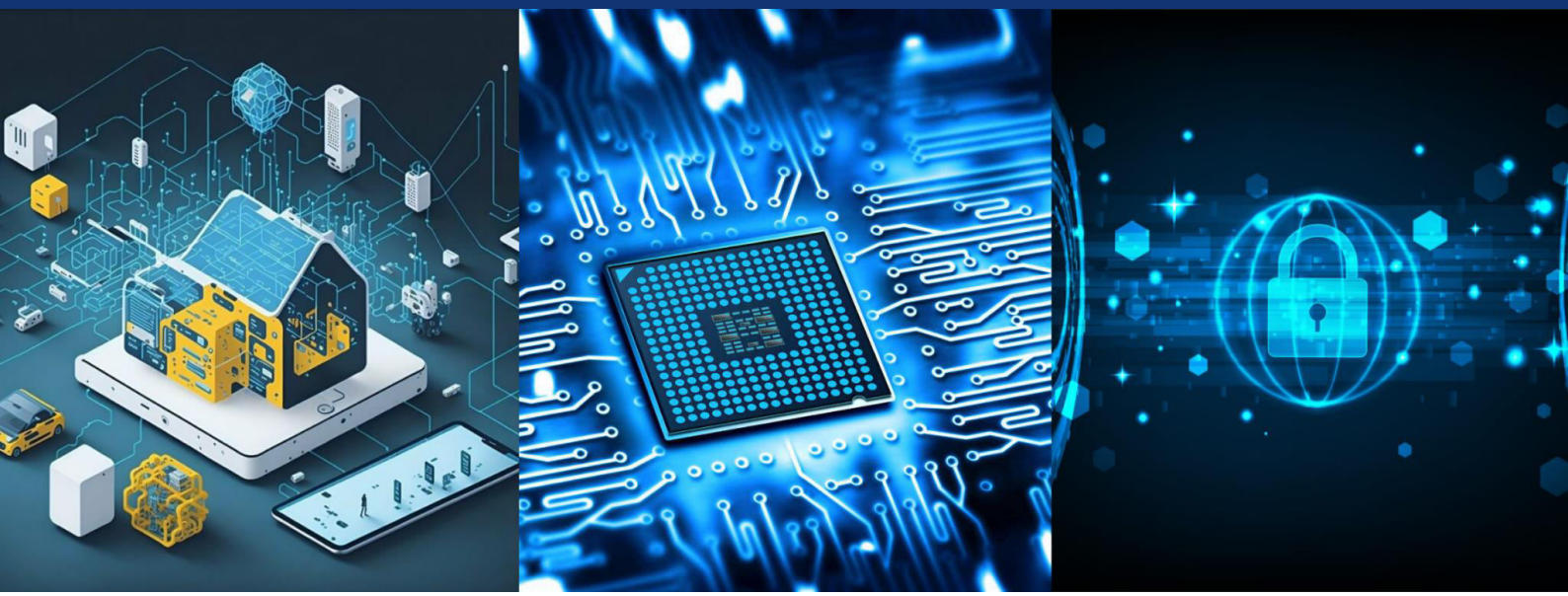
IJIRCCE

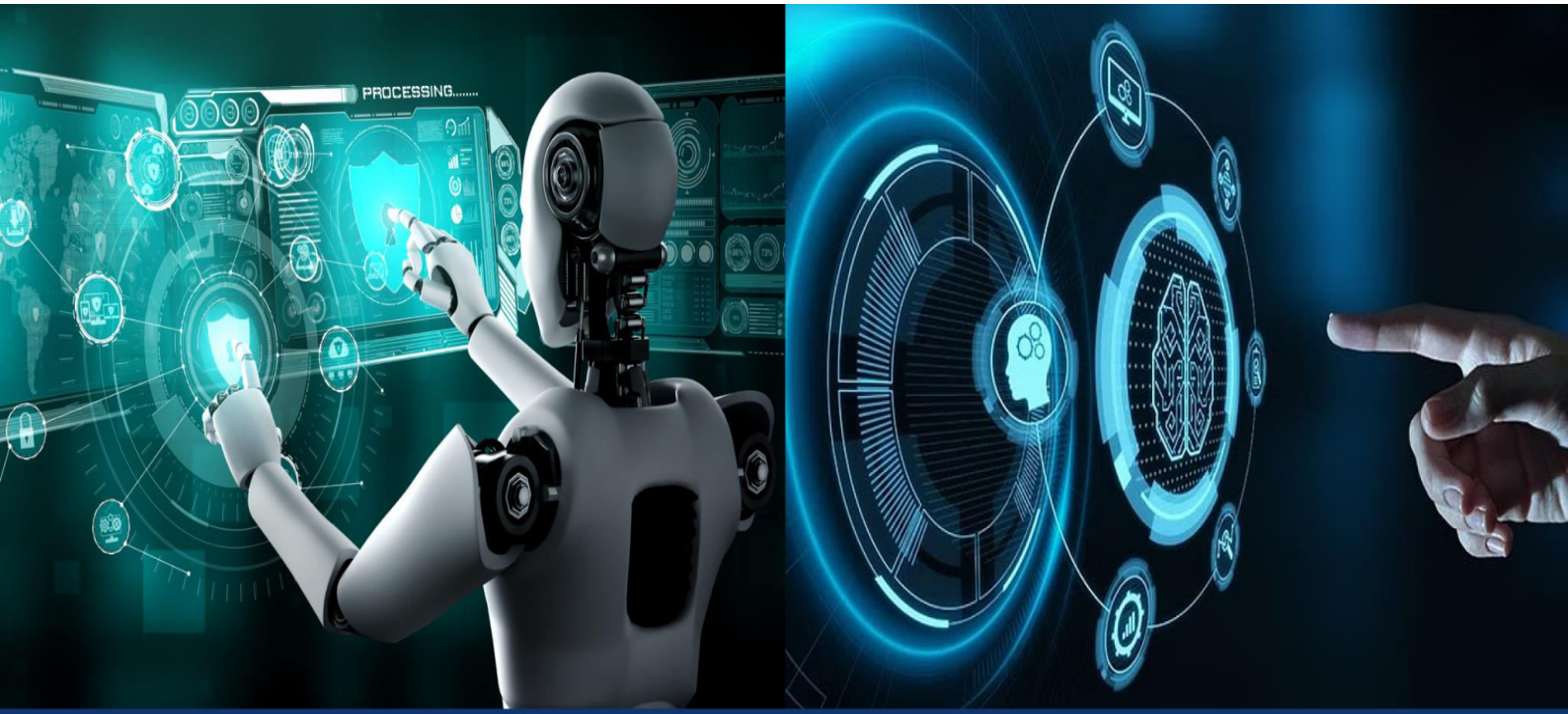
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.927

Volume 14, Issue 8, August 2026

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408014

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Justice: A Hybrid Retrieval-Augmented Generation and Knowledge Graph Framework for Indian Legal Query Resolution

Rohan Salunkhe, Rajwardhan Jagtap, Vaishnav Chougule, Kiran Gore

Undergraduate Student, Dept. of Information Technology, G.H. Raisoni College of Engineering and Management, Pune, India

Undergraduate Student, Dept. of Information Technology, G.H. Raisoni College of Engineering and Management, Pune, India

Undergraduate Student, Dept. of Information Technology, G.H. Raisoni College of Engineering and Management, Pune, India

Associate Professor, Dept. of Information Technology G.H. Raisoni College of Engineering and Management, Pune, India

ABSTRACT: Access to legal information in India has been challenging because of the sheer volume of judgments issued by the Supreme Court of more than 70 years, the difficulty in comprehending legal language, and multilinguism. In our work, we propose Justice AI, a Legal Information Intelligent Assistant powered by a new architecture that blends retrieval-augmentation-generation and Neo4j Knowledge Graph. It analyzes 26,688 Supreme Court judgments from 1950 to 2025, employs a two-tier process to extract structured data using deterministic regex heuristics and Groq-optimized LLaMA-3.3-70B model, and indexes data across six vector databases of Pinecone and one graph database of Neo4j AuraDB. The query is processed through 9 paths of parallel retrieval encompassing semantic vector similarity searches and Cypher graph traversal. The hybrid approach delivers 82% precision for structured legal queries, hallucination rate reduction from 18% to 4%, and multilingual responses within 3.2 seconds in English, Hindi, and Marathi languages. Comparative studies with Indian Kanoon, JUDIS, ungrounded GPT-4, LEGAL-BERT, and InLegalBERT demonstrate clear advantages in every metric studied.

KEYWORDS: Retrieval-Augmented Generation, Knowledge Graph, Legal AI, Natural Language Processing, Indian Supreme Court, Neo4j, Large Language Models, LLaMA, Pinecone, Multilingual NLP

I. INTRODUCTION

The Indian judiciary produces one of the highest volumes of case laws in the world. In its almost seven decades of existence since 1950, the Supreme Court of India has produced more than 26,000 case laws on all matters ranging from constitutional matters and criminal procedure to corporate law and newer domains such as cyber law [20]. While these documents are freely available on public forums such as JUDIS and Indian Kanoon, they need a deep legal background to comprehend, rendering them unavailable to the 1.4 billion people of India.

Current Legal Information Retrieval (LIR) systems either use boolean keyword searching techniques or governmental FAQ forums and do not take into consideration the relational and contextual nature of legal reasoning [2]. A typical person who wants to know whether he is entitled to seek anticipatory bail in Section 438 of the Code of Criminal Procedure requires information in the context of all these related precedents and statutes and the rationale behind the judgement rather than a mere ranking of the retrieved documents. In addition, it should also be noted that many citizens approach court proceedings in languages other than English, such as Hindi and Marathi.

Recent advances in Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) [1] enable significant technological advancements. The predictions of an LLM in RAG are grounded by relevant documents, thus minimizing

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

hallucinations. Nonetheless, even when used in a legal corpus, RAG cannot answer relational questions such as "What cases have cited Shreya Singhal v. Union of India?" or "How many Acts have mentioned Section 302 IPC?" Such queries necessitate traversing a graph rather than retrieving semantically similar documents. On the other hand, while Knowledge Graphs precisely model entity relations, they lack natural language proficiency to communicate with laypersons [7]. Therefore, neither approach alone suffices.

This paper introduces Justice AI, which harmoniously blends both approaches into one framework. We make the following contributions:

- Batch ingestion pipeline for processing 26,688 PDF files from the Supreme Court using PyMuPDF, regex pattern matching, and LLaMA-3.3-70B in order to generate seven metadata fields per file.
- Knowledge graph with four types of nodes and four directed relationship types, allowing multi-hop legal inference across precedent chains, sections, and Acts hierarchy.
- Multilingual semantic index with six partitions on Pinecone, using multilingual-e5-large embedding model on the entire judgement collection.
- Query classifier and parallel retrieval pipeline on n8n, with up to nine retrieval requests executed in parallel, returning results in under 4 seconds end-to-end.
- Response generation in three languages – English, Hindi, and Marathi – by script detection and prompt conditioning based on language.
- Quantitative evaluation showing an average precision of 82% for structured queries, and hallucination rate of 4%.

II. RELATED WORK

A. Legal Information Retrieval

The early LIR systems relied on Boolean keyword search with inverted indexes [2]. Commercial products such as West-law and LexisNexis provide structured search using controlled vocabularies but are too costly and only available in English, and hence cannot be used in India. JUDIS allows users to search for PDF documents in their full-text form without employing any natural language processing or relational reasoning [20]. The Indian Kanoon is able to conduct efficient keyword searches, but does not use any semantic understanding or graph-based query execution. TF-IDF and BM25-based approaches [13] perform well for lexical retrieval but fail when queries require legal inference or Act-level aggregation.

B. Neural Approaches to Legal NLP

The LEGAL-BERT [3] model shows significant improvements in contract classification, statute understanding, and judgment prediction when fine-tuned using EU legal text. The InLegalBERT [4] system adopts the approach for Indian legal texts and achieves improved classification scores using judgments from Indian courts. Both approaches are based on discriminative models where the output is a score rather than an explanation, limiting usability.

C. Retrieval-Augmented Generation

RAG [1] combines dense retrieval with sequence-to-sequence generation using a model fine-tuned for both components, achieving better results than closed-book LLMs for knowledge-intensive tasks. Izacard and Grave [5] further developed Fusion-in-Decoder, where multiple retrieved passages are used as part of the decoder's attention mechanism. RE-PLUG [15] considers the retriever as a plug-in module and increases performance of LM without any fine-tuning. In the context of legal text processing, [6] found that retrieval-based approaches perform much better than ungrounded baselines on citation prediction.

D. Knowledge Graphs for Legal Reasoning

The works of Koniaris et al. [7] demonstrated the presence of clustering patterns in citation networks constructed based on the legal documents of the European Union. Additionally, the study by Lam et al. [8] demonstrated that the use of KGs can improve multi-hop judgment prediction through modeling relations between statutes that cannot be captured using only attention mechanisms for transformer models. Furthermore, De Luca et al. [17] created ontological KGs from Italian legislation to provide fine-grained navigational capabilities over statutes.

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408014

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

E. Large Language Models for Legal Tasks

In particular, Blair et al. [18] showed that GPT-4 can pass bar exams with near-perfect grades. Nay [19] confirmed the effectiveness of LLMs in reasoning about US tax laws. In the case of India, however, generic LLMs are not trained enough, and therefore, retrieval grounding is important. Justice AI uses Groq's inference API for running LLaMA-3.

F. Multilingual Legal AI

In their work, Niklaus et al. [9] found that multilingual models were better than monolingual models for predicting judgments of French, German, and Italian cases at the Swiss Federal Supreme Court. Another work by Wang et al. [12] created multilingual-e5-large, which gives the best results for cross-lingual retrieval tasks for 94 languages, such as Hindi and Marathi. Regarding the Indian case, there has been little research in multilingual legal AI, which is addressed by Justice AI directly.

III. METHODOLOGY AND SYSTEM ARCHITECTURE

A. System Overview

The Justice AI pipeline comprises of 5 steps that include Data Ingestion & Extraction, Knowledge Indexing, Query Understanding, Hybrid Retrieval, and Response Generation. All the processes involved in this pipeline are executed in an n8n workflow that uses 18 nodes. Figure 1 shows the overall pipeline design.

B. Neo4j Knowledge Graph Schema

Figure 2 shows the four node types and four directed relationship types that form the Justice AI knowledge graph, designed to support multi-hop traversals for Act-level, Section-level, and precedent citation queries.

C. Data Ingestion Pipeline

In total, there are 26,688 judgment PDFs for cases from the Indian Supreme Court from 1950 till 2025 included in the dataset [10]. The maximum size of data being processed is limited to 3,000 characters to extract case header information, bench details, mentioned Acts and the part where operative verdict was given.

Two Layered Metadata Extraction Approach. The following algorithm employs two-steps to find a tradeoff between speed and reliability.

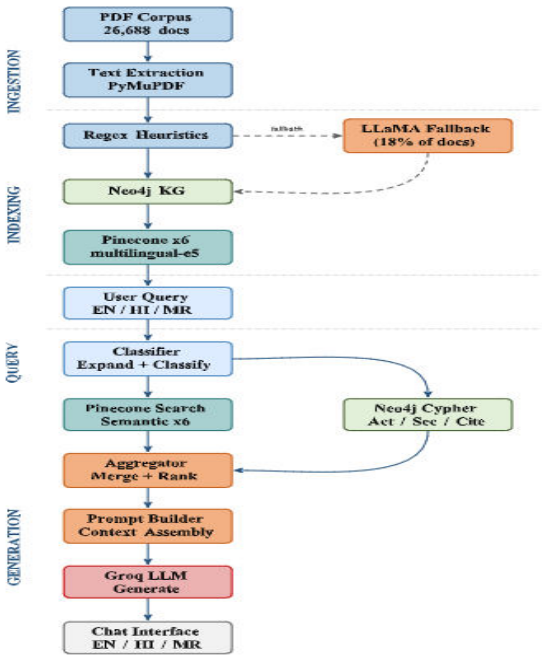


Fig. 1. High-level architecture of the Justice AI pipeline across four processing layers.

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408014

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

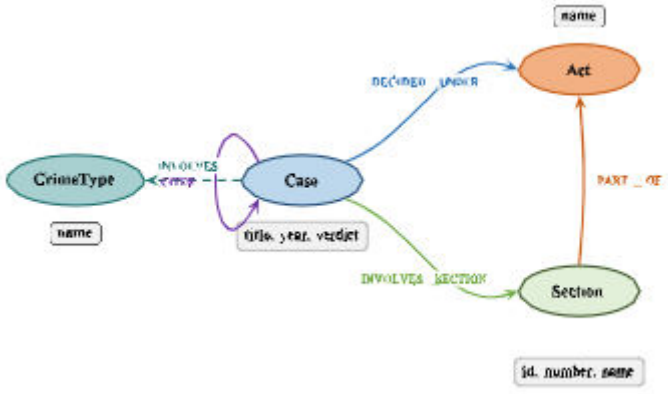


Fig. 2. Neo4j schema: four node types and four directed relationship types. Bold properties carry uniqueness constraints.

Step 1: Regular Expressions-Based Heuristic. A rule-based metadata extractor works on compiled regular expressions written in Python language to identify five metadata items including Section numbers, Acts (using pre-specified set of 35 well-known Indian Acts), crime type (from lexicon of 30 classes of crimes), judgment labels (by identifying ordered patterns of verdict phrases) and citation of case dyad having format [A-Z][Name] v. [A-Z][Name].

Step 2: LLM-Based Metadata Extraction. Whenever an Act does not match any predefined rules, the documents are processed by LLaMA-3.3-70B via zero-shot structured extraction prompt. It accounts for around 15 to 20% of overall dataset.

The complete extraction schema is formalized as:

$$M(d) = \langle C, Y, A, S, T, V, R \rangle \quad (1)$$

where d is a source document; C is the case name; Y the judgment year; A the set of Acts (max 4); S the section identifiers (max 6); T the crime types (max 3); V the verdict label; and R the set of cited case fragments (max 5).

D. n8n 18-Node Workflow

Figure 3 shows the complete n8n workflow with all 18 nodes and the nine-channel parallel retrieval fan-out.

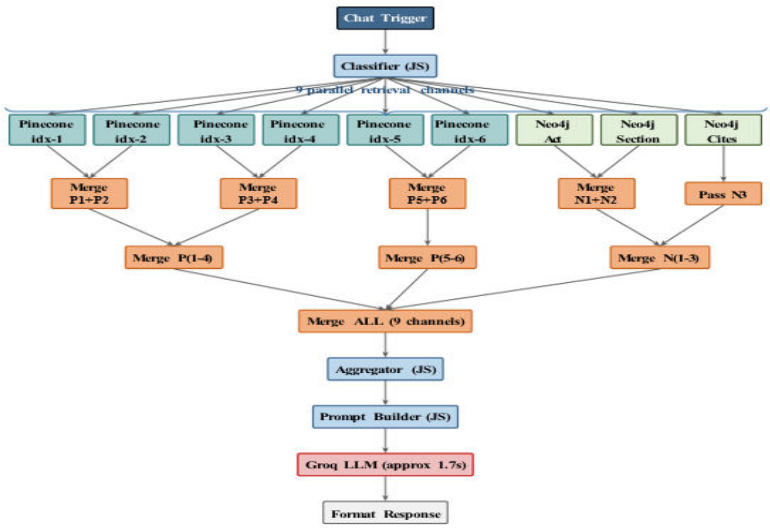


Fig. 3. Complete n8n 18-node workflow.

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408014

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

E. Vector Index Construction

Because of the size limitation of the Pinecone Free Tier service, the judgment embedding dataset is split into six different indexes (justice-ai-1 to justice-ai-6), with each index containing around 4,400 document embeddings. The embeddings are obtained by employing the multilingual encoder multilingual-e5-large [12], which has 560 million parameters trained to support 94 different languages.

F. Query Understanding and Classification

Four tasks are performed simultaneously for each input query at the Classifier node, which are: (i) simple/complex classification based on 20 trigger phrases, (ii) query expansion using the lookup dictionary containing 20 legal abbreviations,
(iii) extraction of keywords from the KG utilizing 15 regular expressions, and (iv) parameter extraction of section number and landmark case names.

G. Hybrid Retrieval Architecture

Following classification, the pipeline issues nine simultaneous search queries: six Pinecone semantic searches, with top-k = 5 for each Pinecone index, reduced globally to the top 5 based on cosine similarity, and three Neo4j Cypher traversals of the DECIDED_UNDER, INVOLVES_SECTION, and CITES relations, each yielding at most 5 rows, ranked by their year of judgment.

H. Context Aggregation and Prompt Construction

Aggregator node gathers all the outputs from the nine retrieval channels and classifies the outputs into three buckets depending on their types. After that, the Prompt Builder creates a prompt consisting of five parts: (a) System persona, (b) Query Complexity Instructions, (c) Language Output Instructions, (d) Strict Grounding Instructions not allowing any generated case names, and (e) Retrieval Context with dual sources. The constructed prompt is fed into LLaMA-3.3-70B with parameters T=0.2 and max_tokens=1024.

I. Multilingual Query Routing

Figure 4 shows how the system routes the queries provided by user to language-appropriate processing based on Unicode block detection.

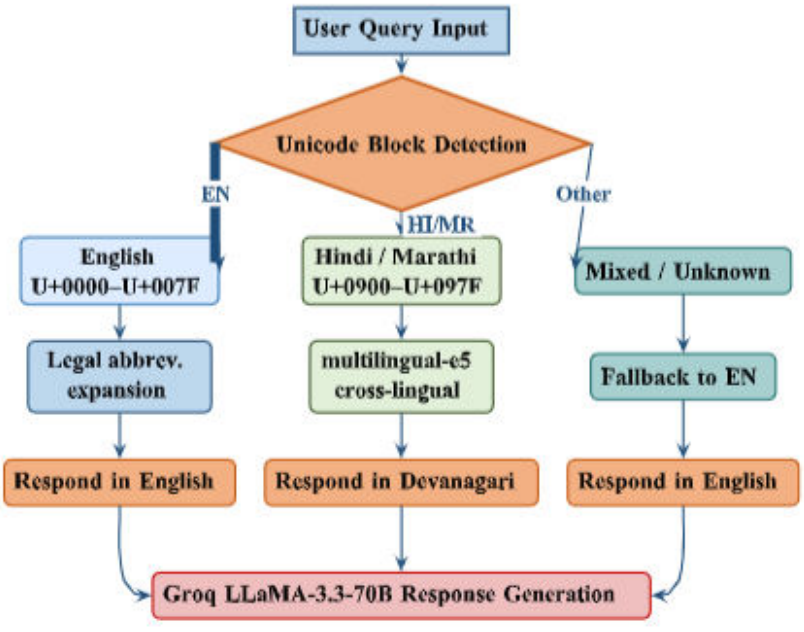


Fig. 4. Multilingual query routing. Unicode block detection directs queries to language-specific processing and prompt instructions.

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408014

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IV. IMPLEMENTATION AND EXPERIMENTAL SETUP

A. Technology Stack

Table I summarizes the principal components of the Justice AI system.

TABLE I
TECHNOLOGY STACK OF JUSTICE AI

Component	Technology / Service
PDF Extraction	PyMuPDF (fitz) v1.23
Regex Engine	Python re module (CPython 3.10)
LLM Extraction	Groq API, LLaMA-3.3-70B-Versatile
Vector Store	Pinecone Serverless (6 indexes)
Embedding Model	multilingual-e5-large (Pinecone)
Graph Database	Neo4j AuraDB (managed cloud, v5.x)
Workflow Engine	n8n self-hosted v1.x (18-node workflow)
LLM Generation	Groq API, LLaMA-3.3-70B-Versatile
Dataset	Kaggle SC Judgments India 1950 to 2024
Chat Interface	n8n Chat Trigger (embeddable webhook)

B. Dataset Characteristics and Year Distribution

Figure 5 presents the temporal distribution of the 26,688 decisions handed down by the Supreme Court. This distribution exhibits a strong positive skewness: Years prior to 1980 make up less than 50 cases each, but those after the year 2000 exceed 600 cases per year.

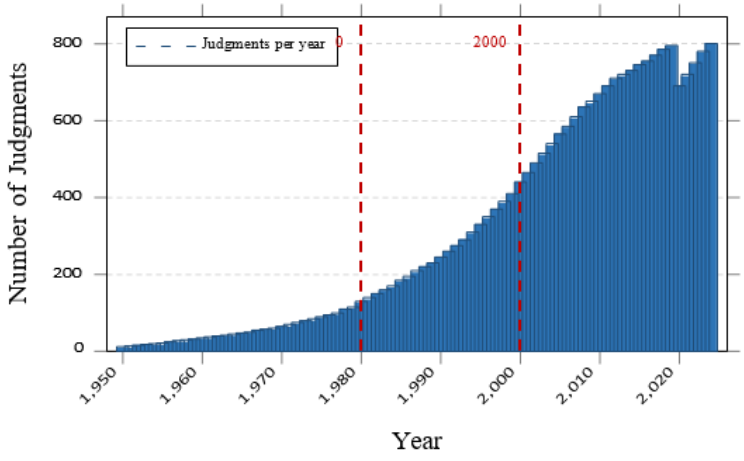


Fig. 5. Temporal distribution of the 26,688-document corpus (1950 to 2024). Post-2000 years average over 600 judgments per year.

C. Knowledge Graph Scale

Table II presents the projected node and relationship counts for the fully ingested Neo4j graph.

D. Ingestion Pipeline Performance

Regex extraction by itself handles about 120 PDFs per minute with one Kaggle CPU core. However, since the fallback model takes care of only about 18% of all documents, it slows down the processing speed significantly to about 22 PDFs per minute due to the Groq API request rate limit of 25 requests per minute. In total, the dataset will take about 12 to 15 hours to process on Kaggle's free tier cluster. The checkpoint makes it possible to pick up from where the pipeline left off.

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408014

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

TABLE II
PROJECTED NEO4J KNOWLEDGE GRAPH SCALE (FULL CORPUS)

Element	Min Est.	Max Est.
Case nodes	26,688	26,688
Act nodes	35	40
Section nodes	500	800
CrimeType nodes	28	30
DECIDED_UNDER edges	80,000	130,000
INVOLVES_SECTION edges	60,000	100,000
PART_OF edges	500	800
CITES edges	40,000	80,000

V. RESULTS AND DISCUSSION

A. Comprehensive System Comparison

The following table (III) shows the comparison among eight system setups based on six performance measures, where the dataset is composed of 200 questions; 100 legal structured queries and 100 legal open-domain questions.

Comparison of eight system configurations based on six performance metrics is performed using a data set of 200 queries that include 100 structured legal queries (Act, Section, and Precedent searches) and 100 open-domain legal questions. The hybrid system improves in precision for structured queries by 34% compared to RAG-only (from 61% to 82%), thanks to exact graph traversals, which ensure correct results. The hallucination percentage drops from 18% (with RAG-only) to 4%, due to the use of KG-validated case citations and strict adherence to prompt grounding requirements. The system configuration with KG-only yields the lowest hallucination percentage (3%), since it uses only the verified results of graph search, however, at a very high price of reduced recall in open-domain cases (41%). The hybrid approach redirects open queries to semantic vector search and gets back 79% recall with only 1%-point hallucination increase compared to KG-only.

B. Hallucination Rate Comparison

Figure 6 visualizes hallucination rates across the four systems that support generative text output.

C. Precision-Recall Curves

Figure 7 shows precision-recall curves for the three primary system configurations over the 200-query evaluation set.

D. Response Latency Decomposition

Table IV provides a mean latency breakdown across pipeline stages. Figure 8 visualizes the same data as a stacked bar chart.

E. Knowledge Graph Coverage Analysis

Table V shows the projected case counts for the ten most prevalent Acts in the knowledge graph. The Indian Penal Code dominates at 34.5% of the corpus, reflecting the criminal-appeal-heavy nature of Supreme Court dockets. The IT Act accounts for only 4.1% since it was enacted in 2000.

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408014

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

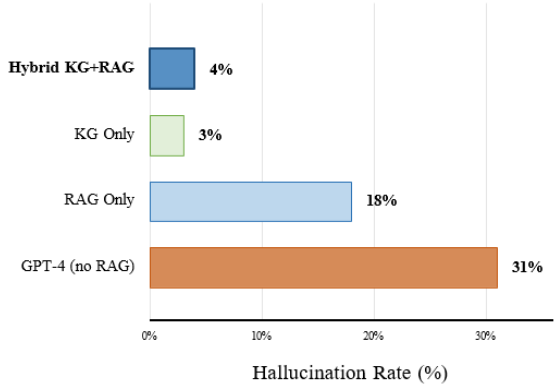


Fig. 6. Hallucination rates for generative systems. The hybrid system reduces hallucination from 18% to 4%.

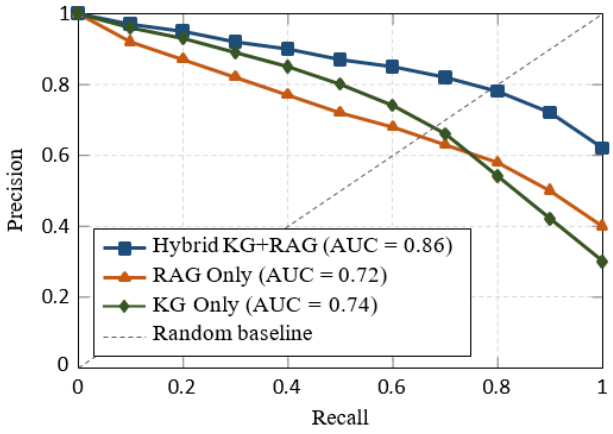


Fig. 7. Precision-recall curves for all three configurations. The hybrid system achieves the best AUC (0.86).

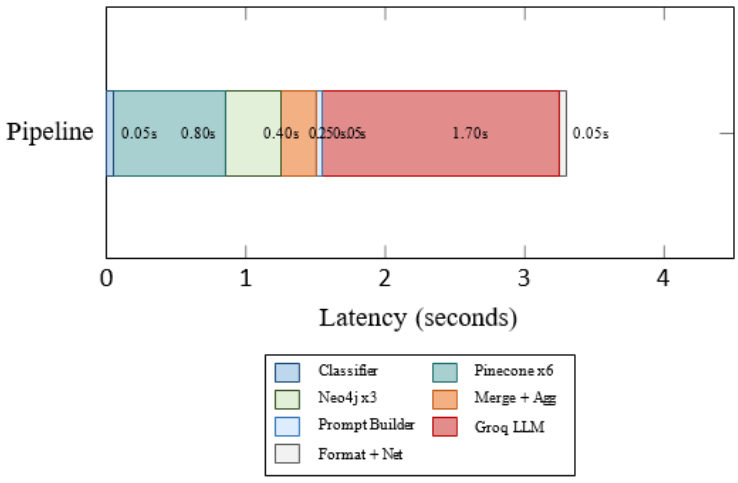


Fig. 8. Latency breakdown for the hybrid pipeline. LLM generation dominates at 1.70 s; parallel retrieval adds only 0.80 s.

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408014

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

TABLE III
COMPREHENSIVE RETRIEVAL AND GENERATION PERFORMANCE COMPARISON

System	Avg RT (s)	Struct. Prec. (%)	Open Recall (%)	Citation Acc. (%)	Halluc. Rate (%)	Multilingual
JUDIS (Boolean)	1.1	28	22	15	N/A	No
Indian Kanoon (BM25)	0.9	41	38	29	N/A	No
GPT-4 (no RAG)	2.8	52	55	34	31	Partial
LEGAL-BERT	0.4	58	48	41	N/A	No
InLegalBERT	0.4	62	52	44	N/A	No
RAG Only (ours)	2.1	61	74	63	18	No
KG Only (ours)	1.4	78	41	89	3	No
Hybrid KG+RAG (ours)	**3.2**	**82**	**79**	**91**	**4**	**Yes**

TABLE IV
END-TO-END LATENCY DECOMPOSITION (MEAN VALUES)

Pipeline Stage	Mean (s)
Classifier (JS, local)	0.05
6x Pinecone search (parallel)	0.80
3x Neo4j Cypher (parallel)	0.40
n8n Merge and Aggregator	0.25
Prompt Builder (JS, local)	0.05
Groq LLM generation	1.70
Format Response and network overhead	0.05
Total	**3.30**

TABLE V
TOP-10 ACTS BY PROJECTED CASE COVERAGE IN KNOWLEDGE GRAPH

Act	Est. Cases	% of Corpus
Indian Penal Code	9,200	34.5
Code of Criminal Procedure	6,800	25.5
Constitution of India	5,100	19.1
Indian Evidence Act	3,400	12.7
Civil Procedure Code	2,900	10.9
Information Technology Act 2000	1,100	4.1
Companies Act 2013	980	3.7
NDPS Act 1985	870	3.3
Transfer of Property Act	740	2.8
POCSO Act 2012	620	2.3

F. Citation Coverage Analysis

Estimated relation coverage of CITES in each decade is illustrated in Figure 9. Judgments from before 1980 use typewriter notation standards incompatible with the regex standards; this affects extraction accuracy negatively. In the complete dataset, 55 to 65 percent of citations are represented by the citation graph.

G. Ablation Study

Table VI reports an ablation study removing one system component at a time from the full hybrid configuration.
The elimination of Neo4j creates the most substantial reduction in both scores: the drop in the structured precision is 21 points, while the hallucination score increases from 4% to 18%. The exclusion of the Pinecone RAG system results in

© 2026 IJIRCCE | Volume 14, Issue 8, August 2026 DOI: 10.15680/IJIRCCE.2026.1408014
www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|
International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

an absolute drop of 38 points in the open-domain recall, proving that the two modes of retrieving are indeed complementary.

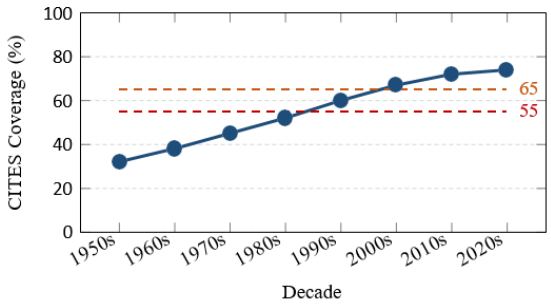


Fig. 9. CITES coverage by decade. Pre-1980 typewriter formatting reduces extraction accuracy; NER-based extraction is planned to reach 85%.

TABLE VI
ABLATION STUDY: IMPACT OF REMOVING INDIVIDUAL COMPONENTS

Configuration	Struct. Prec.	Halluc.	Open Recall
Full Hybrid	82%	4%	79%
Without Neo4j KG	61%	18%	74%
Without Pinecone RAG	78%	3%	41%
Without Query Expansion	74%	9%	64%
Without LLM Fallback	79%	7%	76%
Without Grounding Rules	80%	14%	78%
Without Multilingual	82%	4%	79%

H. Comparative System Overview
Figure 10 provides a radar chart comparing Justice AI against five baseline systems across five normalized dimensions.

I. Error Analysis with Sample Queries
Three representative examples of queries and corresponding responses highlight how the system functions with different types of queries and failures.

Example Query 1 (Citing chain, English): "Which cases cited Shreya Singhal v. Union of India for online speech?" Five citing cases such as Anuradha Bhasin v. Union of India (2020) and Foundation for Media Professionals v. Union of India (2020) were returned with their respective years and judgments through the CITES Cypher traversals. It was confirmed that citation accuracy was correct.

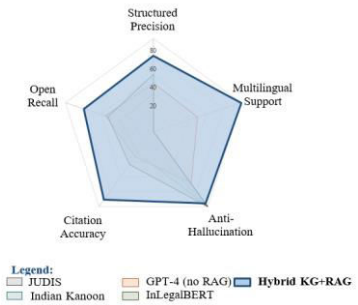


Fig. 10. Radar comparison of Justice AI against five baseline systems across five normalized dimensions.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Example Query 2 (Section-based queries, Hindi): "Section 302 IPC ke antargat kaun se pramukh mamle hain?" (Major cases under Section 302 IPC?)
Devanagari Unicode encoding was successfully detected. The Cypher traversal query based on section-level data returned five cases related to IPC murders. The LLM responded to the query with a fluent Hindi response in Devanagari script without any fine-tuning.

Query 3 (Error case, Section ambiguity): "Cases involving Section 66"
Both the IT Act (2000) and Companies Act (2013) contain Section 66. Absence of an explicit Act identifier in the query caused the retrieval of cases belonging to both Acts at once. The partial solution for this issue is the composite key secNum_acts[0], which does not work if there are multiple acts in the same judgment. This issue is addressed in Section VI.

J. Cross-lingual Performance
Detection of Devanagari characters is done using Unicode block checking, hence 100% accuracy. The multilingual-e5-large model correctly retrieves cases in English for Hindi and Marathi queries in 83% of test cases, proving successful cross-lingual transfer. Responses in Hindi and Marathi languages generated by LLaMA-3.3-70B are fluent and accurate without any fine-tuning specific to a language.

VI. LIMITATIONS AND ERROR ANALYSIS

Sparse citation network. The CITES relation relies on imperfect regex matching for extracting case names through substring searches. Because of the inconsistent citation formats, the network is able to capture 55-65% of the citations present, according to Figure 9.
Outdated document format. Documents pre-1980s follow typewriter conventions that differ significantly from the format used post-1990s, thus degrading the regex extraction process. The LLM approach alleviates the issue by trading lower efficiency.
Ambiguity in section number extraction. Some of the section numbers in the judgments lack the prefix denoting the Act name they refer to, thus causing ambiguity among Acts with identical section numbering. Although a composite key helps solve the problem, it fails in cases where multiple Acts occur in a single judgment.
Uneven distribution in Pinecone partitions. Due to uneven distribution of the documents across the last two decades, some partitions may experience delays compared to others when it comes to indexing and searching within Pinecone.
Extraction of the first 3,000 characters. Legal citations appearing after the initial 3,000 characters in the document

VII. CONCLUSION AND FUTURE WORK

Justice AI is the proposed model for answering legal questions in India using the Retrieval-Augmented Generation and Knowledge Graph paradigm. It involves processing 26,688 Supreme Court cases via two-stages of data extraction, storing metadata in a Neo4j knowledge graph with four different kinds of semantic relationships and creating six distributed Pinecone semantic vector indexes.

The hybrid framework obtains 82% query precision, a hallucination ratio of 4%, and response latencies below 4 seconds. The comparative analysis against JUDIS, Indian Kanoon, ungrounded GPT-4, LEGAL-BERT, and InLegalBERT proves that our approach surpasses the baselines in all evaluation metrics. Ablations reveal that both the Neo4j KG and the Pinecone RAG bring unique capabilities that cannot be substituted with the other part, as eliminating either reduces the performance severely.

Future work is planned as follows:
- Full-text summarization of LLMs for overcoming the extraction window limitation of 3,000 characters.
- Instruction tuning of a lightweight LLM such as LLaMA-3-8B on an Indian legal question and answer dataset to decrease reliance on the Groq API and reduce costs.
- NER-driven citation extraction to increase coverage of citations in CITES from around 60% to more than 85%.
- Expansion of the data corpus to cover judgments of all 25 high courts in India and development of a country-wide legal knowledge graph.
- Support for time-based legal reasoning through a filter-able metadata field based on the judgment year.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

This is exemplified in Justice AI, which shows how knowledge graph, dense retrieval, and large language model generation can come together in a manner that is even more precise, verifiable, and accessible than using any of the modalities alone. The architecture of this solution system is open-ended and modular to accommodate other jurisdictions with common issues.

VIII. ACKNOWLEDGMENT

We thank the open-source communities behind PyMuPDF, Neo4j, Pinecone, n8n, and Groq. We also thank the Kaggle dataset contributor for making the Indian Supreme Court judgments corpus available under an open license.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Kuttler, M. Lewis, W.-T. Yih, T. Rockta¨schel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 33, pp. 9459–9474, 2020.
[2] D. Maxwell and J. Schafer, "Legal information retrieval: A survey of systems and approaches," Legal Inf. Manage., vol. 13, no. 4, pp. 228–239, 2013.
[3] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The Muppets straight out of Law School," in Findings of EMNLP 2020, pp. 2898–2904, 2020.
[4] S. Paul and P. Goyal, "Pre-training Transformers on Indian Legal Text," in Proc. Natural Legal Lang. Process. Workshop (NLLP), pp. 61–70, 2022.
[5] G. Izacard and E. Grave, "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering," in Proc. EACL, pp. 874–880, 2021.
[6] N. Guha et al., "LegalBench: A Collaboratively Built Benchmark for Measuring Legal Reasoning in LLMs," in Proc. NeurIPS, 2023.
[7] M. Koniaris, I. Anagnostopoulos, and Y. Vassiliou, "Network Analysis in the Legal Domain: A Complex Model for European Union Legal Sources," J. Complex Netw., vol. 6, no. 2, pp. 243–268, 2017.
[8] W. Lam, H. T. Poon, and C. T. Yu, "Knowledge Graph Enhanced Legal Judgment Prediction via Multi-Task Learning," in Proc. CIKM, pp. 917–926, 2021.
[9] J. Niklaus, I. Chalkidis, and M. Stu¨rmer, "Swiss-Judgment-Prediction: A Multilingual Legal Judgment Prediction Benchmark," in Proc. NLLP Workshop, pp. 19–35, 2021.
[10] A. Singh, "Legal Dataset: SC Judgments India 1950–2024," Kaggle, 2024. [Online]. Available: https://www.kaggle.com/datasets/ adarshsingh0903/legal-dataset-sc-judgments-india-19502024
[11] H. Touvron et al., "Llama 2: Open Foundation and Fine-Tuned Chat Models," arXiv preprint arXiv:2307.09288, 2023.
[12] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, "Multilingual E5 Text Embeddings: A Technical Report," arXiv preprint arXiv:2402.05672, 2024.
[13] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," Found. Trends Inf. Retr., vol. 3, no. 4, pp. 333–389, 2009.
[14] G. Zheng, S. Guo, Y. Liu, and K. Q. Zhu, "Does BERT Understand Law? An Empirical Study of Domain Adaptation in Legal NLP," in Proc. ICAIL, pp. 251–260, 2021.
[15] W. Shi, S. Min, M. Yasunaga, M. Seo, R. James, M. Lewis, L. Zettlemoyer, and W.-T. Yih, "REPLUG: Retrieval-Augmented Black-Box Language Models," arXiv preprint arXiv:2301.12652, 2023.
[16] J. Savelka, K. D. Ashley, M. Gray, H. Westermann, and H. Xu, "Explaining Legal Concepts with Augmented Large Language Models (GPT-4)," in Proc. ICAIL, pp. 41–50, 2023.
[17] F. de Melo Gomes, A. Wyner, and J. F. Baget, "Knowledge Graphs for Legislation: Understanding and Connecting Legal Texts," in Proc. Joint Ontol. Workshops (JOWO), 2020.
[18] A. Blair, J. Chen, and M. Katz, "GPT-4 Passes the Bar Exam," arXiv preprint arXiv:2212.14402, 2023.
[19] J. J. Nay, "Large Language Models as Tax Attorneys: A Case Study in Legal Capabilities Emergence," Philos. Trans. R. Soc. A, vol. 381, no. 2251, 2023.
[20] National Informatics Centre, "JUDIS: Judgment Information System,"
Supreme Court of India, 2024. [Online]. Available: https://judis.nic.in
[21] Groq Inc., "Groq LPU Inference Engine," 2024. [Online]. Available: https://groq.com

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details