LexChronos: An Agentic Framework for Structured Event Timeline Extraction in Indian Jurisprudence

Anka Chandrahas Tummepalli, Preethu Rose Anish
TCS Research
ankachandrahas.t@tcs.com, preethu.rose@tcs.com

Abstract

Understanding and predicting judicial outcomes demands nuanced analysis of legal documents. Traditional approaches treat judgments and proceedings as unstructured text, limiting the effectiveness of large language models (LLMs) in tasks such as summarization, argument generation, and judgment prediction. We propose *LexChronos*, an agentic framework that iteratively extracts structured event timelines from Supreme Court of India judgments. *LexChronos* employs a dual-agent architecture: a LoRA-instruct-tuned extraction agent identifies candidate events, while a pre-trained feedback agent scores and refines them through a confidence-driven loop. To address the scarcity of Indian legal event datasets, we construct a synthetic corpus of 2000 samples using reverse-engineering techniques with DeepSeek-R1 and GPT-4, generating gold-standard event annotations. Our pipeline achieves a BERT-based F1 score of 0.8751 against this synthetic ground truth. In downstream evaluations on legal text summarization, GPT-4 preferred structured timelines over unstructured baselines in 75% of cases, demonstrating improved comprehension and reasoning in Indian jurisprudence. This work lays a foundation for future legal AI applications in the Indian context, such as precedent mapping, argument synthesis, and predictive judgment modelling, by harnessing structured representations of legal events.

Keywords

Agentic AI, Event Extraction, Legal Document Analysis, Indian Jurisprudence.

1 Introduction

Legal reasoning in high-stakes contexts demands not only interpretive precision but also a structured understanding of complex case narratives. This requirement underscores the critical role of event extraction—the process of identifying and structuring key facts, actors, and temporal relationships (Xian et al. 2024). Despite its importance, event extraction remains a persistent challenge for legal AI systems due to the inherent complexity and unstructured nature of judicial documents, particularly at the document level where tracking events across lengthy texts is essential.

Globally, AI has advanced beyond theoretical applications to practical tools in the legal sector, supporting tasks such as transcription, translation, case management (Press Information Bureau, Government of India 2025), smart scheduling and predictive analytics for backlog reduction. Large language models (LLMs) are increasingly deployed for legal research, drafting and analysis (Deeks and Hollis 2025). However, extracting structured event timelines from court judgments remains largely unexplored—especially within the intricate landscape of the Indian judiciary, one of the largest in the world, with over 86,742 pending cases in the Supreme Court alone as of 2025 (Rohit Singh 2025).

Indian legal texts pose unique challenges: dense statutory language, common law references, constitutional citations, and procedural dependencies. Traditional NLP techniques often fail to capture the temporal, causal, and hierarchical structures embedded in these documents. While sentence-level event extraction has seen progress, document-level extraction requiring co-reference resolution, entity tracking across paragraphs, and temporal-causal linkage remains a bottleneck (Huang and Peng 2021). The absence of structured representations significantly limits downstream tasks such as precedent mapping, argument generation, and judgment prediction.

Although recent advances in LLMs for legal applications (Tiwari et al. 2024) have improved text understanding, most approaches treat legal documents as monolithic text blocks, overlooking the nuanced event structures that underpin legal reasoning. Even as the Supreme Court of India adopts AI for administrative efficiency, comprehensive event extraction from judgment remains an underexplored frontier with transformative potential.

This paper addresses two key gaps:

(1) Limitations of LLMs in extracting and structuring events from unstructured court documents. (2) the lack of publicly available event-level annotated datasets for Indian Supreme Court judgments, hindering reproducibility and benchmarking.

To bridge these gaps, we introduce *LexChronos*, an agentic framework for iterative extraction of structured event timelines from legal texts. Our dual-agent pipeline comprises:

1. An extraction agent (LoRA-instruct-tuned LLMs $< 4B$ parameters) propose candidate events, including timestamps, descriptions, judge(s) and precedent references.

2. A feedback agent (LLMs $< 4B$ parameters) evalu-

ate candidates, assign confidence scores, and refine outputs through a stopping-criteria–driven loop. This meta-cognitive feedback ensures semantic coherence and factual accuracy, converging toward high-confidence timeline.

To establish a benchmark, we construct a synthetic corpus of 2000 Supreme Court judgments using DeepSeek-R1 (Guo et al. 2025) and GPT-4 (Achiam et al. 2023), complete with gold-standard event annotations.

Our key contributions are:

- A dual agent iterative refinement framework for extracting structured event timelines from unstructured legal texts.
- A synthetic, annotated dataset of 2000 Indian Supreme Court judgments tailored for even-level legal NLP tasks.

By introducing structured event timelines as an intermediate representation, *LexChronos* enhances transparency and reliability in legal reasoning, paving the way for advanced tools in precedent mapping, argument generation, and judgment prediction. This work lays the foundation for intelligent and accountable AI integration in India's judicial system.

The rest of the paper is organized as follows: Section 2 reviews Related work, Section 3 explains Dataset creation, Section 4 presents Methodology, Section 5 presents Experimentation and Evaluation results, Section 6 provides Data and Code Availability, Section 7 presents Threats to Validity and Section 8 concludes with Conclusion and Future work.

2 Related Work

Event extraction plays a pivotal role in transforming unstructured text into structured representations, enabling downstream tasks such as summarization, timeline generation, and decision support. Across domains, from biomedical literature to financial disclosures, news media, and legal documents, event extraction has evolved to address domain-specific challenges such as nested structures, long-distance dependencies, and semantic ambiguity. This section reviews key developments in three segments: event extraction across domains, specialized approaches in the legal domain, and emerging efforts tailored to the Indian legal system.

2.1 Event Extraction Across Domains

In the biomedical domain, to handle nested event structures and long-distance dependencies, the BEESL model (Alan et al. 2020) reframed biomedical event extraction as a unified sequence labeling task using BERT-based encoding, multi-label decoding and multi-task learning enabling end-to-end handling of structural complexity. Additionally, CNNs integrated with dependency path embeddings and diverse feature types have proven effective in pipeline-based event and relation extraction (Björne and Salakoski 2018).

In the financial domain, documents such as corporate announcements present challenges due to their length and semantic density. The DocFEE dataset (Chen et al. 2025) supports document-level Chinese financial event extraction, while the Matrix-Chunking Method (MACK) (Huang et al. 2024) introduces fault-tolerance mechanisms for robust extraction from raw text. Similarly, in news media, hybrid approaches combining pattern-based heuristics, machine learning models and word embeddings have been used to maintain high trigger accuracy in real-time risk assessment scenarios (Han, Hao, and Huang 2018).

2.2 Event Extraction in the Legal Domain

Legal event extraction has evolved from rule-based systems to sophisticated language model-driven frameworks. Early work by (Sierra et al. 2018) used part-of-speech tagging and grammatical pattern matching to extract core event attributes from Mexican Spanish legal texts. (Filtz et al. 2020) benchmarked CRFs, deep learning and fine-tuned BERT variants on the ECHR dataset, revealing that CRFs excel at actor identification, while BERT variants outperform in temporal and procedural classifications. (Xian et al. 2024) introduced DLEE, the first large-scale Chinese legal document-level event extraction dataset. They employed a semi-automated annotation pipeline combining expert-designed schemas, trigger lexicons, and a BERT-based QA model to ensure fine-grained, high-quality labeling.

(Xu 2024) combined BERT, BiLSTM, and CRF architectures to extract event entities in economic legal texts, achieving robust cross-genre performance. Recent advances in LLMs have enabled more flexible and context-aware extraction. (Yue et al. 2024) proposed a cooperative framework for criminal court view generation, using LLMs to construct event narratives supporting verdict explanations.

Graph-based approaches, such as those by (Zhao, Zhao, and Mao 2024), use event entity extraction to construct judicial knowledge graphs, enhancing document interpretability and legal reasoning.

2.3 Indian Legal Event Extraction

In Indian Legal context, (Naik, Patel, and Kannan 2023) developed a spaCy-based NER model to extract key entities such as court names, petitioners, and legal acts from Indian legal documents. (Kalamkar et al. 2022) introduced a large, annotated corpus of Indian court judgments containing 46,545 legal named entities across 14 fine-grained types. They developed and released a baseline NER model trained on this dataset to support downstream legal NLP tasks like relation extraction and knowledge graph construction. (Tiwari et al. 2024) introduced AALAP, a Mistral 7B model fine-tuned on a bespoke legal corpus for tasks such as issues generation, argument generation and event timeline construction. While outperforming GPT-3.5 on several tasks, AALAP underperformed in summarization and constitutional Q&A. (Hussain and Thomas 2024) conducted a comparative evaluation of large language models, including Mistral and Gemma, for extracting judicial entities from Indian case law. They fine-tuned and tested these models to assess their effectiveness in automating structured information extraction tasks relevant to legal NLP. Systems like Legal Sarathi (Shivananda et al. 2024) integrates LLMs with machine learning algorithms to extract critical events and participants from unstructured legal documents. Its Retrieval-Augmented Generation (RAG) framework and Streamlit interface support real-time query-based retrieval and decision making.

Judgment Component	*LES Coverage*	*Justification of Coverage*
Facts	Timestamp, Event	Factual elements are encoded as **Events** with corresponding **Timestamps** establishing the case chronology.
Issues	Event	The framing of issues is a critical procedural **Event** which dictates the focus of the subsequent legal analysis.
Petitioner's Arguments	Event, Precedent	Arguments constitute procedural **Events**; their basis relies heavily on **Precedent** or statutory references.
Respondent's Arguments	Event, Precedent	Similar to petitioner's claims, these are timed procedural **Events** grounded in **Precedent**.
Analysis of the Law	Event, Precedent	The application of law is a judicial **Event** and is intrinsically linked to the **Precedent** being discussed.
Precedent Analysis	Event, Precedent, Judge	This analytical **Event** explicitly involves the cited **Precedent** and the authoritative **Judge** performing the analysis.
Court's Reasoning	Timestamp, Event, Judge	The reasoning process forms the decisive **Event**, rendered by the **Judge**, leading up to the final **Timestamp** of the order.
Conclusion	Timestamp, Event, Judge	The final order is the conclusive **Event**, delivered by the authoritative **Judge** at the final **Timestamp**.

Table 1: Mapping core Judgment components to LES

Despite these advances, there remains a significant gap in resources tailored to the Indian judicial context. Existing works often focus on narrow event categories and lack comprehensive coverage. Crucially, no publicly available dataset captures the full spectrum of events in Indian Supreme Court documents.

To address this, we introduce *LexChronos*, an agentic framework designed to iteratively extract structured event timelines from Indian Supreme Court judgment documents. We construct a synthetic corpus of 2,000 documents using reverse-engineering techniques with DeepSeek R1 and GPT-4, generating high-quality gold-standard event annotations. *LexChronos* employs a dual-agent architecture: a LoRA instruct-tuned extraction agent identifies candidate events, while a feedback agent refines them using confidence-driven stopping criteria.

3 Dataset Creation

To address the critical lack of publicly available resources for event-level annotation in the Indian legal domain, we constructed a synthetic dataset tailored to the structure and semantics of Indian Supreme Court judgments. This dataset was generated using a reverse-engineering pipeline designed to produce high-fidelity, structured representations of legal proceedings.

The dataset creation pipeline involves three key stages: Case category selection, Event timeline generation, and Judgment text generation.

3.1 Case Category Selection

Case categories in the Supreme Court of India refer to distinct classifications of cases based on factors such as the legal issues involved, the subject matter of the dispute, the composition of the bench (coram), or other relevant considerations (Supreme Court of India 2025b). The dataset creation pipeline begins by randomly selecting a case category from a curated set of 25 case categories defined by the Supreme Court's official case classification system (Supreme Court of India 2025a). These categories were chosen because they represent high-volume, socially impactful, and specialized domains—ranging from foundational areas like Criminal and Civil Law to emerging fields such as Cyber Law and Intellectual Property Rights. This diversity ensures that the synthetic dataset avoids narrow algorithmic biases and reflects the procedural and topical breadth of Indian legal cases. The complete list of case categories is provided in Appendix A.

3.2 Event Timeline Generation

Once a case category is selected, the next step involves generating a structured timeline of legal events. In this section, we explain the process of event timeline generations and its validation.

Structured Event Timeline Construction

For each selected case category, we use a prompt ($Prompt_E$) to generate a structured timeline of events using DeepSeek-R1 and GPT-4

$$E = Prompt_E(case\ category) \quad (1)$$

Each output E is a JSON array of event objects, where each object contains four attributes:

- *Timestamp*: Time or date of the event.
- *Event*: Narrative description of the event.
- *Judge*: Name(s) of the presiding judge(s), if applicable.
- *Precedent*: Legal precedents cited during the event.

This four-attribute schema (henceforth referred to as LexChronos Event Schema—LES) is derived through an inductive analysis of Supreme Court of India judgment documents. By examining the structural components of judicial reasoning—such as temporal progression, judicial actors, facts and cited precedents—we abstracted these into four core attributes in LES. This enables a granular, machine-readable representation of the judgment's legal history.

Model	*Avg Length of Judgments*	*Avg Events per Case*	*Avg Precedents per Case*	*Unique Vocab Size*
DeepSeek-R1	1010.96	27	6	34623
GPT-4	552.92	19	3	17080

Table 2: Structural and linguistic comparison of DeepSeek R1 and GPT-4 judgments

Core Judgment Component with Judgment text
Facts
"The genesis of the dispute traces back to January 2020, when Proxima Inc. acquired a substantial stake in Zenith Corp, leading to disputes concerning corporate governance and the rights of minority shareholders."
Issues
"..the principal issues discussed revolved around corporate governance, the rights of minority shareholders, and the extent of judicial oversight permissible over corporate decisions."
Petitioner's Arguments
"In February 2020, the minority shareholders of Zenith Corp alleged oppression and mismanagement by the board members of Proxima Inc., leading to the filing of a complaint. . . "
Respondent's Arguments
"Proxima Inc. contended that their actions were in compliance with the Companies Act, 2013, and cited the precedent in Furlong Steel Ltd. v. Cherry Steel Corp. 5 SCC 739 to support the legality of their board decisions."
Analysis of the Law
"Upon careful consideration of the arguments and evidence presented, we find that the need to protect minority shareholders and ensure transparency in corporate governance is paramount."
Precedent Analysis
"Proxima Inc., aggrieved by the injunction, appealed to the High Court in March 2022, invoking the precedent set in Bright Solar Ltd. v. Gloomy Electric Co. 9 SCC 600..."
Court's Reasoning
"It appears that the NCLT's findings were well-founded on the basis of the evidence, particularly the forensic audit report, which revealed significant financial mismanagement."
Conclusion
"In light of the foregoing, we uphold the judgment of the NCLT, affirming the restructuring of Proxima Inc.'s board and the compensation awarded to the affected minority shareholders. The appeal is therefore dismissed..."

Table 3: Structural Component Mapping for Synthetic Judgment Text

LES Validation

To validate the sufficiency of the LES, we map it against the eight core components typically found in Indian Supreme Court judgments, as identified in legal information retrieval systems such as IndianKanoon (Indian Kanoon 2025). Table 1 presents this mapping, demonstrating that the four-attribute schema effectively captures all substantive legal content—facts, reasoning, and conclusions required for comprehensive legal understanding. This confirms the schema's adequacy for event-level extraction and its non-reducibility.

3.3 Judgment Text Generation

With the structured event timeline in place, we proceed to generate the corresponding judgment text. This step ensures that the dataset includes both structured annotations and realistic legal narratives.

Judgment Text Construction

Using the structured event timeline E, a second prompt ($Prompt_J$) generates a synthetic judgment document:

$$J = Prompt_J(E) \tag{2}$$

$Prompt_J$ is guided by patterns derived from authentic judgments sourced via platforms like IndianKanoon. This ensures that the generated judgment text (J) emulates the linguistic complexity, structural conventions, and formatting of real Supreme Court judgments, an essential requirement for downstream legal reasoning tasks.

By repeating the entire pipeline 2,000 times, we construct the complete synthetic dataset:

$$D = \{(E_i, J_i)\}_{i=1}^{2000} \tag{3}$$

This dataset pairs richly annotated event timelines with corresponding judgment texts across all 25 case categories. The size of 2,000 samples was chosen to ensure sufficient coverage of linguistic and structural variability across categories while maintaining computational feasibility for generation and downstream processing. DeepSeek-R1 generated 1,000 samples, while GPT-4 produced the remaining 1,000. DeepSeek-R1 was selected for its strength in structured output and logical consistency, while GPT-4 was chosen for its fluency and rhetorical fidelity in long-form legal text generation (Guo et al. 2025; Achiam et al. 2023). Table 2 compares key structural and linguistic characteristics of judgments generated by DeepSeek R1 and GPT-4.

Validation

To validate the structural fidelity of the synthetic corpus, we present a detailed component breakdown of a randomly selected judgment text J from the dataset. Table 3 illustrates the presence and clarity of all eight core judgment components within the synthetic document. This confirms that the generative constraints embedded in $Prompt_J$ successfully replicate authentic judicial rhetoric and organization.

Both $Prompt_E$ and $Prompt_J$ utilize Zero-shot prompting, enhanced with Role Prompting (Wang et al. 2024) and Style Prompting (Lu et al. 2023). These prompts are provided in the GitHub repository (Tummepalli 2025). This strategy ensures consistency between the structured event annotations and the generated judgment texts, functioning as a constraint satisfaction mechanism that guarantees high quality outputs. By integrating a diverse set of case categories, a rigorously defined event schema, and a dual-model generation strategy, this synthetic dataset supports robust training and evaluation of legal NLP models—particularly for event extraction tasks within the Indian legal context. Figure 1 illustrates the complete pipeline used for dataset creation.

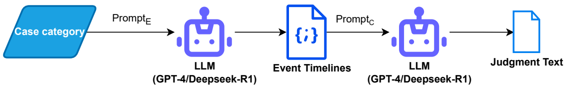

Figure 1: Reverse-Engineering Pipeline for Gold-Standard Event timeline Dataset

4 Methodology

LexChronos, employs a dual-agent architecture comprising an extraction agent and a feedback agent operating in an iterative refinement loop governed by predefined stopping criteria. This collaborative design ensures systematic improvement in event timeline extraction through progressive validation and feedback.

4.1 Event Timeline Extraction

LexChronos follows a structured, iterative process where each Judgment document J undergoes progressive refinement. An instruct-tuned open-source LLM ($<= 4B$ parameters) trained via LoRA on our synthetic dataset, acts as the extraction agent. Given a Judgment document J, the extraction agent generates an initial set of candidate events $E_0 = (e_1, e_2, ..., e_n)$, where each event e_i includes timestamp, description, judge, and precedent information, forming a preliminary.

A pre-trained language model ($< 4B$ parameters) serves as the feedback agent, evaluating E_0 against J and generating feedback F_0 across multiple quality dimensions. The process iterates as follows:

$$E_0 = ExtractionAgent(J) \tag{4}$$

$$F_0 = FeedbackAgent(J, E_0) \tag{5}$$

$$E_{i+1} = ExtractionAgent(J, E_i, F_i) \tag{6}$$

$$F_{i+1} = FeedbackAgent(J, E_{i+1}) \tag{7}$$

where E_i, F_i denotes intermediate stages of refinement process. E_{i+1} and F_{i+1} will be last pair of refinement steps where E_{i+1} will be the final extracted event timeline if the stopping criteria is met.

The iterative process terminates when either of the two stopping conditions (Patience limit and tolerance threshold limit) is satisfied.

- The **patience limit** is triggered when the confidence score fails to exceed the best achieved score S_best for 3 consecutive iterations:

$$\forall j \in \{i-2, i-1, i\} : S_j \leq S_{best} \tag{8}$$

- The **tolerance threshold limit** activates when confidence scores remain identical for 3 successive iterations:

$$S_{i-2} = S_{i-1} = S_i \tag{9}$$

Empirical evaluation showed three iterations provided the best trade-off between performance and computational cost. This process results in a refined event timeline E_k accompanied by final feedback F_k, ensuring systematic quality improvement through collaborative agent interaction.

The feedback agent scores extracted events on seven dimensions ($0.00 - 1.00$ scale): *Narrative Relevance* (alignment with case story), *Temporal Accuracy* (correctness of dates), *Chronological Flow* (logical sequencing), *Event Detail* (adequacy of descriptive details), *Repetition* (detection of duplicates), *Character Identification* (clarity of participant roles), and *Confidence Score* (overall extraction quality). A textual critique accompanies scores, highlighting improvement areas. The Confidence Score serves as the primary convergence signal. This systematic thresholding signals when the output quality has maximized its precision and recall, allowing the system to halt confidently rather than continuing costly, minimally productive generations. This structured iterative mechanism ensures that the final extracted timeline, E_k, possesses the highest possible verifiable reliability before being passed to downstream applications. For initial extraction and refinement, we used Zero-shot prompting with Role and Style prompting. For feedback generation, we used Zero-shot prompting with self-criticism (Huang et al. 2023) and linear scale (Liu et al. 2023) techniques. The overall framework architecture and iterative refinement process are illustrated in Figure 2. These prompts are provided in the GitHub repository (Tummepalli 2025).

4.2 Downstream Task Evaluation

To demonstrate the practical utility of structured event timeline representations in legal AI applications, we evaluate its impact on a critical downstream task: judgment summarization. We randomly select 200 sample judgments from the synthetic dataset introduced in Section 3, denoted as $J = (J_1, J_2, ..., J_{200})$, each paired with its corresponding ground-truth event timelines $E = (E_1, E_2, ..., E_{200})$.

We conduct comparative summarization experiments using two input representations.

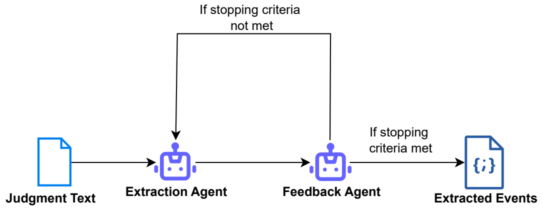

Figure 2: The LexChronos Framework for Iterative Extraction and Feedback-Driven Refinement

- Unstructured approach: Original judgment documents as provided as input requiring the LLM to extract key arguments and outcomes from verbose, noisy text.

$$S_{unstructured} = LLM(J_j) \tag{10}$$

- Structured approach: Generated event timelines serve as input, offering a concise, temporally ordered narrative.

$$S_{structured} = LLM(E_j) \tag{11}$$

Both approaches employ identical open-source language models for summary generation, ensuring fairness and isolating the contribution of structured representations of summarization quality. Summarization prompts use Zero-shot prompting combined with Role and Style prompting. We adopt zero-shot prompting for summary generation to ensure fairness and reproducibility across models, isolating the effect of input representations (structured vs unstructured) from model adaptation. While few-shot or fine-tuned models may improve performance, our objective is to benchmark the contribution of structured representations independently of adaptation effects, making zero-shot prompting the most appropriate setting for controlled evaluation.

For evaluation, we adopt GPT-4 as the judge LLM, and employ a pairwise comparison protocol, where the evaluator receives the original judgment text alongside two candidate summaries and determines which better satisfies legal quality standards. GPT-4 was selected due to its demonstrated strong alignment with human evaluators in summarization assessment, with pairwise comparison identified as the most effective evaluation method (Liu et al. 2024). Given this high correlation, human evaluation was not included, as GPT-4 offers a reliable and cost-effective alternative for large scale experiments.

Our evaluation protocol is guided by a detailed checklist comprising eight equally weighted criteria—*Streamlined Narrative* (clear legal storyline), *Conciseness* (compact yet complete), *Outcome clarity* (explicit final decision), *Focus on key elements* (highlights core issues), *Citation economy* (minimal necessary citations to avoid legal jargon), *Readability for practitioners* (plain professional language), *Balanced detail* (essential context only), *Chronological efficiency* (logical sequencing). These criteria collectively capture the balance between accuracy, readability and practical utility required for legal summaries. The Judge LLM selects the summary that satisfies the majority of these criteria while maintaining fidelity to the original judgment text J. To mitigate bias, the evaluation prompt incorporates zero-shot prompting with self-criticism, requiring GPT-4 to justify its preference with an interpretable rationale. All summarization and evaluation prompts are publicly available in our GitHub repository (Tummepalli 2025).

5 Experiments and Evaluation Results

We evaluate *LexChronos* in three sequential stages: (i) instruct-tuning the extraction agent, (ii) assessing the feedback agent's contribution to extraction quality, and (iii) measuring the impact of structured event timelines on legal judgment summarization. All experiments leverage the synthetic dataset described in Section 3.

5.1 Extraction Agent Performance

To identify the most effective extraction agent, we instruct-tuned eight open-source language models (each under 4 billion parameters) on the training split of our synthetic corpus ($n_{train} = 1600$), reserving $n_{test} = 400$ for evaluation. All models were fine-tuned using LoRA-based instruct-tuning, which updates only a small fraction of parameters, ensuring computational efficiency. Candidate models included: Llama 3.2 3B Instruct (Meta 2024b), Llama 3.2 1B Instruct (Meta 2024a), Gemma 2 2B IT (Google 2024), Gemma 3 1B IT (Team et al. 2025), DeepSeek R1 Distill Qwen 1.5B (Guo et al. 2025), Phi-4-Mini-Reasoning (Xu et al. 2025), Qwen 2.5 3B (Team 2024), and Qwen 3 4B (Yang et al. 2025).

Post tuning, each model was evaluated on the held-out test set using BERT-based Precision, Recall, and F1 (Zhang et al. 2019) against the gold-standard event annotations E_i. The choice of BERT-based metrics is critical in the legal domain: traditional metrics such as ROUGE or BLEU rely on lexical overlap, which often fails to capture semantic nuances in legal language, where subtle word choices carry significant contextual and legal meaning (Kaster, Zhao, and Eger 2021). BERTScore, by leveraging contextual embeddings, measures semantic coherence and equivalence between extracted event and ground truth, offering a stronger correlation with human judgment of textual fidelity. For complex, long-form legal documents, semantic equivalence is a far more reliable indicator of extraction quality than token-level matching, making BERT-based metrics are preferred choice.

Table 4 presents the results, with Llama 3.2 3B Instruct achieving the highest BERT-based F1 score (0.8390) among all candidates. This superior performance underscores that architectural designs optimized for reasoning despite smaller parameter counts significantly enhance structured information extraction. Consequently, Llama 3.2 3B Instruct was selected as the final extraction agent.

5.2 Feedback Agent Contribution

Building on the best-performing extraction agent, we evaluate the role of the feedback agent by integrating multiple feedback models into an iterative refinement loop, with

LLM	***Bert Precision***	***Bert Recall***	***Bert F1***
Llama 3.2 3B Instruct	**0.8450**	**0.8336**	**0.8390**
Gemma 3 1B IT	0.8237	0.7992	0.8113
Gemma 2 2B IT	0.7809	0.7957	0.7880
Llama 3.2 1B Instruct	0.6058	0.6054	0.6052
DeepSeek R1 Distill Qwen 1.5B	0.5548	0.5135	0.5276
Qwen 2.5 3B	0.5628	0.4467	0.4752
Phi 4 Mini Reasoning	0.4802	0.2442	0.3048
Qwen 3 4B	0.4472	0.2506	0.3005

Table 4: Extraction Agent performance on n_{test}

Feedback Configuration	***Bert Precision***	***Bert Recall***	***Bert F1***
Self Feedback	0.8681	0.8371	0.8523
Gemma 2 2B IT	**0.8918**	**0.8590**	**0.8751**
Gemma 3 1B IT	0.8664	0.8348	0.8502
Llama 3.2 3B Instruct	0.8665	0.8351	0.8505
Llama 3.2 1B Instruct	0.8661	0.8344	0.8499

Table 5: Feedback Agent Impact on Extraction Quality

LLM	***$S_{unstructured}$***	***$S_{structured}$***
Llama 3.1 8B Instruct	55	145
Gemma 2 9B IT	**50**	**150**

Table 6: GPT-4 Preference for Structured Input Summaries (n_{sum})

instruct-tuned Llama 3.2 3B Instruct serving as the extraction agent. We compare five configurations: (1) self-feedback (Madaan et al. 2023) using instruct-tuned Llama 3.2 3B Instruct for both extraction and critique; (2)Llama 3.2 3B Instruct; (3) Llama 3.2 1B Instruct; (4) Gemma 2 2B IT; and (5) Gemma 3 1B IT. The latter four feedback agents are pre-trained LLMs without additional tuning. For each configuration, we perform up to 10 refinement iterations per judgment, applying the stopping criteria of patience limit and tolerance threshold as described in Section 4.1. Table 5 reports BERT-based Precision, Recall, and F1 on the same test set (n_{test}).

5.3 Impact on Summarization

Building on our extraction-feedback pipeline, we evaluate the downstream benefits of structured event timelines for legal summarization. From the test set, we randomly sampled 200 judgments ($n_{sum} = 200$) and generated two summaries per judgment using two open-source LLMs: Llama 3.1 8B Instruct (Grattafiori et al. 2024) and Gemma 2 9B IT (Google 2024). For each judgment J_j we produced:

- Unstructured summary: as per equation 10 from section 4.2 with J_j as input
- Structured summary: as per equation 11 from section 4.2 with E_j as input, where E_j denotes the ground-truth event timeline.

To assess summary quality, we employ GPT-4 as an automated evaluator. For each judgment, GPT-4 is given J_j, $S_{unstructured}$, and $S_{structured}$, and asked to select the summary that best captures key information, preserves legal context, and aligns with the original document's intent.

$$Best\ Summary = GPT\ 4(J_j, S_{unstructured}, S_{structured}) \tag{12}$$

As shown in Table 6, structured event timelines markedly improve summarization quality across both LLM architectures, with Gemma 2 9B IT exhibiting the highest preference (75%). This strong preference demonstrates the transferability and utility of structured timelines across different foundational models (Meta vs. Google architectures). Our findings highlight that the extraction process acts as a knowledge preprocessor, creating a high-quality bottleneck. By supplying the summarization models with a clean, temporally ordered legal narrative, we enable them to focus on linguistic generation and high-level reasoning, rather than exhaustive document comprehension and filtering of verbose legal text.

Collectively, the extraction-feedback evaluations validate the effectiveness of our dual-agent framework in producing high-quality event timelines, while the summarization experiments demonstrate the downstream advantages of structured representations in legal AI tasks.

6 Data and Code Availability

The code, prompts, hyper-parameters, instruct-tuned model adapters from section 5.1, and the complete synthetic dataset are available in GitHub (Tummepalli 2025). All resources are publicly accessible.

7 Threats to Validity

While LexChronos demonstrates promising results, certain factors may influence the generalizability of our findings. (1) The synthetic dataset, although designed to emulate the structural and rhetorical characteristics of Supreme Court judgments, is not derived from actual cases. The absence of authentic judicial complexity such as rare procedural anomalies and unpredictable factual patterns may limit its applicability to real-world scenarios. (2) The dataset covers only 25 case categories out of 48 from the Supreme Court's classification framework. While these categories represent high-volume and socially significant domains, excluding others may introduce topical bias and constrain the framework's coverage across the full spectrum of Indian jurisprudence. (3) The current evaluation is restricted to English-language judgments. Given the linguistic diversity of India's judicial ecosystem and the prevalence of vernacular submissions, this limitation may affect the adaptability of the framework in multilingual contexts.

8 Conclusion and Future work

This work introduces and validates *LexChronos*, an agentic AI framework for extracting structured event timelines from Indian Supreme Court judgments. The proposed dual-agent architecture, combining an instruct-tuned extraction agent with a pre-trained feedback agent, offers a systematic approach to iterative output refinement. To address the lack of resources in this domain, we created a synthetic corpus of 2,000 annotated judgments, establishing a foundational benchmark for Indian legal event extraction. The best-performing configuration achieved a BERT-based F1 score of 0.8751, demonstrating high accuracy. Furthermore, downstream summarization experiments confirm the practical utility of structured timelines. Summaries generated from event-based inputs were preferred by GPT-4 in the majority of cases. These findings underscore the importance of moving beyond traditional unstructured approaches and adopting structured representations for legal AI applications.

Looking ahead, several promising research directions emerge. A critical next step is transitioning from synthetic to real-world datasets. While the synthetic corpus enabled initial development and evaluation, a human-annotated Supreme Court dataset would validate the framework's robustness in a practical setting and serve as a valuable resource for the legal AI community. Additionally, expanding *LexChronos* to cover all levels of Indian courts, including High courts and lower tribunals, is essential for achieving generalizability. Addressing linguistic diversity by supporting vernacular languages will further enhance applicability across India's judicial ecosystem.

Beyond data, the framework can be extended to enable advanced downstream tasks. While summarization served as an initial proof of concept, future work will explore applications such as precedent mapping, enabling AI systems to trace the legal history and influence of judgments; argument generation and judgment prediction, which require modelling logical and causal relationships between events; and knowledge base construction, leveraging temporal relationships to transform raw legal text into interconnected networks of legal knowledge. These directions represent a shift from basic information extraction toward task-oriented legal reasoning, positioning *LexChronos* as a foundation for next-generation legal AI systems.

A List of Case Categories

Table 7 lists the curated set of 25 case categories considered for dataset creation, as referenced in Section 3.

Case Categories	
Constitutional Law	Civil Law
Banking and Finance Law	Education Law
Intellectual Property Rights (IPR)	Taxation Law
Labor and Employment Law	Family Law
Consumer Protection Law	Cyber Law
Real Estate and Property Law	Criminal Law
Service Law (Government Employees)	Contract Law
Environmental Law	Corporate Law
Administrative Law	Insurance Law
Health and Medical Law	Maritime Law
Human Rights Law	Election Law
Energy and Mining Law	Telecom Law
Customs and Excise Law	

Table 7: List of Case Categories

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alan, R.; Lombardo, R.; Barbara, P.; et al. 2020. Biomedical event extraction as sequence labeling. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)*, 5357–5367.

Björne, J.; and Salakoski, T. 2018. Biomedical event extraction using convolutional neural networks and dependency parsing. In *Proceedings of the BioNLP 2018 workshop*, 98–108.

Chen, Y.; Zhou, T.; Li, S.; and Zhao, J. 2025. A dataset for document level Chinese financial event extraction. *Scientific data*, 12(1): 772.

Deeks, A.; and Hollis, D. 2025. Large Language Models and International Law. *Chi. J. Int'l L.*, 26: 117.

Filtz, E.; Navas-Loro, M.; Santos, C.; Polleres, A.; and Kirrane, S. 2020. Events Matter: Extraction of Events from Court Decisions. In *Legal Knowledge and Information Systems*, 33–42. IOS Press.

Google. 2024. Gemma.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Han, S.; Hao, X.; and Huang, H. 2018. An event-extraction approach for business analysis from online Chinese news. *Electronic Commerce Research and Applications*, 28: 244–260.

Huang, J.; Gu, S.; Hou, L.; Wu, Y.; Wang, X.; Yu, H.; and Han, J. 2023. Large Language Models Can Self-Improve. In

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 1051–1068.

Huang, K.-H.; and Peng, N. 2021. Document-level Event Extraction with Efficient End-to-end Learning of Cross-event Dependencies. *NAACL HLT 2021*, 36.

Huang, Y.; Hu, N.; Li, K.; Wang, N.; and Lin, Z. 2024. Extracting financial events from raw texts via matrix chunking. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 7035–7044.

Hussain, A. S.; and Thomas, A. 2024. Large Language Models for Judicial Entity Extraction: A Comparative Study. *arXiv preprint arXiv:2407.05786*.

Indian Kanoon. 2025. Indian Kanoon. https://indiankanoon.org. Accessed: 29 September 2025.

Kalamkar, P.; Agarwal, A.; Tiwari, A.; Gupta, S.; Karn, S.; and Raghavan, V. 2022. Named entity recognition in Indian court judgments. *arXiv preprint arXiv:2211.03442*.

Kaster, M.; Zhao, W.; and Eger, S. 2021. Global Explainability of BERT-Based Evaluation Metrics by Disentangling along Linguistic Factors. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 8912–8925.

Liu, Y.; Fabbri, A. R.; Chen, J.; Zhao, Y.; Han, S.; Joty, S.; Liu, P.; Radev, D.; Wu, C.-S.; and Cohan, A. 2024. Benchmarking generation and evaluation capabilities of large language models for instruction controllable summarization. In *Findings of the Association for Computational Linguistics: NAACL 2024*, 4481–4501.

Liu, Y.; Feng, S.; Wang, D.; Zhang, Y.; and Schütze, H. 2023. Evaluate What You Can't Evaluate: Unassessable Quality for Generated Response. *arXiv preprint arXiv:2305.14658*.

Lu, A.; Zhang, H.; Zhang, Y.; Wang, X.; and Yang, D. 2023. Bounding the Capabilities of Large Language Models in Open Text Generation with Prompt Constraints. In *Findings of the Association for Computational Linguistics: EACL 2023*, 1982–2008.

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; et al. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36: 46534–46594.

Meta. 2024a. Llama 3.2 1B Instruct. https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct. Hugging Face Model Hub.

Meta. 2024b. Llama 3.2 3B Instruct. https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct. Hugging Face Model Hub.

Naik, V.; Patel, P.; and Kannan, R. 2023. Legal entity extraction: An experimental study of NER approach for legal documents. *International Journal of Advanced Computer Science and Applications*, 14(3).

Press Information Bureau, Government of India. 2025. Use of AI in Supreme Court Case Management. https://www.pib.gov.in/PressReleasePage.aspx?PRID=2113224. Accessed: 29 September 2025.

Rohit Singh. 2025. SC asks government to integrate AI tools into its case management system. https://www.medianama.com/2025/07/223-sc-ai-tools-case-management-5cr-cases-pile-up-india/. Accessed: 29 September 2025.

Shivananda, S.; Mathews, L. M.; Nikkam, S.; Kambli, P.; Ingale, R. D.; and Koparde, P. 2024. Revolutionizing Legal Case Analysis through Advanced Event Extraction. In *2024 International Conference on Recent Advances in Science and Engineering Technology (ICRASET)*, 1–6. IEEE.

Sierra, G.; et al. 2018. Event Extraction from Legal Documents in Spanish. In *1st Workshop on Language Resources and Technologies for the Legal Knowledge Graph*, 36.

Supreme Court of India. 2025a. Case Category. https://www.sci.gov.in/case-category/. Accessed: 29 September 2025.

Supreme Court of India. 2025b. New Case Categorization Framework. https://cdnbbsr.s3waas.gov.in/s3ec0490f1f4972d133619a60c30f3559e/uploads/2025/05/2025051450.pdf. Accessed: 2025-11-25.

Team, G.; Kamath, A.; Ferret, J.; Pathak, S.; Vieillard, N.; Merhej, R.; Perrin, S.; Matejovicova, T.; Ramé, A.; Rivière, M.; et al. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Team, Q. 2024. Qwen2.5: A Party of Foundation Models.

Tiwari, A.; Kalamkar, P.; Banerjee, A.; Karn, S.; Hemachandran, V.; and Gupta, S. 2024. Aalap: Ai assistant for legal & paralegal functions in india. *arXiv preprint arXiv:2402.01758*.

Tummepalli, A. C. 2025. LexChronos Repository. https://github.com/chandrahas316/LexChronos. Accessed: December 15, 2025.

Wang, N.; Peng, Z.; Que, H.; Liu, J.; Zhou, W.; Wu, Y.; Guo, H.; Gan, R.; Ni, Z.; Yang, J.; et al. 2024. RoleLLM: Benchmarking, Eliciting, and Enhancing Role-Playing Abilities of Large Language Models. In *Findings of the Association for Computational Linguistics ACL 2024*, 14743–14777.

Xian, G.; Du, S.; Tang, X.; Shi, Y.; Jia, B.; Tang, B.; Leng, Z.; and Li, L. 2024. DLEE: a dataset for Chinese document-level legal event extraction. *Neural Computing and Applications*, 36(25): 15581–15597.

Xu, H.; Peng, B.; Awadalla, H.; Chen, D.; Chen, Y.-C.; Gao, M.; Kim, Y. J.; Li, Y.; Ren, L.; Shen, Y.; et al. 2025. Phi-4-mini-reasoning: Exploring the limits of small reasoning language models in math. *arXiv preprint arXiv:2504.21233*.

Xu, N. 2024. Research on Event Entity Extraction in Intelligent Economic Legal System Based on Machine Learning. In *Proceedings of the 2024 International Conference on Image Processing, Intelligent Control and Computer Engineering*, 164–168.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yue, L.; Liu, Q.; Zhao, L.; Wang, L.; Gao, W.; and An, Y. 2024. Event Grounded Criminal Court View Generation with Cooperative (Large) Language Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2221–2230.

Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Zhao, B.; Zhao, Y.; and Mao, Y. 2024. A method for judicial case knowledge graph construction based on event extraction. In *Proceedings of the 2024 9th International Conference on Intelligent Information Technology*, 62–69.