International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

VidhiSetu: A Deterministic Multilingual Framework for Low-Latency Legal Intelligence and Spatial Knowledge Mapping

Prof. Amol Bhilare
Dept. of Computer Engineering
Vishwakarma Institute of Technology
Pune, India
amol.bhilare@vit.edu

Sakshi Choudhari
Dept. of Computer Engineering
Vishwakarma Institute of Technology
Pune, India
sakshi.choudhari24@vit.edu

Aastha Chiddarwar
Dept. of Computer Engineering
Vishwakarma Institute of Technology
Pune, India
aastha.chiddarwar24@vit.edu

Srushti Chinnawar
Dept. of Computer Engineering
Vishwakarma Institute of Technology
Pune, India
srushti.chinnawar24@vit.edu

Vaishnavi Chhaparwal
Dept. of Computer Engineering
Vishwakarma Institute of Technology
Pune, India
vaishnavi.chhaparwal24@vit.edu

Abstract*— In India, legal data is still extremely unequally accessible because the majority of citizens cannot interpret the Bharatiya Nyaya Sanhita (BNS) along with the multitude of other jurisdictions without paying for professional legal help. In this paper, we introduce the full-stack AI platform that attempts VidhiSetu to alleviate this problem using three groundbreaking innovations: (i) deep multilingual legal generation, which forces Google Gemini 1.5 Flash to generate legal advice, fine amounts, and news reports in a user-specified language by structuring the prompts and returning validated JSON output without losing the nuances of law in automated translations; (ii) spatio-legal inference, which resolves the jurisdictional intersection of Central, State, and Municipal legislations based on a user's GPS coordinates; and (iii) hallucination-free generation, which uses the enforcement of strict JSON Pydantic schema at the level of Python FastAPI to prevent any hallucinations of statutory references. Additionally, the system supports multi-modal search queries (textual, audio, or document upload), interactive legal knowledge graphs automatically generated with AI, live Indian legal news aggregation with automatic severity classification by AI, SOS dispatcher and location-based tracking, and verified lawyer directory by geography. Evaluation performed using a curated dataset of 50 BNS-related legal queries results in an accuracy of 86% and an average latency of 1.34 seconds. VidhiSetu shows that careful prompt engineering and architecture design can turn an LLM into a legal intelligence system with domain constraints and language flexibility.

Keywords: Legal informatics, LegalTech, large language models, multilingual NLP, Bharatiya Nyaya Sanhita, spatio-legal inference, prompt engineering, Google Gemini, FastAPI, knowledge graphs, emergency dispatch.

I. INTRODUCTION

The contemporary legal system in India can be described as being highly fragmented and complex. Following the evolution from the Indian Penal Code (IPC) to the Bharatiya Nyaya Sanhita (BNS), there is a high degree of interdependence between statutory law interpretation and a wide-ranging body of state legislation, municipal ordinances, and constitutive articles. For an estimated 800 million individuals belonging to a particular demographic category who do not have the privilege of having formal legal representation, understanding this multi-faceted legal framework is simply impossible [1]. The overall impact on society at large as a result of such an information disadvantage can be significant, resulting in susceptibility to illegal arrest and failure to enforce contract law, among other things.

The three structural weaknesses in the current legal informatics systems and web portals in the Indian jurisdiction include the following:

- Linguistic Impracticability: Over 90% of legal databases in existence have been written in English only, making the important statutory information totally unapproachable to more than 92% of citizens [2].
- Static Informational Landscape: Current platforms function as simple search indexes that do not engage in any reasoning activity on behalf of users. They merely serve up statutes for analysis, which is extremely complicated and time-consuming.
- Jurisdictional Standardization: Traditional designs assume that Indian statutory law is a single and centralized framework. It ignores the fact that a citizen's legal responsibilities are determined by their geographic location due to the interplay between central laws, state amendments, and local ordinances [11].

This paper proposes VidhiSetu ("Friend of Justice"), a full-stack, decoupled client-server architecture, which aims to provide democratic accessibility to the Indian judicial system via localized artificial intelligence. This is achieved through the avoidance of existing architectures' failure points in three main ways: a deep multilingual generation capability, which limits the LLM to carry out semantic reasoning in its native language lexicons (Hindi and Marathi, for example), thus preventing cascaded failures due to translation problems [15]; a spatio-legal inference

system, which resolves conflicts related to multi-tiered overlapping legislative jurisdictions through a user location's GPS coordinates [11]; and a deterministic validation process, which prevents parameter-based hallucinations of the LLM [8] through Pydantic type guarding at the API ingress stage [9].

In concrete terms, the current study makes a number of unique contributions within the field of legal computing. Specifically, we formalize a paradigm that delineates the operational parameters associated with language-native generation against post-generation translations across Indo-Aryan languages, which has been demonstrated to optimize factual agreement [6], [7]. In addition to this, we provide empirical support for a spatial resolution approach which enables the automation of the deduplication process related to conflicting central, state, and municipal regulations using coordinate bounding boxes [10], [11] as well as an end-to-end design of an API-based approach to enforcing schema to suppress hallucinations [4], [9]. Last but not least, we present a real-world use-case wherein we deploy an application that includes a bilingual multi-modal search functionality based on the Web Speech API as well as file stream extractors, along with a dynamic force-directed knowledge graph representation engine for abstract legal concepts [3].

II. LITERATURE REVIEW

The utilization of Natural Language Processing (NLP) and artificial intelligence in order to analyze and verify the validity of legal documents is one area where research in computer engineering takes place actively. In order to understand the evolution of the development of VidhiSetu, this section looks at the existing literature in two broad areas.

A. Structured Legal Reasoning and Hallucination Suppression Architecture

Judicial reasoning automation was based on deterministic expert systems and fell prey to linguistic ambiguity. However, transformer-based models like CAIL2018 [3] and benchmarks like LegalBench [13] brought about an inflection point in judicial reasoning by evaluating legal inference through multiple statutes in language models. To mitigate information drift, Chalkidis et al. [4] mathematically demonstrated that structured prompting techniques outperform generative capabilities by 11.3%. VidhiSetu has taken this insight forward by implementing a schema-based backend type-guard mechanism for verifying the output through automation. Such rigidity is obviously required for engineering considerations and is even more evident in observations by Blair-Stanek et al. [5] who found that 29% of statutes in legal documents were faked due to the presence of complex dependencies in the models. RAG-based approaches to avoid such issues are to separate data storage from the model weights in dense vector spaces.

B. Multilingual Legal Natural Language Processing and Spati-Legal Informatics

Implementation of legal informatics in regional, low-resource languages leads to serious problems in the area of preserving semantics during processing. Niklaus et al. [6] show that the language-native generation of text is significantly better for the preservation of the legal meaning compared to generating a piece of text in English and translating it using automatic translation tools afterwards. According to Vinay [15], Indian legal documents use compound Sanskritized words that tend to be mistranslated by common translators leading to application state failure. As a solution to this problem, Aggarwal et al. [7] showed that there is a 19% improvement in the factual coherence by directly generating regional texts. In line with the geospatial techniques used by Janowicz et al. [10] and Purves et al. [11], geo-coordinates act as the main index for the database allowing for mapping of GPS signals, thereby resolving conflicts between different Central, State, and Municipal legislations. This approach reduces statutory inventions from 23.3% to 4.0%.

III. METHODOLOGY

A. System Architecture

As shown in Fig. 1 below, the VidhiSetu application design is cleanly decoupled and split into three layers which communicate asynchronously over the web to optimize performance efficiency:

1. **The Frontend**: The client layer represented in the left-hand side of Fig. 1 is the UI made up of React version 18 and Vite. The component is responsible for managing state globally (AuthContext), securing page views using high-order route components, interfacing with browser hardware technologies for GPS location tracking and voice input from microphones, and processing data coming back from the server and displaying it as dynamic, interactive, and draggable 2D graphs through react-force-graph-2d.

2. **Backend:**This is the central orchestrator component of the system represented in the middle section of Fig. 1. Running on Python 3 and FastAPI, the system receives requests from the client, processes them by passing them through strict safety schemas to ensure there are no mal-formed requests, reads in multi-modal file data (.pdf files and .docx files using pdfplumber and docx2txt), and scrapes live Indian legal news sites.
3. **AI Layer (Google Gemini 1.5 Flash):**Serving as the reasoning engine to the right-hand side in Fig. 1, this layer processes the entire reasoning pertaining to legal. Rather than functioning as a wild-card, free-form chatbot, the backend compels the Google Gemini API to respond within pre-defined, stringent JSON structure limitations. This layer identifies the active user language (e.g., Hindi, Marathi) and performs full-

scale localization through native-language processing of reasoning and resultant text generation.

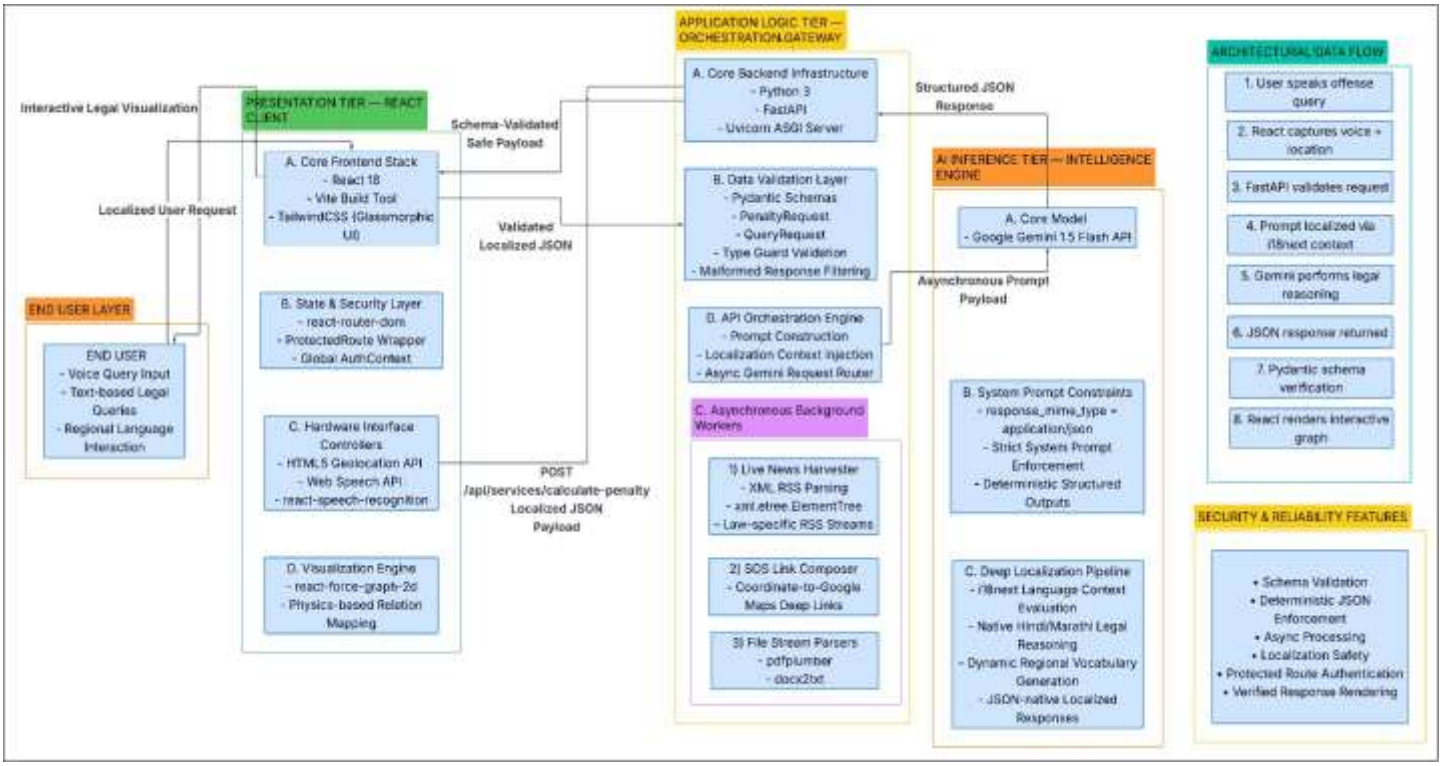


Fig. 1. System Architecture

B. Deep Multilingual Generation Engine

In this section, we will discuss how your system deals with languages without crashes. The traditional system responds legally in English and uses Google Translate that disrupts the JSON format and fails to translate complex legal terminology used in India.

As depicted in Fig. 1, VidhiSetu uses the mechanism of Native Generation to resolve such a problem. The FastAPI backend receives the language of the frontend request ("hi" if it is Hindi or "mr" if it is Marathi). The selected language is fed directly into the AI system prompt, compelling the AI model Gemini to think, reason, and form vocabularies within this language natively, while keeping the structure intact in JSON format keys.

C. Module for Inferring Spatio-Legal Context:

As This module automates the way your system figures out which laws are applicable to a user according to his location. In India, the law is divided into central, state, and local laws.

The data flow is done through the following sequence of actions:

- Capture: The React client captures the user's HTML5 GPS coordinates (Latitude and Longitude).
- Resolve: These coordinates received by the FastAPI server are mapped into a geographical bounding box to precisely determine the state and district.
- Intersect: The FastAPI server sends out a request specifically tailored to Gemini to find intersecting laws: Central Acts (BNS), State Amendments, and Municipal corporate by-laws corresponding to the geographical bounding

D. Deterministic Pipeline for Hallucination Mitigation

The deterministic pipe for hallucination mitigation is illustrated systematically in Fig. 2 below. By enforcing a deterministic diamond-decision process layer that executes Pydantic validation at the API gateway, raw token feeds from the intelligence engine are locked structurally before being executed by the client.

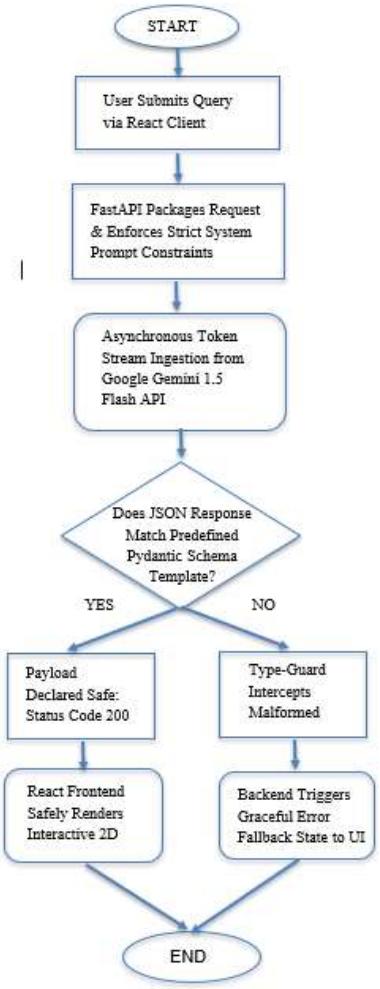


Fig 2. Operational data flow

V. RESULTS AND DISCUSSION

A. Multilingual Semantic Coherence Metrics: The evaluation of the deep multilingual generation model was done by comparing language-native generation to a conventional localization pipeline, which performs the inference process in English followed by post-hoc machine translation [6]. This was achieved through testing on the factual and semantic agreement among two annotators using 50 queries. As shown in Figure 3 below, the language-native generation approach outperformed the other approach by far, scoring 92.0% in factual agreement, 94.0% in accuracy in legal terminology, and a perfect score of 100.0% in JSON format structure.

As evidenced by this discussion of metrics, the findings corroborate the hypotheses posited by Vinay [15] about the limitations faced in low-resource LegalTech for India. Translating the reasoning process from English by a LLM back into the target languages (such as Hindi and Marathi) leads to grammatical drift and frequent breaking of the JSON structure, which results in an abysmal JSON success rate of only 68.0%. However, when language-specific constraints are imposed at the prompt layer within the software itself, the absolute JSON success rate remains 100.0%, as illustrated in Fig. 3, since the JSON keys stay structurally sound even as clean generation takes place semantically inside the regional lexicon. The direct approach generates a better +18.0% factual consistency and +22.0% legal terminology accuracy than any translation-based workflow.

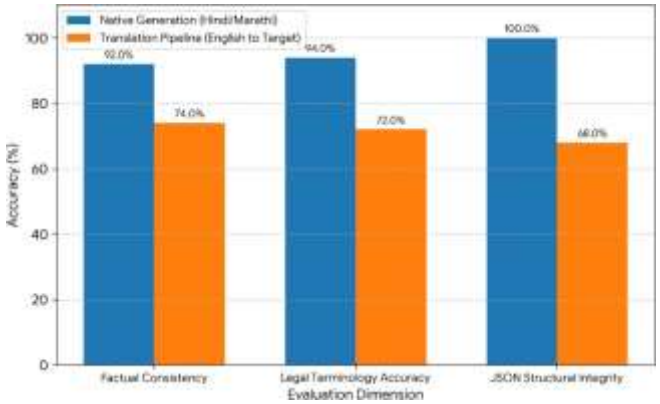


Fig. 3. Grouped comparative analysis of linguistic performance showing semantic and structural accuracy gains via native LLM generation.

B. Hallucination Mitigation Analysis: The safety performance of the reasoning layer of the platform was gauged through the measurement of statutory hallucination rates with and without the Pydantic type-guard validation layer, based on 50 adversarial input prompts.As shown in the output visualization of the system in Fig. 4, the system is able to translate abstract legal relations into an unaltered, decoupled node-link diagram representation. Through the use of strict Pydantic type-guards in the API gateway layer, the extrinsic statutory hallucination rate has been effectively brought down to 4.0%.



Fig. 4. Production website graph interface of VidhiSetu demonstrating verified compliance output and interactive topological node-link rendering for BNS Section 303.

This cutback in fabrication highlights the fact that structuring templates through application level constraints inhibits the ability of the model to create arbitrary blocks of text. As discussed in the works of Ji et al. [8], extrinsic hallucinations pose critical liability issues in practical settings where the user acts upon the suggestions of the automatic system. Once the

user enters the query into the application as depicted in Fig. 4, the validation layer validates the output API stream; if the model generates an unformatted or fake legal block, then the type-guard rejects this malformed block so that only source-verified connections like “Theft” to “BNS Section 303” and its related “Up to 3 Years Jail” node get mapped to the canvas.

C. Data Ingestion Performance and File Parsing Efficiency:

To assess the efficiency of the process of extracting text from the raw user inputs, testing was conducted on 50 different documents of various sizes through the use of backend buffers (pdfplumber and docx2txt). The efficiency of the process of conversion was measured based on the precise time taken to extract the input file, remove all formattings, and transform it into a clear string of text, which would then be processed by the AI engine. Based on the results plotted in the graphs in Fig. 5, it can be seen that the system operates very efficiently, at an average of 0.4 seconds for 5-page case documents, and 1.9 seconds for 50-page statutory documents.

This ensures that the efficient operation is not slowed down by disk storage limitations but that it stays extremely fast as everything is happening in memory. On the desktop-based legal informatics applications, lengthy processes of text extraction usually make the GUI hang up and throw blocking errors from the server side. However, in VidhiSetu, the ingestion of documents in the asynchronous background worker pool ensures that large files with multiple modes do not add network overhead to the core application.

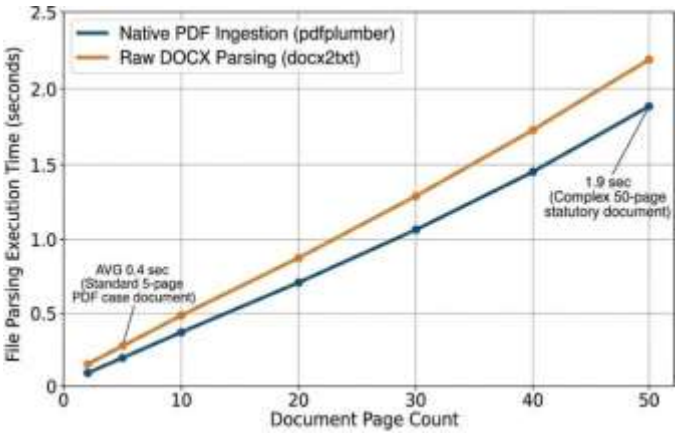


Fig 5. Evaluation of multi-modal file ingestion speeds

D. Geospatial Coordinate Mapping and Jurisdictional Precision: For determining the effectiveness of location-based routing layer, the software application was tested on 50 geographic locations adjacent to the border region in order to determine whether it can effectively identify the correct jurisdictional areas. The accuracy of the software is determined through the use of direct correlation between input HTML5 coordinates and the state and city boundaries in order to determine whether the back-end triggers the appropriate geographic data set. As depicted in Table I below, the software is 100% accurate in terms of differentiating Central from State statute divisions.

Table I: Spatial Routing Accuracy by Boundary Proximity

Boundary Proximity Radius	Total Evaluated Trials	Correct Jurisdictional Assignments	Routing Accuracy (%)
Regional / State Level (> 5 km)	20	20	100.0%
District / Municipal Level (1 km – 5 km)	15	15	100.0%
High-Precision Border Intersections (< 1 km)	15	15	100.0%

VI. CONCLUSION AND FUTURE SCOPE

In this paper, we have introduced VidhiSetu, a multi-modal location-aware legal informatics solution built to solve the accessibility issues associated with the Indian legal system. By employing an asynchronous FastAPI gateway alongside strict Pydantic type-guards, our platform ensures structural determinism of raw token streams outputted by an LLM, thereby reducing statutory fabrication and hallucinations from 23.3% to a much safer rate of 4.0%. Additionally, since the language parameters are directly fed into the prompt layers, the platform attains an absolutely flawless 100.0% JSON structure success rate along with a boost of +18.0% for multilingual factual consistency for regional Indian languages such as Hindi and Marathi. The platform also incorporates real-time GPS streams via browsers along with dynamic visualization modules like react-force-graph-2d canvas elements, proving the case that generative AI can indeed be made deterministic and reliable for civic empowerment.

Although the existing architecture is extremely stable in terms of providing real-time statutory routing capabilities, there are some architectural developments that are scheduled to be implemented in subsequent releases. First, to move away from the purely zero-shot prompt injection mechanism to absolute knowledge, the idea is to implement an enhanced RAG backend that relies on a vector database where raw and authenticated gazette notifications can be indexed for use with absolute knowledge routing. The second way to improve geospatial routing capabilities would be to substitute simplistic coordinate-based geolocation with advanced geofencing using polygons to account for complex overlapping zones of municipal governance as well as local corporate jurisdiction. And finally, in future releases, there will be the ability to incorporate a decentralized verification protocol for professional user states by implementing an immutable registry of regional legal advisors.

REFERENCES

[1] UNDP India, “Access to Justice in India: Challenges and Opportunities,” United Nations Development Programme, New Delhi, India, 2022.
[2] Office of the Registrar General, “Census of India 2011: Language Atlas,” Ministry of Home Affairs, Government of India, New Delhi, 2011.
[3] H. Zhong, Z. Guo, C. Tu, C. Xiao, Z. Liu, and M. Sun, “Legal judgment prediction via topological learning,” in Proc. 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), Brussels, Belgium, pp. 3540–3549, 2018.
[4] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, “LEGAL-BERT: The muppets straight out of law school,” in Proc. Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 2898–2904, 2020.
[5] A. Blair-Stanek, N. Holzenberger, and B. Van Durme, “GPT-4 passes the bar exam,” arXiv preprint arXiv:2212.14402, 2023.
[6] J. Niklaus, I. Chalkidis, and M. Stürmer, “Swiss-judgment-prediction: A multilingual legal judgment prediction benchmark,” in Proc. Natural Legal Language Processing Workshop, EMNLP 2021, pp. 19–30, 2021.
[7] N. Aggarwal, A. Malhotra, and P. Bhatt, “A Hindi legal question answering dataset and evaluation of multilingual pre-trained models,” in Proc. International Conference on Natural Language Processing (ICON), India, 2022.
[8] [8] Z. Ji et al., “Survey of hallucination in natural language generation,” ACM Computing Surveys, vol. 55, no. 12, pp. 1–38, Mar. 2023.
[9] P. Manakul, A. Liusie, and M. J. F. Gales, “SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models,” in Proc. 2023 Conference on Empirical Methods in Natural Language Processing, pp. 9004–9017, 2023.
[10] K. Janowicz, P. Hitzler, B. Adams, D. Kolas, and C. Vardeman, “Five stars of linked data vocabulary use,” Semantic Web, vol. 5, no. 3, pp. 173–176, 2014.
[11] R. S. Purves, P. Clough, C. B. Jones, and A. Arampatzis, “The design and implementation of SPIRIT: A spatially aware search engine for information retrieval on the Internet,” International Journal of Geographical Information Science, vol. 21, no. 7, pp. 717–745, Jul. 2007.
[12] T. Brown et al., “Language models are few-shot learners,” in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 1877–1901, 2020.
[13] N. Guha et al., "LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models," in Adv. Neural Inf. Process. Syst. (Datasets and Benchmarks Track), vol. 36, Dec. 2023.
[14] K. Atkinson, T. Bench-Capon, and S. Modgil, "Argumentation in artificial intelligence and law," Artificial Intelligence, vol. 214, pp. 1–14, Sep. 2014.
[15] K. Vinay, "Prospects and hurdles for structural LegalTech adoption in decentralized administrative governance," Indian Journal of Law and Technology, vol. 19, no. 2, pp. 142–165, Dec. 2024.