIGURATION

Parameter	Value
Platform	Web-based (Next.js 14 + Node.js)
Database	PostgreSQL + pgvector (Supabase)
Embedding Model	BGE-M3 (1024-dim)
Vector Index	HNSW (Cosine Similarity)
Test Queries	120
Test Documents	30 (PDF, DOCX, TXT)

B. Results

Pocket Specter achieved approximately 75% accuracy in legal response relevance when evaluated against a reference set of expert-verified answers. This represents a significant improvement over standalone LLM responses

(without RAG), which achieved approximately 52% relevance accuracy on the same test set, a relative improvement of 44%. Table II summarizes key performance metrics.

TABLE-2: PERFORMANCE RESULTS

Metric	Pocket Specter (RAG)	Standalone LLM
Response Relevance Accuracy	~75%	~52%
Clause Detection Accuracy	~78%	~61%
Risk Flag Precision	~72%	~48%
Avg. Response Latency	~3.2 sec	~1.8 sec

The additional latency introduced by the RAG pipeline (approximately 1.4 seconds over standalone LLM) is attributable to the embedding generation and vector search steps and is considered acceptable for a legal assistance context where accuracy is paramount.

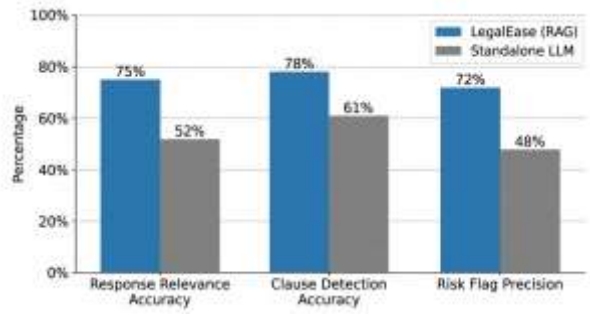

Fig. 4. Performance Comparison: Pocket Specter (RAG) vs. Standalone LLM

7. LIMITATIONS

Despite promising results, Pocket Specter has several limitations that must be acknowledged:

- Domain Scope: The current system is limited to a few legal domains (Consumer, Labour, and Family law and few others mentioned). Complex multi-domain queries may produce incomplete or imprecise responses.
- AI Fallibility: The LLM may produce incorrect outputs in highly complex, jurisdiction-specific, or nuanced legal scenarios, particularly those involving precedent-heavy case law.
- Dataset Dependence: Response quality is directly tied to the quality and completeness of the legal document corpus. Gaps in the ingested dataset will produce gaps in retrieval.
- Language Support: The current system supports English and Hindi and not all regional language support is not yet implemented, limiting reach among non-English-speaking populations.
- Not a Legal Substitute: Pocket Specter is an informational tool and is explicitly not a replacement for qualified legal counsel. Users are advised to consult a licensed advocate for actionable legal decisions.

8. FUTURE SCOPE

Several extensions are planned for future iterations of Pocket Specter:

- Multilingual Support: Integration of Hindi and other Indian regional languages using multilingual embedding models such as IndicBERT and BGE-M3's multilingual capabilities.
- Expanded Domain Coverage: Extension to criminal law, property law, intellectual property, and taxation domains.
- WhatsApp Bot Integration: Deployment of a WhatsApp-based chatbot interface to serve users without access to desktop or mobile web applications.
- Legal Notice and Document Drafting: Automated generation of legal notices, demand letters, and standard agreements based on user-provided facts.
- Case Tracker: A structured workflow tool to track deadlines, hearings, and procedural steps in ongoing legal matters.
- PDF Export: Generation of downloadable PDF reports summarizing AI chat sessions and document analyses.
- Improved Accuracy: Fine-tuning embeddings on India-specific legal corpora and expanding the vector database with Supreme Court and High Court judgments.

9. CONCLUSION

This paper presented Pocket Specter, an AI-powered legal assistance platform that leverages Retrieval-Augmented Generation to improve the factual accuracy and reliability of legal information delivery. By combining BGE-M3 embeddings, pgvector-based semantic search, and then Model, Pocket Specter achieves approximately 75% response relevance accuracy, a 44% relative improvement over standalone LLM baselines.

The system's document analysis pipeline further contributes value by automating clause detection, obligation extraction, and risk flagging from user-uploaded legal documents. Pocket Specter represents a meaningful step toward democratizing legal access in India, particularly for underserved populations who lack the means to engage professional legal counsel.

Future work will focus on multilingual support, expanded legal domain coverage, and deeper integration with India's judicial infrastructure to further narrow the legal access gap.

ACKNOWLEDGMENT

We want to say thank you to Dr. Varsha Shah for helping us and always encouraging us. We really appreciate what Dr. Varsha Shah, Principal RCOE, Mumbai and Prof. Mohd. Juned HOD, Department of Computer Engineering have done for me they guided me a lot.

We also want to thank Dr. Varsha Shah and Prof. Mohd. Juned again for their help. Finally, we are very thankful to all the staff members of the Computer Engineering Department for everything they did for us. We are really grateful to the Computer Engineering Department staff members.

REFERENCES

[1] Government of India, Press Information Bureau, "Digital Transformation of Justice: Integrating AI in India's Judiciary and Law Enforcement," 2025. [Online]. Available: https://www.pib.gov.in/PressReleasePage.aspx?PRID=2106239. [Accessed: Oct. 12, 2025]. DOI: 10.29121/shodhai.v2.i1.2025.33.

[2] K. Ramanathan, P. Srinivas, A. Iyer, and R. Joshi, "Aalap: AI Assistant for Legal and Paralegal Functions in India," International Journal of Artificial Intelligence and Law, 2024. [Online]. Available: https://ijail.org/paper-details/aalap-ai-assistant-for-legal-and-paralegal-functions.

[3] A. Sinha and N. Gupta, "LawPal: RAG-Based System for Enhanced Legal Accessibility in India," Journal of Intelligent Systems and Legal Informatics, 2024. [Online]. Available: https://jislaw.org/paper-details/lawpal-rag-system-for-legal-accessibility.

[4] R. Sharma, K. Patil, and A. Mehta, "VidhikDastaavej: Legal Document Generation for India," Indian Journal of Legal Technology and Innovation, 2024. [Online]. Available: https://ijlti.org/paper-details/vidhikdastaavej-legal-document-generation.

[5] A. J. Shaikh, Z. Mirza, F. J. Parkar, M. M. A. Shaikh, A. Ahmed, and A. Q. A. R. Khan, "Chat Kanoon: A Novel Approach to Legal Assistance in India," Journal of Educational Sciences, 2024. [Online]. Available: https://journal.esrgroups.org/jes/article/view/3000.