Legal Document Summarization: A Zero-shot Modular Agentic Workflow Approach

Taha Sadikot
National Institute of Technology
Kurukshetra
taha.sadikot.m@gmail.com

Sarika Jain
National Institute of Technology
Kurukshetra
jasarika@nitkkr.ac.in

Abstract

The large volume and inherent complexity of Indian Court judgments, which feature nuanced legal arguments and extensive factual details, have created a need for high-quality automated summarization systems. We develop two zero-shot modular agentic workflow frameworks for Indian Court judgment summarization that do not require model fine-tuning: a three-stage Lexical Modular Summarizer (LexA) designed for lexical overlap metrics and a five-stage Semantic Agentic Summarizer (SemA) designed for semantic similarity. We extract a subset of CivilSum and IN-Abs datasets and call it the Sum-IPL-CivilSum test set. On this test set, LexA achieves ROUGE-1 F1 of 0.6326 and BERTScore F1 of 0.8902, comparable to state-of-the-art fine-tuned transformer models while requiring no training data or GPU resources. On the Sum-IPL-IN-Abs test set, LexA achieves ROUGE-1 F1 of 0.1951 and SemA achieves ROUGE-1 F1 of 0.2014 and BERTScore F1 above 0.81, outperforming zero-shot baselines. Our evaluation suggests that modular, zero-shot agentic approaches can achieve competitive results for legal summarization in resource-limited judicial settings.

1 Introduction

The Indian judicial system generates vast volumes of lengthy judgments, often exceeding 5,000 words, making manual summarization a significant bottleneck for legal professionals (Supreme Court of India, 2024; Malik et al., 2024). Current state-of-the-art solutions rely on fine-tuning transformer models like BART (Lewis et al., 2020) and Legal-BERT (Chalkidis et al., 2020), but these approaches are computationally expensive, require large annotated datasets, and lack the interpretability essential for legal trust. Additionally, their rigidity necessitates costly retraining when legal conventions evolve, limiting their utility for smaller stakeholders.

This work proposes zero-shot agentic workflows as a flexible alternative, guided by four core research questions. We investigate whether these workflows can match the performance of fine-tuned models (RQ1) and outperform direct LLM prompting (RQ2). Furthermore, we analyze whether architectural differences within the workflows produce meaningful performance trade-offs (RQ3) and assess their ability to generalize effectively across diverse legal datasets (RQ4).

Research Objective: To evaluate whether modular zero-shot agentic workflows can achieve competitive performance on automated summarization of Indian Court judgments without fine-tuning.

Key Contributions: We make the following key contributions in this paper:

1. **Zero-shot Modular Agentic Framework:** We develop a modular agentic framework for summarizing Indian Court judgments that operates in a zero-shot setting without supervised model fine-tuning. This work applies agentic workflow architectures to automated summarization of full-length judicial decisions.

2. **Competitive Empirical Performance:** We demonstrate empirically that this framework achieves ROUGE and BERTScore metrics at par with leading fine-tuned transformer baselines (ROUGE-1 F1: 0.6326 (Sum-IPL-CivilSum), BERTScore F1: 0.8902 (Sum-IPL-CivilSum)), achieving ROUGE-1 F1 of 0.6326 compared to 0.374 for Llama 2-chat-70B on the same benchmark. This modular architecture provides stepwise process transparency through explicit workflow decomposition and memory-based state management.

3. **Two Complementary Architectures:** We introduce and evaluate two distinct agentic workflow architectures: Lexical Modular Summarizer (LexA) (3-stage modular) and Semantic Agentic Summarizer (SemA) (5-stage

Proceedings of the 1st Workshop on NLP for Empowering Justice (JUST-NLP 2025), pages 29–40

integrated), demonstrating flexibility in design objective without model retraining.

4. **Comprehensive Evaluation and Analysis:** We evaluate our framework extensively using quantitative metrics, qualitative expert assessments, error analysis, and detailed workflow comparisons on our test set Sum-IPL, which is extracted from standard Indian legal datasets (CivilSum, IN-Abs).

Our proposed framework demonstrates that modular, agentic, and zero-shot approaches can be more practical and accessible for judicial systems facing resource constraints. The remainder of this paper is organized as follows: Section 2 provides background and related work. Section 3 details our methodology and the proposed workflow architectures. Section 4 describes the experimental setup. Section 5 presents comprehensive evaluation results, including quantitative performance, qualitative assessments, and error analysis. Section 6 discusses the implications, and Section 7 concludes with key findings and future directions.

2 Background and Related Work

Legal document summarization approaches have evolved from rule-based methods to modern agentic architectures. There have been recent advances in LLM-based reasoning and multi-agent systems that enable task decomposition. In this section, we review the relevant work in these dimensions.

2.1 Legal Document Summarization

Legal document summarization has evolved from early statistical methods to advanced neural architectures, with contemporary research focusing on fine-tuning transformers like BART and Legal-LED on specialized datasets such as IN-Abs and CivilSum. Despite achieving strong metrics, these models face adoption barriers due to high computational and data requirements. Consequently, the field is increasingly prioritizing broader challenges—including multi-granularity, legal reasoning, and multilingualism—highlighted by benchmarks like LegalBench and LEXTREME. Future directions emphasize developing robust methods capable of handling these complexities, particularly within the multilingual Indian legal context, without relying solely on resource-intensive training processes.

Our work explores whether agentic workflows can achieve comparable performance without these requirements.

2.2 Taxonomy of Text Summarization Approaches

Text summarization techniques can be broadly classified into various types (Jain and Saha, 2025; Smith and Wang, 2024).

2.2.1 Based on Summary Generation Strategy

Extractive Summarization involves selecting important sentences and phrases directly from the text and generating a concise summary (Brown and Taylor, 2022; Smith and Wang, 2024). This technique uses ranking algorithms to rank sentences, such as term frequency-inverse document frequency (TF-IDF), sentence position, and keyword occurrence. Classical examples of this approach include TextRank (Mihalcea and Tarau, 2004), LexRank (Erkan and Radev, 2004), and graph-based algorithms to preserve factual accuracy. However, this approach may result in potentially inconsistent output and may not convey the complete picture of the text.

Abstractive Summarization does not pick text from the main source, but generates new sentences that paraphrase and condense the main concepts of the source text, mimicking human summarization behavior (Gupta and Sharma, 2024). Modern abstractive approaches use encoder-decoder architectures, sequence-to-sequence models with attention mechanisms, and transformer-based models like BART (Lewis et al., 2020), PEGASUS (Zhang et al., 2020), and T5 (Raffel et al., 2020). Although abstractive summaries are more human-readable and consistent, they face challenges that include factual inconsistencies (hallucinations), difficulty maintaining legal precision, and higher computational requirements.

Hybrid Summarization combines the strengths of both extractive and abstractive methods (Patel and Singh, 2024). Typically, salient content from source documents is first extracted and then rewritten or paraphrased to improve consistency and readability (Patel and Singh, 2024). This approach balances the merits and demerits of both approaches.

2.2.2 Based on Implementation Methodology

Traditional Rule-based and Statistical Methods employ hand-crafted features, frequency-based heuristics, and graph algorithms (TextRank,

LexRank) (Mihalcea and Tarau, 2004; Erkan and Radev, 2004) without machine learning. These approaches are highly interpretable but limited in their ability to handle complex linguistic patterns of legal texts.

Classical Machine Learning Methods involve early supervised models such as SVMs and decision trees. They use engineered features to predict which sentences to include in a summary.

Neural/Supervised Learning Methods use labeled datasets of document-summary pairs to train deep networks. They include classic sequence-to-sequence models and modern fine-tuned transformer models (e.g., BART (Lewis et al., 2020), T5 (Raffel et al., 2020), PEGASUS (Zhang et al., 2020), Legal-BERT (Chalkidis et al., 2020)). These also include pretrained and fine-tuned Large Language Models (LLMs), such as BERT and GPT. They achieve strong performance results both in extractive and abstractive summarization, but demand a significant amount of annotated data and computational resources for training (Jain and Saha, 2025; Smith and Wang, 2024).

Zero-shot, Few-shot, and Prompt-Based Methods operate without task-specific training data, relying on pre-trained general-purpose language models and prompt engineering. The emergence of instruction-tuned LLMs (GPT-3.5+, Claude, PaLM, Llama 2) (Chung et al., 2022) has enabled zero-shot application to specialized domains. In legal NLP, early results show promise: InstructGPT achieves 85% accuracy on CUAD clause extraction without fine-tuning (Hendrycks et al., 2021), and GPT-4 reaches 70% on LegalBench tasks (Guha et al., 2023). However, for legal summarization, Llama 2-chat-70B achieves only 0.374 ROUGE-1 on CivilSum (Malik et al., 2024), substantially below fine-tuned BART (0.450) on IN-Abs (Bhattacharya et al., 2019). These approaches offer flexibility and ease of deployment but have traditionally underperformed compared to supervised methods.

Agentic Workflow-based Methods represent a modular paradigm where summarization is divided into specialized subtasks, each managed by an agent (Chen and Zhang, 2024). Such systems are flexible and interpretable, combining elements like planning, tool selection, and iterative decision-making, all without requiring model retraining (Johnson and Lee, 2025; Chen and Zhang, 2024).

2.3 Agentic AI Systems: From Single Models to Multi-Agent Workflows

Recent advances in legal summarization have transitioned from monolithic models to multi-agent workflows that decompose complex tasks, leveraging foundational techniques like ReAct (Yao et al., 2023b) and Chain-of-Thought (CoT) (Wei et al., 2022) to improve performance by interleaving reasoning with actions. While methodologies like Tree of Thoughts (Yao et al., 2023a) or Reflexion (Shinn et al., 2023) explore exhaustive search or self-reflection, we adapt core reasoning strategies for targeted paragraph analysis and event extraction within a structured coordinator-executor architecture built on LangGraph (Team, 2024). Distinct from the complex peer-to-peer communication in CAMEL (Li et al., 2023) or dynamic routing in HuggingGPT (Shen et al., 2023), our approach prioritizes predictable production behavior by implementing specialized agent roles with persistent state management—drawing on the component taxonomy of Wang et al. (Wang et al., 2023) and standardized procedures similar to MetaGPT (Hong et al., 2024)—to ensure the verifiable outputs essential for legal applications.

Agentic AI systems can be classified into four levels based on their decision-making autonomy:

- **Level 1: Autonomous Agents** — Models that produce summaries from raw input entirely independently, with minimal external intervention. These agents are aspirational and currently limited to very controlled environments.

- **Level 2: Router/Coordination Workflows** — Modular systems where a core routing component assigns tasks (like fact extraction, event detection) to specialized agents that can act autonomously within predefined boundaries. This allows for powerful orchestration, easy insertion of new subtasks, and fine-grained error handling. Our workflows, Lexical Modular Summarizer (LexA) and Semantic Agentic Summarizer (SemA), operate at this level.

- **Level 3: Output Fusion Workflows** — Multiple summarization agents (e.g., extractive, abstractive, domain-specific models) generate intermediate outputs, which are then aggregated, ranked, or combined by a merging unit to maximize quality, diversity, or reliability.

- **Level 4: Human-in-the-Loop Workflows** — Systems that embed human expertise at key junctures, enabling legal or domain experts to review, correct, or validate intermediate or final outputs for more accountability, safety, and continuous improvement (Wilson and Kumar, 2024).

While agentic frameworks have been successfully applied to software development, their application to full-length legal document summarization remains unexplored. The legal domain presents unique challenges, including extreme length, complex argumentation, and strict accuracy requirements, making it an ideal testbed for evaluating whether modular, zero-shot workflows can match supervised approaches.

2.4 Positioning Our Approach

Our proposed framework operates as a Level 2 Router Workflow with hybrid extractive-abstractive summary generation characteristics. It is a zero-shot, prompt-based approach using general-purpose LLMs without fine-tuning. The key innovation lies in strategic data processing by modular agent orchestration that achieves competitive performance without the resource overhead of supervised fine-tuning (Johnson and Lee, 2025).

Distinguishing Characteristics:

- **Our Approach vs. Fine-tuned Models** (BART, T5+QLoRA, Legal-LED): We eliminate training overhead (120-200 GPU hours, 7,000+ annotations) (Jain and Kumar, 2024; Sharma and Reddy, 2024) while achieving competitive metrics. Our modular design enables rapid adaptation to new case types through prompt modification rather than retraining.

- **Our Approach vs. Direct LLM Prompting** (GPT-4, Llama 2-chat): We decompose summarization into specialized subtasks where zero-shot prompting excels, rather than expecting single-step generation.

- **Our Approach vs. General Agentic Frameworks** (ReAct, HuggingGPT): We design domain-specific workflows optimized for legal document structure (Yao et al., 2023b; Shen et al., 2023). While ReAct uses dynamic action selection and HuggingGPT employs runtime task planning, our fixed pipelines prioritize transparency and predictability for legal applications.

- **Our Approach vs. Contract Analysis Systems** (CUAD, ContractNLI): We address judgment summarization, which requires synthesizing multi-party arguments, chronological reasoning, and abstractive narrative generation, challenges distinct from contract clause extraction (Hendrycks et al., 2021; Koreeda and Manning, 2021).

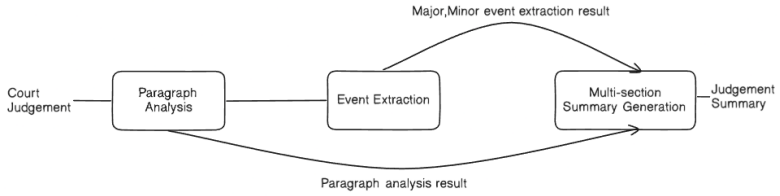
This positioning separates our work from traditional fine-tuned approaches while utilizing the flexibility and interpretability of agentic architectures. To our knowledge, this is the first application of modular, zero-shot agentic workflows to full-length Indian judicial decisions, demonstrating that strategic task decomposition can rival resource-intensive supervised training. By breaking down legal summarization into specialized processing stages, our framework operates at the task-routing level, selecting appropriate processing strategies for different document components while maintaining zero-shot generalization capability.

3 Methodology

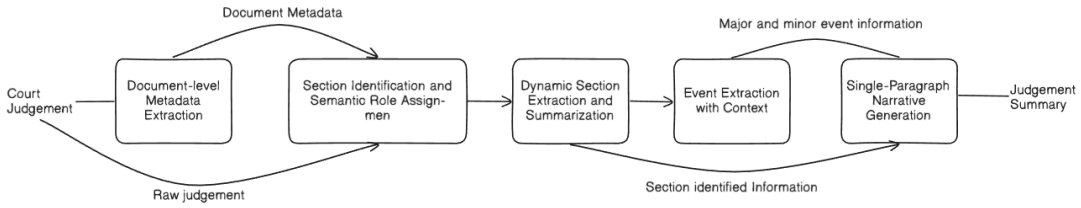
This section details our methodology for legal judgment summarization, describing both proposed modular agentic workflows designed for zero-shot operation.

3.1 Architectural Principles and Implementation Framework

Our architectural design prioritizes reliability, scalability, and maintainability by structuring workflows as a sequence of specialized, independent processing stages that produce intermediate outputs like paragraph classifications and event timelines, thereby creating a durable audit trail for traceability. This modular independence facilitates isolated validation and debugging, while the system's implementation relies on a purely zero-shot methodology orchestrated via the LangGraph framework. Utilizing Google Gemini 2.5 Flash as the primary backend due to its 1M token context window, native JSON support, and cost-effectiveness, the framework avoids task-specific training and remains compatible with other large-context instruction-tuned LLMs such as OpenAI GPT-4 and Claude.



(a) Lexical Modular Summarizer (LexA) (3-stage)



(b) Semantic Agentic Summarizer (SemA) (5-stage)

Figure 1: Architectures of the two proposed agentic summarization workflows.

3.2 The Lexical Modular Summarizer (LexA)

LexA is a three-stage modular architecture (see Figure 1a). This workflow type is categorized as a Level 2 Router Workflow with a hybrid extractive-abstractive strategy, having an extractive bias in early stages. Each processing stage results in an intermediate output with file-based storage at each stage.

1. **Paragraph Analysis:** This stage preprocesses input documents using regex patterns to extract numbered paragraphs using numbered paragraph markers (e.g., "1.", "2.", "3.") commonly found in Indian judgments. For each paragraph, we invoke the LLM with a classification prompt to identify segments as facts, legal issues, arguments, court reasoning, rulings, or procedural history. The process also involves extracting comprehensive metadata, including key topics, a legal significance score, the parties involved, relevant dates, specific legal concepts, and citations. All this information is compiled and exported as structured JSON, featuring detailed paragraph-level annotations. This paragraph-level classification enables downstream stages to focus on legally significant content while preserving exact phrasing for lexical overlap metrics.

2. **Event Extraction:** This stage identifies both major and minor legal events and constructs a chronological timeline to preserve key factual n-grams (dates, procedural terms) that contribute to ROUGE scores while providing narrative structure. Major events include case filings, judgments, appeals, and significant motions; minor procedural events include notices, adjournments, and document submissions. For each paragraph in the JSON file, we invoke the LLM with an event disambiguation and temporal ordering prompt. The final output is a structured event timeline as a JSON file that details the relationships between these events.

3. **Multi-section Summary Generation:** Reference summaries of our benchmark datasets are multi-section. We chose this format to maximize ROUGE overlap. We select high-significance paragraphs and all major events to invoke the LLM with a synthesis prompt, emphasizing n-gram preservation. Explicit instruction to preserve terminology maintains lexical fidelity. This process generates distinct sections, including an executive summary, factual background, legal issues, an event timeline, court reasoning, and the decision. The target length for the summary is determined by matching the 25–30% compression ratio.

3.3 The Semantic Agentic Summarizer (SemA)

SemA is a five-stage integrated architecture (see Figure 1b) designed for BERTScore, which emphasizes semantic similarity and deep legal understanding. This workflow operates as a Level 2 Router Workflow with a hybrid extractive-abstractive strat-

egy, having an abstractive bias in later stages.

1. **Document-level Metadata Extraction:** This stage extracts key case metadata, including case number, court, date, parties involved, and judges from the raw documents. All this information is compiled and exported as structured JSON. Document-level understanding provides context for semantic processing in subsequent stages. It then evaluates the document's overall quality and completeness. Finally, it analyzes the structure of the citation network, identifying connections to statutes, precedents, and regulations.

2. **Section Identification and Semantic Role Assignment:** This stage defines the structural components of the documents, such as sections and subsections (facts, legal issues, arguments, reasoning, and conclusion), using structural markers and semantic analysis, while carefully maintaining the original legal terminology and phrasing. It also identifies clear section boundaries and assigns specific semantic roles to the different content blocks. All this information is compiled and exported as structured JSON. Semantic role identification enables abstractive generation that preserves legal reasoning structure rather than surface form.

3. **Dynamic Section Extraction and Summarization:** This stage extracts paragraphs for each identified section using granular classification and then groups them into sections based on their estimated types. It then generates section-level summaries using abstractive prompting and builds a hierarchical document structure. Section-level abstraction enables semantic compression while maintaining argumentative coherence.

4. **Event Extraction with Context:** This stage identifies a comprehensive event timeline and scores the legal significance of each event on a 1–10 scale based upon the document structure and context (which section, which argument does each event support?). It also analyzes event relationships (causal chains) and integrates these events with the structural layout of the argument. This context-aware event extraction supports semantic coherence in the final narrative by linking events to their legal significance.

5. **Single-Paragraph Narrative Generation:** This stage takes as input all the intermediate outputs stored in the workflow memory (analysis, sections, structure, and events) and invokes the LLM with a semantic synthesis prompt. This generates a single paragraph summary as a coherent narrative, targeting a length of 150–300 words. The single-paragraph format encourages semantic synthesis and conceptual abstraction. The focus on "reasoning chains" and "legal narrative" aligns with BERTScore's semantic similarity measurement.

4 Experimental Setup

To ensure a systematic and reproducible evaluation, we followed a structured experimental procedure encompassing both automated and expert-based assessments. Each test document goes through standardized preprocessing, followed by sequential execution of our proposed workflows. Intermediate results were stored for diagnostic analysis, enabling detailed error tracking and interpretability checks.

4.1 Baseline Models and State-of-the-Art Approaches

To assess the effectiveness of our proposed agentic workflows, LexA and SemA, we systematically compare them against extractive baselines, fine-tuned large transformers, and zero-shot or few-shot LLM baselines. We evaluated strong open-source and commercial models, including Llama 2-chat (70B, 7B), Gemini 2.5 Flash, and GPT-4, using standardized prompts; specifically, Gemini 2.5 Flash and GPT-4 served as zero-shot baselines to establish the direct performance limit of raw LLM API usage.

4.2 Datasets and Evaluation Metrics

4.2.1 Benchmark Datasets

Our workflows are evaluated on the two standard Indian legal datasets which are **CivilSum** and **IN-Abs**

Test Set Composition: We created a test set of a total of 50 Indian Court judgments, including 25 randomly selected documents from each dataset (CivilSum, IN-Abs), which we call Sum-IPL. It

contains two parts: Sum-IPL-CivilSum and Sum-IPL-IN-Abs. We evaluate our workflows on Sum-IPL.

4.2.2 Quantitative Evaluation

For our quantitative evaluation, we employed a suite of automatic metrics to ensure an objective comparison. Specifically, we utilized ROUGE-1, ROUGE-2, and ROUGE-L (via py-rouge v1.1) alongside BERTScore (via bert-score v0.3.11) to assess text quality. Additionally, we tracked practical efficiency and accuracy metrics, including API cost per document, average processing time in seconds, and event coverage to measure the capture of key reference events.

4.2.3 Qualitative Evaluation

Three final-year LLB stusxdents from Kurukshetra University, India, were employed as experts to assess the human-aligned quality of the summary, beyond automatic metrics. Each of these three evaluators independently rated 10 randomly selected outputs (five each from the Sum-IPL-CivilSum and Sum-IPL-IN-Abs test sets) on six key criteria: factual accuracy, legal precision, completeness, coherence, conciseness, and overall quality, using a 1–10 scale. The document selection process ensured that each law student evaluated unique documents, resulting in a total of 30 documents evaluated from Sum-IPL during human evaluation.

The criteria assessed are **Factual Accuracy:** Correctness of facts, dates, parties, and events, **Legal Precision:** Appropriate use of legal terminology, concepts, and reasoning, **Completeness:** Coverage of essential case elements and event coverage, **Coherence:** Logical flow and readability, **Conciseness:** Appropriate length without redundancy, **Overall Quality:** Holistic assessment of summary quality

5 Results

5.1 Quantitative Performance

We attribute Llama 2-chat-70B's lower performance (0.374 ROUGE-1) to its few-shot prompting approach without task decomposition, as reported in prior work (Malik et al., 2024). Our modular workflows demonstrate that strategic task decomposition can substantially improve zero-shot performance over direct LLM prompting.

5.2 Qualitative Expert Evaluation

Table 3 presents the evaluation results, where a Fleiss' kappa of 0.68 indicates substantial inter-annotator agreement on the defined criteria. While both LexA and SemA achieve strong, comparable quality scores (7.5–8.5 range), they remain statistically distinguishable from human-written summaries (9.0 range). Significant gaps persist in Completeness, Legal Precision, and Overall Quality, suggesting that while the automated workflows offer practical utility, they still struggle to match the nuance and comprehensive coverage provided by human experts.

Qualitatively, experts praised both workflows for their accurate chronological structure and use of legal terminology, though they noted specific areas for improvement such as missed legal subtleties, incomplete citations, and occasional verbosity in LexA. The automated systems sometimes oversimplify complex arguments or commit minor factual errors. Overall, the workflows present distinct advantages: LexA excels in structured access and detail preservation, whereas SemA provides superior conciseness, narrative coherence, and conceptual clarity.

5.3 Error Analysis

Six types of failure patterns were identified in 22 out of 50 cases (25 from the CivilSum test set, 25 from the IN-Abs test set).

Complex Multi-Party Cases (6/50 cases): A specific challenge arises in cases involving more than 5 parties. The primary impact of this complexity is potential confusion in party roles and the risk of missing secondary legal issues. For example, in a corporate dispute with 7 parties, SemA misattributed an argument to the wrong party; meanwhile, LexA addresses this with enhanced party tracking, which includes explicit party-argument mapping during the paragraph analysis phase.

6 Discussion

The results demonstrate that modular, zero-shot agentic workflows can achieve competitive performance on legal document summarization without fine-tuning. Our findings offer significant implications for **resource efficiency**, as eliminating the need for GPU-intensive fine-tuning makes advanced legal AI accessible to resource-limited systems. The **modular design** ensures rapid adaptability across domains through prompt modification

Table 1: Comprehensive Performance Comparison Across Multiple Indian Legal Datasets. ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-L (R-L), and BERTScore (BS) F1 scores are reported. All metrics are F1 scores.

Model/Approach	Dataset	R-1	R-2	R-L	BS	Fine-tune
Our Approaches - Zero-shot Agentic Workflows						
Lexical Modular Summarizer (LexA)	Sum-IPL-CivilSum	**0.6326**	0.4563	0.4508	**0.8902**	No
Semantic Agentic Summarizer (SemA)	Sum-IPL-CivilSum	0.4119	0.1253	0.2060	0.8474	No
Lexical Modular Summarizer (LexA)	Sum-IPL-IN-Abs	0.1951	0.0976	0.0928	**0.8299**	No
Semantic Agentic Summarizer (SemA)	Sum-IPL-IN-Abs	0.2014	0.0774	0.1104	**0.8122**	No
Zero-shot and Few-shot LLM Baselines						
Llama 2-chat-70B (Malik et al., 2024)	CivilSum	**0.374**	0.126	0.257	0.851	No
Llama 2-chat-7B (Malik et al., 2024)	CivilSum	0.371	0.126	0.254	0.851	No
Gemini 2.5 Flash (simple prompt)	CivilSum	0.389	0.132	0.184	0.782	No
GPT-4 (simple prompt)	CivilSum	0.412	0.148	0.195	0.804	No
Extractive Baselines						
Oracle Paragraph Extraction (Malik et al., 2024)	CivilSum	**0.331**	0.101	0.220	0.840	No
Random Extraction	CivilSum	0.198	0.042	0.165	0.712	No
Fine-tuned Transformer Models (For Reference)						
BART fine-tuned (Bhattacharya et al., 2019)	IN-Abs	0.450	0.180	0.230	0.820	Yes
T5 + QLoRA (Kumar and Sharma, 2022)	ILC	0.464	—	—	—	Yes
Legal-LED (Kapoor et al., 2024)	IL-TUR	—	—	0.330	0.860	Yes

Table 2: Comparison of Proposed Workflows on Various Aspects

Category	Aspect	LexA	SemA
Architectural Parameters	Workflow Type	Level 2 Router	Level 2 Router
	Summarization Strategy	Hybrid (Extractive-biased)	Hybrid (Abstractive-biased)
	Processing Stages	3-Stage Modular	5-Stage Integrated
	State Management	File-based	Memory-based
	Optimization Target	ROUGE metrics	BERTScore (semantic)
Performance Metrics (CivilSum)	ROUGE-1	**0.6326**	0.4119
	ROUGE-2	**0.4563**	0.1253
	ROUGE-L	**0.4508**	0.2060
	BERTScore	**0.8902**	0.8474
Performance Metrics (IN-Abs)	ROUGE-1	0.1951	0.2014
	ROUGE-2	0.0976	0.0774
	ROUGE-L	0.0928	0.1104
	BERTScore	**0.8299**	0.8122
Operational Characteristics	Processing Speed	Fast (45–60s)	Moderate (60–80s)
	Production Readiness	High	Medium (Research)
	Maintainability	High (Modular)	Medium (Integrated)
	Error Isolation	Excellent	Good
	Interpretability	Very High	High
Best Use Cases	Use Case	Production, high-volume; ROUGE	Research, semantic quality
Processing Efficiency	Avg. Processing Time	52 sec	71 sec
	Median Processing Time	48 sec	65 sec
	Range	28–145 sec	42–198 sec
	API Cost per Document	$0.03	$0.04
	GPU Hours Required	0	0
	Training Data Required	0	0

Table 3: Qualitative Evaluation (Scale: 1–10). Scores represent the average across evaluators. Inter-annotator agreement (Fleiss' kappa): 0.68 (substantial agreement).

Criterion	LexA	SemA	Human
Factual Accuracy	8.5	8.2	9.1
Legal Precision	8.3	7.8	9.3
Completeness	8.1	7.5	8.9
Coherence	7.9	8.1	9.0
Conciseness	7.6	8.4	8.7
Overall Quality	8.1	8.0	9.0

rather than retraining, while the explicit workflow guarantees **transparency** and provides a critical audit trail. Furthermore, the LexA and SemA models demonstrate that different optimization objectives can be achieved purely through architectural variation.

7 Conclusion

This paper demonstrates that modular, zero-shot agentic workflows can achieve competitive performance on legal document summarization without resource-intensive fine-tuning. We introduced two complementary architectures, Lexical Modular Summarizer (LexA) and Semantic Agentic Summarizer (SemA), that decompose summarization into specialized subtasks orchestrated through LangGraph. On the Sum-IPL-CivilSum test set, LexA achieves ROUGE-1 F1 of 0.6326 and BERTScore F1 of 0.8902, comparable to fine-tuned transformer models. Our findings suggest that strategic task decomposition and modular design can rival resource-intensive supervised training.

Future work should extend this approach to other languages, larger document collections, and explore integration with human-in-the-loop feedback mechanisms for continuous improvement. We also plan to investigate more sophisticated architectural patterns for handling complex multi-party cases and subtle legal reasoning.

Limitations

This work has several important limitations that should be considered:

1. **Language Scope:** Our evaluation focuses exclusively on English-language Indian Court judgments. The generalization to other languages or legal systems with different structural conventions remains untested.
2. **Dataset Scale:** Our evaluation uses a test set of only 50 documents. While this allows for detailed qualitative analysis, larger-scale evaluation would strengthen claims of generalizability.
3. **LLM Dependency:** Both workflows depend on the availability and performance of commercial LLM APIs. Changes in model capabilities, availability, or pricing could affect reproducibility and deployment feasibility.
4. **Expert Evaluation Scope:** Qualitative evaluation was conducted by three LLB students, not practicing lawyers. While their domain knowledge is sufficient for this assessment, evaluation by experienced legal practitioners would provide additional validation.
5. **Comparison Limitations:** Direct comparison with fine-tuned models (BART, T5+QLoRA) was not possible due to different evaluation datasets. Comparisons are primarily with zero-shot baselines and our own architectures.
6. **Complex Legal Reasoning:** The workflows struggle with highly complex multi-party cases and subtle legal distinctions, as revealed by error analysis and qualitative evaluation.

Ethics Statement

This work focuses on automating legal document summarization, which has important ethical implications. While automation can improve access to legal information for underserved populations, it also introduces risks of bias, hallucination, and misrepresentation of legal arguments. Our framework maintains interpretability through explicit workflow decomposition, enabling human review and oversight. We strongly recommend that any deployment of this system in real legal settings includes human-expert validation, particularly for cases with high legal stakes. Additionally, the use of LLMs in legal contexts raises privacy concerns regarding document handling by external API providers. Organizations deploying this system should implement appropriate data governance and privacy measures.

Acknowledgements

We thank the law students who participated in the qualitative evaluation and provided valuable feed-

back on summary quality. We also acknowledge the creators of the CivilSum and IN-Abs datasets, which formed the basis for our evaluation. This research was supported by National Institute of Technology Kurukshetra.

References

Paheli Bhattacharya, Kripabandhu Ghosh, Saptarshi Ghosh, Arindam Pal, et al. 2019. Legal case document summarization: Extractive and abstractive methods and their evaluation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*.

David Brown and Susan Taylor. 2022. Extractive methods for legal document summarization: A review. *Legal AI Quarterly*, 12(4):567–589.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The mismatch between domain-specific pretraining and downstream fine-tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904.

Michael Chen and Sarah Zhang. 2024. What are agentic workflows? patterns, use cases, and implementations. *AI Systems Journal*, 8(2):89–112.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Xue, Andrew M Huang, Dmitry Lepikhin, Yuanzhong Xu, Andrew M Dai, Zhifeng Chen, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Neel Guha, Daniel E Ho, Julian Nyrup, Elizabeth Alexander, Andrew D Macey, and Divya Tenney. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Priya Gupta and Rakesh Sharma. 2024. Abstractive text summarization: State of the art, challenges and improvements. *Computer Science Review*, 42:100456.

Dan Hendrycks, Collin Burns, Spencer Chen, Spencer Ball, Frank Basart, Mason DeLucia, Jacob Li, and Radu Soricut. 2021. Cuad: An expert-annotated nlp dataset for legal contract review. In *NeurIPS Datasets and Benchmarks Track*.

Sirui Hong, Mingchen Zheng, Chen Jonathan, Alaa Faqih, Ron Arkin, and Carlos A Gomez-Uribe. 2024. Metagpt: Meta programming for multi-agent collaborative framework. In *Proceedings of the 12th International Conference on Learning Representations*.

Anshika Jain and Sriparna Saha. 2025. A comprehensive survey on legal summarization. *arXiv preprint arXiv:2501.17830*.

Deepak Jain and Sandeep Kumar. 2024. Fine-tuning bart for legal document summarization. In *Workshop on Legal Text Analytics*, pages 45–56.

Robert Johnson and Emily Lee. 2025. What is agentic ai? definition and differentiators in 2025. *Artificial Intelligence Review*, 45:1234–1256.

Abhinav Kapoor, Mohit Goyal, Anand Kumar, et al. 2024. Il-tur: Benchmark for indian legal text understanding and reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 1234–1245.

Yuta Koreeda and Christopher D Manning. 2021. Contractnli: A dataset for document-level natural language inference for contracts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5806–5825.

Anand Kumar and Saurabh Sharma. 2022. Indian legal corpus (ilc): A dataset for abstractive summarization. *arXiv preprint arXiv:2210.10398*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Guohao Li, Hasan Abed Al Kader Hammer, Wenhao Yong, Hao Zhong, and Roberto Togneri. 2023. Camel: Communicative agents for 'mind' exploration of large scale language model society. In *Advances in Neural Information Processing Systems*.

Manuj Malik, Rohan Bhambhoria, Adam Roegiest, and Suzan Verberne. 2024. Civilsum: A dataset for abstractive summarization of indian court decisions. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2457–2467.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Neha Patel and Rajesh Singh. 2024. Fusion of extractive and abstractive text summarization techniques for legal documents. In *International Conference on Natural Language Processing*, pages 234–245.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Priya Sharma and Karthik Reddy. 2024. T5 with qlora for efficient legal summarization. *Indian Journal of AI and Law*, 6(1):78–92.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. In *Advances in Neural Information Processing Systems*.

Noah Shinn, Alfredo Cassirer, Ashwin Bhandwaldar, Natasha Jaques, Justin Tan, Karthik Jakkam, and Carolina Parada. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*.

John Smith and Li Wang. 2024. Extractive and abstractive summarization techniques: A comprehensive overview. *Natural Language Processing Review*, 15(3):145–178.

Supreme Court of India. 2024. Structure and organization of indian judiciary. Official website.

LangChain Team. 2024. Langgraph: Framework for building stateful multi-agent applications. GitHub repository.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Jingning Yang, Jiakai Zhang, Zhuosheng Chen, Jiawei Xie, Yaliang Huang, Dawei Song, et al. 2023. A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Ichien, Ed H Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, pages 24824–24837.

Amanda Wilson and Vijay Kumar. 2024. Addressing ai bias and fairness: Challenges, implications and strategies for ethical ai. *Ethics in AI*, 18(2):234–267.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuandong Cao, and Ankur P Parikh. 2023a. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Ankur P Parikh, and Hongkuk Jiang. 2023b. React: Synergizing reasoning and acting in language models. In *Proceedings of the 11th International Conference on Learning Representations*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

A Appendix: Detailed Prompts and Implementation Details

A.1 Paragraph Analysis Prompt

For each paragraph in the input document, the following prompt is used to classify the paragraph type and extract metadata:

"Classify the following legal paragraph and extract metadata. Paragraph types: Facts, Legal Issues, Arguments, Court Reasoning, Rulings, Procedural History. Extract: key topics, legal significance (1-10), parties, dates, legal concepts, citations."

A.2 Event Extraction Prompt

For event extraction and temporal ordering:

"Extract major and minor legal events from this paragraph. Major events: filings, judgments, appeals, motions. Minor events: notices, adjournments, submissions. Create a chronological timeline with event relationships."

A.3 Final Summary Prompt

For multi-section summary generation:

"Generate a legal summary with sections: Executive Summary, Factual Background, Legal Issues, Event Timeline, Court Reasoning, Decision. Preserve n-grams from source. Target 25-30% compression ratio."

A.4 Semantic Role Assignment Prompt

For SemA semantic role assignment:

"Analyze document structure and assign semantic roles to sections: Facts, Legal Issues, Arguments, Reasoning, Conclusion. Maintain legal terminology. Identify section boundaries."

A.5 Data Format Examples

Example output of LexA Stage 1 (paragraph_analysis.json):

```
{
  "paragraphs": [
    {
      "id": 1,
      "type": "Procedural History",
      "content": "The case was filed...",
```

```
      "metadata": {
        "topics": ["case filing", "
            jurisdiction"],
        "significance": 8,
        "parties": ["Appellant", "
            Respondent"],
        "dates": ["2020-01-15"],
        "concepts": ["writ petition"]
      }
    }
  ]
}
```


Example output of LexA Stage 2 (event_timeline.json):


```
{
  "timeline": [
    {
      "date": "2020-01-15",
      "event": "Case filed",
      "type": "major",
      "significance": 9
    }
  ],
  "relationships": [
    {
      "event1": "Case filed",
      "event2": "Hearing scheduled",
      "type": "follows"
    }
  ]
}
```


Example output of SemA Stage 1 (metadata.json):


```
{
  "case_number": "2020/SC/12345",
  "court": "Supreme Court of India",
  "date": "2023-06-15",
  "parties": ["Appellant", "Respondent
      "],
  "judges": ["Justice A", "Justice B"]
}
```


Example output of SemA Stage 2 (sections.json):


```
{
  "sections": [
    {
      "name": "Facts",
      "role": "factual_context",
      "paragraphs": [1, 2, 3],
      "boundary_markers": ["Para 1", "
          Para 5"]
    }
  ]
}
```