RESEARCH PAPER

Enhancing Legal Question Answering with Evidence-Grounded RAG, Multi-Model Evaluation, and Safe-Template Repair

Pardeep Kumar[1*], Dr. Vijay Dhir[2]
[1]Research Scholar, Department of Computer Science and Engineering, Sant Baba Bhag Singh University, Punjab, India. Email: sharma.pardeep04@gmail.com
[2]Professor, Department of Computer Science and Engineering, Sant Baba Bhag Singh University, Punjab, India. Email: profvijaydhir@gmail.com
***Corresponding Author: Pardeep Kumar**
Research Scholar, Department of Computer Science and Engineering, Sant Baba Bhag Singh University, Punjab, India
Email: sharma.pardeep04@gmail.com
Received: 31st May, 2026; **Revised:** 8th June, 2026; **Accepted:** 10th June, 2026; **Available Online:** 13th June, 2026

ABSTRACT
Legal question answering is essentially based on factual correctness, statute authority and trustworthy interpretation of legal text. While Retrieval-Augmented Generation (RAG) enhances the factual accuracy of LLMs by being able to prompt them to first integrate relevant evidence into their answer before producing their own, retrieval does not necessarily ensure that the produced response is legally correct or supported by the retrieved text. This paper introduces a verified Legal RAG system for Indian Statutory Question Answering. This proposed approach applies the model LayoutLMv3 for structured legal document processing, the BART model for legal text summarization, the hybrid retrieval B&R model DPR, and several models for answer generation (e.g. multiple language models). Generated answers are grounded using evidence grounding, Legal-BERT similarity and rule-based statutory validation is applied to improve reliability. Any answer that does not pass the verification process will have a deterministic safe-template repair module build the answer from verified statutory evidence. This framework was tested with seven models, namely Llama 3.1, Mistral 7B, Phi-3 Mini, DeepSeek 1.5B, the model of GPT-5.1, Legal-Qwen Base, and Legal-Qwen LoRA to assess their performance across various legal questions. Llama 3.1 showed the best overall test performance by repeating the highest number of times with the best score and the smallest dependence on repair, being the best of the top models. The results indicate that verifying and safe template fixing methods boost the dependability of legal RAG systems and diminish unsupported legal answer generation.
Keywords: Legal Question Answering, Retrieval-Augmented Generation, Indian Legal Documents, Hybrid Retrieval, Evidence Grounding, Legal-BERT, Safe-Template Repair, Multi-Model Evaluation.
How to cite this article: Kumar P, Dhir V. Enhancing Legal Question Answering with Evidence-Grounded RAG, Multi-Model Evaluation, and Safe-Template Repair. Int J Drug Deliv Technol. 2026;16(59s): 1086-1093. DOI: 10.25258/ijddt.16.59s.125
Source of support: Nil
Conflict of interest: None

1 Introduction

The legal question answering needs high factual accuracy as correct statutory provisions, sections, articles or clauses need to be provided to support legal answers. Despite their excellent capabilities in text generation, Large Language Models (LLMs) can often deliver responses that are grammatically correct but factually deficient, or even entirely fallacious, an issue that poses a risk for LLM applications in legal contexts [1, 2, 3]. Retrieval-Augmented Generation (RAG) diminishes this problem by retrieving the documents of interest prior to generating the document [4, 5]. However, retrieval alone does not guarantee final document is legally valid or sufficiently supported with retrieved evidence.

There are recent literature on legal RAG, question answering and summarisation, which focuses on legal document analysis and question answering through generative AI [6, 7, 8, 9]. Unfortunately, most present systems address retrieval and generation issues, paying little attention to post-generation verification, statutory rule checking and to repairing unsupported legal answers. This leaves room for the legal option with a higher focus on the reliability of answers rather than fluency.

In this paper, we design a verified Knowledge-Graph Approach for Indian Legal Question Answering platform. The corpus consists of Indian statutes, such as Bharatiya Nyaya Sanhita, Constitution of India, Indian Penal Code, Prevention of Corruption Act, Dowry Prohibition Act, Hindu Marriage Act, Divorce Act, Transfer of Property Act and Protection of Children from Sexual Offences Act [10, 11, 12, 13, 14, 15, 16, 17, 18]. The proposed pipeline involves the following: Structured document preprocessing including LayoutLMv3 [19], legal summarization using BART, hybrid retrieval using DPR, BM25, and multiple LLMs for answer generation.

The novelty of this piece of work is the automatic combination of evidence-grounded verification coupled with a Legal-BERT similarity checking method, a rule-based statutory verification method, and a deterministic safe-template repair method. Evaluation measures are adopted for seven language models including verification score, success rate, usage of repairs, latency, ROUGE, BLEU, BERTScore, and every ablation analysis. The results indicate that verification and repair are effective in enhancing the reliability of legal answers and minimiz- ing unsupported generation.

The contributions of this paper can be summarized as follows:

- Accurate Legal RAG system based on Indian Statutory Question Answering.
- A novel Hybrid retrieval model based on DPR for dense retrieval and BM25 for lexical retrieval.
- Evidence grounding, Legal-BERT similarity, and Rule validation based verification mechanism.
- Safe template repair module for repairing unrecognized or incomplete legal answers.
- Multi-model evaluation using success rate, dependency for repair, latency, evaluation of generation quality metrics.

2 Related Work

2.1 Large Language Models and Legal AI

The LLM has introduced outstanding capabilities in various tasks, including text generation, summarization, question answering, and supporting a fraction of questions from specific areas [1, 2, 3]. The LLM might also provide results that appear coherent, but where some information is missing or not sufficiently backed up, especially in tasks that require accuracy in information or some expertise from the field. This is an important limitation in the legal context as the rule must be followed in legal answers which must refer to text in the statutes/laws, legal terms and the proper reference to the sections. More recently, various works have emerged in the field of generative AI that address reliability, explainability, bias, and controllability issues [20, 21, 22].

2.2 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG): Especially, retrieval of facts can be enhanced by load- ing relevant external documents first before generating an answer. This helps reduce reliance on the internal knowledge of LLMs and aid in generating evidence-based responses [4]. Recently, various studies used improved retrieval, question matching and multi-round retrieval to enhance RAG [23, 9]. Recent studies also indicate, however, But for reliable final answers, retrieval is not sufficient, and test time criticism/optimization is necessary [5].

2.3 Legal RAG and Legal Document Processing

Recently, Legal RAG has attracted attention in the areas of analyzing legal documents, legal summarization, and question and answering. Previous research has focused on legal document analysis using generative AI [6], legal question answering (QA) systems [7], and adaptable RAG-based legal summarization systems[8]. For structured legal documents, the textual information combined with layout information can be retrieved with document understanding models like LayoutLMv3 [19]. Yet, current legal RAG systems primarily emphasize retrieval and generation, with only a few investigations concerning verifying answers, validating the rules and statutes, and repairing unsupported legal responses.

2.4 Research Gap

The literature reviewed here reveals that current legal systems use RAG systems have still not well defined mechanism for verification post generation. Current systems can fetch the relevant evidence but they do not necessarily provide completely sourced, answer-relevant, and reference free answers. Thus, this work presents a proven framework for Legal RAG: It integrates hybrid retrieval, multi-model generation, evidence-driven verification, Legal-BERT similarity check, rule validation approach, and safe-template repair approach.

3 Dataset Description and Corpus Preparation

3.1 Legal Corpus

This work has compiled the legal corpus from the publicly available Indian statuts. Major legal sources are included in the dataset such as the laws of crime, constitutional, corruption, marriage, property, dowry, and child protection. They contain the Bharatiya Nyaya Sanhita, Constitution of India, Indian Penal Code, Prevention of Corruption Act, Dowry Prohibition Act, Hindu Marriage Act, Divorce Act and the Transfer of Property Act as well as Protection of Children from Sexual Offences Act [10, 11, 12, 13, 14, 15, 16, 17, 18].

3.2 Document Preprocessing

The gathered legal records were primarily in a PDF format. Structure of legal documents like reference to chapters, sections etc., explanations, provisos, statutory references etc. is not just simple text and hence simple text extraction is not enough. Thus, the documents' preprocess- ing was performed by LayoutLMv3, which extracted both textual and layout information [19]. They were then all partitioned into structured legal segments, maintaining meaningful meta- data, including the document title, page number, section number, section title, path to hierarchy and legal references.

3.3 Structured Legal Chunks

Each chunk extracted from the text was preprocessed, cleaned and stored in structured format. Some extraneous spaces, special symbols and formatting noise were eliminated by the text cleaning operation. Final Structured Records included the original legal language and context-specific metadata. Such fragmented text fragments were then employed in summariz- ing, retrieving, reranking the data, tuning the dataset and generation of the answer.

3.4 Question Set

The following study was carried out to examine a framework with 30 legal questions. These questions address several types of law such as robbery, theft, extortion, cheating, criminal breach of trust, dowry, Hindu marriage, divorce, equality before law, freedom of speech and protection from arrest etc. as per the statutory law. The questions were employed to evaluate retrieval quality, questions generated, question verification, usage of repair and model-wise performance.

4 Proposed Methodology

The proposed framework integrates retrieval, generation, verification and repair for enhancing the reliability of legal question answering. The proposed system differs from a typical RAG system because it does not immediately accept the result of the AI generation. Rather each answer is subjected to verify it against the retrieved statutory proof before the acceptance of it.

4.1 Overall Framework

The entire pipeline comprises five components—legal document preprocessing, summariza- tion, legal knowledge retrieval (hybrid), answer generation, and verification and repair. First, to ensure that the text content and document structure will remain unchanged during processing, legal PDFs are processed with the use of LayoutLMv3 [19]. Then, BART is applied to create short summaries for legal chunks. During retrieval the actual legal text is retrieved along with these summaries and metadata. The retrieval stage is a combination of DPR-based dense retrieval and BM25-based lexical retrieval. Last, answers are produced by multiple language models, leveraging retrieved evidence, and verified before the final answer is generated.

4.2 Hybrid Retrieval

There are statutory words within the legal query like section, offence, or legal concepts. Hence, while only semantic retrieval may fail to retrieve the exact legal terminology, only keyword retrieval may not retrieve a contextual similarity. To tackle this challenge the proposed system employs dense and lexical retrieval. DPR is employed for retrieving semantically similar legal chunks, and BM25 is employed to retrieve the text of the statutory documents which contain the same keywords. The final hybrid score is computed as:

$$S_{hybrid} = \alpha S_{BM25} + \beta S_{DPR}$$

nent and the dense retrieval component, respectively.

4.3 Answer Generation

Once retrieved, top evidence fragments are provided to varying language models for generating answers. The models evaluated are Llama 3.1, Mistral 7B, Phi-3 Mini, DeepSeek 1.5B, GPT- 5.1, Qwen Base and Legal-Qwen LoRA. A multiple modeling approach was used to compare the skill of generating legal answers in the same retrieval and verification context.

4.4 Verification Mechanism

Answer justification, Legal-BERT similarity score, and rule-based statutory validation are used to validate the generated answer. Evidence grounding is designed to see if the answer that is generated is based on the legal text that is retrieved. Legal-BERT calculates semantic similarity between answer and evidence. Provide rule validation that ensures required legal terms, statutory references and prohibited distractors are managed properly. The final Verification score is:

$$V_{final} = w_1 G + w_2 B + w_3 R$$

where G denotes the score given by the evidence, B denotes the Legal-BERT similarity score, R denotes the score of the rule validation, and w_1, w_2, and w_3 denote assigned weights.

4.5 Safe Template Repair

Applying the safe-template repair module if generated answer cannot be verified. This module does not create any new unsupported content. Rather, it builds a corrected response based on merely the particular statutory evidence picked as well as pre-formed legal guidelines. This step is necessary because recent findings reveal that retrieval is not enough by itself, and that to achieve reliable RAG outputs requiring test-time checking or optimization [5]. Answer is verified, partially verified, or re-retrieval required depending on the verification score.

5 Experimental Setup

5.1 Implementation Setup

The proposed Legal RAG framework adopted is a step-wise pipeline. In the pre-processing stage, the structured legal chunks were created from the available PDF statutory documents. These chunks were summarized, indexed, retrieved, reranked and provided to various language models to generate an answer. Evidence grounding, Legal-BERT similarity, and rule-based statutory validation were used to verify the final answers. The safe-template repair module was used for an answer that did not pass verification before the decision making was done.

5.2 Evaluation Questions

This framework was tested with 30 legal questions prepared from topics mentioned in Indian statute which had also legal sources for the same. These encompassed matters of criminal law, party rights under the Constitution, dowry law, corruption law, Hindu marriage, divorce, domestic violence, consumer complaints and child protection laws. All queries were fed into the same retrieval, generation, verification and repair pipeline.

5.3 Models Used

Seven language models were tested with the same experimental conditions. Available models were chosen to facilitate comparisons of open-source local models, fine-tuned legal models, and API-based generation models. The following models were evaluated:

- Llama 3.1
- Mistral 7B
- Phi-3 Mini
- DeepSeek 1.5B
- GPT-5.1
- Qwen Base
- Legal-Qwen LoRA

5.4 Evaluation Metrics

Two types of assessment – verification based and generation quality assessment – were ap- plied to measure the performance of each model. The metrics that were verified were: verified answers, partially verified answers, re-retrieval required cases, final success rate, repair usage rate, average verification score, latency. The metrics: ROUGE, BLEU, and BERTScore were selected for generation quality measurement. The following metrics measured factual ground- ing, semantic similarity, answer quality, model reliability and computational efficiency.

5.5 Decision Criteria

The following three decision categories were allocated to each answer generated. An answer was deemed to be Verified where it was more explicitly backed up by the statutory evidence found. It was accepted and was Partially Verified because there were minor restrictions. It received Re-retrieval Required when the answer was not well supported or was written for a better evidence retrieval would strengthen the answer. The final success rate took into account both the completely correct and partially correct answers.

6 Results and Discussion

6.1 Overall Performance

Seven language models evaluated the proposed Legal RAG framework by testing on 30 legal questions. Table Table 1 shows the results of the model-wise analysis of the responses. Among all models, Llama 3.1 achieved the best overall performance with 17 verified answers, 8 partially verified answers, and only 5 failed cases. It showed the maximum final Success rate of 83.33%. Mistral 7B placed second with a success rate of 66.67% at the end of the tests, whilst the remaining models had an end-of-test success rate of 60.00%.

Table 1: Model-wise Legal RAG performance.

Model	Verified	Partial	Failed
Llama 3.1	17	8	5
Mistral 7B	13	7	10
Phi-3 Mini	9	9	12
DeepSeek 1.5B	9	9	12
GPT-5.1	9	9	12
Legal-Qwen LoRA	9	9	12
Qwen Base	9	9	12

6.2 Repair-Adjusted RAG Score

The final RAG score after repair adjustments is displayed in Fig. 1. This score is significant because it takes into consideration the final performance of the model as well as the reliance of the model on the repair module. Llama 3.1 scored a perfect 0.666 for repair, followed by Mistral 7B with a score of 0.474. The other models earned lower grades due to poorer original generation or greater repair requirements.

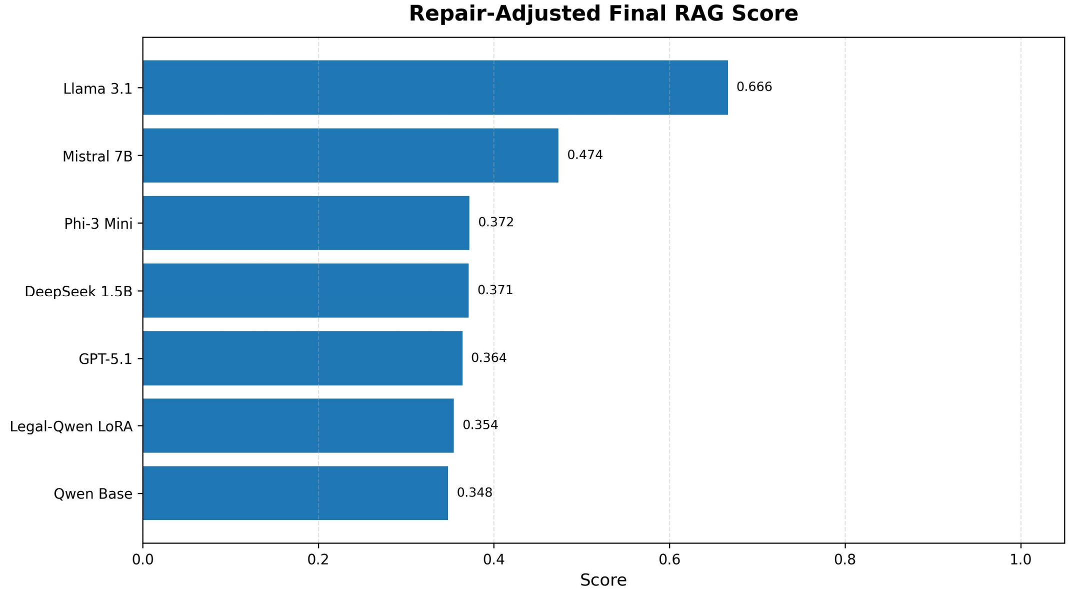
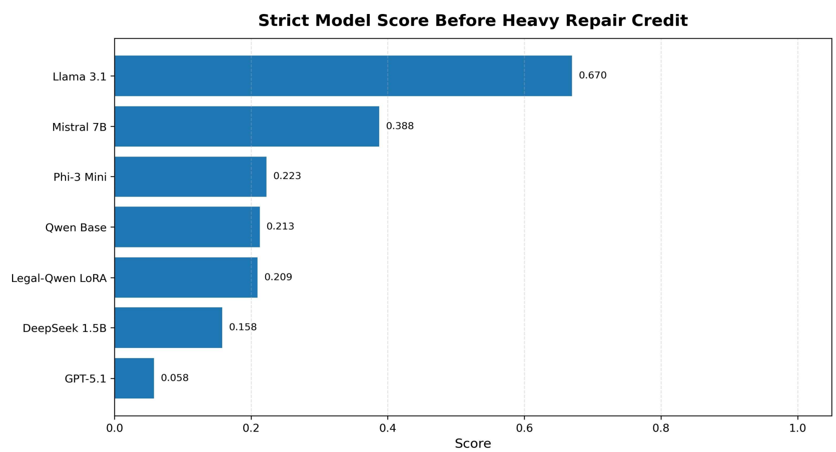


Figure 1: RAG score by repair and models assessement



6.3 Strict Model Score Before Repair

The strict model score was thus examined before heavily crediting the repair module for the actual strength of generation of each of the models. As shown in Fig. 2, Llama 3.1 again achieved the highest strict score of 0.670. This shows that Llama 3.1 not just did well after repair but did well when creating initial responses. Mistral 7B gave it the second best score with the others strict performance being lower.

Figure 2: Model Score prior to heavy repair credit.

6.4 Success Rate and Repair Dependency

Fig. 3 shows the verified rate, the final success rate and the rate of repair usage. Llama 3.1 succeeded 83.33% with only 30.00% repairs. This demonstrates its comparatively more reliable answers prior to repair. The final outcomes for Mistral 7B were 66.67% successful and in 73.33% needed to be repaired. Other models only had a final success rate of 60.00% with the repair used being 100.00%, meaning that a safe-template repair mechanism played the major role in the end results.

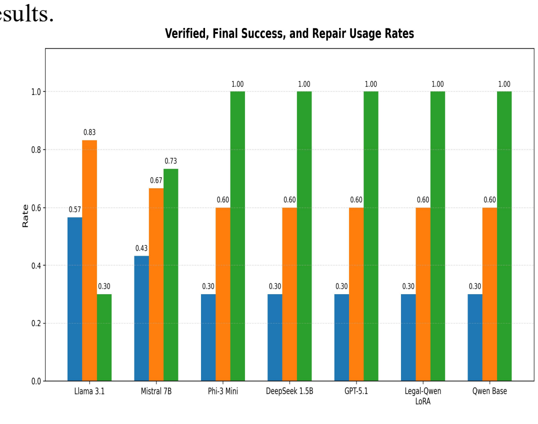


Figure 3: Verified rate, final success rate, and repair usage rate.

6.5 Repair Usage Versus Final Success

In Fig. 4, the rate of use of repair and the outcome of final success are displayed. The target model should be the one that has the greatest success rate and smallest number of repairs needed to get to the end of the run. The only model which falls close to this desired area is Llama 3.1. This indicates that this is the most robust model proposed on its own. Others did moderately well but needed substantial amount of repairs; i.e., they were framework-assisted, not good at generation.

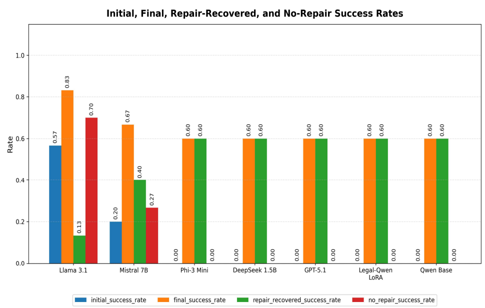


Figure 4: Repair usage rate versus final success rate.

6.6 Latency Analysis

The average latency of each model is shown in Fig. 5. The latency was the lowest for GPT-5.1 while the Legal-Qwen LoRA and Qwen Base models had the longest processing time. In fact, suitability of a model for legal question answering cannot only be judged by latency, since predicted reliability and grounding of the question-answerer in the legal evidence is more crucial for legal question answering. The best compromise of answer quality and computation time was provided by Llama 3.1.

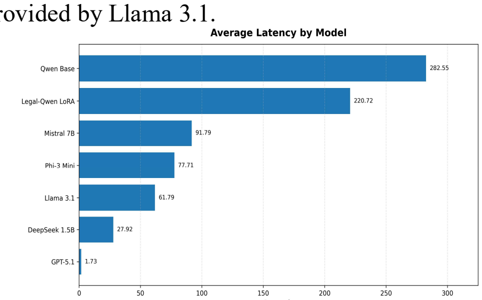
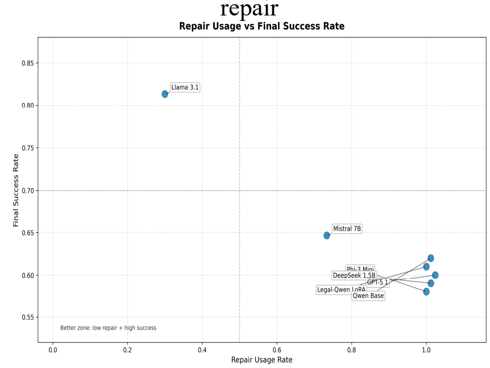


Figure 5: Average latency comparison from the end of the models

6.7 Repair Impact

Safe-template repair has an effect as illustrated in Fig. 6. The findings indicate that repair worked by transforming incomplete and unsupported responses into evidence-based answers to enhance the ultimate success rate. Interestingly, no-repair success rate was the highest for Llama 3.1, demonstrating that it returned more satisfactory answers prior to repair. In most other models, the final success mostly recovered by the repair module.

Figure 6: Initial, final, repair-recovered, and no-repair



success rates.

6.8 Summary of Results

The clearly evident result is that Llama 3.1 is the best performing model in the proposed Legal RAG framework. It had the best overall success rate, best model score, best repair adjusted model score and least repair dependence among the best models. Mistral 7B was in the second place, but needed more repair. The other models had moderate success in their final results, but their results were significantly affected by the issue of safe template repair. Overall, the results validate both the presence of legal question answering beyond retrieval tasks and the presence that verifies the generated legal answer with safe template repair it becomes more reliable.

7 Conclusion

This paper introduced a verified Retrieval-Augmented Generation system for the Indian legal Question Answering. The proposed system involves the following processes: structured legal document preprocessing, BART summarization, DPR retrieval method, BM25 retrieval method, and hybrid retrieval method, multi-model answer generation, stance verification based on evidence, similarity verification based on Legal-BERT, rule-based statutory validation, and safe document repairing based on template. The ultimate goal was to provide more reliable answers to legal questions, even when the ultimate legal answer is retrieved, it is backed by statute support.

Seven language models were employed to evaluate the framework over 30 legal questions. When it comes to best performance, the highest success rate, highest repair adjusted score, and lowest repair dependency, the results indicated that Llama 3.1 was the best performing model. The second highest score was for Mistral 7B, though it needed to have higher repair support. The other models were not as successful overall, but primarily because they had a module for safe-template repair, which was not present in their earlier generations and was poorer in terms of success rate.

The results show that retrieval is not enough for answering legal queries. Answers that are generated may still need to be verified and corrected even if evidence relevant to the responses can be found. The proposed mechanism for the verification and a safe template repair process increases factual grounding, decreases unsupported generation, and offers a better basis for statutory legal question answering. The study will be continued with a larger legal question set, with legal expertise validation, with additional metrics like Recall@K and nDCG for retrieval and and with complex multi-hop tasks of legal reasoning to be tested.

References

[1] P. P. Ray, "Chatgpt: A comprehensive review on background, applications, key chal- lenges, bias, ethics, limitations and future scope," *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 121–154, 2023.

[2] M. A. K. Raiaan, M. S. H. Mukta, K. Fatema, N. M. Fahad, S. Sakib, M. M. J. Mim, J. Ahmad, M. E. Ali, and S. Azam, "A review on large language models: Architectures, applications, taxonomies, open issues and challenges," *IEEE Access*, vol. 12, pp. 26 839– 26 861, 2024.

[3] F. Chiarello, V. Giordano, I. Spada, S. Barandoni, and G. Fantoni, "Future applications of generative large language models: A data-driven case study on chatgpt," *Technovation*, vol. 133, p. 103002, 2024.

[4] M. Klesel and H. F. Wittmann, "Retrieval-augmented generation (rag)," *Business and Information Systems Engineering*, 2025. [Online]. Available: https://doi.org/10.1007/ s12599-025-00945-3

[5] J. Wei, H. Zhou, X. Zhang, D. Zhang, Z. Qiu, N. Wei, J. Li, W. Ouyang, and S. Sun, "Retrieval is not enough: Enhancing rag through test-time critique and optimization," *Advances in Neural Information Processing Systems*, vol. 38, pp. 21 484–21 520, 2026.

[6] P. Saxena, A. Verma, and R. Bhattacharya, "Legalmind-gpt: Domain-specific generative ai for financial legal document analysis," *AI and Law*, 2024, preprint. [Online]. Available: https://arxiv.org/abs/2403.09876

[7] T. Espirito Santo, R. Costa, J. Rodrigues, and A. Lopes, "Cocoruta 1.0: A portuguese legal question-answering and retrieval system using generative ai," *AI and Law*, 2024, preprint version under review. [Online]. Available: https://arxiv.org/abs/2402.01123

[8] S. A. Mukund and K. S. Easwarakumar, "Optimizing legal text summarization through dynamic retrieval-augmented generation and domain-specific adaptation," *Symmetry*, vol. 17, no. 5, p. 633, 2025. [Online]. Available: https://doi.org/10.3390/sym17050633

[9] W. Zhang, H. Zhou, Q. Zhou, Y. Li, Y. Liu, J. Lou, C. Wu, and J. Li, "Towards comprehensive legal document analysis: A multi-round rag approach," in *Proceedings of the 2025 International Conference on Multimedia Retrieval*, 2025, pp. 1840–1848.

[10] Government of India, "The Bharatiya Nyaya Sanhita, 2023," India Code, Ministry of Law and Justice, 2023, act No. 45 of 2023; accessed on 12 June 2026. [Online]. Available: https://www.indiacode.nic.in/handle/123456789/20062

[11] ——, "The Constitution of India," India Code, Ministry of Law and Justice, 1950, accessed on 12 June 2026. [Online]. Available: https://www.indiacode.nic.in/bitstream/ 123456789/19150/1/constitution of india.pdf

[12] ——, "The Indian Penal Code, 1860," India Code, Ministry of Law and Justice, 1860, act No. 45 of 1860; repealed source used for comparative legal corpus; accessed on 12 June 2026. [Online]. Available: https://www.indiacode.nic.in/repealedfileopen? rfilename=A1860-45.pdf

[13] ——, "The Prevention of Corruption Act, 1988," India Code, Ministry of Law and Justice, 1988, act No. 49 of 1988; accessed on 12 June 2026. [Online]. Available: https://www.indiacode.nic.in/bitstream/123456789/15302/1/pc act, 1988.pdf

[14] ——, "The Dowry Prohibition Act, 1961," India Code, Ministry of Law and Justice, 1961, act No. 28 of 1961; accessed on 12 June 2026. [Online]. Available: https://www.indiacode.nic.in/bitstream/123456789/5556/1/dowry prohibition.pdf

[15] ——, "The Hindu Marriage Act, 1955," India Code, Ministry of Law and Justice, 1955, act No. 25 of 1955; accessed on 12 June 2026. [Online]. Available: https://www.indiacode.nic.in/bitstream/123456789/1560/1/A1955-25Eng.pdf

[16] ——, "The Divorce Act, 1869," India Code, Ministry of Law and Justice, 1869, act No. 4 of 1869; accessed on 12 June 2026. [Online]. Available: https://www.indiacode.nic.in/bitstream/123456789/2280/1/A1869-04.pdf

[17] ——, "The Transfer of Property Act,

1882," India Code, Ministry of Law and Justice, 1882, act No. 4 of 1882; accessed on 12 June 2026. [Online]. Available: https://www.indiacode.nic.in/bitstream/123456789/2338/1/A1882-04.pdf

[18] ——, "The Protection of Children from Sexual Offences Act, 2012," India Code, Ministry of Law and Justice, 2012, act No. 32 of 2012; accessed on 12 June 2026. [Online]. Available: https://www.indiacode.nic.in/bitstream/123456789/2079/1/AA2012-32.pdf

[19] Y. Xu, T. Lv, L. Cui, Y. Lu, D. Florencio, C. Zhang, and F. Wei, "Layoutlmv3: Pre-training for document ai with unified text and image masking," in *Proceedings of the 30th ACM International Conference on Multimedia*. ACM, 2022, pp. 4083–4091.

[20] R. Gupta, K. Nair, M. Mishra, B. Ibrahim, and S. Bhardwaj, "Adoption and impacts of generative artificial intelligence: Theoretical underpinnings and research agenda," *Inter- national Journal of Information Management Data Insights*, vol. 4, p. 100232, 2024.

[21] S. Feuerriegel, M. Dolata, and G. Schwabe, "Explainable artificial intelligence for business: State of the art, challenges, and future research opportunities," *Business & Informa- tion Systems Engineering*, vol. 65, no. 4, pp. 457–472, 2023.

[22] X. Liang, H. Wang, Y. Wang, S. Song, J. Yang, S. Niu, J. Hu, D. Liu, S. Yao, F. Xiong, and Z. Li, "Controllable text generation for large language models: A survey," *arXiv preprint arXiv:2408.12599v1*, 2024. [Online]. Available: https://arxiv.org/abs/2408.12599v1

[23] A. Saha, D. Ghosh, N. Mandal, and A. Dey, "Quim-rag: Advancing retrieval-augmented generation with inverted question matching for enhanced qa performance," *IEEE Access*, vol. 12, pp. 123 456–123 469, 2024.