# NyayaMind: A Framework for Transparent Legal Reasoning and Judgment Prediction in the Indian Legal System

**Anonymous ACL submission**

## Abstract

Court Judgment Prediction and Explanation (CJPE) aims to predict a judicial decision and provide a legally grounded explanation for a given case based on the facts, legal issues, arguments, cited statutes, and relevant precedents. For such systems to be practically useful in judicial or legal research settings, they must not only achieve high predictive performance but also generate transparent and structured legal reasoning that aligns with established judicial practices. In this work, we present **NyayaMind**, an open-source framework designed to enable transparent and scalable legal reasoning for the Indian judiciary. The proposed framework integrates retrieval, reasoning, and verification mechanisms to emulate the structured decision-making process typically followed in courts. Specifically, NyayaMind consists of two main components: a *Retrieval Module* and a *Prediction Module*. The *Retrieval Module* employs a RAG pipeline to identify legally relevant statutes and precedent cases from large-scale legal corpora, while the *Prediction Module* utilizes reasoning-oriented LLMs fine-tuned for the Indian legal domain to generate structured outputs including issues, arguments, rationale, and the final decision. Our extensive results and expert evaluation demonstrate that NyayaMind *significantly improves the quality of explanation and evidence alignment compared to existing CJPE approaches*, providing a promising step toward trustworthy AI-assisted legal decision support systems.

## 1 Introduction

Judicial decision-making is an inherently structured process in which courts analyze statutory provisions, examine precedents, evaluate competing arguments from different parties, and ultimately deliver a reasoned decision grounded in legal principles. In practice, faithfully following this deliberative process – identifying legal issues, weighing arguments, grounding conclusions in statutes and precedents, and articulating coherent reasoning – requires significant time and expertise. The challenge becomes even more pronounced in large judicial systems such as India, where over 50 million cases remain pending as of March 2026[1]. The scale of this backlog places immense pressure on courts, often limiting the time available for detailed legal analysis. As a result, there is growing interest in developing AI systems that can assist legal professionals in structuring and analyzing legal information while preserving the transparency and rigor required in judicial decision-making.

Recent advances in LLMs have shown strong potential in complex reasoning and domain-specific language understanding, leading to increasing exploration of LLMs for legal judgment prediction and explanation. Prior work has introduced domain-specific datasets and models for the legal domain, including *PredEx* (Nigam et al., 2024b) and *INLegalLlama* built on the *NyayaAnumana* corpus (Nigam et al., 2025a). These studies demonstrate that LLMs can capture legal semantics and generate explanations for predicted judgments. Other approaches have incorporated retrieval-based techniques, such as RAG, to integrate statutes and precedents into model predictions. While these methods improve contextual grounding, most existing approaches primarily optimize for outcome prediction or surface-level explanation generation rather than explicitly modeling the structured reasoning process followed by courts.

A fundamental limitation of current Legal AI systems is their lack of transparent and verifiable reasoning. LLMs often generate explanations that appear plausible but may not be faithfully grounded in the retrieved legal sources, leading to hallucinated rationales or incomplete legal reasoning. Moreover, existing approaches rarely model the full deliberative structure of judicial reasoning, such

[1] https://ecourts.gov.in/ecourts_home/

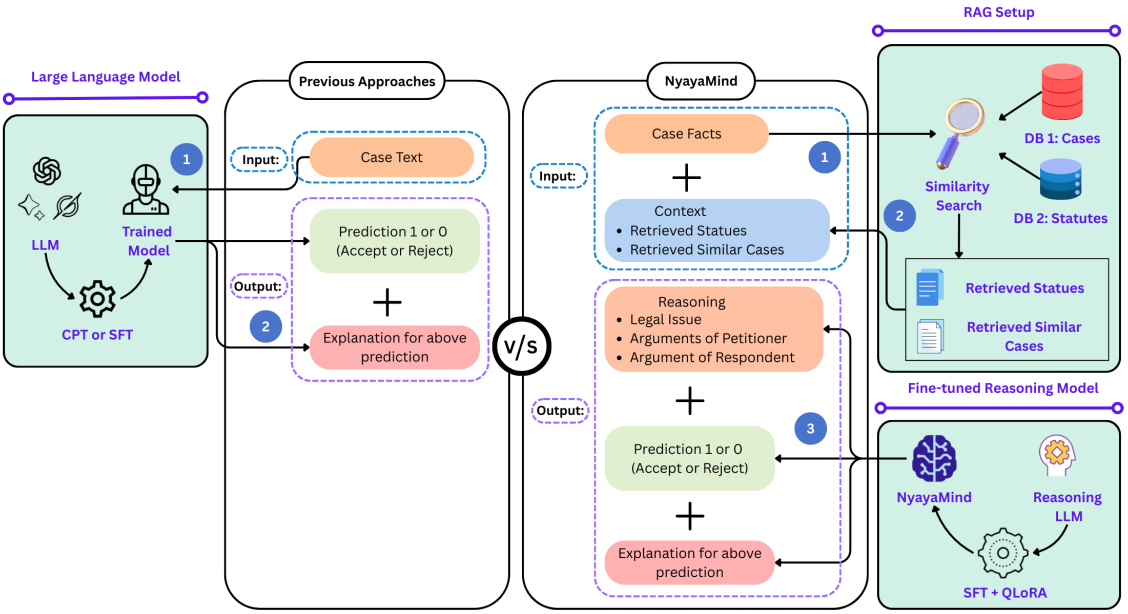


Figure 1: Comparison of our **NyayaMind** System with existing prior approaches.

as identifying legal issues, articulating petitioner and respondent arguments, evaluating statutory provisions, and deriving conclusions through logical justification. As a result, many current systems behave as black-box predictors rather than transparent reasoning assistants capable of supporting legal analysis in practice. Figure 1 illustrates the key differences between our proposed framework and prior approaches.

To address these limitations, we introduce **NyayaMind**, an open-source framework designed to enable transparent and scalable legal reasoning for the Indian judiciary. The proposed system integrates retrieval, reasoning, and verification components to emulate the structured reasoning process typically followed in judicial decision-making. NyayaMind consists of two main modules: a *Retrieval Module* and a *Prediction Module*. The *Retrieval Module* employs a RAG pipeline to identify relevant statutes and precedent cases based on the factual description of a legal dispute. The *Prediction Module* utilizes reasoning-oriented LLMs fine-tuned for the Indian legal domain to generate structured legal outputs, including legal issues, petitioner arguments, respondent arguments, statutory grounding, and the final decision. During fine-tuning, the model is explicitly supervised to follow this deliberative reasoning schema, guiding it to emulate the argumentative structure commonly observed in judicial opinions.

Our key contributions in this work are as follows:

- *A Transparent Legal Reasoning Framework:* We introduce **NyayaMind**, an open-source framework that integrates retrieval, structured reasoning, and verification mechanisms to support transparent legal decision analysis in the judicial context.
- *Structured Reasoning with LLMs for CJPE:* To the best of our knowledge, this is one of the first works in the Indian legal domain to fine-tune reasoning-based LLMs to explicitly follow a structured judicial reasoning paradigm for the CJPE task.
- *Introducing a Large-scale Legal Database:* We introduce **NyayaMind-DB**, a large-scale legal database for the Indian legal domain containing over 16M legal documents that can be leveraged to retrieve relevant statutes and precedents for a RAG-based retrieval module.
- *Comprehensive Evaluation of Retrieval Pipelines:* We systematically evaluate multiple RAG-based retrieval pipelines for statutes and precedents using different vector DBs to analyze their effectiveness in supporting legal reasoning.
- *Comparative Analysis of Reasoning LLMs:* We fine-tune and evaluate four reasoning-oriented LLMs ranging from 4B to 27B parameters, providing insights into their capability to generate structured legal reasoning in the legal domain.
- *Detailed Ablation with Quantization:* We conduct detailed experiments using parameter-efficient LoRA fine-tuning under multiple quantization settings to analyze the tradeoffs between computational efficiency and reasoning quality.
- *Expert Evaluation:* We conduct expert-based

evaluation with legal practitioners to assess the coherence, factual grounding, and legal plausibility of the generated reasoning.

Our codebase is available via this anonymous repository[2].

## 2 Related Work

Legal Judgment Prediction (LJP) has long been studied as an important task in Legal AI, aiming to automatically predict court decisions based on case facts, legal arguments, and supporting evidence. Early work relied on case-based reasoning and symbolic legal models to predict outcomes and provide explanations (Bruninghaus and Ashley, 2003). With the rise of neural methods, researchers began incorporating legal knowledge into deep learning architectures to improve predictive performance and capture complex legal dependencies (Gan et al., 2021; Huang et al., 2021). More recent work has explored combining domain-specific models with large language models (LLMs) to utilize precedents and contextual legal information effectively (Wu et al., 2023).

Another key direction focuses on transparency and explainability, which are critical for deploying AI systems in high-stakes domains such as law. Legal scholars emphasize that AI systems must provide interpretable reasoning to ensure accountability and compliance with legal principles (Papadouli, 2022; Esposito, 2022). Several studies have proposed approaches to enhance explainability by incorporating structured legal information such as entities, citations, or representative prototypes into prediction models (Benedetto et al., 2025; Luo et al., 2023). Policy discussions around emerging regulations, including the EU AI Act, further highlight the importance of transparency and traceability in AI-assisted legal decision-making (Makauskaite-Samuole, 2025).

Recent advances in LLMs have significantly expanded the capabilities of legal reasoning systems. Frameworks such as LegalReasoner integrate knowledge retrieval and reasoning steps for improved judgment prediction (Wang et al., 2024), while discriminative reasoning frameworks help LLMs distinguish between similar legal charges and improve decision accuracy (Deng et al., 2024). Other work explores integrating logical reasoning and semantic knowledge to enhance the reliability of LLM-generated legal responses (Yao et al., 2025). Reinforcement learning and iterative reasoning approaches have also been proposed to strengthen legal reasoning abilities in LLMs, as demonstrated in UniLaw-R1 (Cai et al., 2025). Additionally, emerging agent-based legal AI systems leverage dynamic reasoning and search mechanisms to solve complex legal tasks, such as LawThinker and LRAS (Yang et al., 2026; Zhou et al., 2026). Benchmarks like LEXAM further highlight the need for robust evaluation of legal reasoning capabilities in AI systems (Fan et al., 2025).

In the Indian legal domain, several datasets and models have been introduced to support research in judgment prediction and explanation. PredEx provides expert-annotated data for prediction and explanation tasks (Nigam et al., 2024b), while NyayaAnumana and INLegalLlama offer large-scale corpora and domain-specific language models tailored to Indian legal texts (Nigam et al., 2025a). Furthermore, prior work has studied judgment prediction in realistic legal scenarios where models rely only on information available before the final verdict (Nigam et al., 2024a). Despite these advances, many existing approaches focus either on prediction accuracy or explanation generation independently.

## 3 Task Description

The overall pipeline of `NyayaMind` consists of four sequential sub-tasks: *Retrieval*, *Reasoning*, *Prediction* and *Explanation*.

### 3.1 Retrieval Task

The Retrieval task is formulated as a binary classification problem. Given the case facts $F$, a list of legal statutes $L_s$, and a list of prior cases $L_c$, the objective is to determine which documents are relevant to the current case. Formally, for each statute $s_i$ and case $c_i$ such that $s_i \in L_s$ and $c_i \in L_c$ a label of 1 (accept) or 0 (reject) is generated. A label of 1 (accept) indicates that the document should be included in the set of retrieved documents $L_d$, whereas a label of 0 (reject) indicates that the document should be excluded. The classification decision is based on the semantic similarity between the case facts $F$ and the content of the candidate legal document.

### 3.2 Reasoning Task

Given the case facts $F$ and the retrieved legal documents (statutes and precedents) $L_d$, the system

[2] https://anonymous.4open.science/r/NyayaMind/

performs structured legal reasoning. Specifically, the model generates three key components: the *legal issue* $I$, the *arguments presented by the petitioner* $A_p$, and the *arguments presented by the respondents* $A_r$. These elements together form a structured legal reasoning $R$, where $I, A_p, A_r \in R$. This intermediate reasoning step is essential for improving the transparency, interpretability, and reliability of the NyayaMind system, as it explicitly showcases the argumentative structure underlying legal decisions.

### 3.3 Prediction Task

The Prediction task determines the outcome of the legal appeal. Given the case facts $F$ and the structured reasoning representation $R$, the model predicts a binary decision label $y \in \{0, 1\}$. A label 0 indicates that the appeal is rejected, whereas a label 1 indicates that the appeal is accepted.

This prediction stage leverages both the factual context and the generated reasoning structure, allowing the system to produce decisions that are informed by explicit legal analysis rather than purely statistical correlations.

### 3.4 Explanation Task

The final stage of the pipeline generates a comprehensive legal explanation $E$ supporting the predicted outcome $y$. The explanation integrates information from the case facts $F$, the structured reasoning $R$, and the retrieved legal documents $L_d$. In particular, the generated explanation highlights the legal issues involved, summarizes the arguments presented by both parties, references relevant statutes and precedents, and articulates the rationale behind the final decision.

By jointly modeling retrieval, reasoning, prediction, and explanation, NyayaMind aims to provide interpretable and legally grounded decision support. This structured formulation allows the system to produce transparent outputs that more closely resemble the reasoning process used by human judges in the Indian legal system.

## 4 Dataset

Our study leverages two primary datasets for developing the NyayaMind system: **NyayaMind-DB** and **LegalSeg** (Nigam et al., 2025b). The former is used to support large-scale retrieval of statutes and precedents, while the latter is used for fine-tuning reasoning-oriented LLMs for structured legal reasoning and judgment prediction.

### 4.1 The NyayaMind-DB Dataset

We construct **NyayaMind-DB** as a large-scale legal corpus tailored for the Indian judicial domain. The dataset comprises over 16 million legal documents, including approximately 60K judgments from the Supreme Court of India and nearly 16M judgments from multiple High Courts. In addition, it includes around 1K Central Acts and approximately 7K State Acts. The judicial documents are collected from the *E-Courts* platform[3], while statutory documents are sourced from *IndiaCode*[4].

The NyayaMind-DB serves as the backbone of our Retrieval module, enabling large-scale retrieval of legally relevant statutes and precedents using semantic similarity between case facts and candidate documents. Importantly, the dataset contains not only final judgments but also intermediate court orders associated with each case, thereby providing a richer representation of the judicial process. This diversity and scale allow the system to capture nuanced legal context and improve the grounding of generated reasoning.

### 4.2 Dataset used for fine-tuning

For training the reasoning component of NyayaMind, we utilize the LegalSeg dataset (Nigam et al., 2025b), which focuses on semantic segmentation of legal judgments through rhetorical role classification in the Indian legal context. The dataset consists of 7,120 cases, split into 4,984 training samples, 712 validation samples, and 1,424 test samples. Each case is annotated with structured components such as *Facts*, *Issues*, *Reasoning*, *Arguments of Petitioner*, *Arguments of Respondent*, and *Decision*.

To align with our task formulation, we transform the dataset into a structured input-output format suitable for reasoning-based LLMs. Specifically, we employ an LLM to rewrite the annotated fields into a conversational and coherent format, while preserving their legal semantics. The dataset also includes additional contextual information such as cited statutes (*section_texts*) and cited precedent cases (*cited_case_judgments*), which are incorporated into the model inputs to strengthen legal grounding.

During fine-tuning, the model is trained to generate structured legal reasoning by first identifying issues, then articulating arguments from both par-

[3] https://judgments.ecourts.gov.in/
[4] https://indiacode.nic.in/

ties, and finally producing a decision along with a justified explanation. The prompt templates used for different models are detailed in Tables 9, 10, 11, and 12 in the Appendix. Additionally, a detailed breakdown of token distribution across dataset components, including input and target texts, is provided in Table 4. This structured supervision enables the model to learn a faithful approximation of judicial reasoning rather than relying on shallow pattern matching.

## 5 Methodology

The NyayaMind framework is designed as a multi-stage pipeline that closely mirrors the structured reasoning process followed in judicial decision-making. As illustrated in Figure 2, the system consists of four sequential stages: *User Input*, *Retrieval*, *Reasoning*, and *Prediction & Explanation*. This design ensures that the final decision is not generated in isolation, but is instead grounded in retrieved legal evidence and structured intermediate reasoning.

Given the case facts $F$ as input, the first step involves retrieving relevant legal knowledge. To this end, we implement three distinct Retrieval-Augmented Generation (RAG) architectures: *Milvus*, *Endee*, and *Vespa*. These systems differ in indexing strategies, similarity metrics, chunking mechanisms, and retrieval pipelines. Each RAG setup combines dense vector similarity search with lexical matching (BM25), enabling robust retrieval of statutes and precedent cases that are semantically and contextually aligned with the input case. This multi-system design allows us to systematically evaluate the impact of retrieval quality on downstream reasoning and prediction. For reproducibility, we provide detailed configurations of indexing, retrieval parameters, and integration pipelines in Appendix A, with a comparative summary presented in Table 8.

The retrieved documents $L_d$, along with the original case facts $F$, are then provided as input to the reasoning component. To ensure that the generated outputs follow a principled legal structure, we fine-tune reasoning-oriented Large Language Models (LLMs) using Supervised Fine-Tuning (SFT) combined with parameter-efficient techniques such as LoRA (Hu et al., 2021) and QLoRA (Dettmers et al., 2023). During training, the models are explicitly guided to follow a structured legal reasoning schema, which includes identifying legal issues, articulating arguments from the petitioner and respondent, grounding reasoning in statutory provisions, and deriving a final decision. This process is implemented through structured prompting and supervision within dedicated reasoning segments ("**think tokens**"), encouraging the model to perform step-by-step deliberation rather than direct answer generation.

Finally, the model produces the prediction $y$ (accept/reject) along with a detailed explanation $E$ that is grounded in both the retrieved legal evidence and the generated reasoning structure. By tightly coupling retrieval, reasoning, and prediction, NyayaMind ensures that the generated outputs are not only accurate but also interpretable and verifiable. This integrated design enables NyayaMind to function as a transparent legal reasoning system, bridging the gap between black-box prediction models and the structured analytical processes required in real-world judicial settings.

## 6 Experimental Setup

This section describes the experimental setup used to adapt reasoning-based LLMs for the CJPE task in the Indian legal domain.

### 6.1 Models

We evaluate four recent reasoning-oriented or instruction-tuned LLMs of varying scales: (i) *DeepSeek-R1-Distill-Qwen-14B*[5], (ii) *Phi-4-mini-reasoning*[6], (iii) *Phi-4-reasoning*[7], and (iv) *Qwen3.5-27B*[8]. These models were selected to compare different model families and parameter scales in terms of their ability to perform structured legal reasoning, outcome prediction, and explanation generation.

### 6.2 Training and Inference Setup

Our training setup is designed to adapt reasoning LLMs to the CJPE task while maintaining computational efficiency and preserving their capacity for multi-step structured legal analysis.

**Parameter-efficient Training**: Instead of full-parameter fine-tuning, we adopt a parameter-efficient fine-tuning strategy based on *LoRA* (Hu et al., 2021) and *QLoRA* (Dettmers et al., 2023).

[5] https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-14B
[6] https://huggingface.co/microsoft/Phi-4-mini-reasoning
[7] https://huggingface.co/microsoft/Phi-4-reasoning
[8] https://huggingface.co/Qwen/Qwen3.5-27B

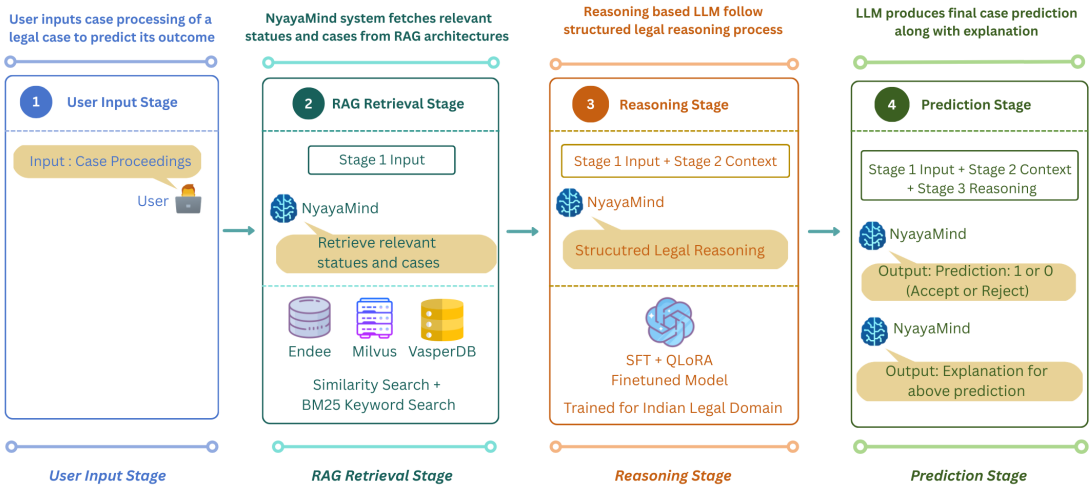


Figure 2: Representation of different stages in the flow of our **NyayaMind** System.

LoRA adapters are inserted into both attention projection layers and feed-forward projection layers, allowing the models to specialize to the legal domain with substantially reduced memory overhead. This setup enables efficient adaptation of large reasoning models while retaining their general reasoning capabilities.

**Quantization Strategy**: To study the trade-off between efficiency and reasoning quality, we experiment with multiple precision settings for *DeepSeek-R1-Distill-Qwen-14B*, namely 4-bit, 8-bit, 16-bit, and 32-bit quantization. The 4-bit version uses the *DeepSeek-R1-Distill-Qwen-14B-unsloth-bnb-4bit*[9] variant provided through the *Unsloth*[10] framework. In contrast, *Phi-4-mini-reasoning*, *Phi-4-reasoning*, and *Qwen3.5-27B* are fine-tuned in 16-bit precision. This design enables a systematic comparison of memory usage, training feasibility, and downstream reasoning fidelity across quantization levels. For *Qwen3.5-27B*, we further use a longer context window of 32K tokens to better accommodate lengthy legal inputs.

**Hyperparameter Configuration**: Training is implemented using the Unsloth framework with distributed training support for efficient optimization. We use gradient accumulation to handle long-context inputs under constrained GPU memory, and apply optimized attention implementations for selected models to improve throughput. The complete training and inference hyperparameters are reported in Table 3 in the Appendix.

**Prompt Design for Legal Reasoning**: To guide the models toward structured judicial reasoning, we design task-specific prompts tailored to the CJPE setting. The prompts integrate multiple legally relevant fields from the training data, including simplified case facts, cited statutory provisions, cited prior judgments, legal issues, arguments from the petitioner and respondent, the final decision, and the associated judicial reasoning. Rather than directly predicting the outcome from raw facts, the model is trained to follow an explicit reasoning trajectory through dedicated intermediate "think tokens," which correspond to key legal reasoning steps such as issue identification and argument analysis. The model is then trained to generate both the final decision and a supporting natural language explanation. Prompt templates for all model variants are provided in Tables 9, 10, 11, and 12.

**Inference Stage**: At inference time, `NyayaMind` takes as input the factual description of a legal case. This input is augmented with relevant statutes and precedent cases retrieved through our RAG-based retrieval module, which combines semantic and lexical retrieval strategies over Indian legal corpora. The retrieved legal context is then passed to the fine-tuned reasoning model, which generates structured intermediate reasoning followed by the final judicial outcome and its explanation. In this way, the system grounds its prediction in both retrieved legal evidence and an explicit reasoning process, improving transparency and legal plausibility. The complete inference pipeline is illustrated in Figure 2.

## 7 Evaluation Metrics

We evaluate the performance of `NyayaMind` using both automatic metrics and expert-based human

[9] https://huggingface.co/unsloth/DeepSeek-R1-Distill-Qwen-14B-unsloth-bnb-4bit

[10] https://unsloth.ai/

evaluation to comprehensively assess prediction accuracy, reasoning quality, and explanation fidelity.

**Automatic Evaluation:** We employ a combination of lexical and semantic similarity metrics to evaluate the generated outputs. For lexical overlap, we report *ROUGE* scores (ROUGE-1, ROUGE-2, and ROUGE-L) (Lin, 2004), *BLEU* (Papineni et al., 2002), and *METEOR* (Banerjee and Lavie, 2005). These metrics capture n-gram overlap, fluency, and word-order similarity between the generated outputs and reference texts.

To assess semantic alignment, we use *BERTScore* (Zhang et al., 2020), which computes contextual similarity between generated and reference texts using pretrained language model embeddings. Additionally, we report *BLANC* (Vasilyev et al., 2020), a reference-free metric that evaluates how well the generated text supports understanding of the original input. Together, these metrics provide a balanced evaluation of both surface-level correctness and deeper semantic consistency of the generated explanations.

**Expert Evaluation:** To evaluate the legal validity and interpretability of the generated outputs, we conduct an expert evaluation with two legal professionals. The evaluation focuses on both intermediate reasoning components, *legal issue*, *arguments of petitioner*, and *arguments of respondent*, and final outputs, *prediction* and *explanation*. Each component is rated on a 1–10 Likert scale based on criteria such as accuracy, relevance, coherence, and completeness. In addition to component-wise scores, an overall score is assigned to reflect the holistic quality of the system output. A score of 1 indicates poor or irrelevant output, whereas a score of 10 indicates high-quality reasoning and explanation, potentially comparable to or exceeding the reference text. This expert-driven evaluation complements automatic metrics by capturing domain-specific correctness and practical usefulness in the Indian legal context.

## 8 Result and Analysis

We evaluate the performance of `NyayaMind` across multiple components of the CJPE pipeline, including legal issue identification, argument generation, prediction, and explanation. The results are reported using both automatic metrics and expert evaluation, as summarized in Table 1.

**Overall Performance:** Across all components, `Qwen3.5-27B` consistently outperforms

| Models | Lexical Evaluation (%) | | | | | Semantic Evaluation (%) | | Expert |
|---|---|---|---|---|---|---|---|---|
| | R-1 | R-2 | R-L | BLEU | METEOR | BERTScore | BLANC | Rating |
| **Generated Legal Issue vs Ground Truth** | | | | | | | | |
| `DeepSeek-R1-Distill-Qwen-14B` | 27.06 | 8.99 | 18.81 | 3.12 | 16.87 | 52.98 | 7.75 | 6.25 |
| `Qwen3.5-27B` | **27.73** | **9.68** | **19.23** | **3.56** | **17.84** | **53.21** | **8.37** | **6.95** |
| **Generated Petitioner Arguments vs Ground Truth** | | | | | | | | |
| `DeepSeek-R1-Distill-Qwen-14B` | 30.65 | 10.08 | 18.30 | 3.90 | **22.19** | 57.09 | **9.85** | 6.50 |
| `Qwen3.5-27B` | **32.18** | **11.29** | **19.20** | **4.22** | 21.23 | **58.25** | 9.36 | **6.90** |
| **Generated Respondent Arguments vs Ground Truth** | | | | | | | | |
| `DeepSeek-R1-Distill-Qwen-14B` | 24.59 | 6.95 | 15.80 | 2.07 | **16.44** | 53.95 | 5.91 | 6.20 |
| `Qwen3.5-27B` | **25.91** | **8.08** | **16.77** | **2.38** | 15.93 | **55.20** | **7.14** | **6.90** |
| **Generated Prediction vs Ground Truth** | | | | | | | | |
| `DeepSeek-R1-Distill-Qwen-14B` | 43.89 | 27.22 | 36.67 | 18.27 | 35.70 | 66.59 | **22.89** | 5.50 |
| `Qwen3.5-27B` | **46.63** | **28.57** | **38.31** | **20.14** | **36.73** | **67.30** | 22.73 | **6.90** |
| **Generated Explanation vs Ground Truth** | | | | | | | | |
| `DeepSeek-R1-Distill-Qwen-14B` | **32.80** | **12.63** | **18.64** | **4.39** | 18.05 | 60.70 | 8.25 | 6.12 |
| `Qwen3.5-27B` | 29.95 | 11.89 | 17.04 | 4.35 | **19.83** | **61.90** | **9.77** | **7.00** |

Table 1: Performance comparison of models on legal judgment explanation tasks. Lexical metrics include ROUGE, BLEU, and METEOR, while semantic similarity is measured using BERTScore and BLANC. Expert scores reflect human evaluation by legal professionals. Best scores are shown in bold.

| Generated Component | Expert 1 | Expert 2 | Average |
|---|---|---|---|
| **`DeepSeek-R1-Distill-Qwen-14B` (32-bit)** | | | |
| Legal Issue | 6.48 | 6.00 | 6.24 |
| Arguments of Petitioner | 6.44 | 6.75 | 6.60 |
| Arguments of Respondent | 6.12 | 6.83 | 6.48 |
| Prediction | 6.60 | 6.25 | 6.43 |
| Explanation | 5.88 | 5.89 | 5.88 |
| **`Qwen3.5-27B`** | | | |
| Legal Issue | 6.40 | 7.80 | 7.10 |
| Arguments of Petitioner | 6.20 | 7.80 | 7.00 |
| Arguments of Respondent | 6.56 | 7.10 | 6.83 |
| Prediction | 6.88 | 7.60 | 7.24 |
| Explanation | 6.24 | 7.20 | 6.72 |

Table 2: Expert evaluation of generated legal components across 25 sampled cases. Scores are reported on a 1–10 Likert scale.

`DeepSeek-R1-Distill-Qwen-14B (32-bit)` on most lexical and semantic metrics. In particular, improvements are observed in ROUGE, BLEU, and BERTScore across all reasoning components, indicating better alignment with ground truth and improved semantic consistency. Moreover, expert ratings also favor `Qwen3.5-27B`, demonstrating that improvements in automatic metrics translate into more coherent and legally meaningful outputs.

**Legal Issue and Argument Generation:** For intermediate reasoning components such as *legal issue*, *petitioner arguments*, and *respondent arguments*, `Qwen3.5-27B` achieves higher scores across most evaluation metrics (Table 1). Notably, gains in ROUGE-2 and BERTScore indicate better capture of key legal phrases and improved contextual understanding. However, `DeepSeek-R1-Distill-Qwen-14B` achieves slightly higher METEOR scores in some cases, suggesting that it may generate more lexically varied

outputs, albeit with slightly lower semantic alignment.

**Prediction Performance:** For the prediction task, `Qwen3.5-27B` achieves superior performance across all lexical and semantic metrics, with notable improvements in BLEU and ROUGE scores. This indicates that larger reasoning-oriented models are better at aligning their predictions with ground truth decisions when supported by structured reasoning and retrieved legal context. Expert evaluation further supports this observation, with `Qwen3.5-27B` achieving higher ratings for prediction quality.

**Explanation Quality:** In the explanation generation task, we observe a trade-off between lexical overlap and semantic richness. While `DeepSeek-R1-Distill-Qwen-14B` achieves slightly higher ROUGE scores, `Qwen3.5-27B` outperforms in METEOR, BERTScore, BLANC, and expert ratings. This suggests that `Qwen3.5-27B` generates explanations that are more semantically meaningful, coherent, and aligned with legal reasoning, even if they do not strictly match the reference text at the surface level.

**Expert Evaluation Analysis:** Table 2 presents detailed expert evaluation across different components. `Qwen3.5-27B` consistently achieves higher average scores across all categories, particularly in *prediction* (7.24) and *legal issue* (7.10), indicating stronger reasoning capabilities and better alignment with judicial interpretation. The relatively lower scores for explanation (6.72) suggest that generating high-quality legal explanations remains a challenging task even for large models. In comparison, `DeepSeek-R1-Distill-Qwen-14B` produces moderately coherent outputs but lacks consistency in explanation quality.

**Impact of Quantization:** We observe a clear degradation in reasoning quality with aggressive quantization. The 4-bit variant of `DeepSeek-R1-Distill-Qwen-14B` frequently produces repetitive and semantically redundant outputs, indicating instability in generation. The 8-bit variant shows partial improvement but still struggles to adhere to structured reasoning templates. In contrast, higher precision variants (16-bit and 32-bit) generate more coherent, logically structured, and template-consistent outputs. This highlights the importance of maintaining sufficient numerical precision for tasks requiring multi-step reasoning and structured generation. Additional qualitative examples illustrating these effects are provided in Table 13 in the Appendix.

**Key Observations:** (i) Larger models demonstrate stronger capability in structured legal reasoning and explanation generation. (ii) Semantic metrics and expert evaluations provide more reliable insights than purely lexical metrics for legal tasks. (iii) Retrieval-grounded reasoning significantly improves both prediction accuracy and explanation quality. (iv) Quantization introduces a trade-off between efficiency and reasoning fidelity, with lower-bit models exhibiting reduced stability and coherence.

Overall, the results demonstrate that `NyayaMind` effectively enables structured, transparent, and evidence-grounded legal reasoning, with significant improvements observed when leveraging larger reasoning-oriented LLMs and higher-precision configurations.

## 9 Conclusion and Future Work

In this work, we presented **`NyayaMind`**, a framework for Court Judgment Prediction and Explanation (CJPE) that integrates retrieval, structured reasoning, and prediction for the Indian legal domain. Unlike prior approaches that focus primarily on prediction or surface-level explanations, `NyayaMind` explicitly models judicial reasoning by incorporating legal issues, arguments, and grounding in statutes and precedents. Our results demonstrate that combining RAG with reasoning-oriented LLMs improves semantic alignment, explanation quality, and expert-rated performance, while highlighting the importance of model scale and precision for stable reasoning.

Despite these contributions, the current system is limited to English-language data and does not fully enforce verification of reasoning against legal sources. Future work will focus on extending the framework to multilingual Indian legal settings, incorporating reinforcement learning and verification mechanisms to improve factual consistency, and exploring agentic or iterative reasoning for deeper legal analysis. These directions aim to further enhance the transparency, reliability, and practical applicability of Legal AI systems.

## Limitations

This study has several limitations that should be considered when interpreting the results. First, `NyayaMind` is constrained by the context length

of the underlying language models. In our current setup, the maximum input length is limited to 16,384 tokens and the output to 4,096 tokens. However, real-world legal documents often exceed these limits, typically ranging from 32K to 64K tokens. As a result, documents must be truncated or selectively processed, which may omit relevant information and affect the completeness of the generated reasoning and explanations.

Second, the framework is limited to English-language legal data. This is primarily due to the scarcity of large-scale, high-quality annotated datasets in Indian regional languages. Consequently, the applicability of the system remains restricted in multilingual judicial settings, particularly in district and subordinate courts where regional languages are widely used.

Finally, while `NyayaMind` improves transparency through structured reasoning and retrieval grounding, it does not enforce strict verification of generated reasoning against legal sources. This may still lead to partially grounded or inconsistent explanations in complex cases.

Addressing these limitations, particularly improving long-context handling, enabling multilingual support, and incorporating stronger verification mechanisms, remains important for enhancing the robustness and real-world applicability of the proposed system.

## Ethical Considerations

The development and deployment of `NyayaMind` in the legal domain raises several important ethical considerations. First, legal decision-making is a high-stakes process, and any AI-assisted system must be used with caution. `NyayaMind` is designed as a decision-support tool and not as a replacement for human judges or legal professionals. The outputs generated by the system should be interpreted as advisory and must always be reviewed by qualified legal experts before use in real-world settings.

Second, the quality and bias of training data can influence model behavior. Although our datasets are sourced from publicly available legal documents, they may reflect historical biases present in judicial decisions. As a result, the model may inadvertently learn and propagate such biases. Addressing fairness and bias mitigation in legal AI systems remains an important direction for future work.

Third, while `NyayaMind` improves transparency through structured reasoning and retrieval grounding, large language models may still generate hallucinated or partially incorrect information. This could lead to misleading interpretations if not carefully validated. To mitigate this, we emphasize the importance of grounding outputs in retrieved legal sources and recommend incorporating additional verification mechanisms in future iterations.

Fourth, privacy and data sensitivity are important considerations. Although the datasets used in this work consist of publicly available judgments and statutes, care must be taken when extending the system to other domains that may involve sensitive or personal information.

Finally, the deployment of Legal AI systems should align with regulatory and ethical frameworks governing AI usage. Ensuring accountability, transparency, and human oversight is critical for building trustworthy AI systems in the legal domain. We advocate for responsible use of such technologies, with clear boundaries on their role in supporting, not replacing, judicial processes.

## References


Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Irene Benedetto, Alkis Koudounas, Lorenzo Vaiani, Eliana Pastor, Luca Cagliero, Francesco Tarasconi, and Elena Baralis. 2025. Boosting court judgment prediction and explanation using legal entities: I. benedetto et al. *Artificial Intelligence and Law*, 33(3):605–640.

Stefanie Bruninghaus and Kevin D. Ashley. 2003. Predicting outcomes of case based legal arguments. In *Proceedings of the 9th International Conference on Artificial Intelligence and Law*, ICAIL '03, page 233–242, New York, NY, USA. Association for Computing Machinery.

Hua Cai, Shuang Zhao, Liang Zhang, Xuli Shen, Qing Xu, Weilin Shen, Zihao Wen, and Tianke Ban. 2025. Unilaw-r1: A large language model for legal reasoning with reinforcement learning and iterative inference. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18128–18142.

Chenlong Deng, Kelong Mao, Yuyao Zhang, and Zhicheng Dou. 2024. Enabling discriminative reasoning in LLMs for legal judgment prediction. In *Find-*

*ings of the Association for Computational Linguistics: EMNLP 2024*, pages 784–796, Miami, Florida, USA. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

Elena Esposito. 2022. Transparency versus explanation: The role of ambiguity in legal ai. *Journal of Cross-disciplinary Research in Computational Law*, 1(2).

Yu Fan, Jingwei Ni, Jakob Merane, Yang Tian, Yoan Hermstrüwer, Yinya Huang, Mubashara Akhtar, Etienne Salimbeni, Florian Geering, Oliver Dreyer, et al. 2025. Lexam: Benchmarking legal reasoning on 340 law exams. *arXiv preprint arXiv:2505.12864*.

Leilei Gan, Kun Kuang, Yi Yang, and Fei Wu. 2021. Judgment prediction via injecting legal knowledge into neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 12866–12874.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Yunyun Huang, Xiaoyu Shen, Chuanyi Li, Jidong Ge, and Bin Luo. 2021. Dependency learning for legal judgment prediction with a unified text-to-text transformer. *arXiv preprint arXiv:2112.06370*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chu Fei Luo, Rohan Bhambhoria, Samuel Dahan, and Xiaodan Zhu. 2023. Prototype-based interpretability for legal citation prediction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4883–4898, Toronto, Canada. Association for Computational Linguistics.

Gintare Makauskaite-Samuole. 2025. Transparency in the labyrinths of the eu ai act: Smart or disbalanced? *Access to Just. E. Eur.*, page 38.

Shubham Kumar Nigam, Deepak Patnaik Balaramamahanthi, Shivam Mishra, Noel Shallum, Kripabandhu Ghosh, and Arnab Bhattacharya. 2025a. NyayaAnumana and INLegalLlama: The largest Indian legal judgment prediction dataset and specialized language model for enhanced decision analysis. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 11135–11160, Abu Dhabi, UAE. Association for Computational Linguistics.

Shubham Kumar Nigam, Aniket Deroy, Subhankar Maity, and Arnab Bhattacharya. 2024a. Rethinking legal judgement prediction in a realistic scenario in the era of large language models. In *Proceedings of the Natural Legal Language Processing Workshop 2024*, pages 61–80, Miami, FL, USA. Association for Computational Linguistics.

Shubham Kumar Nigam, Tanmay Dubey, Govind Sharma, Noel Shallum, Kripabandhu Ghosh, and Arnab Bhattacharya. 2025b. LegalSeg: Unlocking the structure of Indian legal judgments through rhetorical role classification. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1129–1144, Albuquerque, New Mexico. Association for Computational Linguistics.

Shubham Kumar Nigam, Anurag Sharma, Danush Khanna, Noel Shallum, Kripabandhu Ghosh, and Arnab Bhattacharya. 2024b. Legal judgment reimagined: PredEx and the rise of intelligent AI interpretation in Indian courts. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4296–4315, Bangkok, Thailand. Association for Computational Linguistics.

Vasiliki Papadouli. 2022. Transparency in artificial intelligence: A legal perspective. *Journal of Ethics and Legal Technologies*, 4:25–40.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Oleg V. Vasilyev, Vedant Dharnidharka, and John Bohannon. 2020. Fill in the BLANC: human-free quality estimation of document summaries. *CoRR*, abs/2002.09836.

Xuran Wang, Xinguang Zhang, Vanessa Hoo, Zhouhang Shao, and Xuguang Zhang. 2024. Legalreasoner: A multi-stage framework for legal judgment prediction via large language models and knowledge integration. *IEEE Access*, 12:166843–166854.

Yiquan Wu, Siying Zhou, Yifei Liu, Weiming Lu, Xiaozhong Liu, Yating Zhang, Changlong Sun, Fei Wu, and Kun Kuang. 2023. Precedent-enhanced legal judgment prediction with LLM and domain-model collaboration. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12060–12075, Singapore. Association for Computational Linguistics.

Xinyu Yang, Chenlong Deng, Tongyu Wen, Binyu Xie, and Zhicheng Dou. 2026. Lawthinker: A deep research legal agent in dynamic environments. *arXiv preprint arXiv:2602.12056*.

Rujing Yao, Yang Wu, Chenghao Wang, Jingwei Xiong, Fang Wang, and Xiaozhong Liu. 2025. Elevating legal LLM responses: Harnessing trainable logical structures and semantic knowledge with legal reasoning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5630–5642, Albuquerque, New Mexico. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yujin Zhou, Chuxue Cao, Jinluan Yang, Lijun Wu, Conghui He, Sirui Han, and Yike Guo. 2026. Lras: Advanced legal reasoning with agentic search. *arXiv preprint arXiv:2601.07296*.

# A Implementation Details

This section provides implementation details of the retrieval and training components used in `NyayaMind`. We describe three retrieval architectures, Endee, Milvus, and Vespa, along with their indexing strategies, hybrid retrieval mechanisms, and system configurations. Additionally, we present training and inference settings for the reasoning-oriented LLMs, as well as dataset statistics to support reproducibility.

## A.1 Dense Retrieval with Endee

To facilitate effective semantic retrieval of legal documents, we utilized Endee for storing and querying dense vector representations. Each record included a unique string identifier, a 768-dimensional embedding created with the snowflake/snowflake-arctic-embed-m-v2.0, and the original document text (capped at 60,000 characters) along with structured metadata fields. An HNSW (Hierarchical Navigable Small World) index was set up with M = 16, ef_construction = 128, and cosine similarity as the distance metric, employing FP16 (medium) storage precision to optimize both accuracy and memory usage. Documents underwent preprocessing through Unicode normalization, conversion to lowercase, and whitespace consolidation, followed by chunking with a RecursiveCharacterTextSplitter using a chunk size of 4,096 tokens and an overlap of 100 tokens, with the model's tokenizer serving as the length function. During query execution, the input text was encoded using the same snowflake/snowflake-arctic-embed-m-v2.0, and the resulting vector was compared against the HNSW index with ef = 128 and top_k = 10, retrieving the top_k document chunks that were most semantically similar based on cosine similarity.

This RAG setup is composed of four distinct collections, which are as follows: SC_Judgments_DB, Central_Acts, State_Acts, HC_Judgments_DB. The Central_Acts collection comprises 36,991 segments. The HC_Judgments_DB collection holds 69,017 segments, while the SC_Judgments_DB collection contains 81,331 segments. The State_Acts collection includes 1,50,748 segments. In addition to the embedding vector, these collections also retain metadata information. The schema of each collection is detailed in table 5.

| Parameter | Value |
|---|---|
| **System** | |
| GPUs | NVIDIA H200 |
| **LoRA Parameters** | |
| Rank | 16 |
| Alpha | 16 |
| Dropout | 0 |
| Target Modules | q, k, v, o, gate, up, down projections |
| **Training Parameters** | |
| Epochs | 8 |
| Batch Size | 1 |
| Max Sequence Length | 16,384 |
| Learning Rate | $1 \times 10^{-4}$ |
| Scheduler | Cosine annealing |
| Warmup Ratio | 0.1 |
| Weight Decay | 0.01 |
| Early Stopping Patience | 3 |
| Optimizer | AdamW (8-bit) |
| **Inference Parameters** | |
| Max New Tokens | 4,096 |
| No Repeat N-gram Size | 6 |
| Repetition Penalty | 1.1 |

Table 3: Training and inference hyperparameters used for the NYAYAMIND system.

### A.2 Hybrid Retrieval with Milvus

We implemented Milvus as a self-hosted vector database to store and query dense vector representations. Each collection included a unique integer identifier, a 768-dimensional embedding created using the Snowflake/snowflake-arctic-embed-m-v2.0 model, and the original document text along with structured metadata fields. For collections such as Central Acts and Supreme Court, an IVF_FLAT index was set up with nlist = 2048 and L2 (Euclidean) distance, enabling fast approximate nearest neighbor searches by dividing vectors into 2048 Voronoi cells. For collections like High Courts and State Acts, a FLAT (brute-force) index was employed for precise searches. Documents were divided into chunks using a RecursiveCharacterTextSplitter with a chunk size of 4,096 tokens and an overlap of 100 tokens, utilizing the model's tokenizer as the length function. During query time, the input was initially encoded and matched against the Milvus index with nprobe = 10, retrieving the top_k = 500 semantically relevant candidates. BM25 was then selectively applied only to this vector pre-filtered subset, not the entire corpus, to capture lexical overlap while managing computational costs. The BM25 implementation used the Okapi BM25 formula with parameters b = 0.7 and k1 = 1.6, built on sklearn's TfidfVectorizer, and applied Porter stemming during tokenization. Documents were re-chunked into 512-character segments with a 100-character overlap for BM25 scoring. The ranked outputs from vector search and BM25 were combined using Reciprocal Rank Fusion (RRF) with a smoothing constant k = 60, resulting in a unified ranking that integrates both semantic similarity and lexical relevance signals. This three-stage approach — vector retrieval, subset BM25, and RRF fusion, balances retrieval accuracy with computational efficiency.

The RAG configuration comprises four main legal categories indexed in the Milvus vector database: Central Acts, Supreme Court Judgments, High Court Judgments, and State Acts, which together encompass 60 collections. The Central Acts category includes a single collection with 37,202 indexed segments. The Supreme Court category also has one collection, containing 81,625 segments. The High Courts category is divided state-wise into 25 distinct collections, totaling 1,577,859 indexed segments. The State Acts category consists of 33 state-wise collections with 150,878 indexed segments. Overall, the Milvus vector database holds 1,847,564 legal text segments across all collections. Each segment is stored with its corresponding embedding vector and related metadata, such as court name, act title, citation details, jurisdiction, and section identifiers where applicable. The schema design for each collection is outlined in Table 7.

### A.3 Parallel Hybrid Retrieval with Vespa

The Vespa architecture implements a native hybrid approach where semantic and lexical searches operate in parallel across the entire corpus. We implemented Vespa as a self-hosted search engine within a Docker container. Each document schema included an HNSW-indexed 768-dimensional embedding, created using the snowflake/snowflake-arctic-embed-m-v2.0 model, alongside BM25-indexed text fields and structured metadata. The HNSW index was set up with a Euclidean distance metric, max_links_per_node = 16, and neighbors_to_explore_at_insert = 200, utilizing paged attribute storage for efficient memory use. Documents were divided into chunks using a RecursiveCharacterTextSplitter with a chunk size of 8,192 tokens and an overlap of 100–200 tokens, using the model's tokenizer as the length function.

Each schema specified two declarative rank profiles: a semantic profile employing ‘closeness(field, embedding)‘ for ANN-based nearest neighbor retrieval, and a BM25 profile for lexical scoring on text fields. During query time, the input was sent as two separate Vespa queries — a vector-based nearest neighbor search and a BM25 keyword search, each providing the top-10 results. The two ranked result lists were then combined client-side using Reciprocal Rank Fusion (RRF) with a smoothing constant of k = 60, resulting in a unified ranking that integrates both semantic similarity and lexical relevance signals. Unlike the Milvus-based architecture, BM25 was applied natively across the entire Vespa corpus in parallel with vector search rather than on a pre-filtered subset, and no cross-encoder re-ranking was used, leading to a two-stage retrieval pipeline.

The RAG setup is structured into four main legal categories: Central Acts, Supreme Court Judgments, High Court Judgments, and State Acts, all integrated under the legalknowledgebase application. The Central Acts category is represented by a single schema (centralacts) that includes 37,202 indexed segments. The Supreme Court category is managed through the scjudgments schema, which contains 81,625 indexed segments. The High Court collection is housed within a specific judgments schema, comprising 1,577,859 indexed segments sourced from 25 High Courts. The State Acts category is organized by state into 33 distinct schemas (sa_state), together holding 150,878 indexed segments. The schema of each collection is detailed in table 6

The similarities and differences between the 3 RAG approaches are outlined in Table 8.

| Dataset Field | Token Count Percentiles | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **P50** | **P75** | **P80** | **P85** | **P90** | **P95** | **P99** | **Max** |
| **Train Dataset** | | | | | | | | |
| Input Text | 18,295 | 68,606 | 88,391 | 118,401 | 175,102 | 308,669 | 828,490 | 3,117,110 |
| Target Text | 2,096 | 3,274 | 3,747 | 4,407 | 5,485 | 8,289 | 15,243 | 33,839 |
| Case Text | 5,724 | 9,738 | 11,081 | 13,168 | 16,468 | 24,478 | 56,548 | 304,724 |
| Simplified Facts | 992 | 1,561 | 1,744 | 1,994 | 2,383 | 3,084 | 6,528 | 18,000 |
| Simplified Legal Issue | 61 | 116 | 143 | 173 | 218 | 295 | 563 | 8,724 |
| Petitioner Arguments | 360 | 678 | 780 | 946 | 1,235 | 1,679 | 3,993 | 13,970 |
| Respondent Arguments | 259 | 496 | 594 | 732 | 977 | 1,425 | 2,912 | 10,491 |
| Simplified Decision | 96 | 177 | 204 | 244 | 320 | 479 | 1,196 | 9,211 |
| Simplified Reasoning | 1,061 | 1,763 | 2,010 | 2,344 | 3,010 | 4,641 | 8,753 | 19,968 |
| **Validation Dataset** | | | | | | | | |
| Input Text | 17,927 | 61,311 | 78,422 | 109,823 | 177,678 | 282,347 | 1,279,851 | 2,700,242 |
| Target Text | 2,091 | 3,318 | 3,878 | 4,530 | 5,893 | 9,467 | 19,056 | 27,366 |
| Case Text | 5,888 | 9,604 | 11,043 | 12,814 | 16,328 | 25,305 | 68,429 | 413,279 |
| Simplified Facts | 976 | 1,535 | 1,679 | 1,967 | 2,337 | 2,839 | 5,020 | 11,689 |
| Simplified Legal Issue | 64 | 118 | 144 | 187 | 234 | 313 | 782 | 8,678 |
| Petitioner Arguments | 358 | 656 | 792 | 953 | 1,269 | 1,909 | 4,881 | 8,433 |
| Respondent Arguments | 260 | 534 | 634 | 822 | 1,032 | 1,628 | 2,696 | 9,799 |
| Simplified Decision | 100 | 183 | 227 | 279 | 363 | 547 | 828 | 2,877 |
| Simplified Reasoning | 1,032 | 1,774 | 2,043 | 2,345 | 3,027 | 5,010 | 12,303 | 17,994 |
| **Test Dataset** | | | | | | | | |
| Input Text | 18,992 | 73,468 | 95,149 | 125,702 | 186,462 | 328,597 | 865,328 | 2,133,870 |
| Target Text | 2,202 | 3,438 | 3,903 | 4,565 | 5,764 | 8,118 | 13,846 | 29,737 |
| Case Text | 5,982 | 9,933 | 11,247 | 13,010 | 15,893 | 21,284 | 46,887 | 153,313 |
| Simplified Facts | 996 | 1,531 | 1,754 | 2,024 | 2,386 | 3,222 | 8,549 | 11,837 |
| Simplified Legal Issue | 60 | 113 | 139 | 181 | 230 | 309 | 557 | 9,422 |
| Petitioner Arguments | 362 | 667 | 783 | 959 | 1,148 | 1,615 | 3,497 | 9,024 |
| Respondent Arguments | 264 | 504 | 613 | 781 | 1,035 | 1,500 | 3,234 | 10,122 |
| Simplified Decision | 95 | 173 | 202 | 240 | 298 | 443 | 824 | 2,035 |
| Simplified Reasoning | 1,158 | 1,904 | 2,135 | 2,515 | 3,182 | 4,608 | 8,514 | 19,802 |

Table 4: Token distribution statistics across dataset fields. Values represent percentile statistics for token counts in the training, validation, and test splits.

| Collection | #Chunks | #Collections | Metadata & Filterable Fields |
|---|---|---|---|
| `Central_Acts` | 36,991 | 1 | **Metadata:** id, text, case_name, section_no, section_title<br>**Filters:** section_no, title, doc_id |
| `HC_Judgments` | 1,069,017 | 1 | **Metadata:** id, text, court, judge, CNR, date, disposal, state, case_no<br>**Filters:** court, case_no, state, doc_id |
| `SC_Judgments` | 81,331 | 1 | **Metadata:** id, text, case_name, diary_no, type, petitioner, respondent, bench, citations<br>**Filters:** case_no, date, bench, case_name |
| `State_Acts` | 150,748 | 1 | **Metadata:** id, text, state_name, statute, section_no, section_title<br>**Filters:** state, statute, section_no |

Table 5: Collections stored in the Endee vector database used for legal retrieval. Each collection contains chunked legal documents with associated metadata fields and filterable attributes enabling structured retrieval.

| Collection | #Chunks | #Collections | BM25 Fields & Metadata |
|---|---|---|---|
| `centralacts` | 36,921 | 1 | **BM25 Fields:** section_text, section_title, statute<br>**Metadata:** id, section_no |
| `scjudgments` | 52,796 | 1 | **BM25 Fields:** content<br>**Metadata:** case_name, diary_no, type, case_no, petitioner, respondent, bench, date, citations |
| `hcjudgments` | 811,492 | 1 | **BM25 Fields:** text, case_name<br>**Metadata:** id, judgment_date, citations, bench |
| `sa_{state}` | 50,156 | 33 | **BM25 Fields:** section_text, section_title<br>**Metadata:** id, state, statute, section_no, doc_id |

Table 6: Collections stored in the Vespa vector database. Each collection defines searchable BM25 fields along with structured metadata attributes used for filtering and retrieval.

| Collection | #Chunks | #Collections | Metadata Fields |
|---|---|---|---|
| `Central_Acts` | 37,202 | 1 | id, embedding, text, case_name, section_no, section_title |
| `SC_Judgments` | 81,625 | 1 | id, embedding, text, case_name, diary_no, type, petitioner, respondent, bench, citations |
| `HC_{Court}` | 1,577,838 | 25 | id, embedding, case_no, pdf_path, state, disposal, decision_date, text, title, judge, CNR |
| `State_Acts_{Name}` | 146,804 | 33 | id, embedding, doc_id, state_name, statute_name, section_no, section_title, text |

Table 7: Collections stored in the Milvus vector database. Each collection stores embedded legal text chunks along with structured metadata attributes used for filtering and retrieval.

| Feature | Endee | Milvus | VespaDB |
|---|---|---|---|
| **Retrieval Strategy** | | | |
| Retrieval Philosophy | Single-stage Dense Retrieval | Three-stage Hybrid Semantic $\rightarrow$ Lexical $\rightarrow$ Fusion | Parallel Hybrid (Semantic + Lexical) $\rightarrow$ Fusion |
| Fusion Method | – | RRF ($k = 60$) | RRF ($k = 60$) |
| BM25 Retrieval | Not used | Applied on top-500 vector results | Native parallel full-corpus BM25 |
| **Vector Configuration** | | | |
| Embedding Model | `snowflake-arctic-embed-m-v2.0` (768-d) | | |
| Distance Metric | Cosine | L2 | L2 |
| Index Type | HNSW (M=16, ef_con=128) | IVF_FLAT (nlist=2048) or FLAT | HNSW (M=16, ef_ins=200) |
| Precision | FP16 | FP32 | FP32 (Paged attributes) |
| Chunk Size | 4096 | 4096 | 8192 |

Table 8: Comparison of the three retrieval architectures used in NYAYAMIND. The systems differ in retrieval strategy, hybrid fusion mechanisms, and vector indexing configurations.

| **Input Text** |
| --- |
| **System Message** =“SYSTEM:<br>You are a smart and intelligent legal assistant for the Indian legal domain. Based on the user’s instructions, you will have to perform or assist in some tasks related to the Indian legal system. Since these tasks have some legal application, only provide responses you are extremely certain about, and avoid being ambiguous or uncertain. Ensure that your outputs adhere to the user’s instructions or requirements.”<br>**User Content** = “USER:<br>You are a legal expert tasked with making a judgment about whether an appeal should be accepted or rejected based on the provided case proceeding, cited statutes and cited cases. Your task is to evaluate whether the appeal should be accepted (1) or rejected (0) based on the input.<br>### Now, evaluate the following case:<br>Case Proceedings: **{simplified_facts}**<br>Relevant Statutes: **{sections_texts}**<br>Cited Cases Reference: **{cited_case_judgments}**<br>Provide your judgment by strictly following this format:<br>##PREDICTION: [Insert your prediction here]<br>##EXPLANATION: [Insert your reasoning here that led you to your prediction.]<br>Strictly do not include anything outside this format. Strictly follow the provided format. Do not generate placeholders. Just provide the final judgment and explanation.” |
| **Target Text** |
| “ASSISTANT:<br>### Response:<br><think><br>* Legal Issue Analysis:** **{simplified_issue}**<br>**Arguments of Petitioner:** **{simplified_arguments_of_petitioner}**<br>**Arguments of Respondent:** **{simplified_arguments_of_respondent}**<br>**Deliberation:** Weighing the arguments against the relevant statutes and cited cases to form a decision.<br></think><br>##PREDICTION: **{simplified_decision}**<br>##EXPLANATION: **{simplified_reasoning}** ” |

Table 9: Prompt for DeepSeek-R1-Distill-Qwen-14B model.

| **Input Text** |
| --- |
| **System Message** =“<\|system\|> You are a smart and intelligent legal assistant for the Indian legal domain. Based on the user’s instructions, you will have to perform or assist in some tasks related to the Indian legal system. Since these tasks have some legal application, only provide responses you are extremely certain about, and avoid being ambiguous or uncertain. Ensure that your outputs adhere to the user’s instructions or requirements.<br><\|end\|>”<br>**User Content** = “<\|user\|> You are a legal expert tasked with making a judgment about whether an appeal should be accepted or rejected based on the provided case proceeding, cited statutes and cited cases. Your task is to evaluate whether the appeal should be accepted (1) or rejected (0) based on the input.<br>### Now, evaluate the following case:<br>Case Proceedings: **{simplified_facts}**<br>Relevant Statutes:<br>**{sections_texts}**<br>Cited Cases Reference:<br>**{cited_case_judgments}**<br>Provide your judgment by strictly following this format:<br>##PREDICTION: [Insert your prediction here]<br>##EXPLANATION: [Insert your reasoning here that led you to your prediction.]<br>Strictly do not include anything outside this format. Strictly follow the provided format. Do not generate placeholders. Just provide the final judgment and explanation.<\|end\|>”<br>**Assistant** = “<\|assistant\|>” |
| **Target Text** |
| “<think><br>*Legal Issue Analysis:**<br>**{simplified_issue}**<br>*Arguments of Petitioner:**<br>**{simplified_arguments_of_petitioner}**<br>*Arguments of Respondent:**<br>**{simplified_arguments_of_respondent}**<br>*Deliberation:**<br>Weighing the arguments against the relevant statutes and cited cases to form a decision.<br></think><br>##PREDICTION: **{simplified_decision}**<br>##EXPLANATION: **{simplified_reasoning}**” |

Table 10: Prompt for Phi-4-mini-reasoning model.

| **Input Text** |
| --- |
| **System Message** =“<\|im_start\|>system<\|im_sep\|><br>You are a smart and intelligent legal assistant for the Indian legal domain. Based on the user’s instructions, you will have to perform or assist in some tasks related to the Indian legal system. Since these tasks have some legal application, only provide responses you are extremely certain about, and avoid being ambiguous or uncertain. Ensure that your outputs adhere to the user’s instructions or requirements.<br><\|im_end\|>”<br>**User Content** = “<\|im_start\|>user<\|im_sep\|><br>You are a legal expert tasked with making a judgment about whether an appeal should be accepted or rejected based on the provided case proceeding, cited statutes and cited cases. Your task is to evaluate whether the appeal should be accepted (1) or rejected (0) based on the input.<br>### Now, evaluate the following case:<br>Case Proceedings: **{simplified_facts}**<br>Relevant Statutes:<br>**{sections_texts}**<br>Cited Cases Reference:<br>**{cited_case_judgments}**<br>Provide your judgment by strictly following this format:<br>##PREDICTION: [Insert your prediction here]<br>##EXPLANATION: [Insert your reasoning here that led you to your prediction.]<br>Strictly do not include anything outside this format. Strictly follow the provided format. Do not generate placeholders. Just provide the final judgment and explanation.<\|im_end\|>”<br>**Assistant** = “<\|im_start\|>assistant<\|im_sep\|>” |
| **Target Text** |
| “<think><br>*Legal Issue Analysis:**<br>**{simplified_issue}**<br>*Arguments of Petitioner:**<br>**{simplified_arguments_of_petitioner}**<br>*Arguments of Respondent:**<br>**{simplified_arguments_of_respondent}**<br>*Deliberation:**<br>Weighing the arguments against the relevant statutes and cited cases to form a decision.<br></think><br>##PREDICTION: **{simplified_decision}**<br>##EXPLANATION: **{simplified_reasoning}**” |

Table 11: Prompt for Phi-4-reasoning model.

| **Input Text** |
| --- |
| **System Message** =“<\|im_start\|><\|system\|><br>You are a smart and intelligent legal assistant for the Indian legal domain. Based on the user’s instructions, you will have to perform or assist in some tasks related to the Indian legal system. Since these tasks have some legal application, only provide responses you are extremely certain about, and avoid being ambiguous or uncertain. Ensure that your outputs adhere to the user’s instructions or requirements.<br><\|im_end\|>”<br>**User Content** = “<\|im_start\|><\|user\|> You are a legal expert tasked with making a judgment about whether an appeal should be accepted or rejected based on the provided case proceeding, cited statutes and cited cases. Your task is to evaluate whether the appeal should be accepted (1) or rejected (0) based on the input.<br>### Now, evaluate the following case:<br>Case Proceedings: **{simplified_facts}**<br>Relevant Statutes:<br>**{sections_texts}**<br>Cited Cases Reference:<br>**{cited_case_judgments}**<br>Provide your judgment by strictly following this format:<br>##PREDICTION: [Insert your prediction here]<br>##EXPLANATION: [Insert your reasoning here that led you to your prediction.]<br>Strictly do not include anything outside this format. Strictly follow the provided format. Do not generate placeholders. Just provide the final judgment and explanation.<\|im_end\|> ”<br>**Assistant** = “<\|im_start\|><\|assistant\|>” |
| **Target Text** |
| “<think><br>*Legal Issue Analysis:**<br>**{simplified_issue}**<br>*Arguments of Petitioner:**<br>**{simplified_arguments_of_petitioner}**<br>*Arguments of Respondent:**<br>**{simplified_arguments_of_respondent}**<br>*Deliberation:**<br>Weighing the arguments against the relevant statutes and cited cases to form a decision.<br></think><br>##PREDICTION: **{simplified_decision}**<br>##EXPLANATION: **{simplified_reasoning}**” |

Table 12: Prompt for Qwen3.5-27B model.

| Model | Reasoning, Decision and Explanation |
|---|---|
| Ground Truth | **Legal Issue**<br>Is the appellant considered to be in "possession" of the lands as khudkasht. . .<br>**Arguments of Petitioner**<br>It can be contended by Mr. J. P. Goyal, learned counsel for the appellant. . .<br>**Arguments of Respondent**<br>The respondentś arguments could be that the appellant may not be able to challenge the final. . . **Decision**<br>We may hold that the judgment. . . **appeal can be set aside**.<br>**Explanation** It can be inferred that the normal principle of possession by one co-sharer. . . |
| DeepSeek-R1-Distill-Qwen-14B (4-bit) | ****Legal Issue Analysis:****<br>Whether the plaintiff was in cultivatory possession of the plots in dispute on the date of vesting?<br>****Arguments of Petitioner:****<br>It can be inferred that the petitioner may argue that the plaintiff's counsel. . .<br>****Arguments of Respondent:****<br>The respondent may contend that the plaintiff's possession of the plots in dispute may have been limited to a mere physical presence . . .<br>The respondent may argue that the plaintiff's possession of the plots in dispute may have been subject to the rights of the defendants. . . **(Sementic_Redundancy)**<br>**##PREDICTION:** 16. The appeal may be dismissed.<br>**## EXPLANATION:** It can be inferred that the plaintiff's possession of the plots in dispute was not cultivatory possession, as per the findings of the High Court.<br>It can be inferred that the plaintiff's possession of the plots . . . **(Repetitive Text)** |
| DeepSeek-R1-Distill-Qwen-14B (8-bit) | Alright, Iḿ trying to figure out whether the appeal in this case should be accepted or rejected.<br>First, looking at the case proceedings, the plaintiff, Kailashrai. . . **(Wrong output format)**<br>**##PREDICTION:** 0<br>**##EXPLANATION:**<br>The High Court correctly determined that the plaintiff was not in cultivatory. . . |
| DeepSeek-R1-Distill-Qwen-14B (16-bit) | ****Legal Issue Analysis:****<br>Is the plaintiff entitled to a declaration that he is entitled to one-fourth share in the properties? . . .<br>****Arguments of Petitioner:****<br>It is contended that the plaintiff can establish his right to cultivate the plots in question . . .<br>****Arguments of Respondent:****<br>The respondent's arguments could be that the plaintiff may not be able to establish his right . . .<br>**##PREDICTION:** The appeal can be dismissed with costs. . . **(Correct Prediction)**<br>**##EXPLANATION:** It can be inferred that the plaintiff's claim is based on the . . . |
| DeepSeek-R1-Distill-Qwen-14B (32-bit) | ****Legal Issue Analysis:****<br>Is the plaintiff entitled to a declaration that he is entitled to one-fourth share in the properties? . . .<br>****Arguments of Petitioner:****<br>It is contended that the plaintiff can establish his right to cultivate the plots in question, . . .<br>****Arguments of Respondent:****<br>The respondent's arguments could be that the plaintiff may not be able to establish his right . . .<br>**##PREDICTION:** The appeal may be dismissed. . . **(Correct Prediction)**<br>**##EXPLANATION:** It can be inferred that the plaintiff's claim to the properties . . . |
| Phi4-mini-reasoning (16-bit) | ****Legal Issue Analysis:****<br>Is the defendant liable? . . .<br>****Legal Process Analysis:**<br>Respondent Arguments Against Witness Statements . . . art therapy play therapy bibliotherapy music therapy . . . DeFi stablecoins tokenization NFTs metaverse Web3 . . . **(Random text)** |
| Phi4-reasoning (16-bit) | User instructs: "Evaluate whether the appeal should be ACCEPTED (1) or REJECTED (0)"<br>Now, analyze the case. Let me re-read the case summary carefully.. . . **(Wrong output format)**<br>Letś check relevant statutes: . . . Also cited precedents: . . .<br>Explanation: Because High Courtś decision . . . But note: The Abolition Act . . . Double-check: Precedent 2 . . . Thus, answer: Prediction: 0 (rejected) . . . **(Multi-step legal reasoning**<br>Explanation: Because High Court correctly . . . IÍl produce answer accordingly.<br>**##PREDICTION:** 0<br>**##EXPLANATION:** The High Court's . . . |
| Qwen3.5-27B (16-bit) | ****Legal Issue Analysis:****<br>What are the grounds for challenging the High Court's judgment . . .<br>****Arguments of Petitioner:**** The petitioner may contend that the High Court . . .<br>****Arguments of Respondent:**** The respondent's arguments could be that the High Court . . .<br>**##PREDICTIONS:** 12. The appeal may be dismissed. . . . **(Correct Prediction)**<br>**##EXPLANATION:** It can be inferred that a person . . . |

Table 13: Comparative analysis of output generated by different reasoning-oriented models, showing signs of hallucination, repetitive text, sementically redundant text, correct prediction and multi step legal reasoning.