

Evaluating BERT and RAG Frameworks for Legal Information Retrieval and Reasoning

Ayesha Maniyar
KLE Technological University
Hubballi, India
01fe23bci007@kletech.ac.in

Michelle Hoolgeri
KLE Technological University
Hubballi, India
01fe23bci042@kletech.ac.in

Madhav Nidagundi
KLE Technological University
Hubballi, India
01fe23bci036@kletech.ac.in

J Vijay Kumar
KLE Technological University
Hubballi, India
01fe23bci083@kletech.ac.in

Uday Kulkarni
KLE Technological University
Hubballi, India
uday_kulkarni@kletech.ac.in

Pooja Gani
KLE Technological University
Hubballi, India
poojagani4@gmail.com

Abstract—This paper provides a comparative analysis of transformer encoders and Retrieval-Augmented Generation (RAG) for the retrieval and reasoning of Indian legal information. We compare four domain-adapted BERT models—LegalBERT, Lawformer, InLegalBERT, and CaseLawBERT—with five RAG variants: Self-RAG, CRAG, KRAG, Iterative RAG, and a unified Self-Correcting RAG. The viber1/indian-law-dataset has 24,607 instruction-response pairs taken from laws and court cases that were used in the experiments. We use sliding-window chunking within the 512-token limit and FAISS to index chunk embeddings for top-k passage retrieval in order to keep long-context evidence. The results show that transformer encoders give strong and stable classification baselines, while RAG models give citation-based answers that are more consistent with the facts and easier to understand. KRAG gets the best generation quality by integrating knowledge in a structured way, and the SelfCorrecting RAG controller makes retrieval more reliable by checking evidence based on feedback. Qualitative analyses demonstrate that encoder-only outputs are frequently relevant but lack citations, whereas retrieval-augmented systems correlate claims with authoritative sources. Overall, combining retrieval with generation makes facts more solid, easier to understand, and easier to trace back to the law. This is a practical way to get reliable AI help for research and writing in Indian law. Bidirectional Encoder Representations from Transformers, Indian law, Large Language Models, Natural Language Processing, Retrieval Augmented Generation.

Index Terms—Bidirectional Encoder Representations from Transformers, Indian law, Large Language Models, Natural Language Processing, Retrieval Augmented Generation.

I. INTRODUCTION

The legal system is the cornerstone of social order and governance because it enables regulation, justice, and the settlement of disputes. Contracts, laws, and case laws that need to be carefully interpreted are among the many pieces of information needed in this field [1]. Language and cultural differences, court inconsistencies, and case backlogs are issues in places like India. As a result of the increased volume of data brought about by the digitization of laws and court decisions, it became necessary to employ computational tools in order to efficiently search, classify, and interpret legal texts. This laid the groundwork for the application of artificial

intelligence (AI) in the legal domain [1]–[3], which provides systematic, data-driven assistance to improve research and decision-making.

Previous legal AI applications [1] used rule-based systems for analytics and decision-making. The difficulties associated with uncertainty and unstructured data prompted the development of Machine Learning (ML) [2]–[4] methods for classification, prediction, and analysis. Semantic retrieval techniques like Document to Vector + Support Vector Machine (Doc2Vec + SVM) [5] and Bidirectional Encoder Representations from Transformers + Best Matching 25 (BERT + BM25) [6] hybrids were among these approaches. Neural networks were used by Deep Learning (DL) [2], [7], [8] techniques to improve document classification and summarization. Domain-specific models like LEGAL-BERT [7], CaseLawBERT [10], and Lawformer [11] enhanced semantic comprehension, while Generative AI (Gen AI) [9], which included Large Language Models (LLMs) [2], enabled context-aware content production and legal reasoning.

Gen AI further revolutionized legal natural language processing by empowering models to comprehend and generate logical, context-aware legal content. Generative Pre-trained Transformer 3 (GPT-3) and other LLMs showed impressive generalization and reasoning skills when drafting, summarizing, and responding to open-ended legal questions. To address the specialized nature of legal text, domain-specific models were developed, such as LEGAL-BERT [7], CaseLawBERT [10], Lawformer [11], and InLegalBERT [15]. By providing more contextual representations that are suited to legal semantics, these models increased the precision of information extraction and classification. Despite being static and reliant on prior knowledge, pretrained models offer strong linguistic representations [16].

The methods for retrieving legal texts can be divided into three categories, as illustrated in Fig. 1, which shows the progression from rule-based search to representation learning and retrieval-augmented generation: traditional models, pre-trained models, and RAG-based models. Context-aware and

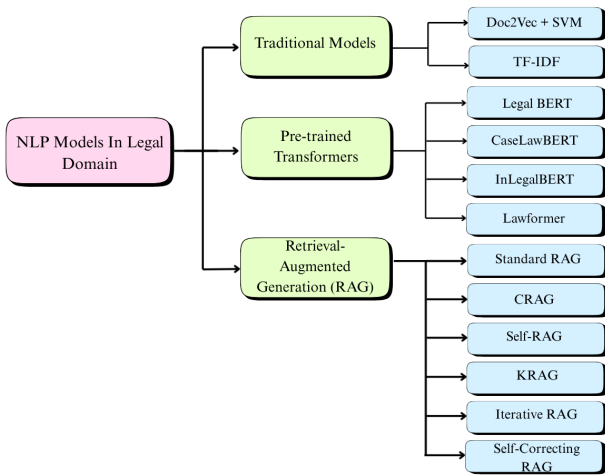


Fig. 1. Taxonomy of NLP Models in the Legal Domain [12].

knowledge-grounded outputs are made possible by the RAG-based models, which integrate retrieval and generation.

RAG frameworks have been developed to support improved legal NLP capabilities by combining generative models with external document retrieval [12]. This method improves factual reliability and explainability in legal applications by guaranteeing that outputs are based on reputable sources like statutes or precedent cases [13], [14]. While RAG combines reasoning with evidence-based generation, BERT-based architectures offer fundamental semantic understanding. BERT and RAG are important technologies in developing reliable AI for the legal field because they combine representation learning and knowledge-grounded generation.

Building upon this framework, we have evaluated four BERT-based models designed for legal text comprehension on the Indian legal dataset. Of these, InLegalBERT outperforms Lawformer by roughly 2.33 and 2.45 percent, achieving the strongest baseline with accuracy 0.9623 and F1 0.9644. The most reliable end-to-end configuration among the RAG variants is Self-Correcting RAG, which shares the top ROUGE-1 score of 0.8000 and achieves the highest retrieval effectiveness with an F1 of 0.5333, indicating strong factual alignment. With the highest BLEU score of 47.6870 and the best ROUGE-L performance, Krag exhibits exceptional generation quality. While Iterative RAG performs relatively poorly, suggesting room for improvement in multi-step refinement, Self-RAG offers balanced results across retrieval and generation.

This is how this paper is structured. The legal domain background and relevant literature on domain-specific language models and retrieval techniques, including BERT-based and RAG-based methods, are reviewed in Section II. The methodology, including the Self-Correcting RAG framework and InLegalBERT architecture and workflow, as well as the setup for training and evaluation, is described in Section III. Using metrics for classification, retrieval, and generation, results for specific BERT and RAG variants are shown in Section IV. The conclusion is given and future work is outlined

in Section V. The paper’s references are listed at the end.

II. LITERATURE SURVEY

The statistical and rule-based methods that were the mainstay of early legal text processing research were unable to adequately capture the semantic and contextual complexity of legal language. Legal NLP has been transformed by the introduction of Transformer-based models, especially BERT and its domain-specific variations, which allow for a profound contextual understanding of contracts, statutes, and case law. To tackle issues like legal terminology, document length, jurisdictional variation, and retrieval efficiency, several prominent models have been developed, including LegalBERT [7], CaseLawBERT [10], Lawformer [11], and InLegalBERT [15]. These models’ contributions to the development of legal-domain natural language understanding are reviewed in the ensuing subsections.

1) *LegalBERT*: LegalBERT [7] is pretrained with a masked language modeling objective on extensive legal corpora. The model learns contextual token embeddings by predicting masked tokens based on surrounding context, as equation (1) illustrates.

$$L_{MLM} = - \sum_{i \in M} \log P(x_i | x_{\setminus M}) \quad (1)$$

In this case, $x_{\setminus M}$ stands for the unmasked context, and M indicates the set of masked tokens. Fig.2, which depicts the architecture of LegalBERT, demonstrates how domain-specific pretraining improves contextual embeddings for legal language comprehension [7].

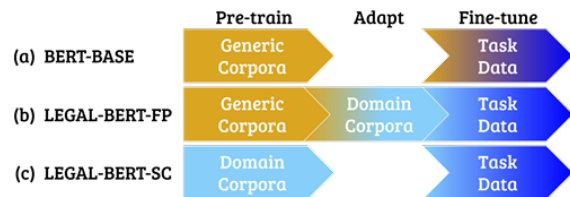


Fig. 2. LegalBERT Architecture — Domain-specific pretraining on legal corpora enhances contextual embeddings [7].

2) *CaseLawBERT*: By concentrating on court rulings from the Harvard Case Law Corpus, CaseLawBERT [10] expands on LegalBERT. In order to support precedent linkage and hierarchical reasoning across lengthy documents, it integrates contextual citation modeling and paragraph-level embeddings. Fig.3 illustrates the process. Superior performance in case entailment, precedent retrieval, and legal judgment prediction is made possible by this design [10].

3) *Lawformer*: A Longformer-based architecture is presented by Lawformer [11] to manage legal documents that are larger than the 512-token limit of BERT. Equation (2) illustrates a sliding-window attention mechanism that enables the model to maintain global dependencies while concentrating on local context.

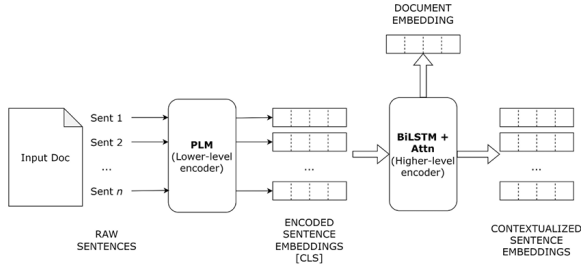


Fig. 3. CaseLawBERT Workflow — Pretraining on court rulings facilitates reasoning across citation chains and case hierarchies [10].

$$h_i = \text{Attention}(Q_i, K_{i-w:i+w}, V_{i-w:i+w}) \quad (2)$$

Here, w defines the local attention window. The combined local and global attention mechanism is illustrated in Fig.4, demonstrating Lawformer’s ability to process long legal texts efficiently [11].

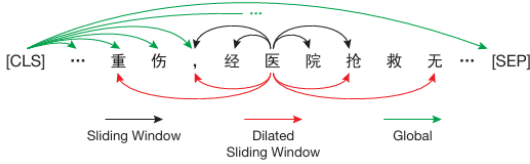


Fig. 4. Lawformer Attention Mechanism — Combines local sliding-window and global token attention for processing long legal texts [11].

4) *InLegalBERT*: InLegalBERT [15] extends LegalBERT for multilingual and jurisdiction-aware legal understanding. It is pretrained on diverse Indian legal corpora, including judgments, acts, and constitutional articles. Its retrieval-augmented fine-tuning framework allows the model to retrieve relevant legal precedents using contextual embeddings, improving performance on query-driven legal analysis tasks. The workflow is shown in Fig.5 [15].

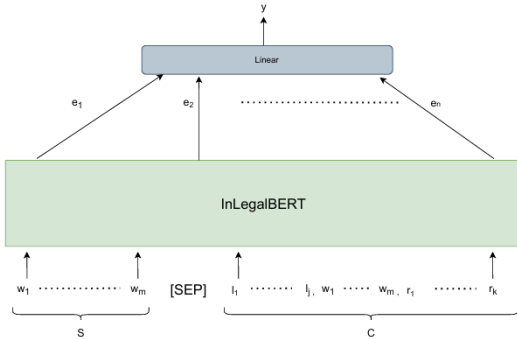


Fig. 5. InLegalBERT Workflow — Combines document retrieval with transformer-based contextual reasoning for domain-grounded legal analysis [15].

The BERT-based models discussed above provide strong contextual representations for statutes, case laws, and contracts, performing well for classification, extraction, and analysis tasks. In many legal workflows, however, it is equally important that answers be accompanied by explicit citations to authoritative sources. A complementary direction is to make use of retrieval and controlled generation so that responses remain grounded in evidence. In this context, RAG serves as a framework that integrates retrieval with generation to support citation-aligned outputs and downstream legal reasoning.

RAG [12] marked a key milestone in knowledge-intensive NLP by combining retrieval and generation in a unified framework. Unlike static transformer models, RAG couples a retriever that fetches relevant knowledge from a non-parametric memory with a generator that produces factual context-grounded responses. Given a query x , the retriever identifies a set of top- K documents $\{z_i\}$, and the generator conditions on them to produce an output y . The learning objective maximizes the marginal likelihood over retrieved evidence as shown in equation. (3).

$$\mathcal{L}_{\text{RAG}}(x, y) = -\log \sum_{i=1}^K p_{\theta}(y | x, z_i) p_{\phi}(z_i | x) \quad (3)$$

Here, $p_{\phi}(z_i | x)$ represents the retriever distribution and $p_{\theta}(y | x, z_i)$ denotes the generator likelihood. RAG thus grounds generation in external evidence, enhancing factual accuracy for open-domain question answering and knowledge reasoning.

5) *Self-RAG*: Self-RAG [19] extends RAG by introducing self-reflective retrieval. Instead of performing retrieval at fixed intervals, Self-RAG learns to decide when and what to retrieve using reflection tokens such as Is Relevant (ISREL), Is Supportive (ISSUP), and Is Useful (ISUSE). These tokens guide dynamic evidence selection during generation, allowing the model to critique and refine its own outputs. The learning objective jointly models both reflection and generation as shown in equation(4).

$$\mathcal{L}_{\text{Self-RAG}}(x, r, y) = -\log p_{\theta}(r, y | x) \quad (4)$$

where r denotes reflection decisions interleaved with text tokens. The total training objective combines reflection and generation components as in equation (5).

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{reflect}} + \lambda_2 \mathcal{L}_{\text{gen}} \quad (5)$$

This reflective mechanism enables adaptive retrieval and reduces hallucinations in generative reasoning tasks. The overall Self-RAG process is illustrated in Fig. 6.

6) *Corrective RAG (CRAG)*: To improve retrieval consistency even more, CRAG [20] presents a correction procedure that is driven by feedback. With regard to the query and output, CRAG assesses the retrieved documents for factual consistency and relevance, starting a new retrieval if evidence is found to be insufficient or contradicting. Equation (6) incorporates the correction mechanism into the objective

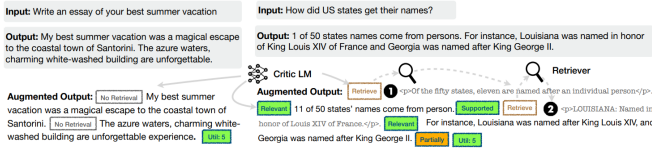


Fig. 6. Self-RAG uses reflection tokens to dynamically select and critique retrieved documents during generation, enabling adaptive evidence retrieval and reducing hallucinations [19].

$$\mathcal{L}_{\text{CRAG}} = \mathcal{L}_{\text{RAG}} + \beta E z \sim p\phi[\text{Rel}(z, x, y)] \quad (6)$$

where the relevance score of the retrieved evidence is quantified by $\text{Rel}(z, x, y)$. Fig. 7 shows the correction pipeline.

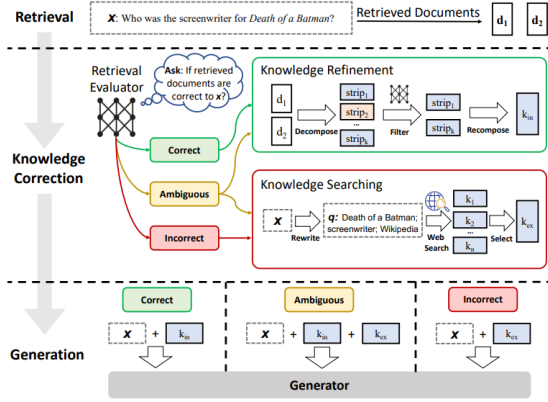


Fig. 7. To improve factual accuracy and robustness, CRAG assesses retrieval and initiates re-retrieval when evidence is insufficient or contradictory [20].

7) *Knowledge RAG (KRAG)*: By combining structured knowledge graphs with unstructured text retrieval, KRAG [27] expands on CRAG. The model can reason over relational graph structures and factual text thanks to its hybrid design. Equation (7) is used to calculate the joint embedding.

$$h = f_{\text{text}}(z_i) \oplus f_{\text{graph}}(G_i) \quad (7)$$

where

\oplus represents that text-based and graph-based feature embeddings have been combined. In law and medicine, KRAG enhances interpretability and supports challenging reasoning tasks.

8) *Iterative RAG*: Multiple refinement phases are brought about by iterative RAG methods, such as Self-Guided Iterative Calibration (SGIC) [28]. As represented by equation (8), iterative RAG keeps on retrieving, generating, and calibrating the response by feedback-guided iterations instead of generating output in one pass.

$$y^{(t+1)} = \text{Gen}(x, \text{Retrieve}(y^{(t)})) \quad (8)$$

Contextual coherence and factual accuracy are progressively improved through this iterative calibration process. Figure 8 shows the multi-step refinement flow.

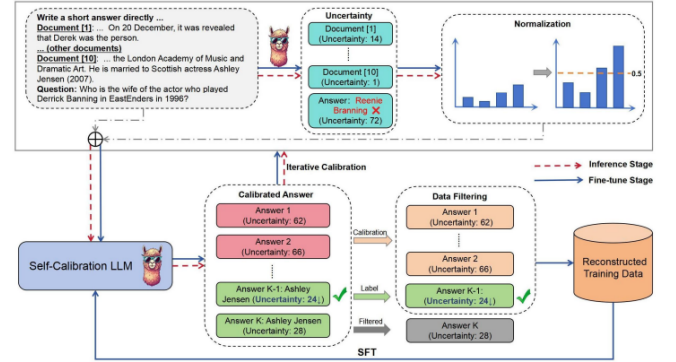


Fig. 8. Multi-step retrieval and refinement using an iterative RAG (SGIC) framework [28].

Iterative RAG multi-step refinement, Self-RAG adaptive reflection, CRAG corrective feedback, KRAG knowledge integration, and the core RAG factual grounding all illustrate the evolution of retrieval-augmented language modeling. Collectively, these approaches enhance interpretability, factual reliability, and reasoning depth as well as address challenges such as latency, retrieval noise, and joint optimizer-generator module optimization.

III. METHODOLOGY

The suggested approach combines two essential elements, which are Self-Correcting RAG and BERT. Self-Correcting RAG strengthens factual reliability through iterative retrieval, generation, and self-evaluation, while BERT extracts domain-specific linguistic and contextual representations from legal texts. Both retrieval and generation metrics, as well as domain-specific legal relevance evaluations, are used to assess the combined framework.

This section explains the overall system framework, which includes the Self-Correcting RAG model's and BERT's architecture, workflow, and evaluation approach. It also emphasizes how the combination of retrieval-augmented generation and representation learning enhances factual consistency, contextual coherence, and accuracy in jurisdiction-aware legal question answering.

A. BERT

1) *Architecture of BERT*: BERT is a deep Transformer-based language model that learns bidirectional contextual representations by jointly considering both left and right contexts of a sentence. It is pre-trained on large text corpora and later fine-tuned for specific domains. In this work, BERT is adapted to the legal domain to capture domain-specific terminology and reasoning structures.

As shown in Fig. 9, BERT processes input text beginning with the special tokens [CLS] and [SEP]. Each token is represented as a combination of token, segment, and positional

embeddings, forming the input to the Transformer encoder. These embeddings pass through multiple self-attention encoder layers, each comprising a Multi-Head Self-Attention mechanism and a Feed-Forward Network (FFN). The self-attention layers capture long-range dependencies between tokens, while the FFN refines their contextual representations. The final hidden state of the [CLS] token is used by the classification head for downstream tasks such as legal text summarization or classification.

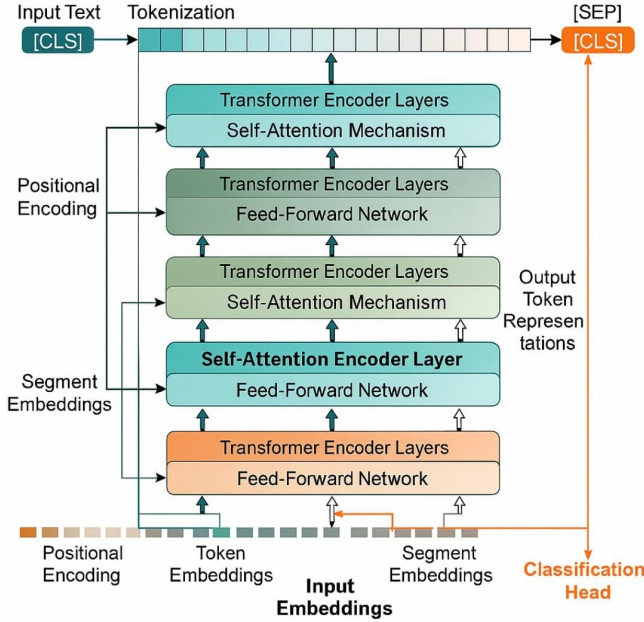


Fig. 9. Transformer layers are used by the BERT architecture to process embeddings for contextualized representations.

2) *Workflow for Legal Domain*: The process of summarization begins by preparing the legal text. This involves tokenizing and segmenting into sentences. The sentences are then embedded with BERT to obtain contextual embeddings representing language as well as legal meaning. The K-Means algorithm clusters these embeddings to identify sentences with similar meaning. We then select representative sentences from each cluster to form the final summary. This method guarantees that we adequately deal with pertinent legal arguments and facts in a simple and concise manner.

3) *Evaluation Metrics*: Quantitative and qualitative measures are applied to measure the performance of the system. ROUGE-1, ROUGE-2, and ROUGE-L assess n-gram and sequence-level overlap between target and generated summaries, as detailed in equation (9). Semantic similarity, calculated as cosine similarity defined in equation (10), measures contextual fidelity, whereas precision, recall, and F1-score measure sentence-level classification accuracy.

In addition, the summaries are also examined by human legal professionals for legal validity, proficiency, and pertinence. When synthesized, these measures ensure that the summaries

generated are legally relevant, contextually consistent, and linguistically correct.

$$\text{ROUGE-N} = \frac{\text{Count of overlapping N-grams}}{\text{Total N-grams in reference summary}} \quad (9)$$

$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\| \|B\|} \quad (10)$$

B. Self-Correcting RAG

1) *Architecture of Self-Correcting RAG*: The Self-Correcting RAG architecture enhances the standard RAG framework through a feedback-driven quality control loop, ensuring factual accuracy, relevance, and completeness of generated outputs, as shown in Fig. 10. The system comprises the following components:

A query preprocessing and analyzer stage cleans and reformulates the user query for effective retrieval. A retriever then uses Facebook AI Similarity Search (FAISS) or embedding-based similarity search to fetch the top-k relevant documents from the knowledge store or vector database. A context or prompt constructor combines the retrieved documents with the query to form a context-rich prompt for the LLM. The generator produces a detailed response using this constructed prompt. A self-critic or quality checker evaluates the generated response for detail, completeness, and relevance and, if deficiencies are found, triggers re-retrieval or re-generation. A verified output is delivered as an accurate citation-backed answer after iterative refinement. Verified outputs can also support fine-tuning through reinforcement learning with human feedback.

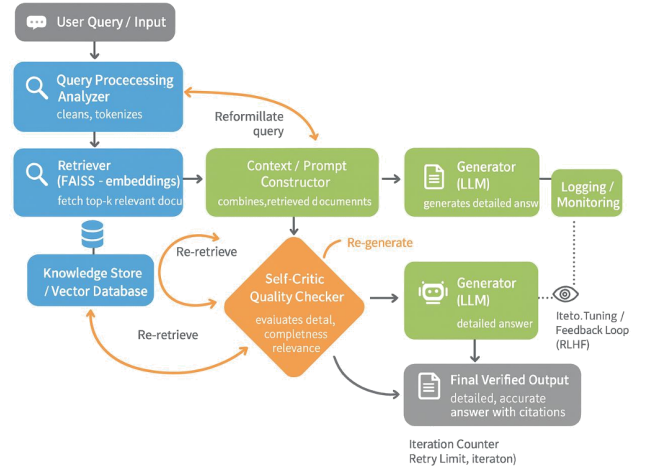


Fig. 10. Architecture of the Self-Correcting RAG System. The feedback loop ensures iterative refinement through retrieval, generation, and self-critique mechanisms.

2) *Workflow for Legal Domain*: The self-auditing RAG model is an improvement over the retrieval and summarization of legislation, precedents, and case-specific judgments in the law arena. Upon input of a legal question by the user, such as a case citation or statutory meaning inquiry, the process

begins. The retriever module fetches the most relevant legal documents based on a legal knowledge base, including acts, court decisions, or constitutional provisions of India. Encoder-based cosine similarity is employed to score every candidate document against the query and select the top K results as context, as given in equation (11).

$$s(z | x) = \frac{f_q(x) \cdot f_d(z)}{\|f_q(x)\| \|f_d(z)\|}, \quad (11)$$

Where z is a candidate document, x is the query, and the query and document encoders are $f_q(\cdot)$ and $f_d(\cdot)$, respectively.

The LLM creates an early draft of the legal response or summary, while the context constructor creates a thorough prompt that includes the legal question and the extracted passages. Three criteria are used by the self-critic to assess this response: the completeness of the reasoning, the relevance of the legal principles, and the factual accuracy of the citations. If any reflection score falls below its threshold, a re-retrieval decision is triggered as shown in equation (12), and the system iteratively refines retrieval or regeneration until a legally coherent and logically sound output is produced

$$\delta_{\text{retrieve}} = 1[\text{Rel} < \tau_r \vee \text{Sup} < \tau_s \vee \text{Use} < \tau_u], \quad (12)$$

with Rel, Sup, and Use denoting reflection estimates of relevance, support, and usefulness. This workflow aligns generated content with judicial reasoning standards to minimize hallucinations or incomplete interpretations often observed in standard LLM outputs.

3) *Evaluation Metrics*: To assess the performance of the Self-Correcting RAG framework in the legal domain, both quantitative and qualitative metrics are used. Recall-Oriented Understudy for Gisting Evaluation (ROUGE) scores, specifically ROUGE-1, ROUGE-2, and ROUGE-L, evaluate n-gram and sequence overlap between generated summaries and ground-truth legal texts. Precision, recall, and F1-score measure the accuracy of retrieved and generated legal content at the sentence and document level. Retrieval accuracy quantifies the proportion of relevant legal documents retrieved standards to minimize error. Semantic similarity uses cosine similarity between generated outputs and reference summaries to assess contextual alignment. Human evaluation involves legal experts who rate the responses for relevance, factual correctness, and legal coherence. These metrics collectively validate the reliability and legal soundness of the system, ensuring that outputs are both contextually rich and jurisprudentially accurate.

IV. RESULTS

This section presents the empirical evaluation conducted for Indian legal text analysis. The dataset and training setup are described first, followed by comparative results for BERT-based classifiers and RAG-variant frameworks under a unified experimental protocol.

A. Dataset Description and Error Analysis

We utilize the *viber1/indian-law-dataset* [29], available on the Hugging Face Hub. The dataset contains 24,607 legal instruction–response pairs designed for supervised training of legal conversational systems. Every record includes a legal response to a user query.

With an average document length of 68.5 words, a median of 58 words, and a 90th-percentile of 112 words, the corpus is short, which indicates that it is better suited for research involving questions than for retrieving full-length documents. This release does not include any explicit metadata, such as jurisdiction or legal-entity annotations.

To manage lengthy legal documents without losing important content, we used a sliding-window chunking technique, which splits each document into overlapping chunks within the 512-token model limit. The predictions from each chunk are then combined to arrive at the final decision. Furthermore, a retrieval-based approach was integrated by generating embeddings for each chunk and indexing them using FAISS, which allowed the model to retrieve the most relevant legal passages during inference. This combination ensures that the model will capture all document information and benefit from contextual reference to similar prior cases.

The average input length after tokenization was about 83.8 tokens, while the maximum observed document length was 3,875 tokens. Sequences were truncated to 512 tokens for transformer encoders such as LegalBERT and Lawformer, while the retrieval context window for RAG-based architectures extended up to 2,048 tokens to accommodate supporting evidence. The corpus was separated into overlapping segments of 512 tokens using a stride of 128. During the generation process, the top- $k = 5$ retrieved passages were concatenated with the query text. Preprocessing included deleting boilerplate text, normalizing Unicode, and keeping section markers and citation identifiers in order to guarantee legal traceability.

When evaluating Self-RAG and Self-Correcting RAG, two primary error types were identified: **(1) Retrieval errors**: happened when responses mentioned laws or portions that weren't fully supported by the retrieved passages. **(2) Generation**: comprised citation errors or insufficient factual reasoning, which were present in 6.3% of Self-RAG and 4.1% of Self-Correcting RAG outputs.

These findings underscore the challenge of maintaining factual alignment in legal reasoning tasks. Future developments could include richer metadata or structured retrieval to lessen hallucinations and strengthen the foundation of statutory references.

B. Experimental Setup

An NVIDIA DGX-1 with eight Tesla V100 GPUs was used for all experiments. High-throughput training and repeatable evaluation across all model classes were guaranteed by the configuration.

C. Training and Validation

BERT-based encoders were fine-tuned for document classification on the dataset using a held-out validation split. Evaluation metrics included Accuracy, Precision, Recall, and F1-score. RAG-based systems were evaluated for retrieval and generation effectiveness using the same corpus. Retrieval was assessed at rank 1 using Recall (R1), Precision (P1), and F1, while generation quality employed BLEU and ROUGE (ROUGE-1, ROUGE-2, ROUGE-L). All values correspond to best-performing checkpoints.

D. Results of BERT-Based Models

Table I reports the results of CaseLawBERT, InLegalBERT, Lawformer, and LegalBERT. LegalBERT achieves the highest overall accuracy and F1-score, confirming the benefit of domain-specific pretraining, while Lawformer performs comparably and InLegalBERT provides strong cross-domain consistency. CaseLawBERT serves as the baseline.

TABLE I
PERFORMANCE OF BERT-BASED MODELS ON INDIAN LEGAL DATASET

Model	Accuracy	Precision	Recall	F1-Score
CaseLawBERT	0.5238	0.5201	0.5238	0.5213
InLegalBERT	0.9204	0.9217	0.9204	0.9202
Lawformer	0.9390	0.9410	0.9390	0.9393
LegalBERT	0.9623	0.9709	0.9623	0.9644

E. Results of RAG-Based Models

Retrieval performance of all RAG variants is reported in Table II, and generation quality in Table III.

TABLE II
RAG VARIANTS — RETRIEVAL METRICS AT RANK 1

Model	R1	P1	F1
Self-RAG	0.4800	0.4800	0.4800
CRAG	0.3333	0.3333	0.3333
Iterative RAG	0.3011	0.3011	0.3011
KRAG	0.5000	0.5000	0.5000
Self-Correcting RAG	0.5333	0.5333	0.5333

TABLE III
RAG VARIANTS — GENERATION METRICS

Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
Self-RAG	25.9905	0.3808	0.3099	0.3549
CRAG	7.2137	0.8000	0.4250	0.5333
Iterative RAG	3.9868	0.2458	0.1408	0.1842
KRAG	47.6870	0.7961	0.5159	0.7405
Self-Correcting RAG	38.2746	0.8021	0.5214	0.6829

As observed in Table II and Table III, KRAG achieves the highest overall generation quality (BLEU and ROUGE-L) due to structured knowledge integration, while Self-Correcting RAG demonstrates the most consistent retrieval accuracy (R1 = 0.5333).

The Self-Correcting RAG that we present proposes a feedback-based correction loop that reexamines generated

output against retrieved evidence. The proposed system re-evaluates through retrieval verification and generative re-evaluation, improving factual grounding and semantic coherence over pre-existing CRAG and Self-RAG approaches. The performance of the model in its evaluation metrics of (BLEU = 38.2746, ROUGE-1 = 0.8021) showcases the improved contextual understanding and trustworthiness on long-form inferences, namely legal reasoning.

In conclusion, RAG neural network models outperformed BERT-based baselines as a result of retrieval and generation mechanisms, allowing greater factual consistency and adaptability in knowledge-heavy contexts.

F. Interpretability, Fairness, and Legal Responsibility

In addition to metrics, interpretability and responsible deployment are focal points of this study. In RAG models like KRAG and Self-Correcting RAG, all generated responses are backed by explicit retrieval citations of the legal source, allowing for traceability of legal sources. In encoder-based baselines like LegalBERT, visualizations of attention-weights allow for transparency of token relevance.

Data includes texts across different criminal, civil, and contractual domains to help reduce bias, although there are still small jurisdictional imbalances. The proposed systems are simply assistive tools; all outputs contain inline legal citations and disclaimers to make clear that they are not legal advice. These methods promote accountability and ethical compliance in using AI for judicial research.

V. CONCLUSION AND FUTURE SCOPE

The BERT architecture serves as the foundation for InLegalBERT, which excels at extractive understanding by efficiently capturing syntactic and semantic dependencies in intricate legal texts.

The Self-Correcting RAG framework, on the other hand, combines generative and retrieval reasoning to produce outputs that are factually consistent and contextually rich. This comparative study emphasizes how retrieval-based augmentation and generative modeling can be combined to provide deeper interpretability, factual reliability, and adaptability—all of which are essential for legal AI applications. In order to improve explainability, future research will investigate expanding the Self-Correcting RAG framework to multilingual and cross-jurisdictional legal corpora by utilizing knowledge graphs and reinforcement learning with human feedback. Deploying these systems in real-time legal research pipelines may also help close the gap between judicial decision support and academic innovation.

REFERENCES

- [1] K. D. Ashley, *Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age*. Cambridge University Press, 2017.
- [2] T. Brown *et al.*, “Language Models Are Few-Shot Learners,” in *Proc. NeurIPS*, vol. 33, 2020, pp. 1877–1901.
- [3] I. Chalkidis, A. Jana, D. Hartung, M. Bommarito, I. Androutsopoulos, D. Katz, and N. Aletras, “LexGLUE: A Benchmark Dataset for Legal Language Understanding in English,” in *Proc. ACL*, 2022, pp. 4310–4330.

- [4] D. Hendrycks, C. Burns, A. Chen, and S. Ball, “CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review,” in *Proc. NeurIPS Datasets and Benchmarks*, 2021.
- [5] Author Names, “Doc2Vec + SVM based semantic search for legal documents,” in *Proc. JURIX*, 2019.
- [6] Author Names, “BERT + BM25 hybrid models for legal information retrieval,” in *Proc. ECIR*, 2021.
- [7] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletas, and I. Androutopoulos, “LEGAL-BERT: The Muppets straight out of Law School,” in *Findings of EMNLP*, 2020, pp. 2898–2904.
- [8] V. Mamakas, I. Chalkidis, and D. Tsarapatsanis, “Processing Long Legal Documents with Pre-trained Transformers: Modding LegalBERT and Longformer,” *arXiv preprint*, 2022.
- [9] L. Wang *et al.*, “A Comprehensive Survey of LLM-based Autonomous Agents (2024 Update),” *arXiv preprint*, 2024.
- [10] L. Zheng, N. Guha, B. Anderson, P. Henderson, and D. Ho, “When Does Pretraining Help? Assessing Self-Supervised Learning for Law and the CaseHOLD Dataset,” in *Proc. ICAIL*, 2021.
- [11] C. Xiao, H. Zhu, and M. Li, “Lawformer: A Pre-trained Language Model for Chinese Legal Long Documents,” *arXiv preprint*, 2021.
- [12] P. Lewis *et al.*, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” in *Proc. NeurIPS*, vol. 33, 2020, pp. 9459–9474.
- [13] M. Trapp and F. Armann, “Large Language Models in Law: A Survey of Opportunities and Challenges,” *preprint*, 2024.
- [14] A. Balaguer *et al.*, “RAG vs Fine-Tuning: Pipelines, Tradeoffs, and a Case Study on Agriculture,” *preprint*, 2024.
- [15] T. Xu, Z. Zhang, and H. Li, “Domain-Specific Pretraining for Legal Judgment Prediction in Chinese AI Courts,” in *Proc. AAAI*, 2020.
- [16] S. Paul, A. Mandal, and P. Bhattacharyya, “Pre-trained Language Models for the Legal Domain: A Case Study on Indian Law,” *arXiv preprint*, 2023.
- [17] J. Wei *et al.*, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” in *Proc. NeurIPS*, 2022.
- [18] S. Yao *et al.*, “ReAct: Synergizing Reasoning and Acting in Language Models,” in *Proc. AAAI*, 2023.
- [19] A. Asai *et al.*, “Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection,” in *Proc. ICLR*, 2024.
- [20] J. Shi *et al.*, “Corrective Retrieval Augmented Generation (CRAG),” *arXiv preprint*, 2024.
- [21] W. Yu *et al.*, “Chain-of-Note (CoN): Enhancing Self-Correction in Large Language Models,” *arXiv preprint*, 2024.
- [22] A. Zeng *et al.*, “AgentTuning: Enabling Generalist Agents with Curated Agent-Tuning Data,” *arXiv preprint*, 2024.
- [23] M. Silveira, L. Pereira, and A. Santos, “Using Topic Modeling in Classification of Brazilian Lawsuits via Legal-BERT,” *ResearchGate preprint*, 2025.
- [24] “SemEval-2023 Task 6: LegalEval – Understanding Legal Texts,” in *Proc. SemEval*, 2023.
- [25] H. Zhu, C. Li, and M. Wu, “A Hybrid Summarization Method for Legal Judgment Documents based on Lawformer,” in *Proc. SIGIR*, 2021.
- [26] A. Asai, S. Min, Z. Zhong, D. Chen, and H. Hajishirzi, “Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection,” in *Proc. ICLR*, 2024.
- [27] N. H. Thanh and K. Satoh, “Krag Framework for Enhancing LLMs in the Legal Domain,” *arXiv preprint arXiv:2410.07551*, 2024.
- [28] G. Chen, Y. Yao, and T. Liu, “SGIC: A Self-Guided Iterative Calibration Framework for Retrieval-Augmented Generation,” *arXiv preprint arXiv:2506.16172*, 2025.
- [29] Hugging Face Datasets, “viber1/indian-law-dataset,” Available: <https://huggingface.co/datasets/viber1/indian-law-dataset>, Accessed: Oct. 2025.