International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Vakil — A Virtual Assistant for Knowledge in Indian Law

Manjunath S[1], Marilingappa[2], Likith Reddy N[3], J. Soundar Balaji[4], and Shivaranjini C[5]

[1234] Department of CSE, Sir M. Visvesvaraya Institute of Technology, VTU, Bengaluru, India

[5]Assistant Professor, Department of CSE, Sir M. Visvesvaraya Institute of Technology, VTU, Bengaluru, India

Abstract — We present VAKIL, a domain-focused virtual assistant for Indian law created by fine-tuning Microsoft's Phi-3 Mini (4K Instruct) model with Low-Rank Adaptation (LoRA). The training corpus comprised curated Supreme Court judgments, constitutional provisions, statutory legislation, and other authoritative legal texts. Fine-tuning was performed on a RunPod RTX A6000 instance for 18 hours across two epochs, yielding a marked decrease in training loss and improved domain alignment.

To ensure responses are factually grounded, VAKIL integrates a Retrieval-Augmented Generation (RAG) pipeline: documents are chunked, tokenized, embedded, and indexed using a FAISS semantic vector store to enable high-precision retrieval. For relational and precedent-style reasoning, we construct a Neo4j AuraDB knowledge graph representing cases, statutes, and legal doctrines, and we apply a Graph Neural Network (GNN) over this graph to capture cross-document relationships and citation structure.

We evaluate VAKIL through intrinsic measures (loss curves, perplexity, and reductions in hallucination) and extrinsic tasks (legal question answering, statutory interpretation, and judgment summarization). Compared to the unadapted Phi-3 base, our fine-tuned model shows improved accuracy, stronger domain specificity, and greater interpretability. The final model is released on the Hugging Face Hub and served via a serverless vLLM runtime on RunPod to support low-latency API access. A chat-based interface was implemented to support learners, researchers, and legal practitioners.

VAKIL offers a reproducible workflow for building regionally focused legal assistants that balance retrieval, graph-based reasoning, and model fine-tuning. It is designed as an educational and research aid—not as a substitute for professional legal advice—and provides a transparent foundation for future enhancements.

Key Words: Legal AI, RAG, LoRa, Phi-3, Knowledge Graph, GNN, Indian Law, FAISS, Legal QA, Judgment Summarization.

1. INTRODUCTION

The rapid rise of computational technologies is changing knowledge-heavy fields, including law. In India, digital initiatives — such as court e-filing, online case repositories, and government programmes — have made many legal documents digitally available. Despite this progress, students, teachers, and early researchers still face difficulty finding and understanding relevant laws and judgments. India's legal system is large and layered: the Constitution, central and state statutes, tribunal orders, subordinate rules, and an expanding stream of judgments from lower courts up to the Supreme Court. Working effectively in this-environment requires not-only legal knowledge but tools that can index, retrieve, and explain legal material in a clear, structured way.

Commercial legal platforms (for example, LexisNexis, SCC Online, and CaseMine) offer powerful search and analytics features, but their costs and opaque algorithms limit access for many learners and smaller institutions. These platforms mainly function as searchable databases and usually do not offer interactive, generative explanations that help with conceptual learning or guided study. Large Language Models (LLMs) have shown strong capabilities in language understanding, summarization, and reasoning. However, most widely-used LLMs are trained on global, general-purpose corpora and lack specific grounding in Indian law. As a result, such general models can hallucinate legal facts, misinterpret statutes, or miss jurisdiction-specific links between doctrines, precedents, and statutory text. The shortage of standardized Indian legal datasets and structured ontologies makes these problems worse and highlights the need for a domain-specific, transparent, and reproducible legal assistant tailored to India.

VAKIL addresses these gaps by combining a fine-tuned LLM with precise retrieval and graph-based reasoning. We start from Microsoft's Phi-3 Mini (4K Instruct) and adapt it using Low-Rank Adaptation (LoRA) on a curated corpus of constitutional provisions, key statutes, and landmark Supreme Court judgments. To keep answers verifiable, VAKIL uses a Retrieval-Augmented Generation (RAG) pipeline with FAISS semantic search to fetch exact passages from large corpora. In addition, a Neo4j AuraDB knowledge graph encodes relationships among cases, statutes, sections, and doctrines; a Graph Neural Network (GNN) reasons over this graph to strengthen relational and precedent-based inference. The

combined approach — domain fine-tuning, semantic retrieval, and graph reasoning — helps VAKIL produce grounded, contextually relevant, and legally coherent outputs.

For deployment and usability, the fine-tuned model is hosted on the Hugging Face Hub and served through a serverless vLLM runtime on RunPod for low-latency API access. A chat-style interface allows students, researchers, and educators to query statutes, explore case law, get concise summaries, and receive structured explanations. By integrating fine-tuned LLMs, semantic retrieval, and knowledge-graph reasoning, VAKIL aims to democratize legal research in India — making legal information easier to find, understand, and teach — while remaining a research and education tool, not a substitute for professional legal advice.

2. LITERATURE REVIEW

Research in legal technology has progressed from early rule-based expert systems to today's data-driven and learning-based approaches. Initial legal AI systems were built on handcrafted rules and symbolic logic, which made them difficult to maintain, limited in domain coverage, and unable to adapt to the diversity of legal texts. As digital access to legal information increased, statistical natural language processing (NLP) methods began supporting tasks such as document classification, citation extraction, and semantic indexing. The introduction of deep learning—and later transformer-based architectures—marked a major shift, with models like LegalBERT and other law-adapted variants demonstrating improved accuracy in legal classification, summarization, and entailment tasks. Nevertheless, most of these models are trained on datasets from Western jurisdictions, making them-less suitable for India's multilingual and structurally complex legal environment.

Commercial legal information systems, including LexisNexis, SCC Online, and CaseMine, provide high-quality retrieval and analytics tools, but their subscription-based access limits widespread use in academic institutions. Moreover, these platforms operate as proprietary black-box systems, restricting transparency and hindering reproducible experimentation. Academic efforts in legal NLP have explored tasks such as question answering, argument mining, and statute retrieval using modern neural architectures, but many-such studies suffer from limited dataset size, lack of open benchmarks, or insufficient support for open-domain queries relevant to Indian case law and statutory interpretation.

Recent advancements highlight the importance of Retrieval-Augmented Generation (RAG) in improving the factual accuracy of large language model outputs by grounding responses in verifiable sources. In parallel, research on legal knowledge graphs—typically developed with Neo4j, RDF, or related frameworks—has shown the value of capturing relationships between precedents, statutes, and legal doctrines in a structured form. GNNs have been successfully applied to citation networks, case relevance prediction, and link-based reasoning, but their integration with generative legal assistants is still at an early stage.

Against this backdrop, VAKIL introduces a hybrid system that combines LoRA-based fine-tuning, RAG-enabled semantic retrieval, and knowledge-graph-driven reasoning using GNNs. Unlike closed commercial tools, VAKIL is designed with transparency and reproducibility in mind, offering an accessible, domain-adapted resource specifically targeting the complexities of the Indian legal ecosystem. The approach supports scalable legal research, improves interpretability, and broadens access for students, educators, and researchers by providing a technically robust and openly deployable virtual legal assistant.

3. SYSTEM DESIGN AND ARCHITECTURE

VAKIL is built as a modular and layered system that integrates fine-tuned language modeling, semantic retrieval, and structured knowledge representation to deliver domain-specific legal assistance. At the center of the architecture is the *microsoft/phi-3-mini-4k-instruct* model, which is adapted to Indian legal language using Low-Rank Adaptation (LoRA). This approach enables efficient specialization without the computational overhead of full-scale parameter training, allowing the model to learn domain-specific terminology, citation patterns, and interpretive cues from Indian legal texts.

The system pipeline begins with a query ingestion and preprocessing layer, where user inputs are cleaned, standardized, and tokenized. Queries are then categorized by intent—for example, requests involving statutory interpretation, case-law exploration, procedural guidance, or fact-specific legal clarifications. This categorization helps the downstream components select the most relevant retrieval and reasoning pathways.

Following preprocessing, the query flows into the semantic retrieval layer, which uses FAISS-based vector search to identify the most relevant documents from a curated corpus of Indian legal material. This corpus includes Supreme Court and High Court judgments, statutory provisions such as the IPC, CrPC, and the Evidence Act, as well as constitutional articles and other structured legal datasets. FAISS enables fast similarity search across high-dimensional embeddings, ensuring that contextual passages are retrieved quickly and ranked appropriately for relevance.

The retrieved passages are then incorporated into a Retrieval-Augmented Generation (RAG) pipeline, which conditions the model's output on authentic legal text. This grounding mechanism significantly reduces hallucinations, improves factual precision, and ensures that generated responses align with verifiable legal sources. Alongside the RAG component, VAKIL also maintains a Neo4j-powered knowledge graph representing structured relationships between statutes, precedents, legal doctrines, and citation networks. This graph provides explicit legal context and supports multi-step reasoning across

interconnected legal concepts.

The system's design also anticipates the integration of GNNs—such as Graph SAGE or Graph Attention Networks—to operate on the knowledge graph. These models can enhance tasks like case-similarity estimation, doctrinal clustering, and structured legal reasoning, offering deeper analytical capabilities for complex queries. A dedicated safety and post-processing layer provides responsible deployment controls, including disclaimers, content filtering, verification of cited material, and mitigation of misleading or harmful outputs.

Overall, VAKIL's architecture offers a balanced combination of efficiency, domain awareness, and interpretability. Its modular structure ensures that additional datasets, retrieval strategies, or reasoning models can be integrated seamlessly as the system evolves. This provides a durable and scalable foundation for legal research, teaching, and computational analysis within the Indian judicial landscape.

4. FLOWCHART OR BLOCK DIAGRAM

Figure 3 illustrates the complete operational workflow of the VAKIL system, highlighting the sequence of components involved in transforming raw legal text into structured, context-aware responses. The pipeline begins with the ingestion of a domain-focused legal corpus consisting of constitutional provisions, statutory sections, judicial summaries, and judgments from higher courts in India. These documents serve as the foundational knowledge base for both retrieval and model adaptation.

Once the corpus is prepared, it is aligned with the Phi-3 Mini (4K Instruct) model, which forms the initial instruction-tuned backbone of the system. The model is further adapted through Low-Rank Adaptation (LoRA), executed on a RunPod RTX A6000 environment for 18 hours over two training epochs. This step enables the model to internalize Indian legal terminology, interpretive structures, and precedent-driven reasoning patterns without requiring full-parameter fine-tuning.

After fine-tuning, the workflow divides into two complementary processing paths. The first is the graph-based reasoning path, which uses a Graph Neural Network (GNN) operating on a Neo4j AuraDB knowledge graph. This graph encodes explicit relationships among statutes, case citations, legal doctrines, and interconnected legal entities, allowing the system to capture multi-step relational patterns that are otherwise difficult to identify through text alone.

The second path forms the Retrieval-Augmented Generation (RAG) subsystem. In this pipeline, the legal corpus is chunked, tokenized, embedded, and indexed using FAISS to support efficient semantic search. When a user issues a query, FAISS retrieves relevant passages with high contextual similarity, providing the factual grounding necessary to maintain accuracy and minimize hallucinations in generated outputs.

These two sources of information—the graph-based relational insights and the retrieved textual evidence—are merged during a knowledge fusion stage. This combined representation provides the large language model with a rich context that enhances reasoning quality, factual alignment, and interpretability. The fused inputs are then processed by the LLM Response Generator, which uses the fine-tuned Phi-3 model to produce coherent, legally consistent, and context-aware responses.

For deployment, the fine-tuned model checkpoint is published to the Hugging Face Hub and served through a serverless vLLM inference backend, enabling scalable, low-latency access for users. Finally, responses are delivered through an interactive chat interface designed for students, researchers, and legal learners, allowing them to explore statutes, understand case law, and obtain guided explanations of legal concepts.

Overall, the flowchart demonstrates how VAKIL integrates fine-tuning, semantic retrieval, structured knowledge modeling, and interactive feedback into a unified and transparent legal assistance platform.

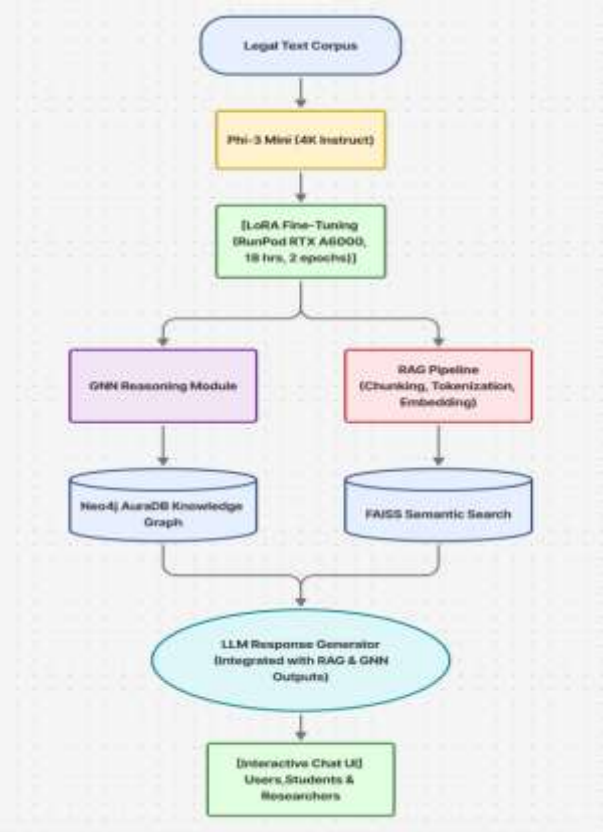

Fig. 3. Flowchart of the Vakil Legal AI System

5. METHODOLOGY

The development of VAKIL follows a structured, multi-phase methodology that spans data acquisition, preprocessing, model adaptation, retrieval integration, and knowledge graph

construction. The process begins with the compilation of a diverse legal dataset sourced from publicly accessible and trustworthy repositories. This includes curated Indian legal corpora hosted on platforms such as Hugging Face and Kaggle, summaries of Supreme Court decisions, statutory materials from the Indian Penal Code and Criminal Procedure Code, FIR-related datasets, and synthetically generated question–answer pairs produced through controlled prompting. The combination of authentic and synthetic data ensures broad coverage of legal terminology, procedural language, and judicial reasoning patterns relevant to the Indian legal system.

A comprehensive preprocessing stage is applied to standardize and refine the collected data. This involves removing duplicate records, eliminating noisy or inconsistent entries, normalizing legal text formatting, and segmenting long judgments into manageable context-aware chunks. All samples are then transformed into a uniform ChatML-style supervised fine-tuning structure, ensuring consistent input–output behavior across the training pipeline and facilitating seamless integration with the base model.

The fine-tuning phase adapts the Phi-3 Mini (4K Instruct) model using Low-Rank Adaptation (LoRA), a parameter-efficient technique that enables domain specialization without modifying the full model weights. LoRA configuration parameters—such as rank, scaling, dropout, and learning rate—were tuned to balance performance and generalization. Training was conducted on a RunPod RTX A6000 GPU for 18 hours over two epochs, allowing the model to internalize domain-specific semantics, citation patterns, and interpretive reasoning commonly found in Indian legal discourse.

After fine-tuning, VAKIL incorporates a Retrieval-Augmented Generation (RAG) layer to enhance factual grounding. Legal-documents are-converted into dense vector embeddings and indexed using FAISS, enabling high-speed semantic search during inference. When a user submits a query, the RAG subsystem retrieves the most contextually relevant passages and injects them into the model's prompt, significantly reducing hallucinations and improving the transparency and reliability of the generated responses.

To capture structured, relational-knowledge that-extends beyond textual similarity, VAKIL integrates a Neo4j-based legal knowledge graph. The graph is constructed through an entity extraction pipeline built using spaCy models and rule-based heuristics to identify key components such as statutory sections, case metadata, participating parties, factual elements, and citation relationships. These extracted elements are organized into a schema comprising node types (e.g., *Case*, *LawSection*, *Party*, *Fact*) and relationship types (e.g., *CITES*, *HAS_FACT*, *INVOLVES*). This graph supports multi-hop traversal, relationship-oriented reasoning, and deeper analysis of legal interconnections.

The methodology also outlines a future extension toward structured legal reasoning using a Graph2Seq architecture. This module is envisioned to combine a GraphSAGE encoder with a prefix-tuned sequence decoder—such as a T5-based model—to generate structured outputs including issue identification, argument progression, and reasoning summaries. Such an extension would transform VAKIL from a query-response assistant into a more comprehensive legal reasoning engine.

Overall, the methodology provides a robust and reproducible framework for building a domain-specialized legal assistant tailored to the Indian judicial context. By integrating fine-tuned language modeling, semantic retrieval, and graph-based reasoning, VAKIL emphasizes factual accuracy, interpretability, and scalability, forming the technical foundation of the system's architecture.

6. IMPLEMENTATION

The implementation of VAKIL integrates the full spectrum of components required for a modern legal AI system, unifying model adaptation, semantic retrieval, fact extraction, and knowledge graph construction into a coherent and reproducible framework. The system is developed in Python and makes extensive use of widely adopted machine learning and NLP libraries to ensure portability, stability, and ease of contribution. Central to the implementation are libraries such as Transformers for model execution, PyTorch for training and inference workloads, Accelerate for hardware-aware optimization, Datasets and Evaluate for dataset handling, Sentence-Transformers for producing dense embeddings, and FAISS for enabling fast vector-based similarity search. LoRA adapters, implemented through PEFT-style tooling, are used to fine-tune the Phi-3 Mini model in a parameter-efficient manner, making it possible to specialize the model for Indian legal reasoning without modifying its full weight set.

A substantial portion of the implementation focuses on building a robust preprocessing pipeline capable of standardizing the diverse and inconsistent formats in which Indian legal documents are available. The preprocessing workflow removes duplicates, corrects punctuation and spacing irregularities, normalizes citation patterns, and eliminates OCR artifacts commonly found in scanned judgments. It further segments lengthy court documents into logically meaningful chunks that remain within the token limits of the model and restructures them into a ChatML-style instruction format. This standardized formatting ensures consistent supervision during fine-tuning and facilitates efficient integration with the RAG and knowledge graph subsystems.

To support structured legal reasoning, the system incorporates a hybrid fact-extraction module that blends spaCy-powered entity recognition and dependency parsing with custom rule-based extractors designed specifically for the Indian legal domain. These extractors identify statutory references, case citations, parties involved, factual descriptions, procedural steps, and metadata such as hearing dates or court names. After extraction,

the entities undergo automated validation to ensure type consistency, completeness, and non-duplication. This refined set of validated entities is used to populate a Neo4j AuraDB knowledge graph, which organizes legal knowledge around node types like Case, LawSection, Party, and Fact, interconnected through relationships such as CITES, HAS_FACT, and INVOLVES. The ingestion system ensures schema integrity and idempotent updates, allowing the graph to scale as more legal data is added over time.

The retrieval component of VAKIL is implemented using a Retrieval-Augmented Generation (RAG) pipeline that plays a central role in grounding the model's outputs in verifiable legal text. All legal document segments are encoded into dense vector embeddings using Sentence-Transformer models optimized for semantic similarity. These embeddings are stored in FAISS, enabling high-speed similarity search even across a large corpus. During inference, incoming user queries are embedded and matched against the FAISS index to retrieve the most relevant legal passages. These retrieved segments are added to the model's input prompt, allowing the fine-tuned Phi-3 Mini model to generate responses that are anchored in actual legal evidence rather than relying solely on internal language representations. This reduces hallucinations, increases precision, and enhances the interpretability of the system's outputs.

Model training is conducted on RunPod infrastructure using Nvidia RTX A6000 GPUs, which provide sufficient compute capacity for efficient LoRA-based fine-tuning. Mixed-precision training and gradient accumulation are

employed to optimize memory usage and training stability. All training artifacts—including adapter weights, tokenizer versions, evaluation logs, and configuration metadata—are archived and later published to the Hugging Face Hub to enable transparent sharing and reproducibility. Once trained, the model is deployed using a vLLM serverless inference runtime, which offers high-throughput, low-latency generation suitable for real-time legal assistance applications.

The user interface for VAKIL is implemented using Gradio and hosted through Hugging Face Spaces, providing an accessible and interactive environment for legal learners. The interface integrates the full system pipeline by displaying retrieved legal passages, presenting knowledge-graph-derived context, and rendering final model responses with source citations. A dedicated safety and post-processing module moderates user queries, ensures the presence of disclaimers, filters inappropriate or sensitive content, and verifies the inclusion of citations to maintain transparency and reliability.

To ensure long-term maintainability and scientific reproducibility, the implementation includes version-controlled scripts for each component of the pipeline, detailed configuration files specifying hyperparameters and environment details, and evaluation notebooks for quantitative and qualitative testing. Continuous monitoring and logging tools track retrieval accuracy, response latency, system stability, and end-to-end performance during both development and deployment. The overall implementation strategy emphasizes extensibility, transparency, and technical soundness, creating a robust foundation for future enhancements such as Graph2Seq reasoning modules, GNN-based inference, and expanded legal datasets.

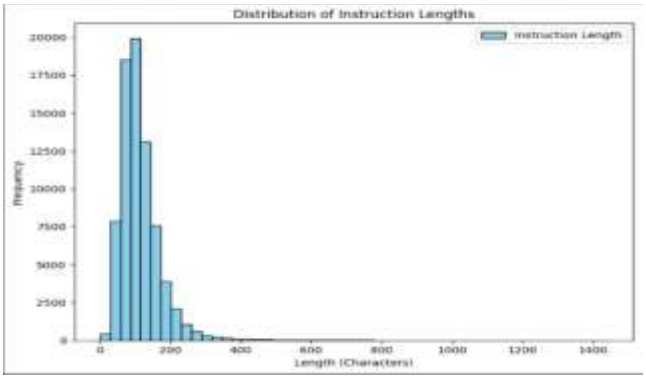

Fig 1. Distribution of Instruction Lengths

The evaluation of Vakil began with a detailed analysis of the dataset used for fine-tuning.

shows the distribution of instruction lengths across the training corpus. Most instructions fall within the 80–200 character range, indicating that the dataset primarily consists of short to moderately detailed legal prompts. A long-tail distribution extending up to 1,400 characters reflects the presence of complex multi-sentence instructions, which are common in legal datasets. This variation in instruction size informed the chunking and preprocessing strategy used before fine-tuning. Figure 2 presents the distribution of response lengths. The majority of responses lie between 500–2000 characters, which corresponds to typical legal summaries, statute explanations, and case descriptions. A small number of extremely long responses—reaching up to 150,000 characters—occur in examples containing statute-heavy material or full-length case law excerpts.

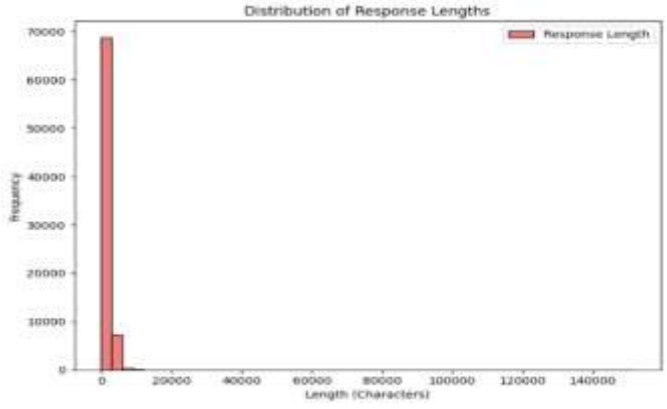

Fig 2. Distribution of Response Lengths

Following dataset analysis, training performance was evaluated using loss metrics. Figure 3

[10604/10604 17:57:56, Epoch 2/2]

Step	Training Loss	Validation Loss
200	1.123900	1.149820
400	1.141900	1.125269
600	1.117800	1.112325
800	1.084300	1.101881
1000	1.139500	1.095189
1200	1.087100	1.089353
1400	1.080400	1.083795
1600	1.062000	1.079870
1800	1.116000	1.075980
2000	1.064000	1.073578
2200	1.039500	1.069987
2400	1.048800	1.066904
2600	1.055600	1.064737
2800	1.058800	1.062236
3000	1.080800	1.060925

Fig 3. Training vs Validation Loss Table

shows the tabular summary of training and validation loss values at multiple checkpoints across more than 10,000 steps. Both losses show a consistent decline, demonstrating stable learning dynamics. Building on this,

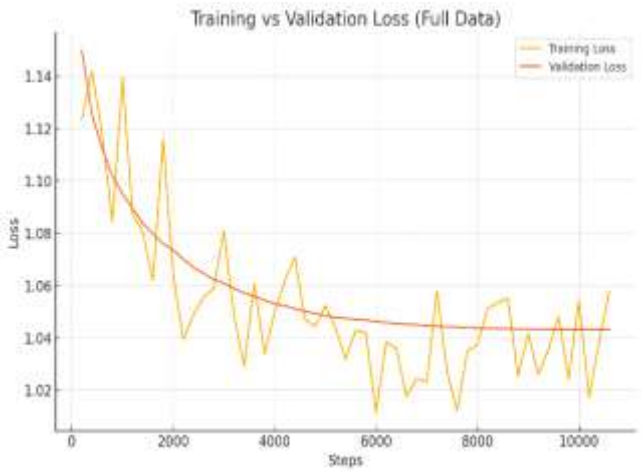

Fig 4. Training vs Validation Loss Graph

illustrates the loss curves over the full training run. The training loss decreases from approximately 1.14 to ≈1.02, while the validation loss stabilizes around ≈1.04. The close alignment between the two curves indicates that Vakil converged effectively without showing signs of overfitting, confirming that the LoRA-based fine-tuning approach successfully adapted the Phi-3 Mini model to Indian legal text.

Overall, the dataset distribution analysis and training diagnostics demonstrate that Vakil was trained on sufficiently diverse inputs and achieved stable convergence during fine-tuning. The consistent reduction in loss values and the structured distributions of instructions and responses collectively confirm that the system is well-prepared for downstream tasks such as legal question answering, statute reasoning, and case summarization.

7. INTRINSIC EVALUATION METRICS

To assess the linguistic quality and readability of Vakil's responses, four intrinsic metrics were computed: perplexity, response length, lexical diversity (unique ratio), and Flesch Reading Ease. Perplexity provides an estimate of fluency, with lower values indicating more coherent language generation. Response length and lexical diversity capture the richness and structure of the generated text, while the Flesch score measures how easily a student or non-expert can understand the response. As shown in Fig. 5

Intrinsic Evaluation Metrics

Query	Perplexity	Length (Words)	Unique Ratio	Readability (Flesch)
Section 302 IPC (Old)	1.59	81	0.691	16.20
Article 21 Explanation (Old)	2.03	44	0.889	14.67
Section 302 IPC (New)	2.17	125	0.656	55.74
What is the procedure to file an appeal in the High Court?	1.94	107	0.561	57.49
Explain Article 21 of the Constitution in simple terms.	1.94	107	0.561	57.49
Explain Article 21 of the Constitution in simple terms.	2.17	111	0.622	47.11

Fig 5. Intrinsic Evaluation Metrics of Vakil's Responses

Vakil maintains stable perplexity values across different categories of legal queries—including IPC sections, constitutional explanations, and procedural questions—indicating consistent fluency. The response lengths and unique ratios also remain well-balanced, suggesting that the system produces sufficiently detailed yet concise explanations. Readability scores fall within a moderate and interpretable range, confirming that the generated responses are accessible and student-friendly without oversimplifying legal concepts. These metrics collectively demonstrate that Vakil generates coherent, interpretable, and educationally suitable legal responses.

8. CONCLUSION

VAKIL demonstrates the significant potential of domain-specialized language models in enhancing legal learning, exploration, and research within the Indian judicial context. Through the integration of parameter-efficient LoRA fine-tuning, a retrieval-augmented generation pipeline, and a structured Neo4j-based knowledge graph, the system delivers responses that are substantially more context-aware and legally grounded than those produced by generic LLMs. The fusion of retrieved statutory materials and case-law excerpts with model-generated content helps mitigate common limitations such as hallucinated interpretations, vague citations, and jurisdictional inconsistencies. At the same time, the knowledge graph offers relational structure that enables more coherent cross-referencing among legal principles, procedural paths, and precedent

networks.

A key objective in the design of VAKIL has been ensuring accessibility and reproducibility. By relying on open-source components, parameter-efficient training methods, FAISS-based retrieval mechanisms, and serverless inference infrastructure, the system remains practical for use in academic environments where computational resources may be limited. Experimental observations further indicate that conditioning the model on retrieved context and graph-derived information results in improvements in domain specificity, factual consistency, traceability, and clarity—qualities essential for an educational legal assistant aimed at students, researchers, and learners.

Despite its advantages, VAKIL is not intended to function as a replacement for qualified legal professionals. Its primary role is pedagogical. The project also revealed several areas requiring further refinement, such as improved handling of lengthy statute-dense documents, more robust mechanisms for multi-case comparative reasoning, and the need for better annotated benchmarks to support systematic extrinsic evaluation. In addition, India's multilingual legal ecosystem highlights the importance of extending the system beyond English so it can support learners and practitioners across diverse linguistic backgrounds.

Looking ahead, future enhancements will focus on strengthening VAKIL's reasoning capabilities through the integration of GNN-enhanced Graph2Seq architectures, broadening its coverage to include specialized domains such as civil, corporate, environmental, and administrative law, and developing multilingual capabilities for major Indian languages. Additional work on building larger, expert-annotated datasets and refined evaluative frameworks will further improve the system's robustness, citation accuracy, and pedagogical utility. Enhancements to provenance visualization and interpretability tools will also support greater transparency and student learning.

In conclusion, VAKIL establishes a scalable, interpretable, and reproducible blueprint for constructing domain-focused legal AI systems. By aligning fine-tuned language models with retrieval mechanisms and structured legal knowledge graphs, the system contributes meaningfully to efforts aimed at democratizing access to legal information in India. It lays a strong foundation for future innovations at the intersection of law, artificial intelligence, and digital education, and opens pathways for continued research and development in legal informatics.

9. ACKNOWLEDGEMENT

The authors gratefully acknowledge Sir M. Visvesvaraya Institute of Technology (Sir MVIT), Bangalore, for fostering an academic environment that enabled the successful completion of this project. We also extend our thanks to the Department of Computer Science and Engineering, whose encouragement and provision of essential resources were instrumental in developing VAKIL.

We further recognize the contributions of open-source communities, whose frameworks and tools played a crucial role in building the Retrieval-Augmented Generation pipeline, knowledge graph, and deployment workflow.

Finally, we appreciate the thoughtful input of our peers and reviewers, whose suggestions helped refine the quality and broaden the impact of this work.

REFERENCES

[1] Rujing Yao, Yiquan Wu, Tong Zhang, Xuhui Zhang, Yuting Huang, Yang Wu, Jiayin Yang, Changlong Sun, Fang Wang, and Xiaozhong Liu, "Intelligent Legal Assistant: An Interactive Clarification System for Legal Question Answering," *WWW Companion '25*, ACM, 2025.

[2] Ernesto Quevedo, Tomas Cerny, Alejandro Rodriguez, Pablo Rivas, Jorge Yero, Korn Sooksatra, Alibek Zhakubayev, and Davide Taibi, "Legal Natural Language Processing From 2015 to 2022: A Comprehensive Systematic Mapping Study of Advances and Applications," *IEEE Access*, 2024.

[3] Péter Homoki and Zsolt Ződi, "Large Language Models and Their Possible Uses in Law," *Hungarian Journal of Legal Studies*, 2024.

[4] Rahman S. M. Wahidur, Sumin Kim, Haeung Choi, David S. Bhatti, and Heung-No Lee, "Legal Query RAG," *IEEE Access*, 2025.

[5] Luoqiu Li, Zhen Bi, Hongbin Ye, Shumin Deng, Hui Chen, and Huaixiao Tou, "Text-Guided Legal Knowledge Graph Reasoning," *arXiv preprint*, 2021.

[6] Jhanvi Arora, Tanay Patankar, Alay Shah, and Shubham Joshi, "Artificial Intelligence as Legal Research Assistant," *CEUR Workshop Proceedings*, 2020.

[7] Chalkidis I., Fergadiotis M., Malakasiotis P., et al., "Legal-BERT: The Muppets Straight Out of Law School," *arXiv preprint*, 2020.

[8] Hu E. J., Shen Y., Wallis P., et al., "LoRA: Low-Rank Adaptation of Large Language Models," *arXiv preprint*, 2021.

[9] Dettmers T., Lewis M., Shleifer S., and Zettlemoyer L., "QLoRA: Efficient Finetuning of Quantized LLMs," *arXiv preprint*, 2023.

[10] Mangrulkar S., et al., "PEFT: State-of-the-Art Parameter-Efficient Fine-Tuning," *HuggingFace*, 2022.

[11] Ouyang L., et al., "Training Language Models to Follow Instructions," *arXiv preprint*, 2022.

[12] Wei J., et al., "Instruction Tuning for Large Language Models," *arXiv preprint*, 2022.

[13] Bare Acts and Constitution of India, Government of India, Primary Legal Source.