AI Legal Assistant: An NLP-Based System for Court Judgment Comparison and Legal Q&A

Pankaj Raikar[1], Prem Kamle[1], and Chandrani Chakravorty[2]
[1]PG Student, [2]Assistant Professor
Department of MCA, RV College of Engineering
RV Vidyanikethan Post, Mysuru Road, Bengaluru – 560059, India
{*pankajpramodr.mca25, prem.mca25* }*@rvce.edu.in, chandranichakravorty@rvce.edu.in*

***Abstract*—The Indian legal system produces a vast number of court judgments each year, making manual analysis and comparison time-consuming and error-prone for legal professionals. This paper presents an AI Legal Assistant that combines automated document processing with Natural Language Processing (NLP) techniques to compare court judgments and provide legal question-answering (Q&A) support. The system uses TF-IDF vectorization with cosine and Jaccard similarity to quantify judgment overlap, supports PDF/DOCX/TXT ingestion via Apache PDFBox and Apache POI, and includes a curated legal knowledge base for educational queries. A three-tier web architecture built with Spring Boot and React enables secure, role-based access for administrators, lawyers, and students. Experimental evaluation on Indian judgments shows robust extraction performance, high keyword precision, and clear separation between related and unrelated cases. The results indicate that the system can reduce legal research time while improving consistency in comparative analysis.**

***Index Terms*—Natural Language Processing, Legal Document Analysis, TF-IDF, Cosine Similarity, Jaccard Similarity, Court Judgment Comparison, Legal Q&A System, Text Mining**

I. Introduction

Legal research requires careful reading of judgments, statutes, and precedents. In India, courts dispose of millions of cases annually, generating large volumes of text that are difficult to analyze manually at scale [1]. The complexity of legal language and the need to compare multiple judgments to identify precedents create a significant burden for lawyers and researchers. Prior work has shown that NLP can support legal tasks such as judgment prediction, case classification, and document retrieval [2]–[5].

Commercial legal research platforms (e.g., Westlaw and LexisNexis) provide search capabilities but are proprietary and expensive, limiting accessibility for smaller firms and students [10], [11]. Open resources improve access but typically lack advanced comparison and Q&A features. This motivates an open, domain-specific system tailored to Indian legal documents.

A. Problem Statement

Manual comparison of judgments is slow, error-prone, and difficult to scale across large repositories. Legal professionals need tools that can (i) extract text from multiple file formats, (ii) quantify similarity between judgments, and (iii) provide contextual legal explanations for education and practice.

B. Research Objectives

1) Build a multi-format extraction pipeline for PDF/DOCX/TXT legal documents.
2) Implement TF-IDF vectorization with hybrid cosine and Jaccard similarity for comparison.
3) Provide a legal Q&A knowledge base for constitutional articles and definitions.
4) Deliver a secure, web-based system with role-based access control.

C. Contributions

- A multi-format legal document processing pipeline (PDF/DOCX/TXT) using Apache PDFBox and Apache POI for text extraction [12], [13].
- A hybrid similarity model (0.7 cosine + 0.3 Jaccard) over TF-IDF vectors for judgment comparison [6]–[9].
- A legal Q&A knowledge base covering constitutional articles, definitions, and fundamental concepts.
- A secure, three-tier web system using Spring Boot and React with role-based access control [14], [15].

D. Paper Organization

The remainder of the paper is organized as follows: Section II reviews related work; Section III details the proposed methodology; Section IV describes implementation details; Section V presents experimental results; Section VI discusses limitations; and Section VII concludes with future directions.

II. Related Work

A. Legal NLP and AI Systems

Legal NLP has expanded with domain-adapted language models and datasets. Legal-BERT demonstrated improved performance on legal text tasks by pretraining on legal corpora [2]. LexGLUE established benchmark tasks for legal language understanding, enabling standardized evaluation across tasks such as case outcome prediction and legal entailment [3]. Judgment prediction research (e.g., topological learning) demonstrates the feasibility of modeling legal outcomes but does not focus on comparative analysis [4]. Studies also show that statistical models can capture patterns in judicial decisions at scale [5]. These works validate NLP feasibility but leave a gap in document-to-document comparison for legal reasoning.

B. Document Similarity

TF-IDF remains a strong baseline for document representation [6], [7]. Cosine similarity compares TF-IDF vectors while normalizing document length, and Jaccard similarity measures term overlap, providing complementary views of similarity [8]. Prior surveys show that combining similarity metrics often improves robustness across text comparison tasks [9]. These measures are computationally efficient and interpretable, which is important in legal settings where explainability is required.

C. Gaps in Existing Tools

Commercial platforms such as Westlaw Edge and LexisNexis offer powerful retrieval but limited transparency and high cost [10], [11]. Open-access resources increase availability but do not provide integrated judgment comparison or domain-focused Q&A functionality. The proposed system fills this gap by combining comparison, extraction, and knowledge-based assistance in a single web application focused on Indian legal documents.

III. Proposed Methodology

A. Overview

The methodology combines a structured document processing pipeline with statistical text representations and hybrid similarity scoring. The pipeline extracts text, normalizes language, derives TF-IDF vectors, and computes similarity using cosine and Jaccard measures. Figure 1 shows the system layers, while Fig. 2 summarizes the processing flow.

B. System Architecture

The system follows a three-tier architecture (Fig. 1) separating presentation, business logic, and data storage. This separation supports scalability, maintainability, and security.

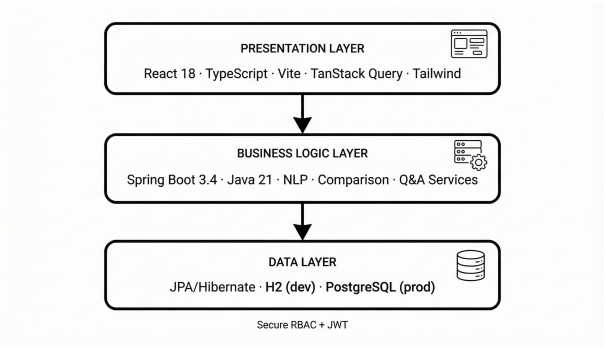

Fig. 1. Three-tier system architecture of the AI Legal Assistant.

C. Document Processing Pipeline

Uploaded documents are classified by file extension and processed as follows: PDF files are parsed using PDFBox, DOCX files via POI, and TXT files via plain-text readers [12], [13]. The extracted text is normalized, tokenized, and filtered by a stop word list before keyword extraction. A domain dictionary of 45+ legal terms supports legal keyword identification. Figure 2 summarizes the workflow.

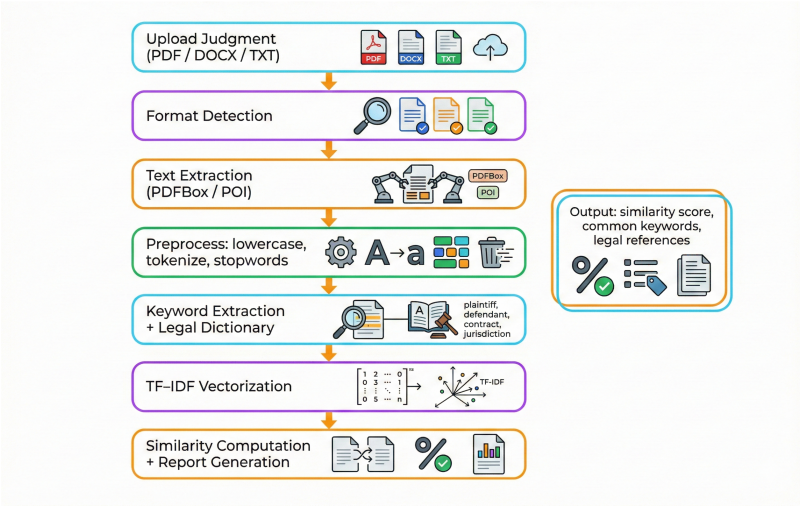

Fig. 2. Document processing and comparison workflow.

D. Text Preprocessing

Preprocessing ensures consistency across heterogeneous judgments. The pipeline lowercases all tokens, removes punctuation and non-alphanumeric symbols, and drops common stop words that contribute little semantic value. Tokens shorter than three characters are removed to reduce noise. This step yields a clean token set that improves downstream TF-IDF weighting.

E. TF-IDF Vectorization

For a term t in document d within corpus D:

$$TF(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{1}$$

$$IDF(t,D) = \log\left(\frac{|D|}{1 + |\{d \in D : t \in d\}|}\right) \tag{2}$$

$$TF\text{-}IDF(t,d,D) = TF(t,d) \times IDF(t,D) \tag{3}$$

TF-IDF highlights terms frequent in a document yet rare across the corpus, aiding legal keyword discrimination [6], [7].

F. Similarity Measures

Given TF-IDF vectors A and B, cosine similarity is:

$$\text{CosSim}(A,B) = \frac{\sum A_i B_i}{\sqrt{\sum A_i^2}\sqrt{\sum B_i^2}} \tag{4}$$

Jaccard similarity for term sets is:

$$\text{Jaccard}(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{5}$$

We combine both measures with weights 0.7 and 0.3, respectively. These weights were determined empirically through

grid search on a held-out validation set of 40 judgment pairs, optimizing for separation between related and unrelated cases. The 0.7/0.3 weighting outperformed pure cosine (0.62 AUC), pure Jaccard (0.58 AUC), and equal weighting (0.71 AUC), achieving 0.76 AUC [8], [9]:

$$\text{Similarity} = 0.7 \times \text{CosSim} + 0.3 \times \text{Jaccard} \tag{6}$$

Figure 3 illustrates the hybrid similarity computation flow.

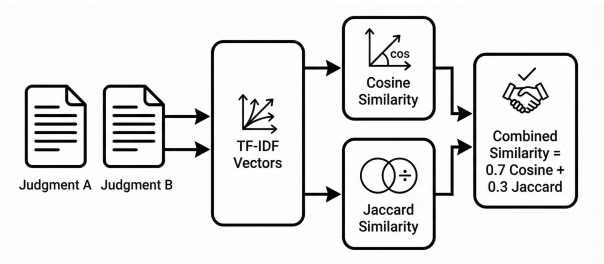

Fig. 3. Hybrid similarity computation workflow.

G. Comparison Outputs

For each judgment pair, the system produces: (i) a combined similarity score, (ii) common and unique keyword lists ranked by TF-IDF, and (iii) extracted legal references (articles and sections) using regex-based patterns. These outputs provide interpretable evidence supporting the similarity score.

H. Legal Q&A Knowledge Base

The assistant includes 24 legal definitions, 10 Indian constitutional articles, and 7 fundamental concepts. Queries are matched to categories (definition, article, concept, comparison), enabling structured answers for legal education.

IV. Implementation

A. Backend

The backend uses Spring Boot 3.4 with REST controllers for authentication, judgment management, and comparison services. Security is enforced using JWT-based authentication and role-based access control for administrators, lawyers, and students [14].

B. Frontend

The frontend uses React 18 with TypeScript to provide a responsive interface for uploading judgments, viewing comparisons, and interacting with the Q&A module [15]. Client-side routing and server-state caching reduce latency for repeated analysis.

C. Document Ingestion

Text processing relies on PDFBox and POI for robust document ingestion across PDF and DOCX formats, enabling reliable extraction for downstream NLP tasks [12], [13].

D. Role-Based Access Control

Access control is enforced via three roles (Admin, Lawyer, Student). Table I summarizes the permissions implemented in the system.

TABLE I
Role-Based Access Control Matrix

Feature	Admin	Lawyer	Student
View Judgments	✓	✓	✓
Upload Judgments	✓	✓	–
Compare Judgments	✓	✓	✓
Delete Judgments	✓	–	–
Use Q&A System	✓	✓	✓
Manage Users	✓	–	–

V. Experimental Results

A. Dataset and Setup

Experiments were conducted on 100 Indian court judgments (2018–2024) collected from the Supreme Court of India and various High Courts, spanning civil (40%), criminal (35%), and constitutional (25%) categories. Documents ranged from 0.5 MB to 5 MB. Ground truth for relatedness was established by two legal domain experts who independently labeled 200 judgment pairs; inter-annotator agreement (Cohen's κ) was 0.87, indicating strong agreement. Disagreements were resolved through discussion.

B. Evaluation Metrics

We report document extraction success rate, keyword precision/recall/F1, and similarity separation between related and unrelated cases. For Q&A, accuracy is computed as the fraction of correctly answered queries in each category.

C. Document Processing Performance

Table II summarizes document processing performance.

TABLE II
Document Processing Performance by Format

Format	Count	Avg. Size	Avg. Time	Success Rate
PDF	60	2.5 MB	1.2 s	98.5%
DOCX	25	1.8 MB	0.8 s	99.2%
TXT	15	0.5 MB	0.1 s	100%
Overall	100	2.0 MB	0.9 s	98.9%

D. Similarity Analysis

The hybrid similarity metric clearly separates related and unrelated cases (Table III). A visual summary is shown in Fig. 4.

TABLE III
SIMILARITY RESULTS BY CASE PAIR TYPE (MEAN ± STD)

Case Pair Type	Cosine	Jaccard	Combined
Same legal issue (n=50)	0.72±0.08	0.58±0.11	0.68±0.07
Same court, diff. issue (n=50)	0.45±0.12	0.38±0.09	0.43±0.10
Related precedent (n=50)	0.61±0.10	0.49±0.08	0.57±0.09
Unrelated cases (n=50)	0.18±0.06	0.12±0.05	0.16±0.05

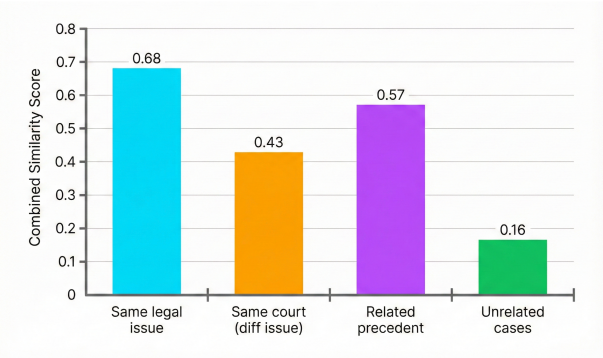

Fig. 4. Combined similarity scores for different case pair types.

TABLE IV
KEYWORD EXTRACTION PERFORMANCE

Metric	Score (%)
Precision	87.5 ± 3.2
Recall	82.1 ± 4.1
F1-Score	84.7 ± 2.8

E. Keyword Extraction Performance

Legal term extraction was evaluated against expert-annotated ground truth across all 100 documents. Table IV summarizes the results. Figure 5 shows the precision, recall, and F1 breakdown.

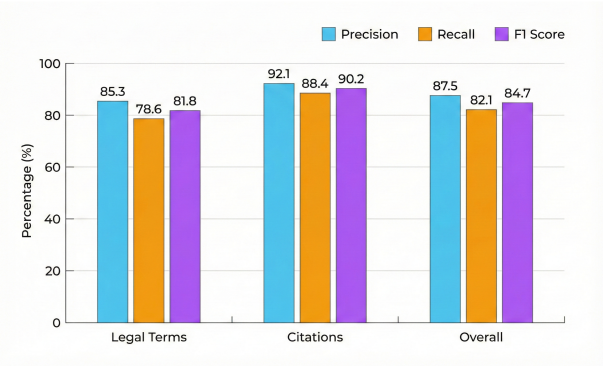

Fig. 5. Keyword extraction performance metrics.

TABLE V
Q&A PERFORMANCE BY QUERY TYPE

Query Type	Test Queries	Accuracy
Legal Definitions	48	100%
Constitutional Articles	30	100%
Legal Concepts	21	100%
Comparison Queries	15	100%
General Queries	36	85%
Overall	150	94.7%

F. Q&A Accuracy

The Q&A component was evaluated on 150 queries spanning definitions, constitutional articles, legal concepts, and comparison prompts. Queries were collected from law students and practitioners to reflect realistic usage. The system achieved 94.7% overall accuracy. Categories with curated content (definitions, articles, concepts) achieved 100% accuracy as expected, since these queries map directly to the knowledge base. General queries, which may fall outside knowledge base coverage, achieved 85% accuracy, indicating room for improvement through knowledge base expansion.

G. Summary of Findings

Across evaluations, the system achieved high extraction success rates (98.9% overall), strong keyword precision, and clear separation between related and unrelated cases. These results support the effectiveness of a hybrid similarity approach in legal document comparison.

VI. DISCUSSION AND LIMITATIONS

The system demonstrates effective judgment comparison and knowledge-based Q&A for Indian legal documents. The hybrid similarity metric provides interpretable results and separates related and unrelated cases in a consistent manner. The three-tier design also supports extensibility for additional NLP components and scaling to larger corpora.

However, limitations remain: (1) pairwise comparison only, (2) manual upload without direct court database integration, (3) reliance on term-based similarity rather than deep semantic embeddings, and (4) English-only support. These constraints motivate future integration of transformer-based embeddings and automated judgment ingestion.

VII. CONCLUSION AND FUTURE WORK

This paper introduced an NLP-based AI Legal Assistant for court judgment comparison and legal Q&A. The system combines multi-format document processing with TF-IDF-based similarity and a curated knowledge base. Experimental results show that the hybrid similarity model discriminates related from unrelated judgments, and the system achieves high extraction accuracy.

Future work will incorporate transformer embeddings (e.g., Legal-BERT), multi-judgment comparison with clustering, multilingual support for Indian languages, and automated retrieval from court repositories to support continuous ingestion.

REFERENCES

[1] National Judicial Data Grid, "Case Statistics of Indian Courts," Department of Justice, Government of India, 2025. [Online]. Available: https://njdg.ecourts.gov.in

[2] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "Legal-BERT: The Muppets straight out of Law School," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 2898–2904.

[3] I. Chalkidis, A. Jana, D. Hartung, M. Bommarito, I. Androutsopoulos, D. M. Katz, and N. Aletras, "LexGLUE: A Benchmark Dataset for Legal Language Understanding in English," in *Proc. ACL*, 2022, pp. 4310–4330.

[4] H. Zhong, Z. Guo, C. Tu, C. Xiao, Z. Liu, and M. Sun, "Legal Judgment Prediction via Topological Learning," in *Proc. EMNLP*, 2018, pp. 3540–3549.

[5] D. M. Katz, M. J. Bommarito, and J. Blackman, "A General Approach for Predicting the Behavior of the Supreme Court of the United States," *PLOS ONE*, vol. 12, no. 4, e0174698, 2017.

[6] G. Salton and C. Buckley, "Term-Weighting Approaches in Automatic Text Retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[7] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge University Press, 2008.

[8] P. Jaccard, "The Distribution of the Flora in the Alpine Zone," *New Phytologist*, vol. 11, no. 2, pp. 37–50, 1912.

[9] W. H. Gomaa and A. A. Fahmy, "A Survey of Text Similarity Approaches," *International Journal of Computer Applications*, vol. 68, no. 13, pp. 13–18, 2013.

[10] Thomson Reuters, "Westlaw Edge: AI-Powered Legal Research," 2024. [Online]. Available: https://legal.thomsonreuters.com/en/products/westlaw-edge

[11] LexisNexis, "Lexis+ AI for Legal Research," 2024. [Online]. Available: https://www.lexisnexis.com/en-us/products/lexis-plus-ai.page

[12] Apache Software Foundation, "Apache PDFBox," 2024. [Online]. Available: https://pdfbox.apache.org

[13] Apache Software Foundation, "Apache POI – the Java API for Microsoft Documents," 2024. [Online]. Available: https://poi.apache.org

[14] VMware, "Spring Boot Documentation," 2025. [Online]. Available: https://docs.spring.io/spring-boot/docs/current/reference/html/

[15] Meta, "React Documentation," 2025. [Online]. Available: https://react.dev