IBPS: Indian Bail Prediction System

Uddeshya Raj[1] **Puspesh Kumar Srivastava**[1] **Praveen Patel**[1]
Shubham Kumar Nigam[1,3†] **Noel Shallum**[2] **Arnab Bhattacharya**[1]
[1]Indian Institute of Technology Kanpur, India [2]Symbiosis Law School Pune, India
[3]University of Birmingham, Dubai, United Arab Emirates
raj.uddeshya157@gmail.com puspeshsrivastava@gmail.com prav906715@gmail.com
shubhamkumarnigam@gmail.com noelshallum@gmail.com arnabb@iitk.ac.in

Abstract

Bail decisions constitute a large share of criminal proceedings in Indian courts, yet they often suffer from delays, inconsistency, and limited transparency. More than 75% of India's prison population consists of undertrial prisoners, highlighting the urgency of improving the efficiency of bail adjudication. In this work, we present the Indian Bail Prediction System (IBPS), an AI-assisted framework that predicts bail outcomes and generates legally grounded rationales from factual case attributes and statutory context. We construct a large-scale dataset of 89,969 bail judgments from 5 High Courts of India, enriched with structured annotations such as statutes, custody duration, criminal history, health conditions, and judicial reasoning. Using this dataset, we design a two-stage architecture that combines a transformer-based classifier for outcome prediction with a fine-tuned large language model for rationale generation. Our results demonstrate the feasibility of combining structured legal data and large language models to develop interpretable AI systems that support data-driven analysis of bail decisions.

1 Introduction

Bail adjudication lies at the core of the criminal justice system, directly impacting individual liberty, judicial efficiency, and societal fairness. In India, bail proceedings constitute a substantial portion of criminal litigation, yet they remain affected by delays, inconsistencies, and subjectivity. As per Prison Statistics Report India, 2021[1], more than 75% of the prison population comprises undertrial prisoners—individuals who have not been convicted but remain incarcerated while awaiting bail decisions. This not only contributes to severe prison overcrowding but also raises critical concerns about procedural justice, particularly for economically and socially disadvantaged groups.

The scale of the problem is further exacerbated by the growing backlog of criminal cases in Indian courts, where bail applications form a significant fraction of daily judicial workload, especially in lower courts. Despite their frequency, bail decisions require careful consideration of multiple factors, including the nature of the offence, prior criminal history, statutory provisions, health conditions, and risk assessments such as flight risk or evidence tampering. The inherently discretionary nature of these decisions, combined with increasing case volumes, creates an urgent need for systems that can assist legal practitioners in making consistent, transparent, and timely decisions.

Despite this pressing need, computational research in this space remains limited. Most prior work in Legal NLP has focused on general court judgment prediction and rationale generation tasks, often in non-Indian jurisdictions or under civil law settings. In the Indian context, datasets such as HLDC (Kapoor et al., 2022) and IL-TUR (Joshi et al., 2024) include only limited subsets of bail-related cases, sourced from district courts and are in Hindi, restricting their applicability for large-scale modeling and downstream NLP tasks. Furthermore, existing approaches for judgment prediction are not directly transferable to bail cases due to the fact-intensive, context-dependent, and highly individualized nature of bail decisions.

In this work, we address this gap by introducing the **Indian Bail Prediction System (IBPS)**, a comprehensive AI-assisted framework designed to predict bail outcomes and generate legally grounded rationales based on structured case attributes and statutory context. Unlike prior approaches, IBPS focuses specifically on bail adjudication, modeling it as a combined task of outcome prediction and rationale generation. To support this, we curate a large-scale dataset of **89,969** High Court bail judgments, enriched with structured features such as statutes, custody duration, criminal history, health

†Corresponding author
[1]Prison Statistics India (PSI) - 2021

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 1236–1260
November 6-10, 2026 ©2026 Association for Computational Linguistics

conditions, and judicial reasoning.

Our approach leverages advances in large language models to learn fact-based legal reasoning while maintaining interpretability through structured inputs and rationale generation. Importantly, IBPS is designed as a decision-support system, not a replacement for judicial authority, enabling human-in-the-loop usage for enhanced transparency and efficiency.

Our key contributions in this work are as follows:

- We curate a large-scale English dataset of **89,969** Indian High Court bail judgments with structured legal attributes, including statutes, criminal history, custody duration, and judicial outcomes, addressing a major gap in Indian Legal NLP resources.
- We design **Indian Bail Prediction System `IBPS`**, a unified framework that performs both bail outcome prediction and rationale generation using structured case features and large language models.
- We systematically evaluate existing NLP techniques for statute extraction and rhetorical role identification in bail judgments, demonstrating their limitations and motivating the use of LLM-based approaches.
- We construct a gold-standard test set of **100** cases annotated by legal experts, enabling reliable evaluation of both predictive performance and the quality of the rationale.

For reproducibility, we release the code and IBPS implementation through our GitHub repository[2] and dataset on the Huggingface repository[3].

2 Related Work

Research at the intersection of Artificial Intelligence and law (AI4Law) has grown rapidly with the digitisation of legal records and advances in natural language processing (NLP). Legal AI systems have been applied to tasks such as court judgment prediction (CJP) and explanation (CJPE), statute retrieval, question answering, contract analysis, and legal summarisation. Among these, CJPE has received particular attention due to its potential to support judicial decision-making.

Early work on legal judgment prediction relied on rule-based, statistical or classical ML-based methods. For example, Aletras et al. (2016) predicted decisions of the European Court of Human Rights using textual features from case documents. Other works like EURLEX57K (Chalkidis et al., 2019), CAIL2018 (Xiao et al., 2018), SwissJudgment (Niklaus et al., 2021), JTD (Yamada et al., 2024), and LegalBench (Guha et al., 2023) furthered the research in this domain, but their datasets focus on civil-law jurisdictions and do not reflect the characteristics of common-law systems such as India. ILDC (Malik et al., 2021) is among the earliest attempts at the Indian CJPE task, providing a dataset of Supreme Court judgments with outcomes and explanations. Later, the IL-TUR benchmark tasks for Indian legal text (Joshi et al., 2024) and HLDC (Kapoor et al., 2022) were released for training CJP systems. The PredEX and NyayaAnumana datasets (Nigam et al., 2024b,a), which came later, facilitated leveraging LLMs to provide explanations alongside outcome predictions. InLegalLlama was one such attempt. A different kind of dataset is IndicLegalQA (K and Mishra, 2025), which is structured as a case comprehension task that provides case judgment documents and a set of related questions and answers for training a legal question-answering system.

In contrast to prior work, our study focuses on predicting bail outcomes in the Indian legal system. We introduce a large-scale dataset of Indian High Court bail judgments and develop models that jointly predict outcomes and generate rationales grounded in factual case attributes and statutory context.

3 Task Description

We study the task of *Bail Prediction*, which consists of two sequential components: (i) predicting the outcome of a bail application and (ii) generating a rationale that justifies the predicted outcome. Our study focuses on two types of bail applications in the Indian legal system: *Regular Bail* and *Anticipatory Bail*.

Outcome Prediction The first task is to predict whether a bail application is granted or denied, given the details of a case. Each case consists of structured and textual information that describes the circumstances of the bail application. Although a case may involve multiple accused individuals, prediction is performed assuming that case details are common for all accused, thereby generating a common outcome for all accused and making it a binary classification task.

[2] https://github.com/Uddeshya-Raj/IBPS-v0.2

[3] https://huggingface.co/datasets/Killer-C0wboy/IBPS-v0.2-dataset

Formal Definition Let $\mathcal{D}_R$ and $\mathcal{D}_A$ denote the sets of documents corresponding to *Regular Bail* and *Anticipatory Bail* applications, respectively. Given a case document $D \in \mathcal{D}_R \cup \mathcal{D}_A$, the task is to predict a binary outcome $y \in \{0, 1\}$, where

- $y = 1$ indicates that bail is granted, and
- $y = 0$ indicates that bail is not granted.

Rationale Generation The second task is to generate a natural-language rationale of the predicted outcome. The rationale generation uses factual and legal factors present in the case and reflects the type of reasoning typically provided in judicial orders. This step improves the interpretability of the system and allows legal practitioners to understand the basis of the model's prediction.

4 Data Preparation

4.1 Raw Data Collection

We began with the Daksh database[4], which provides metadata for bail cases across Indian High Courts, including case identifiers (CNR numbers), statutes applied, filing and judgment dates, and bail outcomes. The metadata covers 927,897 cases. Using this metadata, we scraped the corresponding full-text judgments (sample present in Figure 14) from the eCourts High Court portal[5] through custom Python scripts.

To support statutory legal context, we also collected texts from the India Code portal[6], including provisions from the Indian Penal Code (IPC), the Code of Criminal Procedure (CrPC), and relevant Central and State Acts.

4.2 Structured Feature Extraction

Court judgments are highly unstructured and vary substantially across jurisdictions, time periods, and individual judges. To construct a structured dataset suitable for downstream modelling, we extract attributes describing case facts, statutes applied, criminal history, the accused's personal conditions, custody duration, bail outcomes, and judicial reasoning.

Model Selection We evaluated several approaches for extracting structured information from judgments, including classical NLP models and open-source large language models (LLMs). Preliminary experiments indicated that existing rhetorical-role extraction models did not generalise effectively to our dataset. Consequently, we adopted an LLM-based extraction approach.

We conducted a pilot comparison of several candidate LLMs on a small set of 100 manually verified *Gold standard* cases and selected `microsoft/phi-4` based on extraction quality, output consistency, and inference efficiency. Detailed evaluation procedures and additional experimental results are provided in the Appendix C.

Extraction Strategy Structured information was extracted using a one-shot prompting strategy, in which the model generates outputs conforming to a predefined JSON schema. We evaluated zero-shot and one-shot prompting approaches and found that one-shot prompting provided the most reliable balance between extraction accuracy and computational efficiency.The comparison and example output of zero shot and one shot extraction (using the sample judgement mentioned in Appendix I) is present in Appendix K.The full prompt template used for this extraction is provided in the Appendix H.

4.3 Data Cleaning

Using the Daksh metadata, we processed over 350,000 bail cases, of which approximately 250,000 had corresponding judgment texts available through the eCourts portal. After removing documents lacking substantive factual details (e.g., orders containing only a brief grant/reject decision), we extracted structured information from 208,293 cases. Cases with multiple accused, where bail decisions differed across accused, were also removed to align the dataset for the binary classification task.

We observed that many judgments contain generic statements that provide little explanatory information (e.g., decisions issued without explicit reasoning). To identify such cases, we embedded the reasoning sections using *InLegalBERT*(Paul et al., 2023) and clustered them using *HDBSCAN* (Campello et al., 2015). Clusters corresponding to non-informative justifications were manually identified and removed. This filtering step, resulted in a final dataset of 89,969 bail judgments for downstream prediction and rationale generation task.

5 Dataset Analysis

Our dataset consists of bail judgment documents collected from High Courts in India namely, Allahabad, Bombay, Chhattisgarh, Jharkhand, and Ker-

[4] https://database.dakshindia.org
[5] hcservices.ecourts.gov.in/hcservices/main
[6] www.indiacode.nic.in

Bail Type	Total	Accepted	Rejected
Anticipatory Bail	48,204	33,126	15,078
Regular Bail	41,765	29,789	11,976
Overall	**89,969**	**62,915**	**27,054**

Table 1: Bail judgement type and outcome breakdown.

ala. The final dataset contains 89,969 bail cases, including 48,204 anticipatory bail and 41,765 regular bail judgements. The dataset is moderately imbalanced towards bail-granted decisions, with approximately 2.3 bail-granted cases for every bail-denied case. Table 1 shows the detailed breakdown of the classes in the dataset.

5.1 Dataset Attributes

Each record in the dataset contains thirteen attributes describing the case facts, legal context, and the final bail decision.

- `cnr`: Unique case identifier corresponding to the original judgment on the eCourts portal.
- `bail_type`: Type of bail application (*regular* or *anticipatory*).
- `age_available`: Boolean indicator of whether the accused's age is mentioned in the judgment.
- `ages`: List of ages of the accused individuals (when available).
- `health_condition`: Health-related information about the accused.
- `past_criminal_record_exists`: Indicator of whether the accused has prior criminal records.
- `past_criminal_record_charges`: Statutes associated with past criminal cases of the accused.
- `statutes`: Criminal statutes applied in the current case.
- `case_details`: Textual description of the incident and arguments presented by the prosecution and defence.
- `days_in_custody`: Number of days the accused spent in custody before the bail decision.
- `outcome`: Final decision of the court (*bail granted* or *bail not granted*).
- `reasoning`: Judicial reasoning explaining the bail decision.
- `statute_details`: Descriptions of relevant statutes obtained from the India Code portal.

Detailed definitions of each attribute, along with extraction rules and examples, are provided in the Appendix A.

High Court	Granted	Not Granted	Total Cases
Allahabad	3,916	2,529	6,445
Bombay	13,621	4,667	18,288
Chhattisgarh	3,864	364	4,228
Jharkhand	17,620	7,067	24,687
Kerala	23,894	12,427	36,321

Table 2: Case Distribution Across High Courts

Statistic	Value
Total # cases	89,969
Total # sentences	1,290,546
Avg. sentences per sample	14.34
Avg. tokens per sentence	69.07
Sentence Count per Label	
`case_details`	675,171
`reasoning`	198,640
`statute_details`	416,735
Avg. Tokens per Label	
`case_details`	213.24
`reasoning`	61.03
`statute_details`	760.12

Table 3: Statistics of the final dataset. Microsoft/Phi-4's tokenizer was used for counting tokens

5.2 Dataset Metrics

Table 3 summarizes additional statistics of the final dataset. To support downstream modelling and reasoning generation tasks, we also analyse the distribution of sentences and tokens across different segments of the judgments. The distribution histogram of cases across the five High Courts is shown in Table 2. The detailed analysis of the dataset is presented in Appendix B.

6 Methodology

The IBPS is implemented as a two-stage pipeline for bail outcome prediction and rationale generation (Figure 1). Since the dataset is severely imbalanced between the *bail granted* and *bail not granted* classes, we undersample from the *bail granted* class. The dataset comprises 46,143 samples categorized into four classes: regular bail granted (11,513), regular bail denied (11,275), anticipatory bail granted (11,745), and anticipatory bail denied (11,610). The train-val-test split of

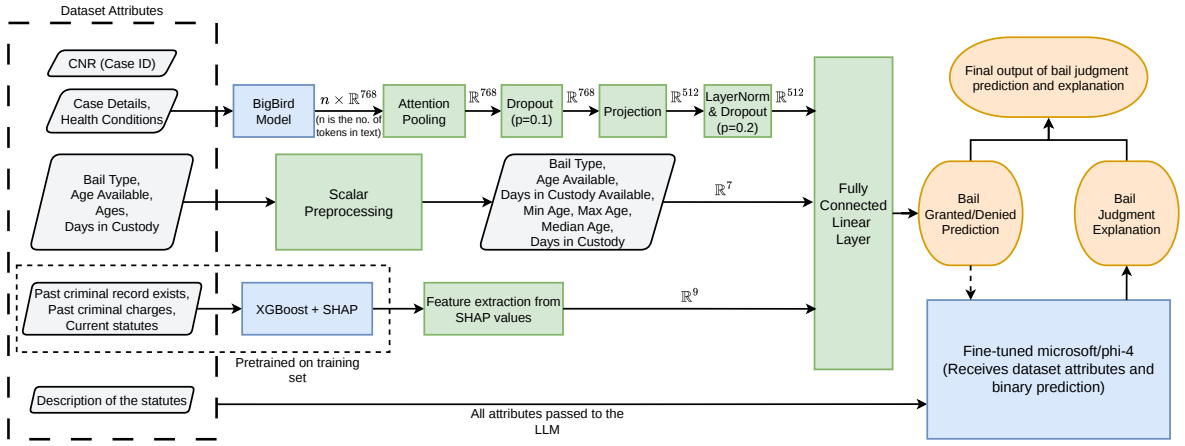


Figure 1: The classification module (left) fuses text encodings from BigBird with standardised numeric attributes, and XGBoost+SHAP extracted features to predict bail outcomes. The generative module (right) receives this prediction alongside raw case attributes, and a fine-tuned microsoft/phi-4 LLM synthesizes the final judgment rationale.

data is $8:1:1$. The near-uniform distribution across classes helps mitigate class imbalance issues during model training. Given a case document D and its associated attributes X, the classifier at the first stage predicts outcome $y = f(D, X)$. The predicted outcome y is then provided to a generative module, where a fine-tuned large language model produces a judicial-style rationale $e = g(D, X, y)$, consistent with the prediction. This design separates prediction and reasoning tasks, allowing encoder-based models to handle classification while leveraging LLMs' generative capabilities for rationale synthesis.

6.1 Bail Outcome Prediction

The objective of the classification module is to predict whether a bail application is granted or rejected based on information in the case record. Bail judgments combine multiple forms of information, including long textual descriptions of the case, structured numerical attributes describing the accused and the proceedings, and legal context in the form of statutes applied to the case. To model these heterogeneous signals, we encode each information type separately and fuse the resulting representations into a unified feature vector for predicting the bail outcome.

6.1.1 Baseline Setup

As a baseline setup for prediction module we used a Logistic Regression model and a 2-layer MLP with 16 nodes in input layer, 64 in middle and 1 node in output layer, trained on structured numeric attributes like, statute numbers, age, days in custody and bail type(1-regular bail and 2-anticipatory bail) from our dataset.

6.1.2 Textual Feature Encoding

The textual attributes `case_details` and `health_condition` contain long narrative descriptions of the case and therefore require contextual semantic representations. Since standard BERT models are limited to 512 tokens, we employ *BigBird-RoBERTa-Large*[7], which supports long-context encoding with sequences up to 4096 tokens.

Given an input sequence of n tokens, the encoder produces hidden states

$$H = [h_1, h_2, \ldots, h_n], \quad h_i \in \mathbb{R}^{768}$$

To obtain a fixed-length representation, we apply attention pooling over the token embeddings. A learnable attention vector $v \in \mathbb{R}^{768}$ computes relevance scores

$$e_i = h_i^T v,$$

which are normalised using a softmax function

$$\alpha_i = \frac{\exp(e_i)}{\sum_{j=1}^{n} \exp(e_j)}$$

The final text representation is the attention-weighted sum of token embeddings

$$r_{\text{text}} = \sum_{i=1}^{n} \alpha_i h_i$$

The resulting vector is projected to $\mathbb{R}^{512}$ using a fully connected layer to produce the textual feature embedding used by the classification module.

[7] huggingface.co/google/bigbird-roberta-large

6.1.3 Numerical Feature Representation

Several structured attributes describing the accused and the proceedings are represented as numerical features. These include `bail_type`, `age_available`, `ages`, and `days_in_custody`. All continuous variables are standardised before being passed to the classification layer. Since the `ages` field may contain multiple values when a case involves multiple accused individuals, we derive three scalar features: `min_age`, `max_age`, and `median_age`. This ensures a consistent representation across cases with varying numbers of accused. In addition, a binary indicator `days_in_custody_available` is introduced to distinguish between custody durations explicitly provided in the judgment and values imputed with median during preprocessing.

6.1.4 Statutory Feature Representation

The attributes `statutes`, `past_criminal_record_charges`, and `past_criminal_record_exists` represent categorical legal information associated with each case. These attributes correspond to a large multi-label space containing approximately 2,500 unique statutes. Representing them using multi-hot encoding would produce a highly sparse feature space that could dominate the textual representations.

To address this, we train an auxiliary XGBoost model using only these statute- related attributes and derive compact feature summaries using SHAP (*SHapley Additive exPlanations*[8]) values. The SHAP outputs are aggregated into nine scalar features capturing the magnitude and direction of statutory influence on bail outcomes:

- `shap_sum_pos`
- `shap_sum_neg`
- `shap_max_pos`
- `shap_min_neg`
- `shap_pos_count`
- `shap_neg_count`
- `shap_l1_total`
- `shap_top3_abs_sum`
- `xgb_pred`

These features are used as a compact representation of the statutory context and are concatenated with the textual and numerical representations for outcome prediction. Detailed definitions of each SHAP-derived feature are provided in the Appendix D.

6.1.5 Outcome Prediction Layer

The textual embedding (512 dimensions) and the scalar attribute vector (16 dimensions) are concatenated to form a unified 528-dimensional feature representation. This vector is passed through a fully connected layer to produce a single logit representing the likelihood of bail being granted. The model is trained using binary cross-entropy with logits (BCEWithLogitsLoss[9]), which operates directly on the predicted logits while maintaining numerical stability.

6.1.6 Training Procedure

The classification model is trained using a two-phase fine-tuning strategy to stabilise training and preserve the pretrained representations of the BigBird encoder. The model is optimised using the AdamW optimiser with gradient clipping and mixed-precision training.

In the first phase, the BigBird encoder is frozen and only the task-specific layers, including the attention pooling vector and classification head, are trained. This allows the newly initialised parameters to converge without perturbing the pretrained encoder.

In the second phase, the encoder is unfrozen, and the entire network is fine-tuned end-to-end using differential learning rates for the encoder and task-specific layers. Additional optimisation details and hyperparameters are provided in the Appendix E.

6.2 Bail Rationale Generation

The rationale generation module produces a judicial-style rationale for the predicted bail outcome. We fine-tune the *microsoft/phi-4* language model to generate reasoning conditioned on case attributes and the classification module's predicted decision.

6.2.1 Baseline Setup

For the baseline setup we are using vanilla *microsoft/phi-4* model. The LLM itself decides whether bail should be granted and provides the rationale behind its prediction.

6.2.2 LoRA Fine-tuning

For each case, the input prompt concatenates the textual case description with structured attributes, such as the accused's ages, past criminal charges,

[8] github.com/shap/shap

[9] BCEWithLogitsLoss-PyTorch documentation

and the statutes applied in the case. Descriptions of the relevant statutory provisions are also included to provide legal context. The model is trained to generate the court's reasoning text, enabling it to learn patterns in judicial justification and argumentation.

To efficiently adapt the model while preserving its pretrained knowledge, we employ Low-Rank Adaptation (LoRA) during fine-tuning. The model is trained with a long-context setting (up to 8,192 tokens) to accommodate lengthy legal documents. Additional training configuration and infrastructure details are provided in the Appendix G.

7 Evaluation Metrics

We evaluate IBPS on two tasks: bail outcome prediction and explanation generation. For the prediction task, we report standard classification metrics including Accuracy, Precision, Recall, and F1-score. For the explanation generation task, we employ the following evaluation methods:

1. **LLM-based Evaluation (LLM-as-a-Judge):** We employ an LLM-as-a-Judge framework (Zheng et al., 2023) using G-Eval (Liu et al., 2023). GPT-4o scores each generated explanation on a 1–10 scale relative to the reference explanation, evaluating factual accuracy, completeness and coverage, and clarity and coherence. Prompt templates are provided in Appendix H.
2. **Expert Evaluation:** Legal professionals evaluate a subset of generated explanations using a 1–10 Likert scale across three criteria: factual accuracy, completeness and clarity of reasoning. A score of 1 denotes poor or misleading reasoning, while 10 indicates strong legal fidelity and argumentative soundness. We involved two final-year LLB students as legal experts in the evaluation of the results.

8 Results and Analysis

The prediction module achieves 89% accuracy, and the rationale generation module showed a significant improvement in rationale quality compared to a vanilla model. The detailed breakdown of the results is given in tables 4 and 6.

8.1 Prediction Performance Breakdown

The prediction module used three types of feature encoders, each evaluated separately. To evaluate the predictive power of structured attributes with and without SHAP values, we train and test the performance of a standard logistic regression classifier trained using L2 regularisation, a single fully-connected layer mapping the input feature vector directly to a binary prediction, trained using binary cross-entropy loss, and a small feed-forward neural network consisting of two fully connected layers.

Model	Accuracy	Precision	Recall	F1
Logistic Regression (numeric features)	0.57	0.58	0.57	0.58
2-layer MLP (numeric features)	0.56	0.57	0.61	0.59
2-layer MLP (numeric+SHAP features)	0.60	0.57	0.85	0.69
IBPS (Frozen BigBird)	0.82	0.85	0.79	0.82
IBPS (Full-Training)	**0.88**	**0.88**	**0.89**	**0.89**

Table 4: Performance comparison on the bail outcome prediction task.

To test BigBird fine-tuning performance, we recorded the prediction module's performance with the BigBird encoder frozen and unfrozen.

As shown in table 4, the numeric features alone provided little signal for bail prediction. Even a 2-layer MLP performed no better than a logistic regression model. But using categorical data on statutes and past criminal records drastically boosted the model's recall, suggesting it became very good at identifying cases with an outcome labelled 1 (bail granted). Adding textual features from the BigBird encoder and attention pooling improved the model's semantic understanding, boosting overall performance. Allowing full-fine-tuning of BigBird during model training increased accuracy and F1-score to 88% and 0.89, respectively.

Fine-tuned LLM guided by these highly accurate predictions generates a rationale that is closer in quality to the rationale given by actual judges on the decision of a bail petition. Quality improves across the board on all metrics: Factual Accuracy, Completeness and Clarity, as seen in table 6.

8.2 Rationale generation performance breakdown

The rationale generated was judged by LLM (GPT-4o) and expert evaluators on three criteria: Factual accuracy, Completeness of rationale, and Clarity. We also found that our dataset contained two distinct case types based on information in the `case_details` attribute. we divide the evaluation cases into *flagged* and *unflagged* subsets. This flag was set and unset by the expert legal evaluators and helped us determine whether the performance drop

was due to poor training or incomplete data.

Subset	Metric	Baseline	Fine-tuned
Flagged Cases	Factual Accuracy	5.62 (1.32)	7.40 (1.03)
	Completeness	4.23 (1.41)	6.17 (1.18)
	Clarity	3.35 (1.73)	6.82 (1.40)
Unflagged Cases	Factual Accuracy	6.76 (2.19)	7.92 (1.40)
	Completeness	5.99 (2.41)	7.37 (1.56)
	Clarity	6.10 (3.16)	7.97 (1.36)

Table 5: Human expert evaluation averaged across the two experts for flagged and unflagged cases. Values represent mean (standard deviation).

Flagged cases A case is marked as *flagged* when the original court judgment introduces a new factual detail in the reasoning section that is not present in the case description extracted from the judgment text. In such cases, the model is forced to reason without access to the complete factual context, making the task inherently more difficult.

Unflagged cases *Unflagged* cases correspond to judgments where the reasoning relies only on facts already present in the extracted case details.

The table 5 shows the improvement in reasoning from flagged to unflagged cases, and table 6 shows overall performance improvement over a vanilla LLM. It can be seen that LLM consistently gives higher scores than the human evaluators and is more optimistic in its evaluation. This undermines the general usefulness of evaluation using GPT and other LLMs.

8.3 Inter-annotator Agreement (IAA)

Table 7 shows the average scores given by each evaluator on different criteria for baseline and fine-tuned models. Inter-annotator agreement score is calculated using Krippendorff's α, Spearman ρ, and Mean Absolute Difference (MAD). It can be seen in table 8 that there is strong agreement on the sub-par performance of the baseline model and moderate agreement on the reasons generated by the fine-tuned model, as can be seen from α and ρ scores(Appendix J). The MAD scores, however, indicate that the average difference between the scores given by the two evaluators is close to 1.

9 Conclusions and Future Scope

In this work, we introduced the IBPS, a comprehensive AI framework for predicting bail outcomes and generating fact-based legal rationales. By leveraging a newly curated dataset of 89,969 High Court bail judgments, we demonstrated the effectiveness of fine-tuning large language models with structured case features, statutory knowledge, and guidance from a model that can correctly predict bail case outcomes. Our experiments across various inference configurations show that training with legal attributes and statutes, and having the LLM know the bail case outcome, lead to more accurate and explainable decisions. The proposed approach not only achieves strong performance on legal prediction tasks but also lays the foundation for transparent and interpretable legal AI systems.

Looking ahead, IBPS can be extended in multiple directions. First, future work may incorporate post-training RL to fine-tune a chain-of-thought model and further improve the quality of rationale generation. Second, the inclusion of judgments in regional Indian languages will enhance linguistic diversity and applicability across different jurisdictions. We believe IBPS marks an important step toward data-driven legal decision support in India.

Evaluation	Metric	Baseline	Fine-tuned
Human Evaluation	Factual Accuracy	6.71 (2.20)	7.85 (1.38)
	Completeness	5.89 (2.44)	7.25 (1.60)
	Clarity	5.94 (3.28)	7.86 (1.40)
LLM Evaluation	Factual Accuracy	6.68 (1.88)	8.46 (1.34)
	Completeness	6.70 (1.47)	7.86 (1.48)
	Clarity	7.64 (1.44)	9.18 (0.77)

Table 6: Evaluation of generated reasoning. Human scores represent the average across legal experts. Values are reported as mean (standard deviation).

Evaluator	Metric	Baseline	Fine-tuned
Expert 1	Factual Accuracy	6.70 (2.23)	8.20 (1.48)
	Completeness	5.80 (2.45)	7.50 (1.77)
	Clarity	5.68 (3.23)	7.80 (1.64)
Expert 2	Factual Accuracy	6.72 (2.16)	7.50 (1.27)
	Completeness	5.98 (2.43)	7.00 (1.43)
	Clarity	6.20 (3.33)	7.92 (1.16)

Table 7: Detailed expert evaluation scores for generated reasoning. Values represent mean (standard deviation).

Acknowledgments

We would like to thank anonymous reviewers for their insightful comments. We would like to thank our legal experts, who helped us assess the results and prepare the Gold standard set. Finally, we would like to thank Research-I Foundation and BharatGen for providing the resources required for this research. This work would not have been possible without their help.

IAA Metric	Criterion	Baseline	Fine-tuned
Krippendorff's α	Factual Accuracy	0.815	0.486
	Completeness	0.921	0.593
	Clarity	0.948	0.488
Spearman ρ	Factual Accuracy	0.699	0.471
	Completeness	0.842	0.553
	Clarity	0.711	0.382
MAD	Factual Accuracy	0.90	0.96
	Completeness	0.78	1.04
	Clarity	1.04	0.90

Table 8: IAA Scores between the two legal experts.

Limitations

Although the proposed Indian Bail Prediction System (IBPS) promises good outcomes, it has a few limitations to consider when interpreting its results.

The dataset is not fully representative because it comprises 89,969 bail judgments from only five High Courts (Allahabad, Bombay, Chhattisgarh, Jharkhand, and Kerala). The practice, procedural norms, and reasoning among courts, as well as between levels of the Indian judiciary, can differ significantly.

The models are trained on historical bail determinations, which may reflect systemic biases in the criminal justice process. Factors such as socioeconomic background, policing patterns, local enforcement, and prosecutorial strategies can be indirectly related to judicial outcomes and thus implicitly learned by the model. IBPS can therefore replicate historical patterns in the data rather than discover legally normative principles. This is a weakness of predictive modeling in legal decision-making and needs to be addressed with great care when applying model outputs. The standards of law and judicial reasoning are also changing with time through statutory changes, the development of precedents, and policy alteration. Updating and retraining the databases would thus need to be done continuously to ensure their long-term applicability. Lastly, IBPS cannot be seen as a solution that will help ensure legally sound bail decisions. Its just a part of research to understand how good can NLP system get at legal understanding.

Ethical Considerations

We have adhered to all ethical principles in conducting this research by paying close attention to ethical data collection, model development, and potential societal impact. In this study, the dataset comprises bail judgments from the Indian High Court, sourced from publicly available portals on official judicial websites and legally accessible databases. These judgments are publicly available, so there was no intentional collection of personal or confidential information. However, during data processing and analysis, we took precautions to ensure that sensitive personal information included in the judgments was processed and used solely for research purposes. Although such documents might refer to participants in the legal process, the dataset is not intended to profile people but to analyse patterns of legal decisions for academic research.

In this work, legal experts were consulted to evaluate the generated explanations, assess factual accuracy, legal relevance, and the completeness of reasoning, and provide the gold standard for extraction. Their feedback was used solely for evaluation purposes and does not imply endorsement of our system for judicial decision-making. The models created in this paper are intended to serve as analytical decision-support systems for legal NLP and computational law research. They do not serve as substitutes for judges, lawyers, or other legal experts. The decision on bail is a matter of complex legal interpretation and constitutional issues, which can only be determined by a human being. Hence, human control and close interpretation of model results should be a part of any implementation of such systems.

Altogether, this study is ethical in the use of data, model construction, and assessment. We also stress responsible use, openness, and the need to maintain human control over AI systems when they are applied in sensitive areas such as the legal system.

References

Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoţiuc-Pietro, and Vasileios Lampos. 2016. Predicting judicial decisions of the european court of human rights: A natural language processing perspective. *PeerJ computer science*, 2:e93.

Ricardo J. G. B. Campello, Davoud Moulavi, Arthur Zimek, and Jörg Sander. 2015. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Trans. Knowl. Discov. Data*, 10(1).

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2019. Large-scale multi-label text classification on eu legislation. *arXiv preprint arXiv:1906.02192*.

Neel Guha, Julian Nyarko, Daniel E Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N Rockmore, et al. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *arXiv preprint arXiv:2308.11462*.

Abhinav Joshi, Shounak Paul, Akshat Sharma, Pawan Goyal, Saptarshi Ghosh, and Ashutosh Modi. 2024. Il-tur: Benchmark for indian legal text understanding and reasoning. *Preprint*, arXiv:2407.05399.

Veningston K and Apratim Mishra. 2025. Dataset for legal question answering system in the indian judiciary context. *Data in Brief*, 60:111647.

Prathamesh Kalamkar, Astha Agarwal, Aman Tiwari, Smita Gupta, Saurabh Karn, and Vivek Raghavan. 2022. Named entity recognition in Indian court judgments. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 184–193, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Arnav Kapoor, Mudit Dhawan, Anmol Goel, Arjun T H, Akshala Bhatnagar, Vibhu Agrawal, Amul Agrawal, Arnab Bhattacharya, Ponnurangam Kumaraguru, and Ashutosh Modi. 2022. HLDC: Hindi legal documents corpus. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3521–3536, Dublin, Ireland. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *Preprint*, arXiv:2303.16634.

Vijit Malik, Rishabh Sanjay, Shubham Kumar Nigam, Kripabandhu Ghosh, Shouvik Kumar Guha, Arnab Bhattacharya, and Ashutosh Modi. 2021. ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4046–4062, Online. Association for Computational Linguistics.

Shubham Kumar Nigam, Tanmay Dubey, Govind Sharma, Noel Shallum, Kripabandhu Ghosh, and Arnab Bhattacharya. 2025. Legalseg: Unlocking the structure of indian legal judgments through rhetorical role classification. *arXiv preprint arXiv:2502.05836*.

Shubham Kumar Nigam, Balaramamahanthi Deepak Patnaik, Shivam Mishra, Noel Shallum, Kripabandhu Ghosh, and Arnab Bhattacharya. 2024a. Nyayaanumana & inlegalllama: The largest indian legal judgment prediction dataset and specialized language model for enhanced decision analysis. *arXiv preprint arXiv:2412.08385*.

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.

Joel Niklaus, Ilias Chalkidis, and Matthias Stürmer. 2021. Swiss-judgment-prediction: A multilingual legal judgment prediction benchmark. *arXiv preprint arXiv:2110.00806*.

Shounak Paul, Arpan Mandal, Pawan Goyal, and Saptarshi Ghosh. 2023. Pre-trained language models for the legal domain: A case study on indian law. In *Proceedings of 19th International Conference on Artificial Intelligence and Law - ICAIL 2023*.

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, et al. 2018. Cail2018: A large-scale legal dataset for judgment prediction. *arXiv preprint arXiv:1807.02478*.

Hiroaki Yamada, Takenobu Tokunaga, Ryutaro Ohara, Akira Tokutsu, Keisuke Takeshita, and Mihoko Sumida. 2024. Japanese tort-case dataset for rationale-supported legal judgment prediction. *Artificial Intelligence and Law*, pages 1–25.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*.

A Dataset Schema and Attribute Definitions

Each record in the IBPS dataset contains thirteen attributes describing the case facts, legal context, and the final bail decision. Twelve attributes are extracted directly from the judgment documents, while one attribute is obtained from external legal sources.

cnr Unique case identifier corresponding to the bail application. This identifier can be used to retrieve the original judgment document from the eCourts portal.

bail_type Type of bail application. Possible values are *regular-bail* and *anticipatory-bail*. Bail cancellation petitions also exist, but were excluded because of their very low frequency in our metadata resource.

age_available Boolean indicator specifying whether the age of the accused is mentioned in the judgment. As shown in the Data analysis section of Appendix B, there is a positive correlation between the accused's age and the probability of bail, making this an important attribute to include.

ages List of ages of the accused individuals involved in the case (when available).

health_condition Health-related information about the accused as described in the judgment text. Again, as shown in Appendix B, worse health conditions of the accused positively correlate with the probability of bail, making it another important attribute.

past_criminal_record_exists Boolean indicator specifying whether the judgment mentions any prior criminal history for the accused.

past_criminal_record_charges List of statutes corresponding to past criminal cases or ongoing investigations involving the accused.

statutes List of statutes applied to the accused in the current case.

case_details Textual description of the case facts, including the narrative of events and arguments presented by both the prosecution and the defence.

days_in_custody Number of days the accused spent in custody before the bail decision. This value is computed from the arrest date (if available) to the date of the court order.

outcome Final decision of the High Court regarding the bail petition. Possible values are *bail granted* and *bail not granted*.

reasoning Judicial rationale provided by the court describing the rationale behind the bail decision.

statute_details Descriptions of the statutory provisions applied in the case. These descriptions are obtained from the India Code portal and provide additional legal context for the model.

A.1 Example Dataset Record

An example entry from the dataset is shown below.


```
{
  "CNR": "HCBM010291952013",
  "bail_type": "regular-bail",
  "age_available": false,
  "ages": null,
  "health_condition": "none",
  "past_criminal_record_exists": true,
  "past_criminal_record_charges": null,
  "statutes": [
      "406 IPC",
      "34 IPC",
      "409 IPC",
      "420 IPC"
  ],
  "case_details": "The applicant and
  other accused are charged with
  offences related to criminal
  breach ...",
  "days_in_custody": null,
  "outcome": "bail granted",
  "reasoning": "The power to grant or
  refuse bail is not to be...",
  "statute_details": [
    "Section 34 IPC: Acts done ...",
    "Section 409 IPC: Criminal ...",
    "Section 420 IPC: Cheating ...",
    "Section 406 IPC: Punishment ..."
  ]
}
```


B Data Analysis and Correlation Studies

The final dataset contains thirteen structured variables, including case number, bail type, outcome, withdrawal status, age, health condition, criminal history, involved statutes, arrest and judgment dates, judgment length, detailed case facts, and derived crime category. We preprocessed the data by replacing missing values (e.g., `none`, `nan`) with `NaN`, converting age into numeric and categorical formats, calculating custody duration from date differences, and parsing statutes. The dataset includes two major types of bail: Anticipatory (53.6%) and

Regular (46.4%). Figure 2 illustrates the proportion of each category. Regular bail applications had the highest grant rate at 76.7%, while anticipatory bail had a rate of 72.9%. Furthermore, we used FLAN-T5 to semantically classify crimes (e.g., theft, murder, fraud) based on case details. While some attributes were incomplete, only 42.03% had valid age and 28.44% had arrest dates; the dataset provided sufficient information to uncover robust patterns. The age of the accused ranged from 11 to

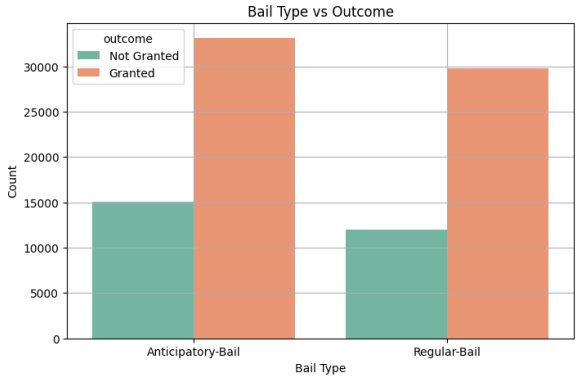


Figure 2: Bail outcome by application type.

95 years, with a mean of 37.3 and a standard deviation of 13.1. The most represented group was aged 30–50, followed by 18–30 and 50–65. A noteworthy pattern emerged linking age to bail outcomes.

A particularly influential variable was criminal history. Applicants with no prior record had a 70.8% success rate, while those with a record received bail in only 48.9% of cases, a 21.9 percentage point drop, as shown in Figure 3.

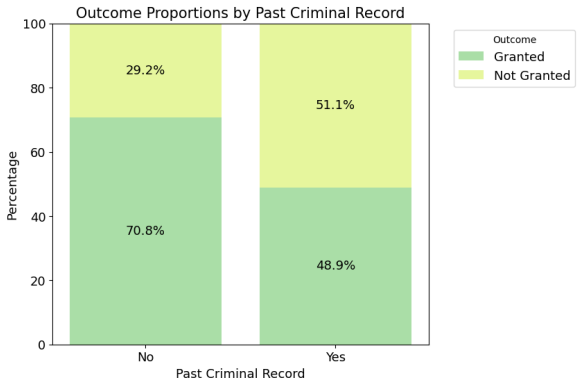


Figure 3: Effect of past criminal record on bail outcome.

Examining statute-wise bail patterns revealed that frequently cited provisions like Section 506 (criminal intimidation), Section 420 (cheating), and Section 34 (common intention) had high grant rates (above 75%), as depicted in Figure 4. Conversely, socially sensitive statutes such as Section 18 of the SC/ST Act and Section 64 of the Abkari Act had some of the lowest grant rates (see Appendix Figure 6).

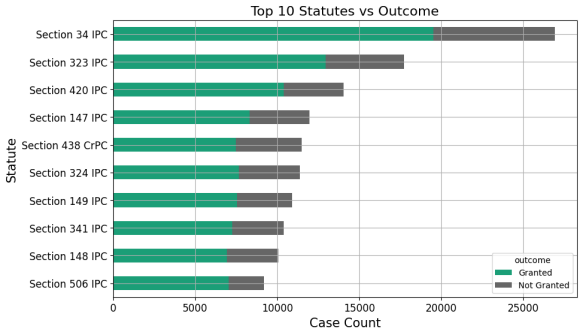


Figure 4: Top statutes and their corresponding bail outcomes.

Crime-wise outcome analysis (Figure 5) showed theft and domestic violence had high grant rates (73.2% and 81.1% respectively), whereas crimes like rape and murder had significantly lower success rates, 55.75% and 61.03% respectively. Appendix Figure 10 further illustrates the age distribution across different crime categories, revealing that youth were disproportionately involved in crimes like kidnapping and assault, while middle-aged individuals dominated in fraud and drug-related offenses.

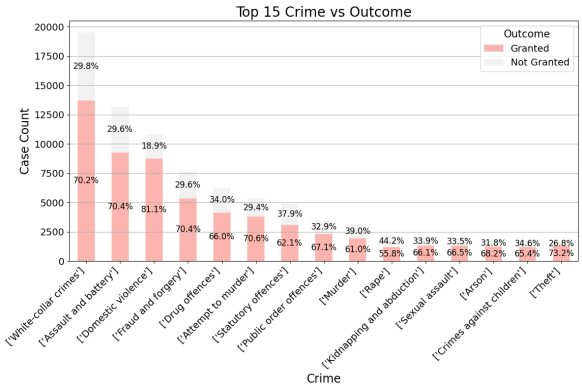


Figure 5: Bail outcomes across crime categories.

Finally, the duration of custody varied significantly by crime type. Figure 9 shows that murder cases had the longest pretrial detentions, with 25.4% of such cases extending beyond two years, while drug offences were resolved within 90 days for 78.89% of cases.

B.1 Statutes with Lowest Grant Rates

Figure 6 presents statutes associated with the lowest bail grant rates. Sections under the Abkari Act

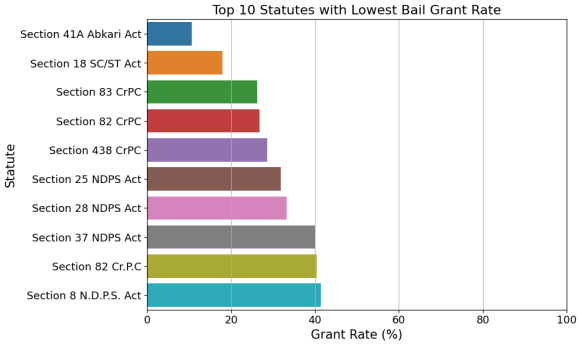


Figure 6: Statutes with lowest bail grant rates.

(e.g., Section 64, Section 41A(b)(ii)) had a 10.7% grant rate. Other socially sensitive laws like Section 18 of the SC/ST Act had grant rates below 20%, indicating stricter judicial standards for such offenses. Temporal analysis based on 25,282 cases with complete arrest and judgment dates revealed substantial delays in pretrial detention. The average custody duration was approximately 272.6 days (median: 88 days), with a maximum of over 17 years in one extreme case (Figure 7).

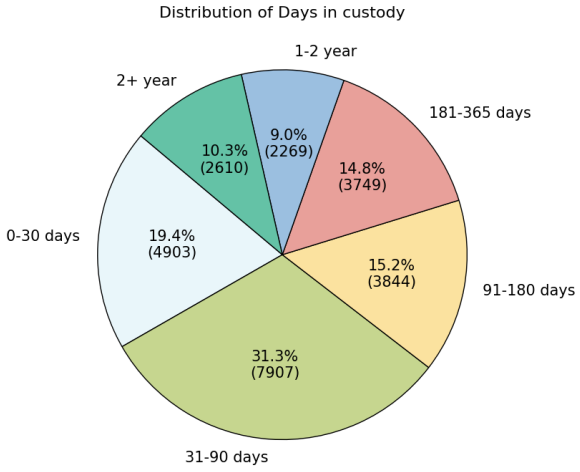


Figure 7: Custody duration distribution before judgement.

B.2 Age-Based Outcome Analysis

Figure 8 highlights how age influences bail outcomes. Applicants aged 65 and above had the highest grant rate (82.5%), while those in the 18–30 age group had the lowest (66.7%). This suggests a potential judicial inclination toward granting bail to older individuals, possibly due to perceived lower flight risk or greater humanitarian concern.

Figure 9 reveals that murder cases had the longest pretrial detentions, 25.4% of such cases extended beyond two years. In contrast, drug and sexual assault cases were resolved more quickly, with over 78% of drug-related cases and 64% of sexual assault cases resolved within 90 days. This highlights disparities in trial speed, possibly linked to the complexity of evidence gathering or legal procedure.

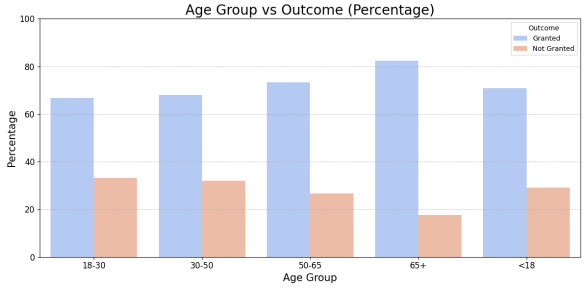


Figure 8: Bail outcomes across different age groups.

B.3 Crime-Specific Custody Duration Patterns

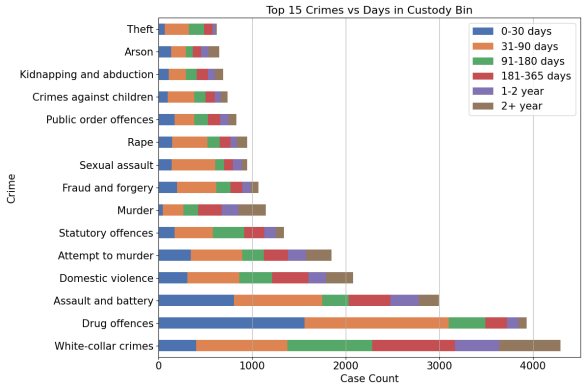


Figure 9: Custody duration distribution by type of crime.

Figure 10 explores age distribution across crime types. Youth (18–30) were predominant in crimes like kidnapping (57.11%), assault (52.7%), and rape (50.6%). Middle-aged individuals (30–50) dominated in drug-related (56.8%), fraud (56.3%), and white-collar crimes (52.4%). These age-crime relationships offer insights into offender profiling and potential rehabilitative strategies.

C Feature Extraction Experiments

This section describes preliminary experiments conducted with existing NLP models for extracting structured information from bail judgment documents. These experiments were performed to evaluate whether traditional named entity recognition (NER) and rhetorical role extraction models could

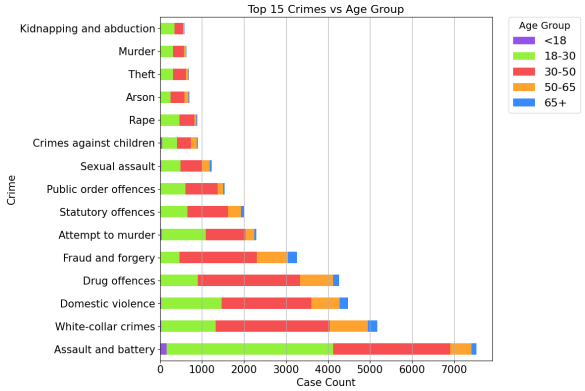


Figure 10: Age distribution of accused across crime categories.

be used for constructing the structured dataset required by IBPS.

C.1 Named Entity Recognition Models

We evaluated two legal-domain NER systems for extracting statute references and related legal entities from bail judgments:

- **en_legal_ner_trf** (Kalamkar et al., 2022)
- A custom SparkNLP-based NER model trained on the *InLegalNER* dataset.

Both models were able to detect section numbers and statute mentions as individual entities. However, they did not reliably capture the association between section numbers and the statutes to which they belong (e.g., linking "Section 420" to "IPC"). This limitation arises because the underlying training datasets annotate entities independently but do not label relationships between them.

To quantify this behaviour, we constructed a small evaluation set of sentences containing statute references and measured three metrics: section detection F1 score, statute detection F1 score, and the percentage of correctly linked sections to their corresponding statutes.

Model	Section f1.	Statute f1.	Linking %.
en_legal_ner_trf	0.90	0.87	0.00
SparkNLP InLegalNER	0.91	0.83	0.00

Table 9: Evaluation of legal NER models for statute extraction and section–statute association.

As shown in table 9, although entity detection performance was reasonable, the models consistently failed to capture the relationship between sections and statutes. As a result, these models were unsuitable for extracting structured statute information required by our dataset construction pipeline.

C.2 Rhetorical Role Extraction

We also evaluated the BiLSTM-CRF model trained on the LegalSeg dataset (Nigam et al., 2025), which is designed for rhetorical role classification in Indian legal judgments. The goal of this experiment was to determine whether the model could reliably extract structured segments such as factual narratives, legal arguments, and judicial reasoning from bail judgments.

Although we were able to reproduce the results in the *LegalSeg* paper using cases from their own dataset and even judgement texts from IndianKanoon[10] using the model, it failed to generalise to our dataset, always labelling sentences with the *None* tag even when they clearly belonged to rhetorical categories such as facts or reasoning.

Table 10 reports the accuracy for each rhetorical role in our evaluation set.

Role	Accuracy
Facts	0.00
Arguments	0.00
Reasoning	0.00
Outcome	0.00
None	0.32

Table 10: Performance of the BiLSTM-CRF rhetorical role extraction model on bail judgments.

These results suggest that models trained on existing rhetorical role datasets do not generalise well to bail judgments, likely due to structural differences between the training corpus and the bail case documents used in our dataset.

Based on these observations, we adopted a large language model based extraction approach, which provided significantly better performance and output consistency for a structured dataset construction.

C.3 Extraction LLM selection

We evaluated several open-source large language models (LLMs) for one-shot rhetorical role extraction across 100 manually verified cases that form our Gold standard. The evaluated models were selected from the top sub-20-billion-parameter open-source models on the MMLU benchmark performance leaderboard. We used MMLU as a proxy for reasoning capability, as the selected model was intended to support both rhetorical role extraction during dataset construction and reasoning generation in the IBPS system. The reason for using

[10] https://indiankanoon.org/

the same model for both tasks is that we intend to launch it as a chat tool in the future, which extracts the required information from the user-provided case details and provides a prediction and rationale for bail.

Model	Inference Time (s)	GPT-eval Score (/10)
gemma-3-12b-it	188	6.81
Llama-3.1-8B-Instruct	70	6.89
phi-4	101	7.56
Mistral-7B-Instruct-v0.3	136	6.95
DeepSeek-R1-0528-Qwen3-8B	139	6.09
Param-1-2.9B-Instruct	—	—

Table 11: Comparison of LLMs on feature extraction quality and efficiency

All experiments were conducted on an NVIDIA A100 GPU with 40GB VRAM. Table 11 summarises the inference time and evaluation scores obtained for each model.

Among the evaluated models, Phi-4's extracted features were scored the highest and had the second-lowest inference time. It was the only model that consistently produced well-structured outputs conforming to the predefined extraction schema. Other models, including LLaMA-3.1 and Mistral-7B, achieved comparable extraction quality in some cases but often produced outputs that deviated from the required format or contained hallucinated fields and symbols, making parsing them very difficult.

Based on these observations, Phi-4 was selected as the backbone model for large-scale feature extraction. We also attempted to evaluate PARAM-1-2.9B-Instruct; however, its smaller context window (2048 tokens) led to repeated failures when processing longer case documents, preventing evaluation.

Attribute	Correct	Incorrect	Incomplete
bail_type	98	0	2
age_available	100	0	0
ages	100	0	0
health_condition	100	0	0
past_criminal_record_exists	95	5	0
past_criminal_record_charges	100	0	0
statutes	100	0	0
case_details	99	0	1
days_in_custody	97	1	2
outcome	99	0	1
reasoning	100	0	0

Table 12: Verification results of extracted attributes over 100 cases

The *microsoft/phi-4* model is then used to extract attributes from our 100 Gold standard cases, and the extracted attributes are then manually verified by GPT-4o. Each extracted attribute was given one of three labels: `Correct`, `Incorrect`, or `Incomplete`. `Incorrect` label was provided if the extracted value of the attribute directly contradicted what was specified in the case judgment text. `Incomplete` label was provided when either the extracted value of the attribute didn't represent the full truth specified in the judgment text, or when the extraction model assumed some information to provide conclusive information when the facts from the case judgment itself didn't provide conclusive information. Table 12 shows the results of the verification by GPT.

D SHAP Feature Definitions

To represent statutory information in a compact form, we trained an auxiliary XGBoost classifier using the categorical attributes `statutes`, `past_criminal_record_charges`, and `past_criminal_record_exists`. The model outputs SHAP (*SHapley Additive exPlanations*) values that quantify the contribution of each statute-related feature to the predicted bail outcome.

Instead of directly using the sparse multi-hot representation of statutes, we aggregate the SHAP values into nine scalar features that summarise the magnitude and direction of statutory influence for each case.

D.1 Feature Definitions

`shap_sum_pos` Sum of positive SHAP values across all statutes in a case. This feature represents the total contribution of statutes that increase the likelihood of bail being granted.

`shap_sum_neg` Sum of negative SHAP values across statutes. This captures the overall contribution of statutes that decrease the likelihood of bail being granted.

`shap_max_pos` Maximum positive SHAP value among all statutes present in the case, representing the strongest statute supporting the bail decision.

`shap_min_neg` Most negative SHAP value among statutes, representing the statute that most strongly argues against granting bail.

`shap_pos_count` Number of statutes whose SHAP values are positive, indicating how many charges contribute positively toward bail.

`shap_neg_count` Number of statutes with negative SHAP contributions, indicating how many charges oppose bail.

`shap_l1_total` Sum of the absolute values of

SHAP contributions across statutes. This captures the overall strength of statutory influence regardless of direction.

shap_top3_abs_sum Sum of the three largest absolute SHAP values for statutes in the case, representing the combined influence of the most impactful charges.

xgb_pred Predicted probability produced by the XGBoost model using only statute- related features. This value summarises the overall effect of the statutory context on the predicted bail outcome.

The XGBoost model trained only on categorical features achieved about 65% accuracy. This was a sufficient amount of signal for xgb_pred to be included in the extracted feature set. These features provide a compact numerical representation of a case's legal context and are concatenated with textual and numerical representations in the final classification model.

E Classification Model Training Details

The bail outcome classifier is trained using a two-phase optimisation strategy that stabilises training while preserving the pretrained representations of the BigBird encoder.

E.1 Optimisation Setup

The model is trained using the AdamW optimiser with a weight decay of 0.01. Gradient clipping with a maximum norm of 1.0 is applied to prevent unstable updates. Training is performed using Automatic Mixed Precision (AMP)[11] to accelerate computation and reduce memory usage.

Input sequences are truncated or padded to a maximum length of 1200 tokens. The training batch size is 4.

E.2 Phase 1: Head Initialisation

In the first training phase, the BigBird encoder is frozen, and only the task-specific layers, i.e. the attention pooling vector and the classification head, are trained. This prevents large initial gradients from distorting the pretrained encoder representations.

Dropout is disabled during this stage ($p = 0$) to stabilise training. The classification head is trained using a learning rate of 1×10^{-3}. Training runs for up to 20 epochs with early stopping if the validation loss does not improve for five consecutive epochs.

[11] AMP-PyTorch documentation

E.3 Phase 2: End-to-End Fine-Tuning

After the classification head converges, the BigBird encoder is unfrozen, and the entire network is fine-tuned end-to-end. Differential learning rates are applied: 5×10^{-6} for the transformer encoder and 1×10^{-3} for the task-specific layers.

Regularisation is reintroduced by enabling dropout ($p = 0.1$ before the text projection layer and $p = 0.2$ after layer normalisation). A linear learning rate scheduler with warmup is used, where the learning rate increases linearly over the first 10% of training steps and then decays linearly thereafter.

Training continues for up to 30 epochs with early stopping if the validation loss does not improve for eight consecutive epochs.

Hyperparameter	Value
Optimizer	AdamW
Weight Decay	0.01
Batch Size	4
Max Sequence Length	1200
Gradient Clipping	1.0
Phase 1 Learning Rate	1×10^{-3}
Phase 2 Encoder LR	5×10^{-6}
Phase 2 Head LR	1×10^{-3}
Dropout (projection layer)	0.1
Dropout (post layer norm)	0.2
Warmup Ratio	10%
Phase 1 Epochs (max)	20
Phase 2 Epochs (max)	30

Table 13: Hyperparameters used for training the bail outcome classifier.

F Confusion Matrix

Fig 11 shows overall confusion matrix , Fig 12 shows confusion matrix for Regular-bail type(denoted as 1) and Fig 13 shows confusion matrix for Anticipatory-bail type(denoted as 2).

The confusion matrix demonstrates balanced model performance. The comparable false-positive and false-negative rates suggest that the model does not exhibit substantial bias toward either class (granted/not granted). In the context of bail prediction, such a balanced error distribution is desirable, as both incorrect approvals and wrongful denials carry significant implications.

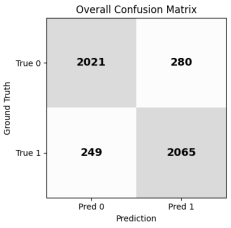


Figure 11: Overall confusion matrix

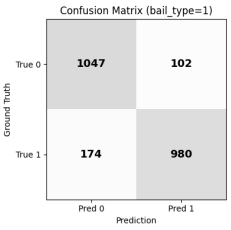


Figure 12: Confusion matrix for Regular-bail.

G LLM Fine-Tuning Configuration

This section describes the training configuration used to fine-tune the language model for bail rationale generation.

G.1 Model and Training Data

The justification module is implemented using the *microsoft/phi-4* large language model. The model is fine-tuned to generate judicial-style rationale for bail decisions, grounded in structured case attributes and statutory context.

For each training example, all available case information (including `case_details`, `ages`, `past_criminal_record_charges`, and other extracted attributes) is concatenated into a single input sequence. Descriptions of the relevant statutes are also appended to provide additional legal context. Table 14 provides the hyperparameters used to fine-tune the phi-4 LLM.

The model is fine-tuned and tested on 46,143 cases that include both structured attributes and reference judicial reasoning.

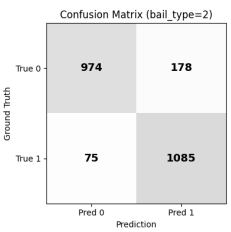


Figure 13: Confusion matrix for Anticipatory-bail.

G.2 Training Infrastructure

Fine-tuning is performed using the Axolotl training framework across four NVIDIA RTX A6000 GPUs (48GB VRAM each). Training required approximately 400 GPU hours to complete two full epochs.

Given the long legal case contexts, the maximum sequence length is set to 8,192 tokens. To support this context window efficiently, the model is trained with bfloat16 precision, Flash Attention, and gradient checkpointing.

Fully Sharded Data Parallel (FSDP) with the `full_shard` strategy is used to distribute model parameters, gradients, and optimiser states across the four GPUs.

G.3 Parameter-Efficient Fine-Tuning

Instead of updating all model parameters, we employ Low-Rank Adaptation (LoRA) to efficiently adapt the model.

LoRA modules are inserted into all linear projection layers within the transformer blocks, including the attention projections (W_q, W_k, W_v, W_o) and the feed-forward layers (W_{gate}, W_{up}, W_{down}). This configuration allows the model to adapt its reasoning behaviour while maintaining most of the pretrained weights unchanged.

G.4 Optimisation Setup

Parameter	Value
Base Model	microsoft/phi-4
Training Framework	Axolotl
GPUs	4 × NVIDIA RTX A6000 (48GB)
Dataset Size	41,528 (training) + 4,615 (testing)
Sequence Length	8192 tokens
Precision	bfloat16
Attention Optimisation	Flash Attention
Parallelisation	FSDP (full_shard)
LoRA Rank (r)	32
LoRA Scaling (α)	64
LoRA Dropout	0.05
Optimizer	AdamW
Peak Learning Rate	5×10^{-5}
Scheduler	Cosine Annealing
Warmup Steps	100
Micro Batch Size	1
Gradient Accumulation	2
Effective Batch Size	8
Training Epochs	2
Training Cost	∼400 GPU hours

Table 14: Hyperparameters used for LLM fine-tuning.

The model is optimised using the AdamW optimiser with a peak learning rate of 5×10^{-5}. A

cosine annealing learning rate scheduler is used following a 100-step linear warmup phase.

To stabilise training across the distributed setup, we use a micro-batch size of 1 per device with 2 gradient accumulation steps, resulting in an effective global batch size of 8.

After training, the distributed FSDP weights are consolidated, and the LoRA parameters are merged into the base model weights to produce the final inference-ready checkpoint.

H Prompt Templates

H.1 Dataset extraction prompt

```
f'''
    given below is a python dictionary format to be filled with information about
    the case given in the raw judgement text. replace the text between the < and > with
    the information extracted from the raw judgement text.
    **DO NOT COPY THE TEXT BETWEEN < AND >, INSTEAD REPLACE IT WITH THE EXTRACTED INFORMATION.**

    python dict format :
    {
        "case":"""
            Applicant applied for <type of application, "Regular-Bail" OR "Anticipatory-
            Bail" OR "Bail-Cancellation", (one of these)>.
            Is it a withdrawal application? <"Yes" or "No" depending upon if it is
            application for withdrawal>.
            Age of the accused is <age of the accused if provided, else write "not
            provided">.
            Health issues for the accused are <description of health issues if provided,
            else "None">.
            There are <"no" if there are no past criminal records, else "some"> past
            criminal records of the accused.
            Statutes mentioned in the judgement are <list of statutes,
            eg: [Section 438 CrPC, Section 294(a) IPC, Section 506(1)(b) IPC, Section 34
            IPC, Section 25 Arms Act],
            do not include the acts/codes/sections that were removed or replaced later>.
            Precedents mentioned in the judgement are <list of precedents, if any, else
            "None">.
            Details of the incident are <details of the incident if provided, else "None">.
            Arguments supporting the bail application are <arguments supporting the bail
            application, if any, else "None">.
            Arguments opposing the bail application are <arguments opposing the bail
            application, if any, else "None">."
        """
        "outcome": "The outcome of the case is <status of the outcome, "Bail granted" OR
        "Bail not granted" OR "Bail cancelled" OR "Bail not cancelled", (one of these)>.
        The bail conditions are <list of bail conditions, if any, else "None">.
      "reasoning": "The reasoning for the judgement is <list of reasoning, if any, else "None">."
        "date_of_arrest": "<date of arrest, if provided, else "not provided">."
        "date_of_judgement": "<date of judgement, if provided, else "not provided">."
    }

    For example, if the raw judgement text is as follows:

    IN THE HIGH COURT OF KERALA AT ERNAKULAM ...

    the output should be as follows:
    ```json
    {
        "case": "Applicant applied for ...
    }
    ```
    similarly given a raw judgement text, extract the information and convert it into
    given json format.
    Respond **ONLY** with valid JSON matching the schema. Do not add explanations or data
    from example itself into the JSON.

    Raw Judgement to process:
    < case to be processed >
'''
```

H.2 G-Eval Prompt for Attribute Validation

```
Instructions:
You are an expert in legal text evaluation. You will be given:

A full case judgement text that is the source of all the information.
An expert evaluated text that has all the detail of the case extracted from the full case text
that serves as the reference.
A machine generated text after extracting all the details of the case from the full case text that
needs to be evaluated.
Your task is to assess how well the machine generated text aligns with the given description while
using the actual document as a reference for correctness.

Evaluation Criteria (Unified Score: 1-10)
Your evaluation should be based on the following factors:

Factual Accuracy (50%) – Does the generated text correctly represent the key legal facts,
reasoning, and outcomes from the full case text, as expected from the description?
Completeness & Coverage (30%) – Does it include all crucial legal arguments, case details, and
necessary context that the expert evaluated text implies?
Clarity & Coherence (20%) – Is the text well-structured, logically presented, and legally sound?

Scoring Scale:
1-3 → Highly inaccurate, major omissions or distortions, poorly structured.
4-6 → Somewhat accurate but incomplete, missing key legal reasoning or context.
7-9 → Mostly accurate, well-structured, with minor omissions or inconsistencies.
10 → Fully aligned with the description, factually accurate, complete, and coherent.

Input Format:
Full case text:
{{full_case}}

Expert Evaluated Text (Reference):
{{Actual_Document}}

Machine Generated Text (To Be Evaluated):
{{Generated_Document}}

Output Format:
Strictly provide only a single integer score (1-10) as the response, with no explanations,
comments, or additional text.
```

H.3 G-Eval Prompt for Rationale Evaluation

```
You are an expert in legal text evaluation. You will be given:

   A full case details that is the source of all the information regarding the events, people and
    objects related to the crime.
   A machine generated text that is the predicted outcome and reason behind the judgement of the
    court that needs to be evaluated.
   A ground truth text that has the actual judgement and reason behind the provided judgement of
    the court that serves as the reference.
    Your task is to assess how well the machine generated text aligns with the given reasoning
    while using the actual reasoning of the court as a reference for correctness.
    Be fair but very strict with grading.

    Evaluation Criteria (Unified Score: 1-10)
    Your evaluation should be based on the following factors:

    Factual Accuracy - Does the machine generated text correctly predict the actual court
    judgement(most important part. without this being correct everything else is useless.) and
   reasoning, while correctly referencing key facts from the full case text? This also includes
    correct understanding of laws and statutes applied in the case.
    Completeness & Coverage - Does the machine generated text take into account all the case
   facts, legal arguments and necessary context that the actual court judgement used to arrive at
    the decision?
```

```
     Clarity & Coherence - Is the text well-structured and logically presented so that the
    connection between the actual events of the case, the laws, the inferences drawn from them and
     the outcome are clearly visible?
    Extra flag - Does the actual court judgement's reasoning uses a new fact that is not mentioned
     in the full case detail text. A "fact" is an actual information about an event or person or
     object related to crime, not an argument or inference that can be derived from available
     information or some intricacies of a law.

     Scoring Scale:
     1-3 → Highly inaccurate, major omissions or distortions, poorly structured.
     4-6 → Somewhat accurate but incomplete, missing key legal reasoning or context.
     7-9 → Mostly accurate, well-structured, with minor omissions or inconsistencies.
     10 → Fully aligned with the description, factually accurate, complete, and coherent.

     Output Format:
     FA: Score representing factual accuracy of the machine generated reasoning.
     C1: Score representing Completeness and coverage of the machine generated reasoning.
     C2: Score representing Clarity and Coherence of the machine generated reasoning.
     Flag: '0' if NO new fact is used in actual reasoning behind the judgement provided by the
     court, otherwise '1'. (don't actually print inverted commas)

     Strictly provide the response comma separated format like this:
     FA, C1, C2, Flag

     Don't output explanations, comments, or additional text.
     Use the reference only to verify factual and inference correctness.
     Do NOT compare writing quality to the reference.
     Score the candidate independently.


Input:
Case Details and Court Judgement:
{{full_case}}

Court's reasoning behind judgement (Reference):
{{Actual_reason}}

Machine Generated reasoning behind judgement (To Be Evaluated):
{{Generated_reason}}
```

I Sample Judgement

IN THE HIGH COURT OF JHARKHAND AT RANCHI
B.A. No. 7892 of 2018

2018:JHHC:27900

Rila Devi ... Petitioner(s).
Versus
The State of Jharkhand … Opp. Party(s).

CORAM : HON'BLE MR. JUSTICE ANANDA SEN.

For the Petitioner(s) : Mr. P.K. Mukhopadhyay, Advocate.
For the State : A.P.P.
…….

03/29.11.2018: Heard the counsel for the parties.

Learned APP opposes the prayer for bail.

The petitioner is an accused for the offence punishable under Section(s) 302, 201, 120-B and 34 of the Indian Penal Code.

It is stated by the informant that one Santosh Manjhi, who happens to be her son-in-law, always threatened them that he will commit murder of the deceased to get the job in BCCL on compassionate ground. The deceased left his house on 15.6.2018 but he did not return and on the next day, his dead body was found. The informant suspected that it is Santosh Manjhi who along with this petitioner has committed the murder of the deceased.

Counsel for the petitioner submits that only the basis of suspicion, Santosh Manjhi was arrested who confessed his guilt. He further submits that there is no material against the petitioner, except confessional statement of Santosh Manjhi before the police, which has got no evidentiary value.

Since there is no material against the petitioner, except confessional statement of one Santosh Manjhi and the fact that the petitioner is a lady, I am inclined to release the petitioner on bail. Accordingly, the petitioner, above named, is directed to be released on bail on furnishing bail bond of Rs.10,000/-(Rupees ten thousand) with two sureties of the like amount each to the satisfaction of the learned Judicial Magistrate, 1st Class, Dhanbad, in connection with Putki (Bhagabandh OP) P.S. Case No. 61/2018, corresponding to G.R. No. 2483 of 2018.

Anu/-C.P.-3 **(ANANDA SEN, J.)**

Figure 14: Sample high court judgement representing raw data before LLM extraction.

J Instructions for Legal Evaluators

You are requested to evaluate legal reasoning generated by a machine learning system.

For each case, you will be provided with the following materials:

- **Full Case Details** – This section contains all factual information regarding the events, persons, and objects relevant to the case.
- **Court's Reasoning** – The reasoning provided by the court in support of its decision. This serves as the reference standard.

- **Two Machine-Generated Justifications** – Two justifications generated by the system. Your task is to evaluate each system-generated text based solely on the quality, correctness, and alignment of the outcome and justification produced by the system.

Evaluation Criteria

1. Factual Accuracy (FA)

- Does the justification accurately reflect the reasoning adopted by the court?
- Are key facts from the case details correctly referenced and interpreted?
- Are relevant laws, statutes, and legal principles correctly understood and applied?
- Does the reasoning avoid factual distortions or incorrect legal inferences?

Note: This is the most important evaluation criterion.

2. Completeness and Coverage (C1)

- Does the justification address all material facts and legal considerations relied upon by the court?
- Are significant arguments, contextual elements, or statutory considerations omitted?
- Does the justification capture the core structure of the court's reasoning?

3. Clarity and Coherence (C2)

- Is the justification logically structured?
- Are the connections between facts, legal principles, and conclusions clearly articulated?
- Is the reasoning internally consistent and easy to follow?

4. Extra Fact Flag

Indicate whether the **court's reasoning** relies on a factual element that is not present in the provided case details. This is intended to identify cases where the information provided in the *Case Details* section may itself be incomplete.

- A **fact** refers strictly to concrete information about an event, person, object, or circumstance relevant to the case.
- It does **not** include legal interpretations, logical deductions, or inferences derived from existing facts.

Flag Guidelines

- Leave the checkbox empty if no new factual element appears in the court's reasoning.
- Mark the checkbox if at least one new factual element appears in the court's reasoning.

Scoring Scale (For FA, C1, C2)

Score Range	Interpretation
1–3	Highly inaccurate reasoning with major omissions or distortions and poor logical structure.
4–6	Partially accurate reasoning but incomplete, with important legal reasoning or contextual elements missing.
7–9	Largely accurate and well-structured reasoning, with only minor omissions or inconsistencies.
10	Fully aligned with the court's reasoning, factually precise, comprehensive, and coherent.

Evaluation Guidelines

- Use the court's reasoning only as a reference for verifying factual and inferential correctness.
- Do not compare writing style or literary quality.
- Evaluate each machine-generated justification independently.
- Focus strictly on the quality of legal reasoning and alignment with the actual judgment.
- Apply strict but fair standards when assigning scores.

K Zero-shot and One-shot Comparison

K.1 Zero-shot

In this technique the extraction output was not consistent and not following the provided schema.It also loses some factual details, precision, and completeness.

K.1.1 Zero-shot output using sample judgement


```
```json
{
   "case": "Applicant applied for Regular-Bail. Is it a withdrawal application? No.
    Age of the accused is not provided. Health issues for the accused are None.
    There are no past criminal records of the accused. Statutes mentioned in the
    judgement are [Section 302 IPC, Section 201 IPC, Section 120-B IPC, Section 34
    IPC]. Precedents mentioned in the judgement are None. Details of the incident
    are The informant suspected that Santosh Manjhi, along with the petitioner,
    committed the murder of the deceased. Arguments supporting the bail application
    are There is no material against the petitioner, except the confessional
    statement of Santosh Manjhi, which has no evidentiary value. Arguments opposing
    the bail application are Learned APP opposes the prayer for bail.",
    "outcome": "The outcome of the case is Bail granted. The bail conditions are
    [Furnishing bail bond of Rs.10,000/- with two sureties of the like amount each
    to the satisfaction of the learned Judicial Magistrate, 1st Class, Dhanbad].",
    "reasoning": ["There is no material against the petitioner, except the
    confessional statement of Santosh Manjhi, which has no evidentiary value.", "The
    petitioner is a lady."],
    "date_of_arrest": "not provided",
    "date_of_judgement": "2018"
}
```
```

K.2 One-shot

In this technique the output is consistent and following the schema along with this it follows the original judgment, preserving detailed facts, exact dates, and complete bail conditions

K.2.1 One-shot output using sample judgement


```
```json
{
   "case": "Applicant applied for Regular-Bail.\nIs it a withdrawal application?
    No.\nAge of the accused is not provided.\nHealth issues for the accused are
    None.\nThere are no past criminal records of the accused.\nStatutes mentioned in
    the judgement are [Section 302 IPC, Section 201 IPC, Section 120-B IPC, Section
    34 IPC].\nPrecedents mentioned in the judgement are None.\nDetails of the
    incident are The petitioner is an accused for the offence punishable under
    Section(s) 302, 201, 120-B and 34 of the Indian Penal Code. It is stated by the
    informant that one Santosh Manjhi, who happens to be her son-in-law, always
    threatened them that he will commit murder of the deceased to get the job in
    BCCL on compassionate ground. The deceased left his house on 15.6.2018 but he
    did not return and on the next day, his dead body was found. The informant
```
```

```
   suspected that it is Santosh Manjhi who along with this petitioner has committed
    the murder of the deceased.\nArguments supporting the bail application are
    Counsel for the petitioner submits that only the basis of suspicion, Santosh
   Manjhi was arrested who confessed his guilt. He further submits that there is no
   material against the petitioner, except confessional statement of Santosh Manjhi
   before the police, which has got no evidentiary value.\nArguments opposing the
    bail application are Learned APP opposes the prayer for bail.",

    "outcome": "The outcome of the case is Bail granted. The bail conditions are
   [furnishing bail bond of Rs.10,000/-(Rupees ten thousand) with two sureties of
   the like amount each to the satisfaction of the learned Judicial Magistrate, 1st
   Class, Dhanbad, in connection with Putki (Bhagabandh OP) P.S. Case No. 61/2018,
    corresponding to G.R. No. 2483 of 2018].",
    "reasoning": "Since there is no material against the petitioner, except
   confessional statement of one Santosh Manjhi and the fact that the petitioner is
    a lady, I am inclined to release the petitioner on bail.",
    "date_of_arrest": "not provided",
    "date_of_judgement": "29.11.2018"
}
```