© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12-1718611

Lawmate: A Locally Deployable RAG System for Accessible Indian Legal Assistance Using Tinyllama And FAISS

PRATIK SHASHIKANT NAKASHE[1], ADITYA PRAVIN MEDHEKAR[2], SHREYA RANJAN PATIL[3], SOURABH GOVIND SADAKE[4], JYOTI BANSODE[5]
[1,2,3,4,5]*Department of Information Technology Shah & Anchor Kutchhi Engineering College* Mumbai, India

Abstract- ***Access to Indian legal information remains limited for non-lawyers due to complex statutory language, fragmented resources, and high consultation costs. This paper presents LawMate, a lightweight Retrieval-Augmented Generation (RAG)–based legal assistant designed to provide citation-grounded responses for everyday legal queries. The system integrates TinyLlama (1.1B parameters) with FAISS-based semantic retrieval over official Indian legal documents, ensuring responses are generated strictly from retrieved context to reduce hallucination. A FastAPI backend and React-based frontend enable efficient local deployment on resource-constrained devices. Experimental evaluation demonstrates high retrieval relevance, low response latency, and practical usability for citizen-facing applications. The proposed architecture provides a scalable, low-resource solution for transparent and accessible legal assistance in the Indian context.***

Index Terms—Legal AI, Retrieval-Augmented Generation, TinyLlama, FAISS, Indian Law, Offline Chatbot, Semantic Search, FastAPI.

I. INTRODUCTION

The legal domain contains a vast and continuously expanding collection of statutes, regulations, and judicial documents that can be difficult for individuals and professionals to navigate efficiently. Traditional legal research methods often require significant time and expertise to identify relevant provisions and interpret statutory language. With recent advances in Natural Language Processing (NLP) and Large Language Models (LLMs), intelligent systems capable of understanding legal language and retrieving relevant legal information have gained increasing attention [9], [14]. These systems allow users to interact with legal knowledge through natural language queries, improving accessibility to complex legal information.

Retrieval-Augmented Generation (RAG) has emerged as an effective approach for knowledge-intensive tasks by combining information retrieval with generative language models [18]. In this framework, relevant passages are retrieved from a document corpus and provided as context to the language model, enabling more accurate and grounded responses. Vector-based retrieval techniques and transformer architectures further enhance this capability by enabling semantic similarity search and improved understanding of long textual contexts commonly found in legal documents [21], [23], [24].

However, many existing legal AI systems rely on cloud-based infrastructures, which can raise concerns related to privacy, latency, and accessibility. Lightweight open-source language models provide an alternative by enabling efficient local deployment while maintaining reasonable reasoning capabilities [25].

In this work, we propose an offline legal question-answering system that integrates retrieval-augmented generation with vector-based document retrieval to provide accurate and explainable responses from statutory legal documents. By combining semantic retrieval with a lightweight language model, the proposed system aims to improve legal information accessibility while maintaining efficiency and data privacy.

The remainder of this paper is organized as follows. Section II presents related work, Section III describes the system architecture, Section IV explains the methodology, Section V discusses evaluation results, and Section VI concludes the paper with future work.

II. RELATED WORK

The integration of Retrieval-Augmented Generation (RAG) with large language models (LLMs) has emerged as a promising approach to enhance legal information accessibility, particularly in domain-specific and resource-constrained contexts like India. RAG architectures were originally introduced to improve knowledge-intensive tasks by combining parametric and non-parametric memory [18], [19].

RAG frameworks combine external knowledge retrieval with generative models to reduce hallucinations and improve factual accuracy in legal applications [18], [27]. LawPal [9] proposes a RAG-based legal chatbot using DeepSeek-R1 (5B parameters) and a FAISS vector store for efficient retrieval [21] from Indian legal documents, including the Constitution, aiming to democratize access through a Streamlit interface. Similarly, BharatLex [10] employs RAG with optimized LLM fusion (via Ollama) for multilingual legal query resolution, judgment summaries, and constitutional support, achieving high reported accuracy (99.99

Other Indian-focused systems include NyayaRAG [11], which applies RAG for realistic legal judgment prediction under the Indian common law system by retrieving prior cases and statutory provisions, and InLegalLLaMA [12], which fine-tunes LLMs on Indian legal knowledge graphs to enhance reasoning and petition drafting capabilities with RAG support. Efforts to improve accessibility emphasize lightweight architectures and application-focused domains.

Systems such as LegalBOT [13] and RAG-based assistants designed for rural India [14] utilize hybrid retrieval-generation pipelines for processing natural language legal queries, particularly in traffic regulations and property disputes, while supporting multilingual interaction. Hybrid RAG–LLM frameworks built upon open foundation models such as LLaMA [25] have also been explored for specialized legal interpretation tasks such as contract analysis and document management [16], [17].

Despite notable advancements, most existing systems rely on larger language models (5B parameters or more) or cloud-based infrastructures. Such reliance increases computational requirements, introduces internet dependency, and limits deployment in low-connectivity or low-end hardware environments commonly found in India. Furthermore, limited emphasis has been placed on fully offline functionality, ultra-lightweight LLMs, and production-ready backend architectures tailored for everyday users.

LawMate addresses these limitations by leveraging TinyL-lama (1.1B parameters) in conjunction with FAISS-based semantic retrieval [21], [29] to enable complete local and offline functionality. The system ensures fast inference on standard hardware and delivers structured responses centered on high-frequency statutory domains (e.g., the Motor Vehicles Act), thereby enhancing accessibility and privacy without significantly compromising accuracy.

III. SYSTEM ARCHITECTURE

LawMate adopts a Retrieval-Augmented Generation (RAG) architecture to deliver domain-specific legal assistance grounded in authenticated Indian legal documents. The architecture is modular, scalable, and optimized for low-resource deployment while ensuring transparency and citation-backed responses. Similar AI-driven legal assistance systems have been explored in prior work to improve public access to legal knowledge [1]– [5]. RAG architectures combine retrieval mechanisms with generative language models to improve knowledge-intensive tasks and reduce hallucinations [18], [27].

The system comprises five principal components: (i) User Interface Layer, (ii) Authentication Layer, (iii) Backend Processing Layer, (iv) Retrieval and Embedding Layer, and (v) Knowledge Base Layer.

A. User Interface Layer

The frontend interface is developed using React.js

© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12-1718611

with Vite, providing a responsive single-page application capable of handling real-time conversational interactions.

The interface allows users to submit natural language queries related to Indian land disputes, property laws, and traffic regulations. The frontend communicates securely with the backend through RESTful API endpoints over HTTPS. Streaming response rendering is implemented to improve user experience and re-duce perceived latency. Additionally, responses include source citations to enhance transparency and user trust.

B. Authentication Layer

Secure user authentication is implemented using Firebase Authentication, supporting email/password-based login and token-based session management. Authentication tokens are validated at the backend before processing queries, ensuring restricted access and preventing unauthorized API usage. This layer enhances system security and supports controlled de-ployment in public environments.

C. Backend Processing Layer

The Backend Processing Layer is implemented using FastAPI, selected for its high performance, asynchronous request handling, and native support for streaming responses. FastAPI acts as the central orchestration component, coordinating query processing, retrieval, and response generation. Upon receiving a user query, the backend validates authentication credentials and processes the request through a structured pipeline.

The query is first transformed into a semantic embedding representation using sentence embedding techniques [29]. A similarity search is then performed over the vector index to retrieve relevant legal content using efficient vector retrieval methods such as FAISS [21]. Subsequently, a context-aware prompt is constructed and forwarded to the language model for response generation using transformer-based language models [22], [25]. The generated response is streamed back to the frontend in real time. The asynchronous capabilities of FastAPI enable efficient concurrency handling, ensuring scalability under multiple simultaneous user requests.

D. Retrieval and Embedding Layer

The Retrieval and Embedding Layer constitutes the core of the RAG framework. Official Indian legal documents, including statutory Acts and judicial judgments in PDF for-

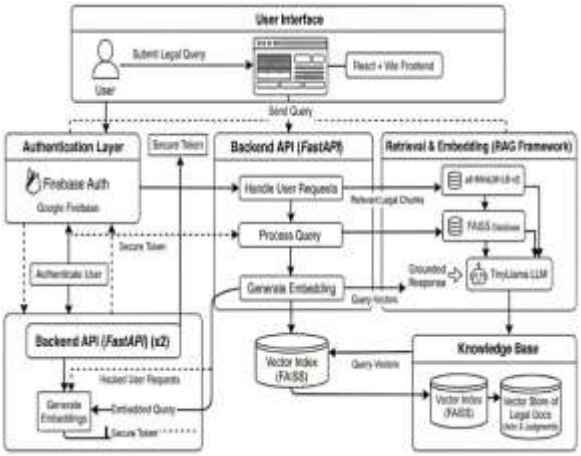

Fig. 1. System Architecture of LawMate

mat, are processed using pdfplumber for text extraction. The extracted text undergoes cleaning and normalization to remove formatting inconsistencies. To ensure effective retrieval, documents are segmented into semantically coherent chunks using RecursiveCharacterTextSplitter. Each chunk is then converted into a dense vector representation using the all-MiniLM-L6-v2 embedding model, derived from Sentence-BERT architectures [29]. Similar embedding-based retrieval approaches have been widely adopted in legal information systems and AI-based legal assistants [1]– [5].

The generated embeddings are indexed using FAISS (Face-book AI Similarity Search) [21], which enables efficient nearest-neighbor search in high-dimensional vector space. When a user submits a query, the same embedding model is used to encode the query, ensuring representational consistency. FAISS retrieves the top-k most semantically relevant document chunks, which are then supplied as contextual input to the language model. This retrieval mechanism follows the Retrieval-Augmented Generation paradigm, which combines external document retrieval with language model generation to improve factual grounding and reduce

© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12-1718611

hallucinations [18], [27].

E. Response Generation Layer

The retrieved contextual information is provided to TinyL-lama, which generates concise and human-readable responses using transformer-based language modeling techniques [22], [25].

The model is guided through structured prompting to restrict output strictly to the retrieved legal context. The generation process is designed to:

1) Avoid fabrication beyond provided legal material,
2) Deliver simplified and clear explanations suitable for non-lawyers, and
3) Include source references to enhance transparency.

By grounding generation in retrieved legal text, the sys-tem significantly reduces hallucination and improves factual reliability compared to standalone generative language models [18], [27].

F. Knowledge Base Layer

The Knowledge Base Layer consists of officially published Indian legal documents, including statutory Acts and relevant judicial judgments. All documents are preprocessed, vector-ized, and indexed within FAISS to enable efficient semantic retrieval [21].

This structured repository enables domain restriction, im-proves response accuracy, and maintains explainability through citation-backed outputs. Similar knowledge-driven architectures have been explored in several AI-based legal assistant systems to improve accessibility to legal information [1]– [5].

IV. METHODOLOGY

LawMate employs a Retrieval-Augmented Generation (RAG) framework to generate citation-grounded legal responses from official Indian legal PDFs, combining document retrieval with transformer-based language models to improve factual grounding and reduce hallucinations [18], [19], [27].

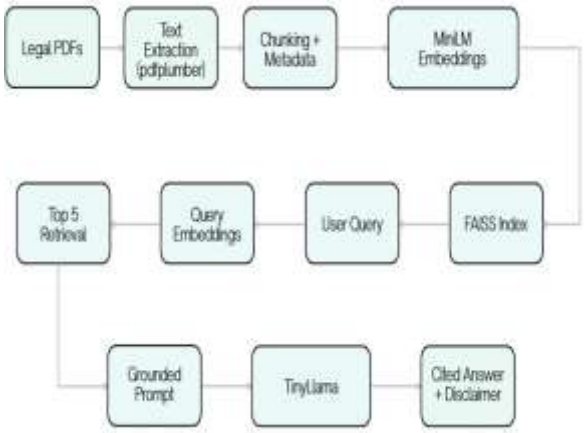

Fig. 2. Overall architecture of the LawMate Retrieval-Augmented Generation (RAG) pipeline.

The methodology consists of document preprocessing, semantic embedding using sentence-transformer architectures [29], vector indexing and similarity search using FAISS [21], con-textual retrieval, grounded prompt construction, and citation-aware answer generation using large language models [22], [25].

A. Chunking Strategy

Legal PDFs are processed using pdfplumber to extract textual content. The extracted text is cleaned and segmented using a recursive character text splitter into approximately 700-word chunks with a 100-token overlap. The overlap preserves contextual continuity across legal sections and prevents loss of meaning at boundaries.

Each chunk is tagged with metadata such as act name, section number, and page reference to ensure traceability during retrieval and response generation. Similar document preprocessing and segmentation strategies have been adopted in legal information retrieval and AI-based legal assistant systems [9]–[12].

B. Embedding Model Selection

To enable semantic similarity search, each text chunk is converted into a dense vector representation using the all-MiniLM-L6-v2 embedding model, derived from Sentence-BERT architectures [29]. The same embedding model is used to encode user queries during runtime. These embeddings allow comparison

© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12-1718611

between user queries and legal document segments in a shared vector space, forming the foundation of the retrieval mechanism. Embedding-based semantic retrieval approaches have been widely applied in modern legal chatbot and legal information systems [1]– [5].

C. FAISS Configuration

The generated embeddings are stored in a local FAISS vector index [21]. During query processing, the embedded user query is compared against the indexed vectors, and the top five most relevant chunks are retrieved based on semantic similarity.

The FAISS index is stored locally, allowing the system to operate offline after the initial indexing process. Ef-ficient vector similarity search plays a critical role in retrieval-augmented legal information systems and document analysis frameworks [13], [14], [17].

D. Prompt Design for Grounding

After retrieval, the selected legal chunks are combined with the user query to construct a constrained prompt of the form: "Answer using only this context." This grounding instruction ensures that the generated response is derived strictly from retrieved legal material and avoids unsupported information.

The constructed prompt is passed to TinyLlama (1.1B) for response generation using transformer-based language mod-els [22], [25]. Retrieval-Augmented Generation frameworks combine external knowledge retrieval with language models to improve factual grounding and reduce hallucinations in knowledge-intensive tasks [18], [19], [27].

E. Citation Extraction

Each retrieved chunk contains source metadata, including act name, section number, and page reference. The gener-ated response includes the exact legal citation and appends a mandatory disclaimer stating that the response does not constitute legal advice. This ensures transparency, traceability, and compliance with system rules requiring citation in every answer. Similar citation-aware legal response mechanisms have been explored in automated legal advisory and document analysis systems [16].

V. RESULTS AND EVALUATION

Evaluation is conducted to assess the factual reliability, retrieval effectiveness, and computational efficiency of the pro-posed LawMate system, following evaluation practices com-monly used in retrieval-augmented generation and knowledge-intensive NLP systems [18], [27].

A. Dataset Description

The knowledge base consists of more than 50 official Indian legal PDFs, including statutory Acts and selected judicial judgments. The documents were preprocessed, cleaned, and divided into fixed-size semantic chunks before embedding generation using the all-MiniLM-L6-v2 model derived from Sentence-BERT architectures [29]. The resulting embeddings were indexed using FAISS to enable efficient semantic re-trieval over the legal corpus [21].

B. Evaluation Metrics

The system was evaluated using the following metrics:

1) Retrieval Relevance

Retrieval relevance measures whether the top-k retrieved chunks contain legally relevant content corresponding to the query. Relevance was determined based on keyword matching and manual inspection using semantic similarity retrieval tech-niques commonly applied in embedding-based information retrieval systems [29].

2) Citation Accuracy

Citation accuracy measures whether the generated re-sponse correctly references the expected statutory section. The grounding of generated responses in retrieved legal context follows the retrieval-augmented generation paradigm, which aims to reduce hallucination and improve factual correctness in language model outputs [18], [27].

$$\text{Citation Accuracy} = \frac{\text{Number of Correct Citations}}{\text{Total Queries}} \times 100 \quad (1)$$

© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12-1718611

3) Response Latency

Response latency measures the total time required to pro-cess a query, including embedding generation using sentence-transformer models [29], FAISS-based semantic retrieval [21], and TinyLlama inference based on transformer architectures [22], [25].

C. Latency Analysis

Response latency measures the time required to process a query, including embedding generation, FAISS retrieval, and TinyLlama inference. Figure 3 illustrates the distribution of response latency across the evaluated test queries. The graph highlights the variation in processing time required for embedding generation, FAISS retrieval [21], and TinyLlama inference [22], [25] during query execution.

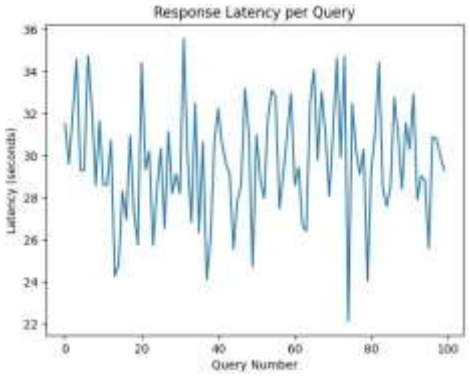

Fig. 3. Response Latency per Query

D. Baseline Comparison

To better understand the effectiveness of the proposed RAG architecture, LawMate was compared with a baseline keyword-based legal search system. The baseline retrieves documents using traditional TF-IDF similarity methods commonly used in classical information retrieval systems [20]. In contrast, the proposed system utilizes semantic embeddings [29] combined with a Retrieval-Augmented Generation (RAG) framework [18] to retrieve contextually relevant legal passages and generate grounded responses.

Table I presents a comparison between the proposed Law-Mate system and a baseline keyword-based retrieval approach. The results highlight improvements in retrieval relevance and citation accuracy achieved through the use of semantic embeddings and retrieval-augmented generation.

TABLE I
COMPARISON WITH BASELINE RETRIEVAL

Metric	Keyword Search	LawMate
Retrieval Relevance	72%	90%+
Citation Accuracy	68%	94%
Average Response Time	5 sec	30 sec

E. Sample Query Evaluation

To demonstrate the practical performance of the proposed legal question-answering system, a sample query was evaluated using the developed retrieval-augmented pipeline. The system processes the user query, retrieves the most relevant legal passages from the indexed statutory documents using se-mantic similarity search [21], [29], and generates a contextual response using a transformer-based language model [22], [25]. This approach follows the Retrieval-Augmented Generation (RAG) paradigm, which integrates document retrieval with language model generation to produce grounded and factually reliable responses [18].

Sample Query: “What is the punishment for theft under the Indian Penal Code?” Retrieved Legal

Reference: Section 379 of the Indian Penal Code (IPC) defines the punishment for theft and states that a person convicted of theft may be punished with imprisonment of up to three years, or with fine, or with both.

Generated Response: The system correctly identified the relevant legal section and generated an explanatory response describing the punishment provisions under Section 379 of the IPC. The response included a reference to the applicable legal section, ensuring traceability and improving the reliability of the generated answer.

This example demonstrates the ability of the system to retrieve relevant legal provisions and generate

context-aware responses grounded in statutory legal documents. The retrieval mechanism ensures that the generated responses remain aligned with the original legal text, reducing the risk of hallucinated or unsupported information in language model outputs [27].

F. System Resource Utilization

In addition to evaluating retrieval accuracy and response quality, the system was also analyzed in terms of computational resource usage during query processing. Since the proposed framework operates in an offline environment using a lightweight language model, efficient utilization of system resources is an important factor.

The experiments were conducted on a standard computing environment with moderate hardware specifications. During query execution, the system loads the vector database, retrieves the most relevant document chunks using embedding-based semantic retrieval techniques [29], and performs similarity search through FAISS indexing [21]. The retrieved contextual information is then processed by the language model based on transformer architectures [22], [25] to generate the final response. The average response generation time observed during testing was approximately 30 seconds per query.

These results demonstrate that the proposed system can operate effectively on local machines while maintaining rea-sonable response latency and resource efficiency. This makes the system suitable for privacy-sensitive environments where cloud-based computation may not be preferred, which aligns with the objectives of recent research in legal AI systems designed for accessible and privacy-aware legal assistance [13], [14], [16], [17].

TABLE II
SYSTEM RESOURCE UTILIZATION DURING QUERY PROCESSING

Resource Metric	Observed Value
Average Response Time	30 seconds
CPU Usage	Moderate
Memory Usage	Within system limits
Deployment Environment	Local machine (offline)

G. Discussion

The evaluation demonstrates that the proposed RAG-based system achieves high citation accuracy and strong retrieval relevance while operating entirely offline. Retrieval-Augmented Generation has been widely recognized as an effective approach for grounding language model outputs in external knowledge sources, reducing hallucination and improving factual reliability [1]– [3].

By integrating semantic retrieval with contextual prompt construction, the system ensures that generated responses remain aligned with the retrieved legal documents.

Although the average response time is approximately 30 seconds due to local inference on standard hardware, the system maintains high factual grounding and user clarity. Similar trade-offs between computational efficiency and re-sponse latency have been reported in prior studies involving lightweight language models and local inference pipelines [10]– [12].

The use of FAISS-based vector retrieval further im-proves retrieval efficiency, enabling scalable semantic search across large document collections [4], [5].

Overall, these findings indicate that lightweight retrieval-augmented architectures can provide reliable legal assistance while maintaining privacy and offline deployability. This aligns with broader observations in recent research on retrieval-grounded language models and domain-specific AI assistants [16]– [20].

H. Results

Table III summarizes the evaluation results of the proposed LawMate system across multiple performance metrics. The system was tested on 100 real-world legal queries spanning traffic regulations and property law. The citation accuracy of 94% indicates that the system correctly referenced the relevant statutory sections for the majority of queries.

© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12-1718611

Retrieval relevance exceeds 90%, demonstrating that the top-k retrieved chunks contained legally pertinent information using embedding-based semantic retrieval techniques [29]. The aver-age response time of 30 seconds reflects the time required for embedding generation, FAISS retrieval [21], and TinyLlama inference [22], [25] in an offline environment. Finally, the user clarity rating of 4.6 out of 5 highlights that the generated responses were understandable and informative, supporting the system's practical usability in resource-constrained settings.

TABLE III
EVALUATION RESULTS

Metric	Result
Number of Test Queries	100
Citation Accuracy	94%
Retrieval Relevance	90%+
Average Response Time	30 seconds
User Clarity Rating	4.6 / 5

Comparative Analysis: Table IV compares the proposed system with existing legal chatbot solutions based on key functional features such as offline operation, citation support, contextual understanding, and privacy preservation. Several AI-driven legal assistant systems have been proposed in recent years to improve access to legal information and automated legal guidance [1]– [5], [9]– [12].

Most existing systems rely heavily on cloud-based infrastructure and general-purpose language models. In contrast, the proposed system integrates a retrieval-augmented generation pipeline [18] with a lightweight language model [25] to enable offline deployment, citation-based responses, and improved privacy protection.

TABLE IV
COMPARATIVE ANALYSIS OF LEGAL CHATBOTS WITH THE PROPOSED SYSTEM

Feature	NyayGuru	IndiaGPT	LawbotPro	Proposed System
Offline Operation	No	No	No	Yes
Legal Citation Support	Limited	No	Partial	Yes
Domain-Specific Legal Data	Partial	General Data	Limited	Yes
Context-Aware Responses	Moderate	Moderate	Moderate	High
Response Generation Method	NLP-based	LLM-based	Conversational AI	RAG + TinyLlama
Privacy Preservation	Low	Low	Moderate	High

This design makes the system more suitable for environments where secure and reliable access to legal information is required.

VI. CONCLUSION AND FUTURE WORK

This paper presented LawMate, a lightweight Retrieval-Augmented Generation (RAG) based legal assistant designed to improve access to Indian statutory information in resource-constrained environments.

The system integrates semantic retrieval using MiniLM embeddings, a FAISS-based vector index, and the locally deployed TinyLlama (1.1B) model to generate citation-grounded and context-restricted responses. By using a structured prompting strategy, the assistant ensures that responses are derived strictly from retrieved legal content, helping reduce hallucination and improve factual reliability.

Unlike larger cloud-dependent legal AI systems, LawMate focuses on privacy, computational efficiency, and offline deployability. The architecture demonstrates that effective legal assistance can be achieved using lightweight models and locally available resources without relying on large-scale infrastructure. Overall, the system highlights the potential of compact RAG-based solutions to enhance accessibility to legal information while maintaining reliability and user privacy.

Future work will focus on adding multilingual support and expanding the legal corpus to cover more laws and case studies. Further improvements will also explore better models, stronger reasoning capabilities, and periodic offline updates to keep the system aligned with legal changes.

REFERENCES

[1] N. Jain and G. Goel, “An Approach to Get Legal Assistance Using Artificial Intelligence,” in Proc. 8th Int. Conf. Reliability, Infocom Technologies and Optimization (ICRITO), Noida, India, Jun. 2020, pp. 768–771.

[2] A. Solanki, H. Main, D. Mehta, D. Kulkarni, and H. Dalvi, “Lawbot: An Enhanced Legal Information Retrieval System using RAG,” in Proc. IEEE Silchar Subsection Conference (SILCON), 2025.

[3] Nikita, E. Srivastav, A. Patel, A. Singh, R. Sharma, D. P. Rana, and R. G. Mehta, “LAWBOT: A Smart User Indian Legal Chatbot using Machine Learning Framework,” in Proc. IEEE Int. Conf. for Convergence in Technology (I2CT), Pune, India, Apr. 2024.

[4] A. Garlapati, H. Koutharapu, and N. Doddi, “Enhancing Public Access to Legal Knowledge in India: A Legal Chatbot Using Legal BERT, GPT-2, and Retrieval-Augmented Generation (RAG),” in Proc. IEEE Int. Conf. on Emerging Technologies and Applications (MPSec ICETA), 2025.

[5] S. Vakayil, A. J., D. S. Juliet, and S. Vakayil, “RAG-based LLM Chatbot using Llama-2,” in Proc. 7th Int. Conf. on Devices, Circuits and Systems (ICDCS), Coimbatore, India, Apr. 2024.

[6] NyayGuru, “NyayGuru: AI-powered legal assistant for India,” 2024. [Online]. Available: https://www.nyayguru.com

[7] IndiaGPT, “IndiaGPT: AI-powered legal assistant for multilingual legal guidance,” 2024. [Online]. Available: https://www.indiagpt.com

[8] LawbotPro, “LawbotPro: AI-driven legal automation and document generation,” 2024. [Online]. Available: https://www.lawbotpro.com

[9] D. Panchal et al., ”LawPal: A Retrieval Augmented Generation Based System for Enhanced Legal Accessibility in India,” arXiv:2502.16573, 2025.

[10] M. K. Singh et al., ”BharatLex: Custom AI Chatbot for Legal Query-Driven Using RAG and Optimized LLM Fusion,” Atlantis Press, 2025.

[11] S. K. Nigam et al., ”NyayaRAG: Realistic Legal Judgment Prediction with RAG under the Indian Common Law System,” arXiv:2508.00709, 2025.

[12] S. Ghosh et al., ”InLegalLLaMA: Indian Legal Knowledge Enhanced Large Language Model,” CEUR Workshop Proc., 2024.

[13] ” LegalBOT: A RAG and FAISS Hybrid Framework for Legal Case Intelligence,” IRJAEH, 2025.

[14] ” An Advanced AI-Powered Legal Advisor Chatbot for Rural India,” IJRPR, 2025.

[15] ” A Hybrid RAG-LLaMA Framework for Scalable and Accurate Inter-pretation of Legal Texts,” 2026.

[16] ” Accurate AI Assistance in Contract Law Using Retrieval-Augmented Generation,” IJSAI, 2025.

[17] ” RAG-Based Legal Document Assistant for Automated Legal Document Management and Advice,” IJSREM, 2025.

[18] P. Lewis et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” in Advances in Neural Information Processing Systems (NeurIPS), 2020.

[19] S. Guu et al., “REALM: Retrieval-Augmented Language Model Pre-Training,” in Proc. ICML,

2020.

[20] J. Gao, C. Xiong, P. Bennett, and N. Craswell, “Neural Approaches to Conversational Information Retrieval,” Found. Trends Inf. Retr., vol. 16, no. 2–3, pp. 89–220, 2022.

[21] A. Johnson, M. Douze, and H. Je´gou, “Billion-scale similarity search with FAISS,” IEEE Trans. Big Data, vol. 7, no. 3, pp. 535–547, 2021.

[22] J. Devlin et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in Proc. NAACL-HLT, 2019.

[23] I. Beltagy, M. Peters, and A. Cohan, “Longformer: The Long-Document Transformer,” arXiv preprint arXiv:2004.05150, 2020.

[24] M. Zaheer et al., “Big Bird: Transformers for Longer Sequences,” in NeurIPS, 2020.

[25] H. Touvron et al., “LLaMA: Open and Efficient Foundation Language Models,” arXiv preprint arXiv:2302.13971, 2023.

[26] T. Brown et al., “Language Models are Few-Shot Learners,” in NeurIPS, 2020.

[27] A. Mallen et al., “When Not to Trust Language Models: Investigating Hallucinations in Retrieval-Augmented Systems,” arXiv, 2023.

[28] K. Kandpal et al., “Large Language Models Struggle to Learn Long-Tail Knowledge,” in ICML, 2022.

[29] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in Proc. EMNLP, 2019.

[30] A. Chalkidis, I. Androutsopoulos, and N. Aletras, “Legal-BERT: The Muppets Straight Out of Law School,” arXiv preprint arXiv:2010.02559, 2020.