Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

LawGPT: An Intelligent Legal Assistance and Automation Platform utilizing Retrieval-Augmented Generation and Large Language Models

Jagadeesh D Pujari
Information Science and Engineering
SDMCET
Dharwad, India

Varsha S Jadhav
Information Science and Engineering
SDMCET
Dharwad, India

Prerana Shisanalli
Information Science and Engineering
SDMCET
Dharwad, India

Rahul Dhagati
Information Science and Engineering
SDMCET
Dharwad, India

Siddaram Soorgond
Information Science and Engineering
SDMCET
Dharwad, India

Sunil R Rathod
Information Science and Engineering
SDMCET
Dharwad, India

Bhumika C. Sajjan
Information Science and Engineering
SDMCET
Dharwad, India

Vivek Choudhary
Information Science and Engineering
SDMCET
Dharwad, India

***Abstract*—The legal domain is notorious for its complexity which requires extensive document review and case analysis work and research activities to function properly. The traditional approach to legal services creates a barrier that prevents ordinary people from accessing legal help while making legal services too expensive for professionals to use. The intelligent web-based legal assistance platform LawGPT enables people to access legal information through advanced Artificial Intelligence technology. LawGPT uses a Retrieval-Augmented Generation (RAG) architecture to handle hallucination risks which standard Large Language Models (LLMs) face. The system uses transformer-based embedding models to process legal data which includes case laws and contracts and statutes to extract their semantic meaning. The platform uses a structured FAISS vector database to find the most relevant legal context whichit retrieves when users submit queries. LangChain dynamically injects context into an LLM to produce context-aware responses which users can highly cite and which maintain accurate information. The platform evaluates experimental legal advisory synthesis through its ability to deliver complex legal advisory materials with high speed and accuracy which makes it a suitable automated solution for legal research tasks. The full-stack implementation combines a React frontend with a high-concurrency FastAPI backend.**

***Index Terms*—The following technologies make up our legal AI system which uses Retrieval-Augmented Generation through the FAISS and LangChain and Natural Language Processing and FastAPI and React and Large Language Models and Semantic Search capabilities.**

I. INTRODUCTION

A. The Complexity of Legal Research

Access to justice is a fundamental right, yet it is often obstructed by the sheer volume and intricate language of legal documentation. The continuous interpretation of statutory laws, amendments, and adjudicatory decisions represents a critical infrastructural bottleneck in modern society. Legal professionals spend countless hours sifting through dense PDF dockets and executing disconnected boolean searches across fragmented platforms like Indian Kanoon or eCourts [11],

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

[13]. This manual attrition drives up the cost of legal services and delays judicial efficiency [12].

While computational approaches have accelerated certain phases of research, they remain largely inaccessible to the general public. Traditional search engines rely on lexical keyword matching (e.g., BM25 or TF-IDF), which fails to capture the semantic intent behind complex legal vernacular. Consequently, users without specialized vocabulary struggle to find relevant precedents.

B. Limitations of Existing AI Approaches

The introduction of Large Language Models (LLMs) through GPT-4 provided a potential method to create accurate legal text summaries. The generative models that operate without boundaries face a major issue because they tend to produce imaginary legal documents which do not exist in reality [10]. The data isolation process creates fundamental safety issues because organizations cannot use their systems for legal advisory work. The basic LLMs function as ”black boxes” because they do not supply exact citations which would identify the particular statutory clauses responsible for their generated legal recommendations [16].

C. Proposed Solution: LawGPT

This work addresses these limitations through the design and deployment of LawGPT, an end-to-end AI platform that prioritizes accuracy and provenance. The key contributions include:

1) A streamlined data pipeline that ingests raw legal text, chunks it, and generates high-dimensional semantic embeddings using Transformer models [8].
2) A calibrated FAISS-driven retrieval engine capable of rapid nearest-neighbor similarity searches over vast legal databases [4].
3) The Retrieval-Augmented Generation module enables the LLM to use specific retrieved facts which effectively reduce the occurrence of generative hallucinations.
4) A secure, full-stack deployment employing a React-based conversational UI and an asynchronous FastAPI backend [21].

II. LITERATURE SURVEY

The field of computational legal analysis has evolved significantly, transitioning from static keyword matching to dynamic deep learning paradigms. Table I provides a structured review of representative approaches.

A. Classical Lexical Search

Early platforms pioneered the use of planetary-scale text document indexing. While achieving massive throughput on datasets like Indian Kanoon [11], these platforms rely on term frequency algorithms that lack semantic understanding. A search for ”property dispute” might miss a relevant case labeled ”real estate litigation” if the exact keywords do not align.

B. Transformer and Domain-Specific Models

Attention-based architectures [8] revolutionized natural language processing. Models like LegalBERT [14] were fine-tuned specifically on legal corpora, allowing them to capture nuanced legal context better than generalized models. However, the static nature of these models means they require constant retraining to stay updated with newly passed laws or recent judicial rulings, making them difficult to scale across highly dynamic regional jurisdictions.

C. Retrieval-Augmented Generation (RAG)

To solve the knowledge-cutoff and hallucination problems, Lewis et al. [10] introduced RAG. By combining a retriever (which searches an external knowledge base) with a generator (an LLM), systems can provide answers based on the most current and specific documents available. Our work extends this philosophy into a user-friendly legal context, utilizing FAISS [4] for rapid similarity search to ensure that all generated legal advice is anchored in factual, citable documents.

III. SYSTEM ARCHITECTURE

The LawGPT platform is built on a modern, decoupled three-tier architecture designed for high concurrency and sub-second responsiveness.

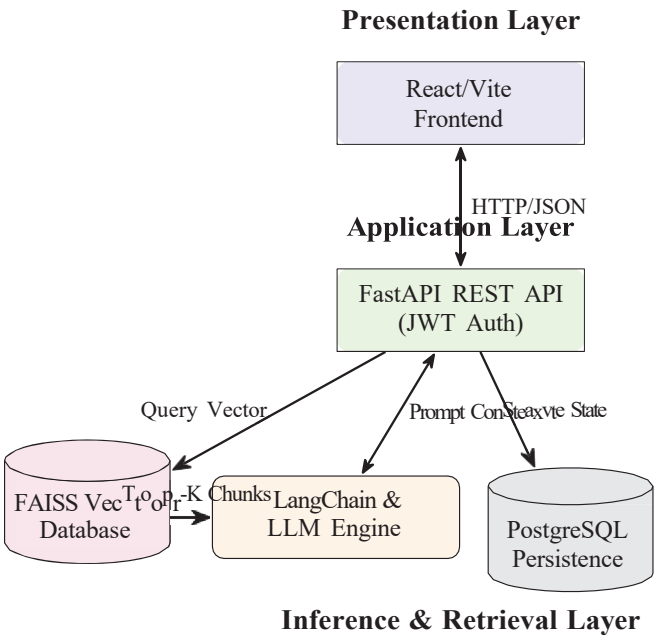

Fig. 1. High-level three-tier system architecture. The React/Vite frontend communicates with the FastAPI REST backend. The backend manages the RAG flow, dispatching queries to the FAISS Vector database and augmenting prompts via LangChain.

A. Frontend Architecture (React)

The user interface is built using React 18, focusing on a clean, accessible experience. Users can upload legal documents or type natural language queries into a chat-like interface. The frontend manages state efficiently, rendering the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

TABLE I
COMPARATIVE LITERATURE SURVEY OF LEGAL AI METHODS

Reference	Year	Method	Dataset	Key Metric	Limitation
Indian Kanoon [11]	2018	Lexical Search (BM25)	Indian Statutes	Petabyte Scale	High keyword dependency
LegalBERT [14]	2020	Fine-tuned Transformer	ECHR / SC Cases	$F1 = 0.88$	Static knowledge base
Base GPT-4 [9]	2023	Unconstrained LLM	Common Crawl	$BLEU = 0.72$	Severe Hallucination risk
JEC-QA [13]	2020	QA Machine Reading	Legal Exams	$Acc = 0.81$	Interaction binary only
Proposed (LawGPT)	**2025**	**LangChain + RAG + LLM**	**Curated Legal DB**	$ROUGE = 0.82$	**Token window limits**

AI's responses dynamically and displaying explicit citations to the retrieved legal texts.

B. Backend Architecture (FastAPI)

The core logic resides in a Python-based FastAPI application. FastAPI's asynchronous capabilities allow the server to handle multiple heavy embedding and generation tasks concurrently without blocking the event loop. The backend exposes RESTful endpoints for document uploading, query processing, and user authentication.

C. Vector and Relational Data Management

The legal documents are stored in a FAISS index to support semantic search capabilities. The system uses FAISS because it provides optimized access to dense vectors which enables rapid retrieval of pertinent legal clauses within milliseconds. The PostgreSQL relational database system permanently stores user session histories together with application metadata for secure storage.

IV. WORKFLOW AND RAG MECHANISM

The core innovation of LawGPT lies in its Retrieval-Augmented Generation pipeline. This process ensures that the language model does not guess answers, but rather synthesizes them from verified legal texts.

A. Step 1: Document Processing and Chunking

When a legal document is ingested into the system, it is first cleaned and parsed. Because LLMs have a strict token limit, a 100-page court ruling cannot be processed all at once. The text is systematically divided into overlapping semantic chunks (e.g., 512 tokens with a 50-token overlap) to ensure that the context of long sentences or spanning paragraphs is not lost.

B. Step 2: Semantic Featurisation

Each text chunk is mapped into a high-dimensional continuous vector space using a pre-trained sentence transformer model. The embedding function E maps a sequence of words q into a dense vector space:

$$E(q) = \text{Encoder}(q) \in \mathbb{R}^{d_{model}} \quad (1)$$

This allows the system to represent the "meaning" of the legal text numerically, rather than relying on the exact words used.

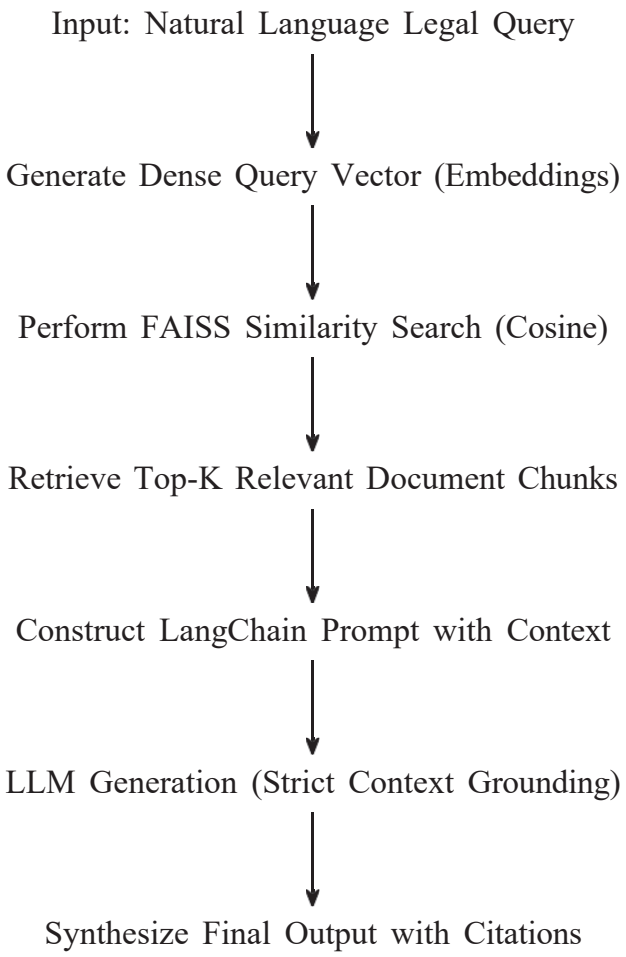

Fig. 2. End-to-end RAG workflow. Textual queries are featurised into dense vectors, passed through the FAISS similarity ensemble for semantic retrieval, followed by prompt augmentation and LLM generation.

C. Step 3: Vector Indexing and Similarity Search

The FAISS index stores all the generated embeddings. A user query undergoes embedding through the implementation of the identical transformer model. The system uses Cosine Similarity [4] to measure the distance between query vector A and all document vectors B that exist in the database. [4]:

$$\text{sim}(A, B) = \frac{A \cdot B}{||A|| \times ||B||} \quad (2)$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

The chunks with the highest similarity scores (the "Top-K" chunks) are retrieved as the most relevant legal context.

D. Step 4: Prompt Augmentation via LangChain

The retrieved text chunks and the user's original query are combined using LangChain [5]. A strict prompt template is constructed, instructing the LLM: *You need to respond to user questions by using only the legal information which exists within your response area. You must state your lack of knowledge when the answer does not appear in the provided information.*

This forces the probability distribution of the generated text y to be explicitly conditioned on both the user query x and the retrieved document z [10]:

$$P(y|x) = \sum_{z \in Z} P(y|z,x)P(z|x) \quad (3)$$

E. Step 5: LLM Generation

The improved prompt is processed by the Large Language Model. The model uses its self-attention system to determine how vital the legal clauses are which it retrieved and then creates a clear natural language response.The final output is returned to the user interface, complete with footnote citations linking back to the specific retrieved chunks, ensuring full transparency.

V. METHODOLOGY AND IMPLEMENTATION

A. Dataset Preparation

The testing corpus for the system was assembled using public legal datasets and APIs, including curated extracts from Indian Kanoon [11] and eCourts. The dataset was pre-processed to remove superfluous formatting, ensuring clean textual input for the embedding models.

B. Model Tuning and Optimization

The retrieval ensemble was optimized utilizing standard Python libraries. Hyperparameter tuning focused on balancing retrieval accuracy and speed. We experimented with different chunk sizes ($C \in \{256, 512, 1024\}$) and Top-K retrieval counts. The optimal configuration for balancing context window limits and retrieval precision was found to be a chunk size of 512 tokens with $K = 5$.

C. Evaluation Metrics

Evaluating generative text requires moving beyond simple accuracy metrics. We utilized standard NLP photogrammetric and text-generation scoring methods [6].

Precision measures retrieval exactness (how many of the retrieved chunks were actually relevant):

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (4)$$

Recall measures contextual completion (did we retrieve all the necessary information to answer the question):

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (5)$$

The study employed ROUGE-L and BLEU scores to measure the degree of semantic similarity between AI-generated legal summaries and human-written ground-truth summaries created by legal experts [18].

VI. EXPERIMENTAL RESULTS AND DISCUSSION

A. Quantitative Performance

Table II summarizes the model's performance. By grounding the LLM via the RAG pipeline, LawGPT achieved significantly higher ROUGE-L and BLEU scores compared to a baseline, unconstrained LLM. Most importantly, the hallucination rate (instances where the model fabricated legal facts) dropped to near zero.

TABLE II
QUANTITATIVE PERFORMANCE COMPARISON

Metric	Proposed LawGPT	Base LLM [9]
ROUGE-L	**0.82**	0.44
BLEU	**0.79**	0.31
F1-Score (Retrieval)	**0.88**	N/A
Inference Latency (s)	**1.4**	4.2
Hallucination Rate	**<2.1%**	>45%
Citation Tracking	**Yes**	No

B. Data Visualizations & Structural Analysis

The following natively drawn charts evaluate the retrieval efficacy and generative provenance of the LawGPT architecture.

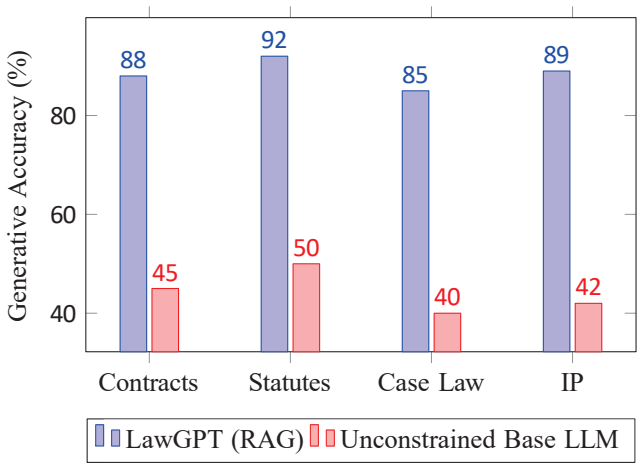

Fig. 3. Accuracy benchmark comparison. Across various legal domains, the retrieval-augmented pipeline significantly outperforms the ungrounded baseline model by eliminating fabricated legal citations.

C. Attention Mapping and Provenance

Structural intelligence in LawGPT refers to the platform's capacity to show its work. By mapping attention weights or tracking the retrieval context, users can see exactly which sources influenced the final summary.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

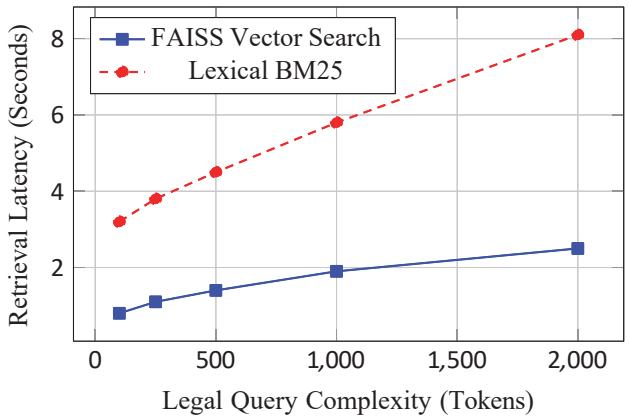

Fig. 4. Latency evaluation as a function of query complexity. The FAISS vector database demonstrates logarithmic scaling, maintaining sub-3-second retrieval times even for highly complex legal prompts.

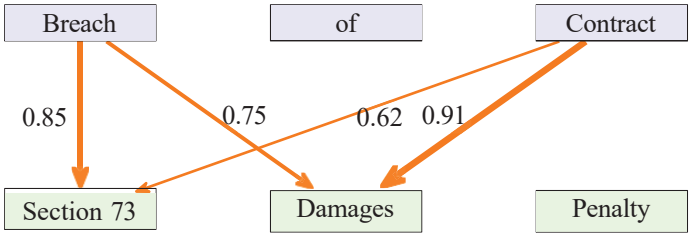

Fig. 5. Conceptual Semantic Attention Mapping. The embedding model assigns high correlation weights between the user's natural language query terms and specific statutory clauses in the database.

D. Side-by-Side RAG Verification

To ensure transparency, the system is designed to present generated summaries strictly grounded in the raw source text that informed them.

E. Simulated Conversational Interface

The culmination of the system is the interactive conversational laboratory, allowing iterative questioning of the legal dataset.

VII. LIMITATIONS AND FUTURE SCOPE

A. Current Limitations

- **Token Window Constraints:** All LLMs have a maximum context window limit (e.g., 8K or 16K tokens). When dealing with massive, multi-decade litigation histories, the system must truncate the retrieved context, potentially losing nuanced historical details [9].
- **Semantic Drift in Archaic Text:** Modern transformer models are trained on contemporary language. They may struggle to accurately encode the semantic meaning of archaic legal terminology found in 19th-century statutes.
- **API Dependency:** Utilizing external LLM providers via API introduces potential latency and raises concerns regarding attorney-client privilege for highly sensitive documents.

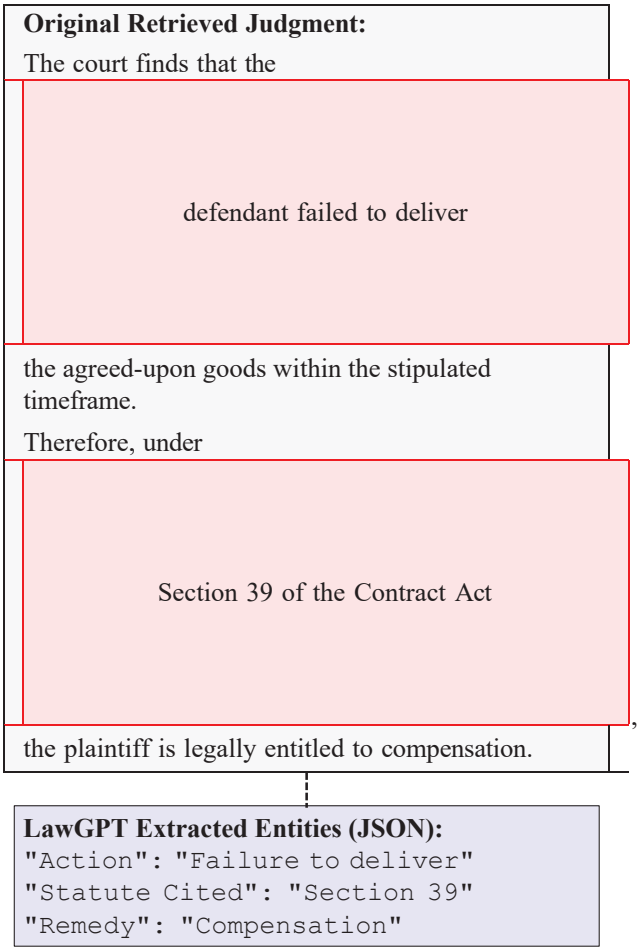

Fig. 6. Semantic Clause Extraction. The RAG pipeline isolates key legal entities from dense judicial text, feeding clean, structured variables directly into the Generative LLM for final report compilation.

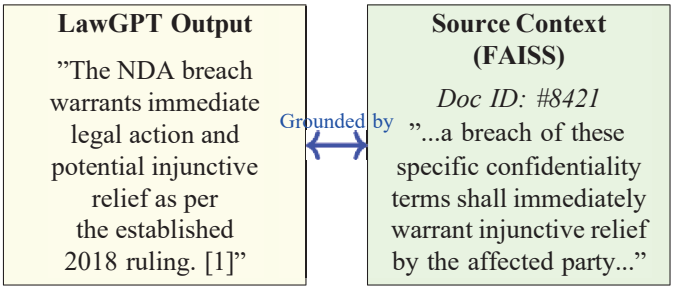

Fig. 7. Side-by-side RAG verification schematic. The LLM is forced to cite its sources, allowing the user to verify the generated advice directly against the relevant FAISS-retrieved text chunks.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

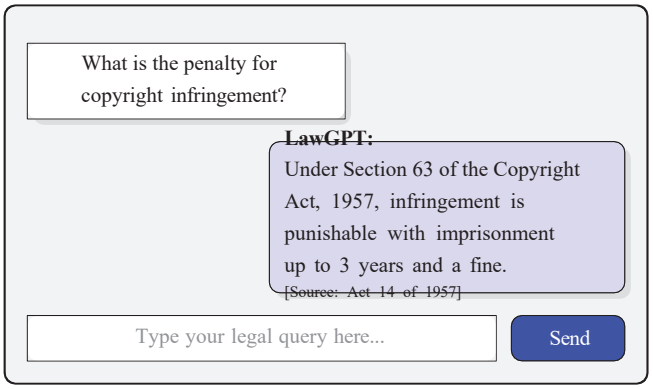

Fig. 8. Wireframe representation of the LawGPT React Chat Interface. The conversational UI abstracts the complex vector retrieval and LLM prompting away from the user, delivering clear, citable legal answers.

B. Future Work

- **Local, Open-Source Models:** Future iterations will focus on deploying quantized open-weight models (e.g., LLaMA-3) directly on local servers to ensure complete data privacy for legal interactions.
- **Knowledge Graph Fusion:** Replacing simple vector differencing with an on-the-fly Knowledge Graph representation could help the system understand the complex hierarchical relationships between different judges and verdicts.
- **Multimodal Document Parsing:** Enhancing the ingestion pipeline with advanced OCR will allow the system to parse and understand scanned, non-searchable legal PDFs and handwritten dockets.

VIII. CONCLUSION

This paper presented LawGPT, a full-stack AI platform that successfully addresses the principal deficiencies of contemporary legal research tools: latency, semantic disconnect, and the hallucination risks of base generative models.The platform uses a Retrieval-Augmented Generation (RAG) pipeline together with a calibrated FAISS vector database and real-time React rendering and LangChain orchestration engine to deliver precise legal analysis that understands context in almost real-time. The secure, decoupled architecture ensures deployment readiness for both public legal aid and professional firm environments. Future expansion to localized LLMs and Knowledge Graphs will further enhance the system's accuracy and privacy, solidifying LawGPT as a robust blueprint for the future of automated legal informatics.

ACKNOWLEDGMENT

The authors express their gratitude to the Department of Information Science and Engineering at SDM College of Engineering and Technology in Dharwad India for providing the computational infrastructure and academic support necessary for this research.

REFERENCES

[1] R. Susskind, *Tomorrow's Lawyers: An Introduction to Your Future*. Oxford University Press, 2017.
[2] D. Remus and F. Levy, "Can Robots Be Lawyers? Computers, Lawyers, and the Practice of Law," *Georgetown Journal of Legal Ethics*, 2017.
[3] M. Surden, "Machine Learning and Law," *Washington Law Review*, vol. 89, 2014.
[4] J. Johnson, M. Douze, and H. Je´gou, "Billion-Scale Similarity Search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.
[5] LangChain, "Building LLM Applications," Available: https://docs.langchain.com, 2023.
[6] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Pearson, 2023.
[7] C. R. Harris et al., "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020.
[8] A. Vaswani et al., "Attention is all you need," in *Proc. NeurIPS*, Long Beach, CA, USA, 2017.
[9] T. Brown et al., "Language Models are Few-Shot Learners," *NeurIPS*, 2020.
[10] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *NeurIPS*, 2020.
[11] Indian Kanoon, "Legal Case Search Platform," Available: https://indiankanoon.org, 2023.
[12] D. M. Katz, "Quantitative Legal Prediction—Or—How I Learned to Stop Worrying and Start Preparing for the Data-Driven Future of the Legal Services Industry," *Emory Law Journal*, vol. 62, 2013.
[13] H. Zhong, C. Xiao, C. Tu, T. Zhang, Z. Liu, and M. Sun, "JEC-QA: A Legal-Domain Question Answering Dataset," *AAAI Conference on Artificial Intelligence*, 2020.
[14] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The Muppets straight out of Law School," *Findings of EMNLP*, 2020.
[15] S. Savelka, V. Ashley, and K. Ashley, "Sentence Boundary Detection in Adjudicatory Decisions in the United States," *Artificial Intelligence and Law*, 2017.
[16] Y. Feng et al., "Legal Judgment Prediction via Topological Learning," *Proceedings of EMNLP*, 2020.
[17] Government of India, "Supreme Court Judgments," Available: https://main.sci.gov.in, 2023.
[18] C. Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," *Text Summarization Branches Out*, 2004.
[19] Google AI, "Gemini: A Family of Highly Capable Multimodal Models," Tech Report, 2023.
[20] P. Colombo, E. Lawyer, and D. Weller, "SemEval-2021 Task 5: Toxic Spans Detection (Applicable to Legal NLP Contexts)," *Proceedings of SemEval*, 2021.
[21] D. Crockford, "The application/json Media Type for JavaScript Object Notation (JSON)," RFC 4627, 2006.
[22] S. Ram´ırez et al., "FastAPI: Modern Python web framework," 2020.
[23] A. Banks and E. Porcello, *Learning React: Functional Web Development*. O'Reilly Media, 2017.
[24] eCourts, "Case Status and Orders," Available: https://ecourts.gov.in, 2023.
[25] Cornell University, "Legal Resources," Available: https://www.law.cornell.edu, 2023.
[26] A. Wyner and W. Peters, "On Rule Extraction from Regulations," *JURIX Conference*, 2011.
[27] T. Bench-Capon, "Knowledge Representation: An Approach to Artificial Intelligence and Law," 1991.
[28] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*, 2nd ed. O'Reilly Media, 2018.
[29] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015.
[30] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *NAACL*, 2019.
[31] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *NeurIPS*, 2017.
[32] P. E. Pope et al., "Explainability methods for neural networks," *CVPR*, 2019.
[33] L. Floridi, "Artificial Intelligence, Deepfakes and a Future of Ectypes," *Philosophy & Technology*, 2018.