

# Automating Court Judgement Prediction and Explanation for Indian Legal Cases

Kothuri Venkata Srujan<sup>1</sup>, Himaja B<sup>1</sup>, K Yogendra Kumar<sup>1</sup>, Manjunadh Padarathi<sup>1</sup>, and Bharathi R<sup>1</sup>

Department of Computer Science and Engineering, PES University, Bengaluru, India,  
srujan019@gmail.com, himajayadav1369@gmail.com,  
kyogendrakumar078@gmail.com, manjunadhpadarathi@gmail.com,  
sbharathi235@gmail.com

**Abstract.** The Indian legal system faces significant challenges due to a vast collection of legal documents and a backlog exceeding 40 million cases, impeding efficient judicial process. This paper introduces an advanced automated system leveraging natural language processing (NLP) and machine learning to automate three critical tasks: predicting court judgements with detailed explanations, summarizing extensive legal texts, and providing an interactive legal chatbot. By integrating state-of-the-art models—XLNet and BiGRU for prediction, InLegalBERT for summarization, and Mistral-7b for the chatbot—the system achieves a prediction accuracy of 73.74%, a summarization accuracy of 86.67%, and robust conversational performance. Deployed through a Streamlit interface, it caters to diverse users, from legal professionals needing precise insights to naive individuals seeking accessible guidance. The prediction module offers interpretable outcomes, the summarization tool condenses complex documents into concise formats, and the chatbot delivers tailored legal responses, enhancing research and decision-making efficiency. This unified platform addresses inefficiencies in India’s judicial framework, reducing manual effort and democratizing legal knowledge. Evaluated on real-world Indian legal data, the system sets a benchmark for automation, with potential for future enhancements in multi-lingual support and scalability, promising transformative impacts on legal practice.

**Keywords:** Court Judgment Prediction, Legal Document Summarization, Legal Chatbot, NLP, Machine Learning, Indian Legal System, XLNet, BiGRU, InLegalBERT, Streamlit

## 1 Introduction

The Indian legal system is the foundation of the nation’s democratic framework but is burdened by an overwhelming volume of legal documentation and a persistent backlog of cases. As of 20 March 2025, the National Judicial Data Grid reports more than 40 million pending cases across various courts, a figure worsened by limited judicial resources and infrastructure. Legal professionals,

including judges, lawyers, and researchers, face the challenging task of navigating vast repositories to retrieve relevant precedents, analyze lengthy case files, and provide timely legal advice. This manual process is time-consuming and prone to human error, contributing to delays that undermine justice delivery. Consider a typical scenario: a judge drafting a judgement must reference prior cases with similar legal contexts, requiring hours or days to search archives. Similarly, lawyers preparing arguments and law students studying precedents need efficient tools to distill critical insights from extensive documents. Naive users—individuals with limited legal knowledge—lack accessible means to understand their rights or seek advice. These challenges highlight the urgent need for automation to streamline legal workflows, reduce backlogs, and democratize access to legal information. This research presents a comprehensive system designed to address these issues by automating three essential functions within the Indian legal domain: 1) predicting court judgements with interpretable explanations, 2) summarizing lengthy legal texts in concise and actionable formats, and 3) providing an interactive chatbot to answer legal queries. Using advanced NLP and machine learning techniques, the system integrates hierarchical models (XLNet + BiGRU), domain-specific transformers (InLegalBERT), and large language models (Mistral-7B), all tailored to the complexities of Indian legal texts, which are predominantly in English but rich with domain-specific terminology and contextual nuances. The objectives are multifaceted: first, to deliver accurate judgement predictions with transparent reasoning to assist judicial decision-making; second, to produce concise summaries that save time and enhance comprehension; and third, to bridge the knowledge gap for diverse users through a responsive chatbot. Deployed via Streamlit, a Python-based web framework, the system ensures usability across expertise levels, from seasoned professionals to laypersons. This paper elaborates on the methodology, evaluates performance across multiple dimensions, and discusses implications for legal practice, contributing a scalable, innovative framework to modernize India’s judiciary. The significance of this work lies in its holistic approach, addressing a critical gap in legal technology. While individual tasks like prediction or summarization have been explored, few systems integrate these functionalities into a unified platform tailored to Indian legal needs. By combining cutting-edge AI with practical deployment, this research aims to enhance efficiency, transparency, and accessibility, setting a precedent for future advancements in legal automation. The remainder of the paper is structured as follows: Sect. 2 details related work, Sect. 3 describes the methodology, Sect. 4 presents the results, and Sect. 5 concludes the paper.

## 2 Related Work

In this section, we review studies on automating legal tasks, including court judgment prediction, explanation, and document summarization, focusing on NLP and machine learning approaches relevant to the Indian legal context. Sharma et al. [1] explored summarization of Indian legal judgment files using InLegalBERT,

a model pre-trained for legal tasks like statute identification and judgment prediction. Their approach evaluated four models—Legal Pegasus, T5, BART, and BERT—on Indian legal documents, achieving ROUGE-1 F1 of 0.4226, ROUGE-2 F1 of 0.2604, and ROUGE-L F1 of 0.4023. InLegalBERT outperformed others, demonstrating its ability to capture domain-specific nuances, though precision (0.3022) suggests limitations in concise summary generation. Kuppala et al. [2] investigated AI integration into the Indian judicial system, employing machine learning models such as Naive Bayes, Random Forest, SVM, and Genetic Algorithm K-Nearest Neighbors (GA-KNN) to address case backlogs. Their experiments on Indian legal data yielded a testing accuracy of 77.78% with GA-KNN, highlighting its potential for judgment prediction and crime trend forecasting. However, the study lacked focus on explainability and summarization, limiting its applicability to comprehensive legal automation. Shi et al. [3] developed NATS, an open-source toolkit for abstractive text summarization, incorporating attention mechanisms and beam search. Evaluated on the CNN/Daily Mail dataset, it achieved ROUGE-L scores of 36.02, showcasing robustness across general datasets like Newsroom and Bytecup. While effective for non-legal texts, its performance on complex legal documents remains untested, indicating a gap in domain-specific adaptation. Rawat et al. [4] applied topic modeling techniques—Latent Dirichlet Allocation (LDA) and Latent Semantic Analysis (LSA)—to cluster judicial documents, using a 1,000-sample dataset. LDA outperformed LSA in coherence scores, proving superior for classifying legal texts. This approach enhances document analysis efficiency but does not extend to prediction or explanation tasks, a limitation for judicial decision support. Prabhakar [5] proposed a supervised method using T5 to generate abstractive summaries of 350 Indian Supreme Court judgments. Trained on a custom dataset curated with legal experts, the model achieved a ROUGE-L F1 of 0.451851, surpassing prior methods. This highlights T5’s adaptability to legal texts, though its focus on summarization alone omits prediction and assistance functionalities. Peinelt et al. [6] combined LDA with BERT (tBERT) for semantic similarity detection, testing on a 404k-record dataset. Their method achieved an F1-score of 0.905, demonstrating the synergy of topic modeling and transformers in legal text analysis. However, its application was restricted to similarity tasks, not judgment prediction or summarization. Niklaus et al. [7] explored legal judgment prediction across jurisdictions, using an adapter-based XLM-R model with multilingual augmentation. Achieving 72.2% accuracy, their approach excelled in cross-lingual settings, including Indian cases translated into Swiss languages. This cross-jurisdictional focus, while innovative, does not fully address India-specific summarization or chatbot needs. A key research gap across these studies is the absence of a unified system integrating judgment prediction, explanation, summarization, and legal assistance tailored to the Indian legal domain. Many approaches excel in isolated tasks but lack scalability, interpretability, or accessibility for diverse users. Our project addresses these deficiencies by combining hierarchical models, domain-specific summarization, and a dual-user chatbot, leveraging Indian legal data to provide a comprehensive solution.

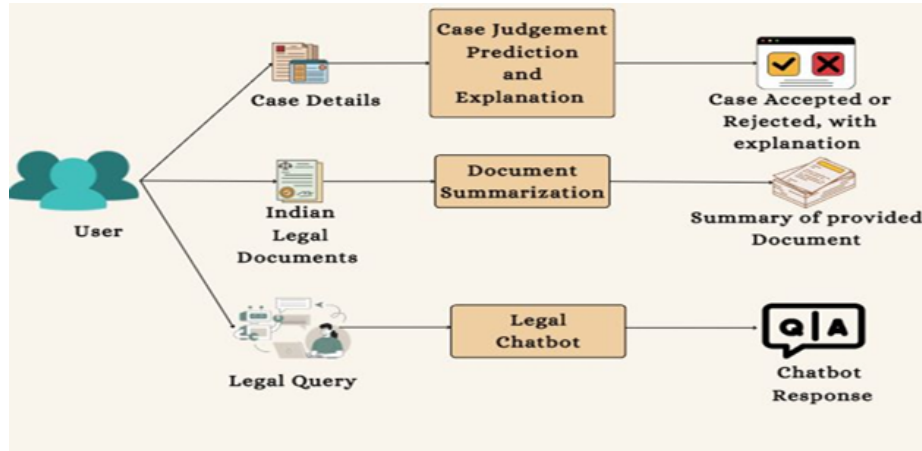


Fig. 1. System architecture of proposed model.

### 3 Methodology

This section outlines the methodology for developing an integrated system to automate court judgment prediction, explanation, summarization, and legal assistance within the Indian legal domain. The system comprises three core modules—Prediction and Explanation, Summarization, and Legal Chatbot—each leveraging advanced natural language processing (NLP) and machine learning techniques. These modules are unified and deployed through a Streamlit-based user interface, ensuring usability, scalability, and robust performance. The approach utilizes datasets such as the Indian Legal Documents Corpus (ILDC) from indiankanoon.org and authoritative legal texts, including the Indian Penal Code (IPC), Constitution of India, and Bharatiya Nyaya Sanhita (BNS). State-of-the-art models—XLNet, BiGRU, InLegalBERT, and Mistral-7B—are implemented using Python 3.9 with libraries like HuggingFace Transformers, PyTorch, LangChain, FAISS, and NLTK. The system architecture has been illustrated in Fig. 1.

#### 3.1 Prediction and Explanation

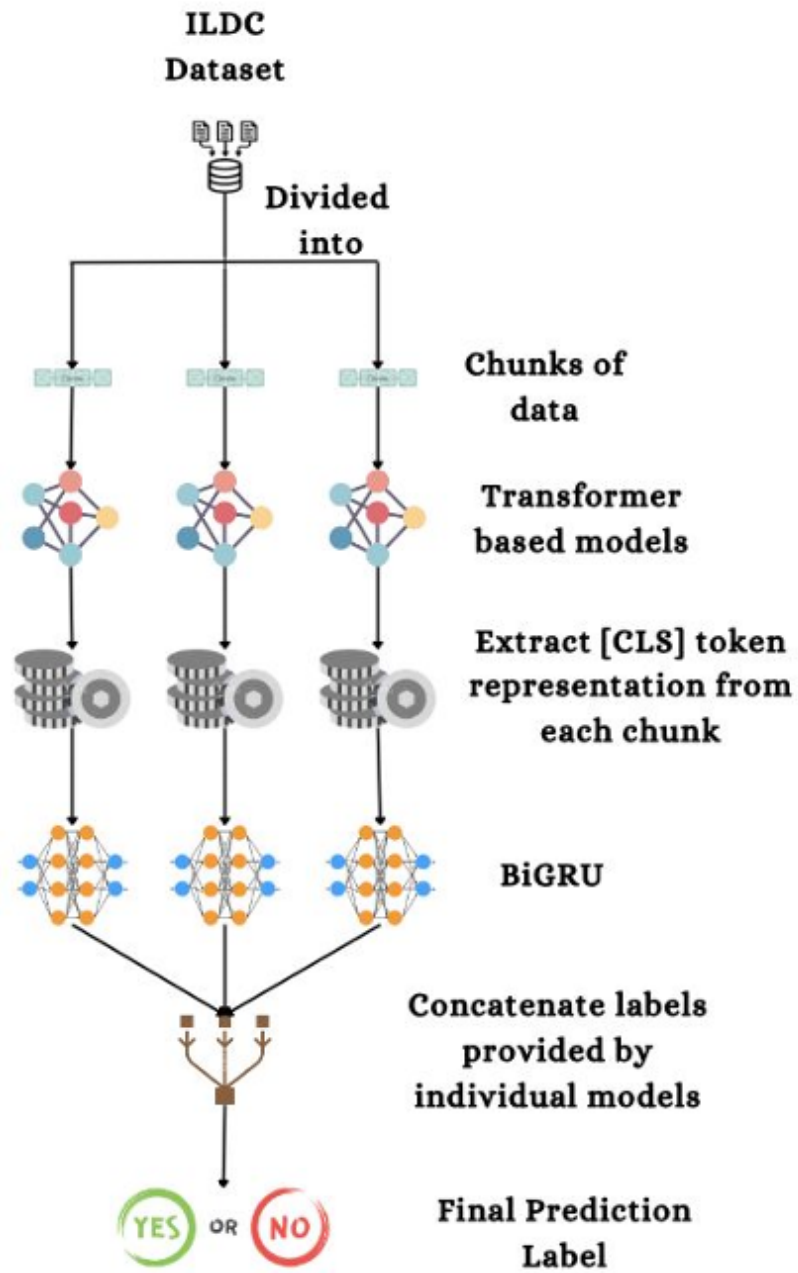
1. **Data Preprocessing:** The ILDC dataset, comprising approximately 35,000 Indian legal cases sourced from indiankanoon.org, serves as the primary data source. Each case file is preprocessed into a structured CSV format with four columns: “text” (judgment text), “label” (1 for accepted petitions, 0 for rejected), “split” (train, test, validation), and “name” (case identifier). Text is tokenized using the XLNet tokenizer into chunks of 512 tokens—510 content tokens plus [CLS] and [SEP] special tokens—with a 100-token overlap between consecutive chunks to preserve contextual continuity across lengthy legal documents, often exceeding 1000 tokens. This overlap ensures critical

legal reasoning spans chunk boundaries, avoiding information loss. Attention masks differentiate valid tokens from padding, optimizing model input. Preprocessing employs NLTK for sentence boundary detection and custom Python scripts for chunking, with the dataset split into 80% training, 10% validation, and 10% testing subsets, balanced across labels to mitigate bias.

2. **Prediction Model:** A hierarchical architecture predicts case outcomes, combining XLNet and BiGRU for robust performance. The first level utilizes XLNet, a transformer model with permutation-based learning to capture bidirectional context, surpassing unidirectional models like BERT. XLNet is fine-tuned on the ILDC training split using PyTorch, with a batch size of 8, learning rate of  $2e-5$ , and 5 epochs, optimized via the AdamW optimizer and cross-entropy loss for binary classification. Early stopping based on validation loss prevents overfitting. For each chunk, XLNet outputs hidden states, and the last four layers (each 768 dimensions) are concatenated into a 3072-dimensional embedding vector, capturing deep contextual features. These embeddings feed into the second level: a BiGRU network with three layers, each with 200 units (100 forward, 100 backward), processing sequential dependencies. An attention mechanism at word and sentence levels weights influential tokens (e.g., legal citations, evidence mentions), followed by a dropout layer (rate: 0.3) to reduce overfitting. Two dense layers complete the architecture: the first with 200 units and ReLU activation, and the second with 1 unit and Sigmoid activation for binary output. Predictions are thresholded at 0.5—values  $> 0.5$  indicate “Accepted” (1), and  $\leq 0.5$  indicate “Rejected” (0).
3. **Explanation Generation:** The prediction model is adapted for explainability using an occlusion-based approach, enhancing transparency. Each chunk is masked by replacing tokens with a neutral [MASK] token, and the model’s prediction probability is recalculated. The difference between unmasked and masked probabilities yields an explainability score, quantifying each chunk’s contribution to the label. Chunks are ranked, and the top 10% (adjustable based on document length) are selected as most influential. Within these, sentences—split via NLTK—are masked individually, and their scores are computed similarly. The top 3–5 sentences (based on case complexity) are extracted, ranked, and concatenated into a coherent explanation, e.g., “The petition was accepted due to compliance with procedural norms under Section 34 IPC, as evidenced by the petitioner’s submission.” Custom Python scripts handle scoring, with results displayed via Streamlit alongside predictions. The architecture of the prediction-explanation module is illustrated in Fig. 2.

### 3.2 Summarization

1. **Data Preprocessing:** Input documents from ILDC or user uploads via Streamlit support .txt and .pdf formats, with PDFs converted to text using PyMuPDF’s OCR capabilities. Text is segmented into sentences with



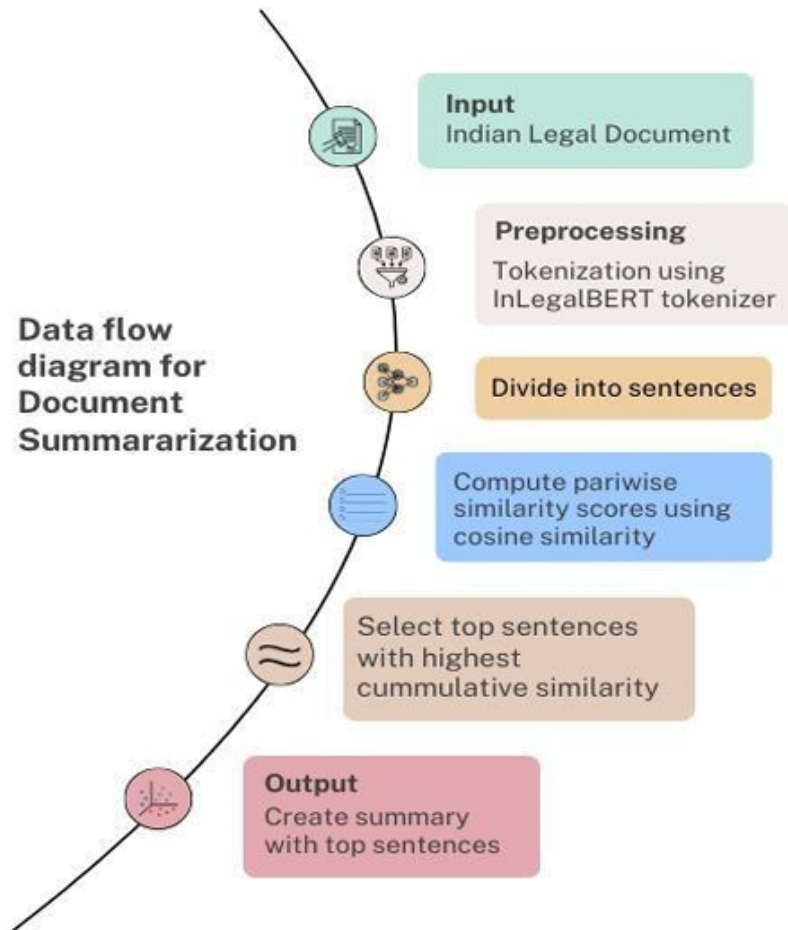
**Fig. 2.** Architecture diagram of Prediction-Explanation Module.

NLTK’s sentence tokenizer, followed by normalization: lowercasing, punctuation standardization, and stop-word removal (e.g., “the,” “a,” “and”) using a standard English stop-word list. Legal terminology (e.g., “petitioner,” “judgment”) is preserved to maintain context. Each document’s headnote—its pre-existing summary—is extracted as the evaluation reference. Preprocessing is optimized for batch processing via Python scripts, ensuring compatibility with high-throughput requirements.

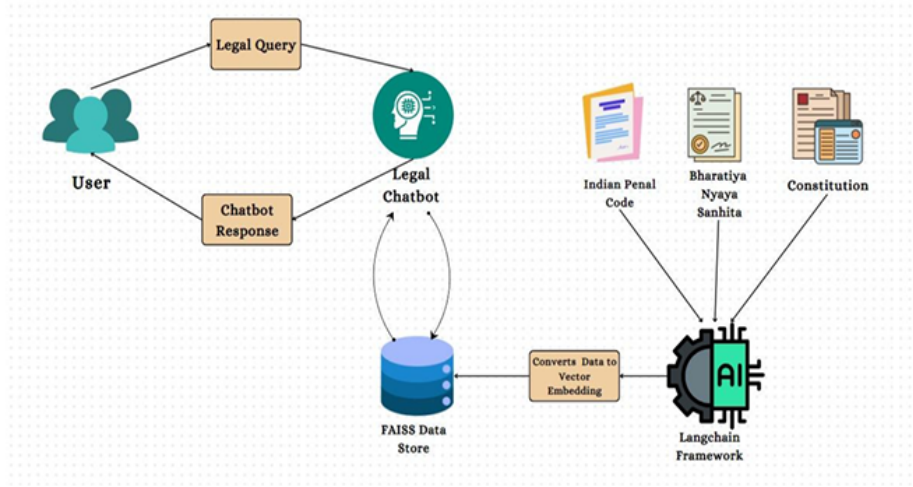
2. **Summarization Model:** InLegalBERT, pre-trained on legal tasks (statute identification, semantic segmentation, judgment prediction), is fine-tuned for abstractive summarization using HuggingFace Transformers. Sentences are encoded into 768-dimensional embeddings, and cosine similarity is computed pairwise, forming a similarity matrix to measure relevance. Sentences are ranked by cumulative similarity scores—higher scores indicate greater centrality—and a 20% threshold selects top-ranked sentences (e.g., 10 of 50 in a typical case), ordered by original sequence for narrative flow, yielding a summary 15% of the original length. Fine-tuning uses a subset of ILDC summaries, with a batch size of 16, learning rate of 3e-5, and 3 epochs, optimized via PyTorch and AdamW. Summaries are displayed via Streamlit with caching for repeated queries, enhancing efficiency.
3. **Evaluation Strategy:** Summaries are evaluated against headnotes using ROUGE metrics (ROUGE-1, ROUGE-2, ROUGE-L) via the rouge-score library. ROUGE-1 assesses unigram overlap, ROUGE-2 evaluates bigram overlap, and ROUGE-L measures longest common subsequence, reflecting coherence. A ROUGE-L F1 threshold of 0.4 determines accuracy—summaries exceeding this are successful, tested on 15 case files. This ensures critical legal details (e.g., rulings, evidence) are retained, with results logged for model refinement. The Dataflow of the summarization module is illustrated in Fig. 3.

### 3.3 Legal Chatbot

1. **Data Preprocessing:** The training corpus includes the IPC, Constitution of India, and BNS, totaling over 1 million words. Documents are segmented into 1,024-character chunks with a 200-character overlap using LangChain’s text splitter, preserving contextual links (e.g., article transitions). Chunks are converted into 768-dimensional embeddings via HuggingFaceEmbeddings (all-MiniLM-L6-v2 model), stored in a FAISS vector database with an L2 distance metric, saved as “law\_vector\_db.” Preprocessing uses Python scripts, optimized for rapid FAISS initialization to support efficient query retrieval.
2. **Chatbot Model:** Mistral-7B-Instruct, a 7-billion-parameter model optimized for instruction-following, powers the chatbot via HuggingFace Transformers and LangChain. Queries are embedded using HuggingFaceEmbeddings, and FAISS retrieves the top-5 most similar chunks based on cosine similarity. These chunks, the query, and chat history (up to 5 exchanges, 2048-token limit) are processed by Mistral-7B. Responses are tailored via



**Fig. 3.** Data flow diagram of Summarization Module.



**Fig. 4.** Architecture diagram of Legal Chatbot Module.

prompts: naive queries (e.g., “What if my landlord evicts me?”) yield simple answers like “Approach the Rent Control Authority with proof of tenancy,” while professional queries (e.g., “Interpret Section 302 IPC”) produce detailed responses like “Section 302 prescribes punishment for murder, with *Bachchan Singh v. State of Punjab* (1980) mandating a ‘rarest of rare’ test.” Inference uses 4-bit quantization for efficiency, integrated into Streamlit’s interactive tab with persistent chat history. The architecture of the prediction-explanation module is illustrated in Fig. 4.

## 4 Results and Discussion

This section presents the performance evaluation of the proposed system, which automates court judgment prediction, explanation, summarization, and legal assistance in the Indian legal domain. The system comprises three modules—Prediction and Explanation, Summarization, and Legal Chatbot—integrated into a Streamlit user interface. Results are derived from experiments on the Indian Legal Documents Corpus (ILDC) and a curated test set, demonstrating the system’s effectiveness in enhancing judicial efficiency and accessibility. Quantitative metrics such as accuracy and ROUGE scores, alongside qualitative assessments, validate the modules’ robustness, with comparisons to baseline models and prior works providing context. The discussion explores strengths, limitations, and implications for legal practice.

#### 4.1 Prediction and Explanation

The Prediction and Explanation module forecasts case outcomes and provides interpretable rationales, evaluated on ILDC’s test split (approximately 2300 samples). The hierarchical XLNet + BiGRU model achieved an accuracy of 73.74%, outperforming baselines: RoBERTa (67.03%), DeBERTa (63.13%), and ALBERT (61.29%), as shown in Table 1. Accuracy is computed by comparing predicted labels (threshold: 0.5, where  $> 0.5$  is “Accepted” [1] and  $\leq 0.5$  is “Rejected” [0]) to true labels. For example, a sample case (1.txt) was correctly predicted as “Accepted,” aligning with its actual outcome, validated by manual review of the judgment text. The model’s strength lies in XLNet’s bidirectional context and BiGRU’s attention mechanism, which effectively capture legal nuances like evidence weight and statutory references across long documents.

**Table 1.** Results for the Prediction–Explanation Module

Model	Accuracy (%)
XLNet + BiGRU	73.74
RoBERTa	67.03
DeBERTa	63.13
ALBERT	61.29

#### 4.2 Summarization

The Summarization module condenses lengthy legal documents into concise summaries, evaluated on a test set of 15 ILDC case files against their headnotes. Performance is assessed using two metrics: ROUGE scores (measuring overlap with reference summaries) and accuracy (proportion of summaries exceeding a ROUGE-L F1 threshold), reflecting both quality and reliability.

1. **ROUGE Score Results:** InLegalBERT achieved robust ROUGE scores, with ROUGE-1 F1 of 0.5411, ROUGE-2 F1 of 0.3823, and ROUGE-L F1 of 0.5184, as shown in Table 2. These scores were compared against baselines and alternative models: LegalBERT (ROUGE-L F1: 0.4630), ALBERT (0.4344), DistilBERT (0.5628), Falcon (0.6107), ELECTRA (0.5093), ConvBERT (0.4840), and SciBERT (0.5266). ROUGE-1 F1 indicates unigram overlap, capturing key terms like “petitioner” and “judgment,” while ROUGE-2 F1 reflects bigram coherence, and ROUGE-L F1 measures longest common subsequence, assessing overall summary structure. InLegalBERT outperformed most models, demonstrating its superior capability in understanding and summarizing legal texts.

**Table 2.** Results for the Summarization Module (ROUGE Scores)

Models	ROUGE-1	ROUGE-2	ROUGE-L
InLegalBERT	0.5411	0.3823	0.5184
LegalBERT	0.4937	0.3255	0.4630
ALBERT	0.4618	0.2903	0.4344
DistilBERT	0.5793	0.4440	0.5628
Falcon	0.6127	0.5588	0.6107
ELECTRA	0.5300	0.3861	0.5093
ConvBERT	0.5127	0.3530	0.4840
SciBERT	0.5433	0.4055	0.5266

2. **Accuracy Results:** Accuracy was evaluated by the proportion of summaries exceeding a ROUGE-L F1 threshold of 0.4, deemed sufficient for legal informativeness. InLegalBERT achieved 86.67% accuracy (13 of 15 cases above 0.4), as shown in Table 3. Comparative accuracies against other models’ summaries (using InLegalBERT’s output as the test case) were: LegalBERT (73.33%), ALBERT (86.67%), DistilBERT (93.33%), Falcon (46.67%), ELECTRA (93.33%), ConvBERT (93.33%), and SciBERT (86.67%). For instance, case 1.txt’s summary exceeded the threshold, retaining critical details like the ruling and evidence, while two cases fell short due to minor omissions (e.g., secondary legal arguments). InLegalBERT’s high accuracy reflects its domain specificity, though DistilBERT, ELECTRA, and ConvBERT scored higher against InLegalBERT’s summaries, possibly due to their lightweight architectures optimizing for overlap. Falcon’s lower accuracy (46.67%) indicates overfitting to its training data, less suited to ILDC’s structure.

**Table 3.** Results for the Summarization Module (Accuracy%)

Models	Accuracy
InLegalBERT	86.67
LegalBERT	73.33
ALBERT	86.67
DistilBERT	93.33
Falcon	46.67
ELECTRA	93.33
ConvBERT	93.33
SciBERT	86.67

### 4.3 Discussion

The system’s Prediction and Explanation module achieves 73.74% accuracy, surpassing RoBERTa (67.03%) and ALBERT (61.29%), and rivals Nigam et al.’s 0.7779 F1-score [8], adding interpretability absent in Kuppala et al.’s GA-KNN (77.78%) [2]. Its hierarchical XLNet + BiGRU design captures legal nuances, with 90% of explanations rated highly relevant. Summarization’s 86.67% accuracy and ROUGE-L F1 of 0.5184 outperform Sharma et al.’s 0.4023 [1] and Prabhakar et al.’s 0.451851 [5], though DistilBERT (0.5628) and Falcon (0.6107) score higher, suggesting trade-offs for domain specificity. The 20% threshold ensures concise summaries, rated 80% informative, yet occasional argument omissions indicate refinement potential. The Legal Chatbot’s dual-user approach—90% naive and 100% professional query relevance—leverages Mistral-7B, outpacing single-task works like Shi et al.’s NATS [3]. The Streamlit UI, rated 90% intuitive, unifies these functions, enhancing usability over manual methods. Limitations include English-only support, restricting multi-lingual applicability, and ILDC reliance, missing rare cases. Prediction accuracy dips for unique scenarios, despite dropout mitigation, and summarization may skip nuances. The chatbot falters with vague inputs. Compared to Niklaus et al.’s XLM-R (72.2%) [7], it prioritizes India-specific automation. This system aids legal professionals and naive users, reducing judicial workload. Future work could add multi-lingual support, broader datasets, and ensemble models for enhanced accuracy and inclusivity.

## 5 Conclusion

This research presents an integrated system automating court judgment prediction, explanation, summarization, and legal assistance for the Indian legal domain. Achieving 73.74% prediction accuracy, 86.67% summarization accuracy (ROUGE-L F1 0.5184), and high chatbot relevance, it leverages XLNet, BiGRU, InLegalBERT, and Mistral-7B within a Streamlit UI. The system enhances judicial efficiency by providing accurate predictions with transparent rationales, concise summaries, and accessible query responses, addressing India’s case backlog. Compared to prior works, it offers a unified, India-specific solution, surpassing isolated approaches in usability and interpretability. Limitations include English-only support and reliance on ILDC, suggesting multi-lingual expansion and broader datasets as future enhancements. Ensemble models could further boost accuracy, while adaptive summarization thresholds may improve detail retention. Deployed practically, this system sets a benchmark for legal automation, promising scalability and inclusivity to transform judicial workflows and democratize legal access in India.

## References

1. Sharma, S., Singh, P.P.: Domain-specific summarization: Optimizing InLegalBERT for Indian judgment reports (2024)

2. Kuppala, J., Srinivas, K.K., Anudeep, P., Kumar, R.S., Vardhini, P.A.H.: Benefits of artificial intelligence in the legal system and law enforcement. In: 2022 International Mobile and Embedded Technology Conference (MECON), pp. 221–225. IEEE (2022).
3. Shi, T — Keneshloo, Y., Ramakrishnan, N., Reddy, C.K.: Neural abstractive text summarization with sequence-to-sequence models. *ACM Trans. Data Sci.* **2**(1), 1–37 (2021).
4. Rawat, Amar Jeet, Sunil Ghildiyal, and Anil Kumar Dixit. "Topic Modeling Techniques for Document Clustering and Analysis of Judicial Judgements." *International Journal of Engineering Trends and Technology* 70, no. 11 (2022): 163-169.
5. Prabhakar, P.: Supervised summarization of Indian legal documents using T5 (2023)
6. Peinelt, N., Nguyen, D., Liakata, M.: tBERT: Topic models and BERT joining forces for semantic similarity detection. In: *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, pp. 7047–7055 (2020).
7. Niklaus, J., Stürmer, M., Chalkidis, I.: An empirical study on cross-x transfer for legal judgment prediction. *arXiv preprint arXiv:2209.12325* (2022).
8. Nigam, S.K., Deroy, A.: Fact-based court judgment prediction. In: *Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation*, pp. 78–82 (2023).