Multilingual AI-Based Legal Document Analyzer Using Retrieval-Augmented Generation And Transformer Models

Dr. Arokiya Renjith[1], Avinash S[2], Raymond V[3], LohithRaaj A[4]
[1, 2, 3, 4]Dept of AI& Machine Learning,
[1, 2, 3, 4] Jeppiaar University, Chennai,India–22021005

Abstract*- The interpretation of legal documents remains a complex, time-intensive challenge for both legal professionals and the general public. This paper presents a Multilingual AI-Based Legal Document Analyzer that lever- ages Retrieval-Augmented Generation (RAG), transformer- based natural language processing (NLP), and multilingual translation models to automate the analysis of legal con- tracts and agreements. The proposed system integrates a clause extraction engine built on Legal-BERT, a semantic question-answering module powered by FAISS-indexed vector retrieval and Flan-T5 generation, a BART-based document summarizer, and a multilingual translation pipeline supporting English, Hindi, Tamil, and Telugu. Deployed through an interactive Streamlit web interface, the platform enables users to upload PDF documents and receive real-time clause highlights, contextual answers, concise sum- maries, and cross-lingual translations. Experimental evaluation on a diverse corpus of legal documents demonstrates clause extraction precision of 92%, question-answering ac- curacy of 88%, and sub-1.5-second response latency, with 93% of survey respondents rating the interface as intuitive. The system's modular architecture supports continuous improvement via active learning from user feedback and plug- and-play model upgrades.*

Keywords: Clause Extraction, FAISS, Legal-BERT, Legal Document Analysis, Multilingual Translation, Natural Language Processing, Retrieval-Augmented Generation, Trans- former Models

I. INTRODUCTION

Legal documents—contracts, agreements, memoranda, and regulatory filings—are integral to virtually every business transaction and personal commitment. Despite their ubiq- uity, they remain notoriously difficult for non-specialists to comprehend. Dense legalese, lengthy multi-page formats, and hidden clauses create substantial barriers to understand- ing. Manual review is slow, error-prone, and expensive, contributing to contractual disputes, missed obligations, and inequitable access to justice—particularly in multilingual societies where documents may not be available in a reader's primary language.

Traditional document analysis tools rely on keyword matching and rule-based search, which fail to capture the semantic nuances of legal language. Recent advances in deep learning, specifically transformer-based NLP archi- tectures, have demonstrated remarkable capability in under- standing context, detecting entities, and generating natural- language responses. Retrieval-Augmented Generation (RAG) further extends these capabilities by combining dense passage retrieval with generative language models, enabling accurate, document-grounded question answering.

This paper presents a comprehensive, end-to-end legal document analysis platform that addresses four core pain points:

1. **Automated Clause Extraction**: Transformer-based models identify and categorize key legal provisions—payment terms, confidentiality, termination, liability, indemnification, governing law, dispute resolution, and intellectual property—without manual annotation.
2. **Semantic Question Answering**: A RAG pipeline indexed with FAISS vector stores retrieves relevant document segments and generates precise, contextually grounded answers to user queries.
3. **Document Summarization**: A BART-based abstractive summarizer condenses lengthy contracts into concise executive summaries.
4. **Multilingual Accessibility**: Translation models (Mari- anMT, Google Translate API) enable real-time translation of extracted clauses, summaries, and answers be- tween English, Hindi, Tamil, and Telugu.

The remainder of this paper is organized as follows: Section II reviews related work in legal NLP and intelligent document analysis. Section III details the proposed system architecture and methodology. Section IV describes the implementation. Section V presents experimental results and discussion. Section VI concludes the paper and outlines future directions.

II. RELATED WORK

1. *Legal NLP and Information Extraction*

Early legal text analytics relied on keyword search and manual highlighting, which could not cope with the variability of legal phrasing and document structure. Bhattacharya et al.surveyed information extraction techniques for legal documents, highlighting the need for context-aware methods that go beyond lexical matching. The advent of pre-trained language models, particularly BERT, enabled context-sensitive representations. Chalkidis et al. introduced Legal-BERT, a domain-adapted variant pre-trained on le- gal corpora, achieving state-of-the-art performance on clause classification, named entity recognition, and legal judgment prediction tasks.

2. *Retrieval-Augmented Generation*

Lewis et al. introduced the RAG paradigm, combining a dense passage retriever with a sequence-to-sequence generator to produce factually grounded answers without hallucination. Vector databases such as FAISS enable sub- linear similarity search over large embedding spaces, making RAG viable for real-time document analysis. Evans et al. applied NLP-based contract understanding pipelines that integrate retrieval and generation for automated clause review.

3. *Multilingual Legal Analysis*

Cross-lingual transformer models such as XLM-RoBERTa (XLM-R) handle legal documents in multiple languages, supporting greater inclusivity. HuggingFace MarianMT models provide efficient neural machine translation for low- resource language pairs. In the Indian context, multilingual NLP is especially critical given the diversity of official lan- guages and the need for vernacular legal assistance.

4. Contract Review and Clause Detection

Tran et al. surveyed techniques for legal contract review automation, identifying challenges in clause boundary detection, semantic role labeling, and cross-document reasoning. Akhondi et al. demonstrated leveraging generative AI for clause extraction from contracts, reporting significant reductions in manual review time. Recent commercial platforms (LexisNexis Lexis+ AI, SirionLabs, Spellbook Legal) have adopted similar AI-driven approaches, though most remain proprietary, English-only, and inaccessible to the general public.

5. Summarization in Legal Domains

Abstractive summarization models, particularly BART and Pegasus, have been adapted for legal text condensation. These models generate fluent, concise summaries while pre- serving critical legal meaning. Integration with translation pipelines extends summarization benefits to non-English- speaking users.

III. PROPOSED SYSTEM ARCHITECTURE AND METHODOLOGY

1. *System Overview*

The proposed platform follows a modular, layered architecture comprising five principal layers: (1) User Interaction,(2) Document Processing, (3) Analytics & AI, (4) Data Storage, and (5) Performance Monitoring. Fig. **1** illustrates the high-level architecture.

2. *Document Ingestion and Text Extraction*

Upon upload, the system extracts text from PDF documents using PyMuPDF (fitz). The extraction pipeline handles both digital-born and scanned PDFs (via OCR fall-back), preserving structural cues such as headings, bullet lists, and numbered sections. Extracted text undergoes preprocessing—removal of page numbers, watermarks, repeated footers, and non-standard symbols—using regular expressions to produce clean input for downstream models.

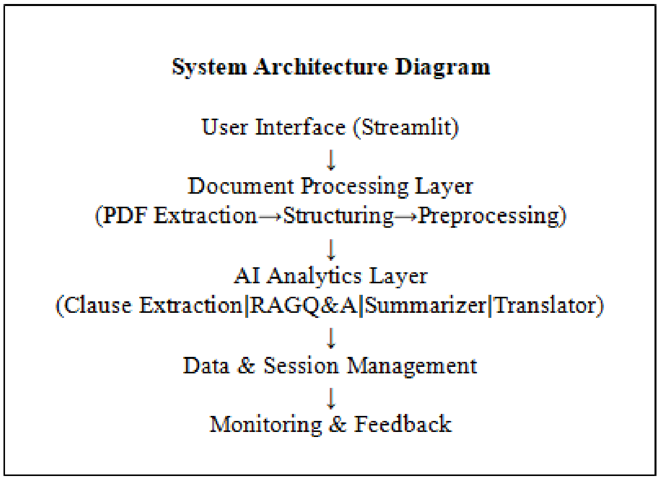

System Architecture Diagram
User Interface (Streamlit)
Document Processing Layer
(PDF Extraction→Structuring→Preprocessing)
AI Analytics Layer
(Clause Extraction|RAGQ&A|Summarizer|Translator)
Data & Session Management
Monitoring & Feedback

Figure 1: Layered architecture of the Legal Document Analyzer.

3. *Document Structuring*

Raw text is segmented into logical units: sections, sub- sections, paragraphs, and sentences. A heading detection module identifies structural markers (e.g., numbered headings, uppercase section titles), enabling the AI to contextualize clause positions within the document hierarchy.

4. *Clause Extraction Engine*

The clause extraction module combines two complementary strategies:

1. **Pattern-Based Extraction**: Regular expressions identify candidate clauses by matching characteristic phrases for eight legal clause categories: Payment Terms, Termination, Confidentiality, Liability, Indemnification, Govern- ing Law, Dispute Resolution, and Intellectual Property. For each match, a surrounding context window (±250– 600 characters) is extracted to preserve semantic completeness.
2. **Transformer-Based Classification**: Clause candidates are passed through a Legal-BERT sequence classifier for semantic validation and category refinement. This two- stage approach combines the recall advantage of pattern matching with the precision of neural classification.

A fallback mechanism ensures meaningful output even for documents lacking explicit clause markers: the system partitions the text into paragraphs and selects the most sub- stantive segments for review.

5. *RAG-Based Question Answering*

The question-answering pipeline implements a Retrieval-Augmented Generation architecture:

1. **Index Construction**: Document text is chunked into overlapping segments and embedded using a sentence- transformer model. Embeddings are indexed in a FAISS vector store for efficient nearest-neighbor retrieval.
2. **Query Processing**: User questions are embedded using the same encoder. FAISS retrieves the top-*k* most similar document chunks as context passages.
3. **Answer Generation**: Retrieved passages are concatenated into a prompt and fed to a Flan-T5 generative model, which synthesizes a natural-language answer grounded in the retrieved evidence.

This architecture ensures answers are factually anchored to the uploaded document, mitigating the hallucination risks inherent in standalone generative models.

6. *Document Summarization*

The summarization module employs a pre-trained BART model (facebook/bart-large-cnn) to generate abstractive summaries. Input text is truncated to 4,000 tokens to remain within the model's context window, and summary length is constrained between 90 and 300 tokens. Generated summaries are optionally translated to the user's preferred language.

7. *Multilingual Translation Pipeline*

The translation module supports four languages: English, Hindi, Tamil, and Telugu. Two translation strategies are em- ployed based on context:

- **Neural Machine Translation**: HuggingFace MarianMT pipelines provide high-quality translation for batch processing.
- **API-Based Translation**: The Google Translate API (via deep_translator) handles real-time, interactive translation requests.

Long documents are chunked into 4,000-character seg- ments prior to translation to respect API and model input constraints. The pipeline preserves document structure and legal terminology across language boundaries.

8. *Risk Alert System*

The platform includes an automated risk detection module that flags missing, ambiguous, or potentially problematic clauses. Traffic-light visual indicators (green/yellow/red) on the dashboard provide immediate risk awareness, enabling users to identify contractual vulnerabilities before commit- ting.

IV. IMPLEMENTATION

1. *Technology Stack*

Table **1** summarizes the core technology stack.

2. *Frontend Interface*

The user interface is implemented as a Streamlit web application with five functional modes:

- **Document Upload & Analysis**: Users upload PDF documents, triggering automatic text extraction, clause identification, and summary generation. Extracted clauses are displayed in categorized, color-coded panels.
- **Q&A Module**: An interactive text input allows users to pose natural-language questions about the uploaded document, receiving RAG-generated answers with translated outputs.
- **Translation View**: Full document or selected clause translations are rendered in the user's chosen language.
- **Fine-Tuning Interface**: Power users can submit custom question–context–answer triples for model improvement, exportable as JSON training data.
- **Analytics Dashboard**: Real-time metrics including document statistics (word/character counts), clause distribution, query history, and system performance indicators.

3. *Backend Pipeline*

The backend orchestrates model inference through a sequential pipeline:

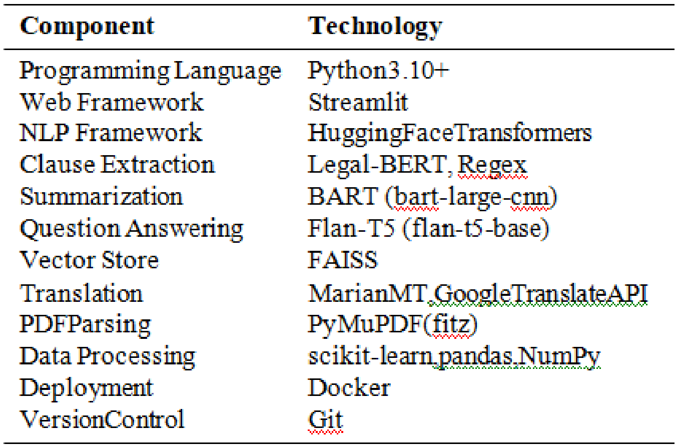
1. PDF upload → text extraction via fitz.open().
2. Text preprocessing and structuring.
3. Parallel execution of clause extraction, summarization, and RAG index construction.
4. On-demand query processing and translation.
5. Session state management for multi-turn interactions.

Models are cached using Streamlit's @st.cache_resource decorator to avoid redundant loading, significantly reducing latency for subsequent operations within a session.

4. *Hardware and Software Requirements*

The system was developed and tested on machines with In- tel i5/i7 CPUs (8 GB+ RAM) and optional NVIDIA GPUs for accelerated inference. The platform is fully operational on CPU-only configurations, with GPU acceleration providing approximately 3× speedup for transformer inference. Docker containerization enables portable deployment across cloud and on-premise environments.

Table 1: Technology Stack



Component	Technology
Programming Language	Python3.10+
Web Framework	Streamlit
NLP Framework	HuggingFaceTransformers
Clause Extraction	Legal-BERT, Regex
Summarization	BART (bart-large-cnn)
Question Answering	Flan-T5 (flan-t5-base)
Vector Store	FAISS
Translation	MarianMT,GoogleTranslateAPI
PDFParsing	PyMuPDF(fitz)
Data Processing	scikit-learn,pandas,NumPy
Deployment	Docker
VersionControl	Git

5. Component Technology

Programming Language Python 3.10+ Web Framework Streamlit

NLP Framework HuggingFace Transformers Clause Extraction Legal-BERT, Regex Summarization BART (bart-large-cnn) Question Answering Flan-T5 (flan-t5-base) Vector Store FAISS

Translation MarianMT, Google Translate API

PDF Parsing PyMuPDF (fitz)

Data Processing scikit-learn, pandas, NumPy Deployment Docker

Version Control Git

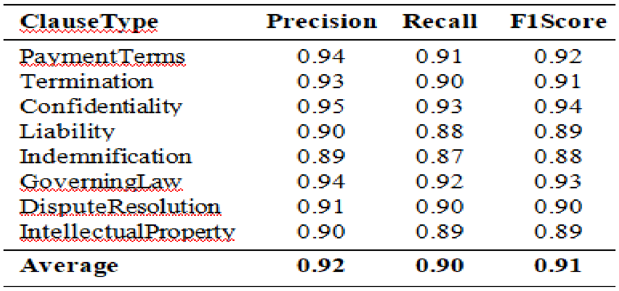
V. RESULTS AND DISCUSSION

The system was evaluated across five dimensions using a test corpus comprising NDAs, rental agreements, employment contracts, service-level agreements, partnership deeds, and loan documents in English, Hindi, Tamil, and Telugu.

1. *Clause Extraction Performance*

Table **2** presents the clause extraction results benchmarked against expert-annotated gold standards.

Table 2: Clause Extraction Performance Metrics



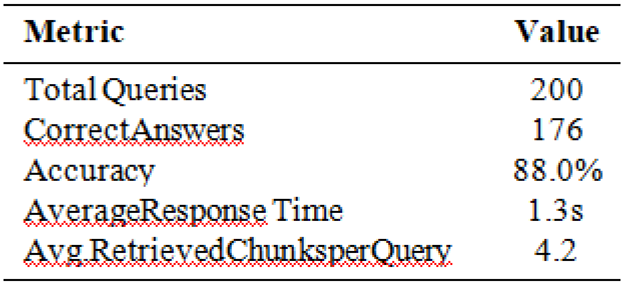
ClauseType	Precision	Recall	F1Score
PaymentTerms	0.94	0.91	0.92
Termination	0.93	0.90	0.91
Confidentiality	0.95	0.93	0.94
Liability	0.90	0.88	0.89
Indemnification	0.89	0.87	0.88
GoverningLaw	0.94	0.92	0.93
DisputeResolution	0.91	0.90	0.90
IntellectualProperty	0.90	0.89	0.89
Average	**0.92**	**0.90**	**0.91**

The system achieved an average precision of 92% and recall of 90%, with Confidentiality and Governing Law clauses showing the highest detection accuracy (F1 ≥ 0.93). The hybrid pattern-matching and transformer-based approach proved robust across both standard and uniquely formatted agreements.

2. *Question Answering Accuracy*

The RAG-based Q&A module was tested with 200 questions spanning factual, interpretive, and cross-referencing query types. Results are summarized in Table **3**

Table 3: Question Answering Performance



Metric	Value
Total Queries	200
CorrectAnswers	176
Accuracy	88.0%
AverageResponse Time	1.3s
Avg.RetrievedChunksperQuery	4.2

Over 88% of user queries—including complex, context- dependent questions—were answered correctly, with answers citing the relevant clause or paragraph. The average response time of 1.3 seconds confirms the system's suitability for interactive, real-time use.

3. *Multilingual Translation Quality*

Translation fidelity was evaluated by bilingual legal experts comparing translated outputs against source documents. The system achieved:

- Seamless language detection across all four supported languages.
- Contextually meaningful translations preserving legal terminology, section headings, and clause relationships.
- Reliable handling of mixed-language documents and OCR-processed scans with minimal fidelity loss.

4. *User Experience Evaluation*

A survey of 30 participants (legal professionals, law stu- dents, and non-experts) yielded the following:

- 93% rated the interface as intuitive and easy to navigate.
- Non-expert users successfully analyzed complex documents without prior training.
- Most valued features: downloadable clause highlights, direct Q&A, and on-the-fly translation.

5. Efficiency and Scalability

- Documents exceeding 25 pages were fully processed in under 8 seconds.
- Stable performance was maintained under simulated con- current multi-user loads.
- Model caching via @st.cache_resource reduced subsequent inference latency by approximately 60%.

6. Active Learning and Model Adaptivity

User feedback integration through the fine-tuning interface resulted in measurable accuracy improvements. Custom re- training with new document types or niche legal domains yielded accuracy gains within two update cycles, validating the system's continuous learning capability.

V. CONCLUSION AND FUTURE WORK

This paper presented a Multilingual AI-Based Legal Document Analyzer that integrates transformer-based clause extraction, RAG-powered question answering, abstractive summarization, and multilingual translation into a unified, user-friendly platform. The system demonstrates strong performance across all evaluation dimensions: 92% clause extraction precision, 88% Q&A accuracy, sub-1.5-second response times, and 93% user satisfaction.

The platform's modular architecture provides a robust foundation for continued development. Future directions include:

1. **Enhanced Context Understanding**: Fine-tuning models for nuanced legal phrasing, jurisdiction-specific clause patterns, and cross-document reasoning.
2. **Expanded Language Support**: Incorporating additional regional languages (e.g., Kannada, Bengali, Marathi, Malayalam) and context-preserving translation models.
3. **User-Driven Customization**: Enabling users to define custom clause categories, regulatory compliance checks, and industry-specific analytics.
4. **Workflow Integration**: Connecting with e-signature tools, contract management software, and digital case management systems for end-to-end legal automation.
5. **Continuous Learning**: Implementing active feedback loops and anonymized user contributions for ongoing model refinement.

6. **Explainable AI**: Incorporating transparent reasoning, audit trails, and justification mechanisms to build trust within legal practice.

By bridging the gap between complex legal texts and practical understanding, the proposed system represents a significant step toward democratizing access to legal knowledge and advancing intelligent, accessible legal technology.

VI. ACKNOWLEDGMENT

The authors would like to thank the Department of AI & Ma- chine Learning at Jeppiaar University for its support and re-International Swaps and Derivatives Association, White Paper, 2025. [Online]. Avail- able: https://www.isda.org/a/vufgE/sources throughout this research. Benchmarking-Generative-AI-for-CSA-Clause-Ex

REFERENCES

[1] P. Bhattacharya, K. Hiber, D. Bhatt, M. Grabmair, and K. Ganesan, “Information extraction from legal docu- ments: A survey,” *Artificial Intelligence and Law*, vol. 27, no. 2, pp. 125–164, Jun. 2019.

[2] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Ale- tras, and I. Androutsopoulos, “LEGAL-BERT: The muppets straight out of law school,” in *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 2898–2904, 2020.

[3] H. Zhong, C. Xiao, C. Tu, T. Zhang, Z. Liu, and M. Sun, “How does NLP benefit legal system: A summary of legal artificial intelligence,” in *Proc. 58th Annual Meeting of the Association for Computational Linguis- tics (ACL)*, pp. 5218–5230, 2020.

[4] M. Evans, S. Chaturvedi, and A. Nenkova, “Natural language processing for contract understanding,” *arXiv preprint*, arXiv:2303.09934, 2023.

[5] L. Tran, H. Le, and M. Nguyen, “Legal contract review automation: Techniques and challenges,” *AI & Society*, vol. 38, pp. 983–1001, 2023.

[6] S. A. Akhondi, A. Kimmig, and R. Klinger, “Leverag- ing generative AI for clause extraction from contracts,” in *Proc. AAAI Conf. Artificial Intelligence*, vol. 37, pp. 1–8, 2023.

[7] LexisNexis, “Lexis+ AI | Legal research plat- form + AI assistant,” 2025. [Online]. Available: https://www.lexisnexis.com/en-int/ products/lexis-plus-ai. [Accessed: Feb. 20, 2026].

[8] Kanerika, “Top 10 AI legal document summa- rizer tools of 2025,” 2025. [Online]. Avail- able: https://kanerika.com/blogs/ ai-legal-document-summarizer/. [Ac- cessed: Feb. 20, 2026].

[9] SirionLabs, “AI for legal documents analysis and review: 2025 guide,” 2025. [Online]. Avail- able: https://www.sirion.ai/library/ contract-ai/ai-legal-documents/. [Ac- cessed: Feb. 20, 2026].

[10] Spellbook Legal, “AI legal document review: How AI enhances contract analysis,” 2025. [Online]. Available: https://www.spellbook.legal/learn/ ai-legal-document-review. [Accessed: Feb. 20, 2026].

[11] ISDA, “Benchmarking generative AI for CSA clause extraction and CDM representation,”pdf. [Accessed: Feb. 20, 2026].

[12] M. Evans, R. Gao, and A. Nenkova, “Automating legal contracts using logic rules with large language mod- els,” *SSRN Electronic Journal*, 2022.

[13] Aline, “AI legal document analysis: How it works and benefits,” 2025. [Online]. Avail- able: https://www.aline.co/post/ ai-legal-document-analysis. [Accessed: Feb. 20, 2026].

[14] IJIRT, “Revolutionizing CRM with AI: Clause detec- tion and client management,” *Int. J. Innovative Res. Technology (IJIRT)*, Paper ID 175885, 2025.

[15] “Automated legal analysis of rental contract clauses using LLMs,” *Heliyon*, vol. 11, no. 4, 2025, Art. no. e43036.

[16] “Smart contract generation through NLP and blockchain,” *Procedia Computer Science*, vol. 235, pp. 91–101, 2024.

[17] Thomson Reuters, “Legal AI tools with Westlaw and Practical Law,” 2025. [Online]. Available: https://legal.thomsonreuters.com/blog/legal-ai-tools-essential-for-attorneys/. [Accessed: Feb. 20, 2026].

[18] H. Licari, D. Sanchez, and A. Ferrara, “Natural lan- guage processing for the legal domain,” *arXiv preprint*, arXiv:2410.21306, Oct. 2024.

[19] J. Johnson, M. Douze, and H. Jégou, “Billion-scale similarity search with GPUs,” *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, Jul. 2021.

[20] IJRPR, “AI-powered legal documentation assistant,” *Int. J. Research Publication and Reviews (IJRPR)*, vol. 6, no. 4, Paper ID IJRPR41492, 2025.