A Multi-Agent RAG system for Legal Information Retrieval in Bangladesh

Abdullah Al Raiyan[1], Md. Sifat Haque Zidan[2], Rahil Mehnaz[3], and Dr. Mohammad Rashedur Rahman[4]

Department of Electrical and Computer Engineering
North South University, Dhaka 1229, Bangladesh
[1]abdullah.raiyan@northsouth.edu
[2]sifat.zidan@northsouth.edu
[3]rahil.mehnaz@northsouth.edu
[4]rashedur.rahman@northsouth.edu

Abstract. Access to legal information presents a significant impediment in developing nations like Bangladesh, where a substantial portion of the populace lacks legal literacy and access to professional counsel. This paper introduces a Multi-Agent Retrieval Augmented Generation (RAG) system specifically designed for Bangladeshi legal information retrieval. The system employs a novel two-agent architecture: a Clarification Agent that refines user queries through Large Language Model (LLM)-driven clarity assessment and interactive questioning, and a primary RAG Agent. The RAG Agent executes a comprehensive process involving document retrieval from a meticulously curated legal corpus, relevance grading, conditional web search, and final answer synthesis. This corpus, compiled from official government websites, legal blogs, and scholarly articles, ensures accurate representation of the Bangladeshi legal landscape. The multi-agent RAG pipeline, through structured collaboration between these autonomous agents, is designed to improve the relevance and contextual accuracy of responses to complex legal queries. A conversational interface built with `Streamlit` facilitates accessible, real-time user interaction. The Agentic RAG model outperforms baseline and general-purpose legal bots (using GPT-4o-mini and Gemma 3B) on 20 diverse legal queries, showing better contextual relevance, accuracy, and fewer hallucinations. It also supports cost-effective local deployment via the Ollama framework. This research demonstrates a scalable and potentially cost-effective approach to democratizing legal knowledge, which is particularly vital in low-resource settings such as Bangladesh.

Keywords: Multi-Agent · Retrieval Augmented Generation (RAG) · Legal Chatbot · Bangladesh Law · LangChain · LangGraph · Ollama · Open Source · Information Retrieval · Natural Language Processing · Low-Resource Settings.

1 Introduction

Legal services are fundamental for protecting individual rights, ensuring social justice and promoting social equity. However, widespread access to reliable legal assistance services remains a significant challenge, especially in underprivileged and densely populated countries like Bangladesh due to the scarcity of legal professionals, lack of public legal awareness, and high associated costs. Even though Bangladesh has made some progress in offering legal aid to its people, the support system is still not enough to meet the growing needs. Many people, especially in rural or underprivileged areas, cannot get help for their simple legal problems or don't know what their legal rights are. This situation brings up an important question: Can smart and automated systems be used to make legal information more available and easier to understand for everyone in Bangladesh?

In recent years, Large Language Models (LLMs) have shown remarkable success across domains such as healthcare, finance, and law. Tools like ChatGPT, LLaMA and other models have been increasingly explored for their potential in the legal domains. However, despite their potential, traditional LLMs frequently generate hallucinated outputs and answers that may seem accurate but are actually incorrect or they are based on outdated information. This can be dangerous in the legal context, where accuracy and up-to-date knowledge are critically important.

To address these limitations, we developed a Multi-Agent Retrieval-Augmented Generation (RAG) chatbot, tailored for Bangladeshi legal information retrieval. Our solution integrates curated datasets including Supreme Court judgments [24], official legislative documents [15], and legal analysis articles [3] into a knowledge base designed to support factual retrieval. By utilizing `LangChain` [12] and `LangGraph` [13] frameworks, the system introduces a multi-agent architecture where the Retriever Agent identifies relevant legal content, evaluates and refines results based on their relevance to the query. Furthermore, the system is equipped with a conditional web search mechanism to extend support when local resources are insufficient.

For user interaction, a `Streamlit`-based conversational interface [23] was developed, allowing individuals to input queries and receive clear, relevant legal information in real time. Through this work, we aim to present a scalable, low-cost blueprint for enhancing public access to legal knowledge in low-resource settings like Bangladesh, potentially transforming how citizens engage with legal information.

2 Related Works

The integration of artificial intelligence and natural language processing technologies into the legal domain has gained significant momentum in recent years with various approaches such as improving the accessibility, reliability, and efficiency of legal services. The development of legal chatbots and AI-driven legal assistance systems has been a growing research focus in recent years. Various approaches have been proposed to address the challenges of legal knowledge retrieval, accurate response generation, and user trust.

Jiaxi Cui et al. proposed `ChatLaw` [6], a multi-agent, Mixture-of-Experts legal QA system for the Chinese domain that integrates RAG and domain-specific expert networks to reduce hallucinations. Mudita Sharma et al. presented a legal chatbot [21] architecture using AWS Lex and Lambda, showing practical deployment of legal bots with crowdsourced data. Isaac and Nwabueze discussed ethical challenges in deploying AI legal tools such as DoNotPay [11], highlighting both accessibility benefits and misinformation risks.

Xu et al. extended the Technology Acceptance Model for AI-powered legal chatbots [26] and found that usability and perceived value were stronger predictors of adoption than legal compliance. Queudot et al. developed chatbots [20] for immigrant legal guidance and workplace compliance, showing practical gains in specialized legal domains. Cirillo et al. proposed a pipeline [5] for parsing and structuring legal corpora to improve retrieval, demonstrating significant improvements in relevance and grounding. Maree et al. built a chatbot [16] for Palestinian cooperative law that achieved high accuracy based on human expert validation. Nikita et al. created `LAWBOT` [17], an Indian legal chatbot combining IR and ML techniques to simplify access to legal information. Amato et al. introduced `CREA2` [2], an agent system that helps users draft legal documents and resolve family and corporate disputes. RA et al. proposed a chatbot [1] that provides real-time legal guidance based on location and financial status. Devaraj et al. outlined foundational design steps for building LangChain-powered legal bots [8], and Feldman et al. analyzed how RAG systems reduce but don't eliminate hallucinations, especially with misleading prompts [10]. Finally, the RAGAS framework [9] introduced by Es et al. automates RAG evaluation and aligns closely with human judgments on faithfulness and relevance.

Our work builds upon and distinguishes itself from these efforts by focusing on Bangladeshi law, combining Clarification and RAG Agents, incorporating human expert evaluation, and prioritizing region-specific legal access.

3 Methodology

In this work, a Multi-Agent Retrieval-Augmented Generation (RAG) chatbot system specialized for Bangladeshi legal information retrieval. The overall methodology can be divided into several key stages:

3.1 Data Collection and Preparation

At first, we collected a set of legal documents, mainly PDF files, from sources like Bangladesh Supreme Court judgments [24], Laws of Bangladesh [15], and BD Law Post articles [3], BD Law Help [14], etc. These PDFs were loaded using the `PyPDFDirectoryLoader` from the `langchain.document_loaders` module [12]. Additionally, to extract more accurate text from complicated PDFs, the `fitz` library from `PyMuPDF` was used [19].

After loading the documents, it needed to split the long texts into manageable chunks for better retrieval. Here, two techniques were used:

– `RecursiveCharacterTextSplitter` [12] to split text based on certain characters like "\n\n" or ".".

Fig. 1: Data Preparation Workflow.

3.2 Embedding Generation

For generating dense vector representations of each text chunk, OpenAI Embeddings was used [18]. This helped in converting text into high-dimensional vectors, making it easier to perform similarity search later on.

3.3 Vector Store Creation

After getting embeddings, they were stored in a vector database called `Chroma` [4]. `Chroma` is a simple and lightweight vector store that supports similarity search. The `ChromaDB` was set up locally using the `LangChain Chroma` wrapper. This made it easy to retrieve the most relevant chunks when a user asks a query.

3.4 Multi-Agent System with LangGraph

One of the core innovations of this study was using `LangGraph` [13] to build a graph-based, multi-agent system. Unlike traditional RAG systems with a single retrieval loop, two specialized agents were implemented:

– **Clarification Agent:** This agent orchestrates the initial user interaction phase, focusing on query refinement. It employs specialized Large Language Model (LLM) configurations to perform structured assessments of query clarity ($assessment\_llm$) and, when necessary, generates targeted follow-up questions ($question\_gen\_llm$) to resolve ambiguity or elicit sufficient detail prior to information retrieval.
– **RAG Agent:** Responsible for executing the core Retrieval-Augmented Generation process upon receiving a sufficiently clear query. This agent integrates several components:
  - It utilizes a *Vector Database Retriever* (`vector_db.as_retriever`) for primary information retrieval from a specialized corpus (Chroma DB).
  - It employs an LLM ($rewriter\_llm$) to potentially rephrase the query for optimal retrieval performance against both the vector database and external search engines.

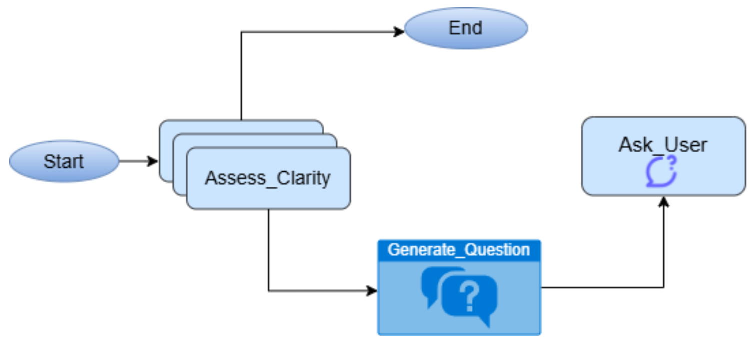

Fig. 2: Clarification Agent Workflow.

- It leverages a structured LLM (*structured_ llm_ grader*) to evaluate the relevance of retrieved document chunks.
- It incorporates an external web search capability via the *Tavily Search Tool* (`TavilySearchResults`) as a fallback or supplementary information source.
- Finally, it synthesizes a comprehensive answer using a generative LLM (*llm*), grounded in the context provided by the retrieved, graded, and potentially web-augmented information.

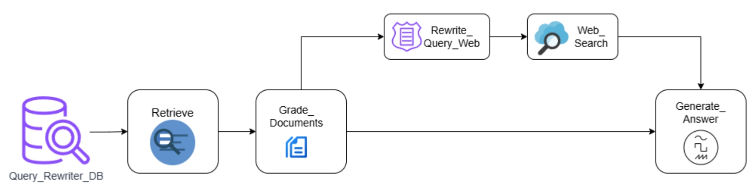

Fig. 3: RAG Agent Workflow.

In aggregate, this multi-agent workflow comprises a total of **two distinct agents**. These agents collectively leverage **seven key functional components**: two external tools (one vector database retriever and one web search tool) and five specialized LLM configurations, each assigned a specific role within the clarification, retrieval, grading, or generation stages of the process. Both agents were implemented using `LangGraph`'s `StateGraph`, with `add_node`, `add_edge`, and `add_conditional_edges` functions [13].

3.5 Query Rewriting

Before sending the user's raw query to the retriever, a Question Rewriter Node reformulates the query to make it clearer and easier to retrieve relevant information.

3.6 Conditional Web Search

Another important part of the system is the structure for conditional web search. If the local database provides enough information, the system answers directly. And if not, it can be extended to perform a real-time web search for additional information. This decision-making logic was handled using a `should_search` function and conditional routing inside the `LangGraph` [13].

3.7 Language Model and Tool Handling

For language understanding and text generation, OpenAI's GPT 4o mini and Gemma3:4b via Ollama was used through the `LangChain`'s `ChatOpenAI` wrapper [18]. Several "tools" were defined, such as: retrieval tool (fetch chunks), search tool (for web search), answer tool (format the answertool). Agents dynamically call these tools based on the need detected from the conversation flow.

3.8 User Interface

To make the system accessible, we built a prototype `Streamlit` UI [23]. Through this interface, users can input their legal queries and receive relevant answers retrieved and generated through the multi-agent RAG system.

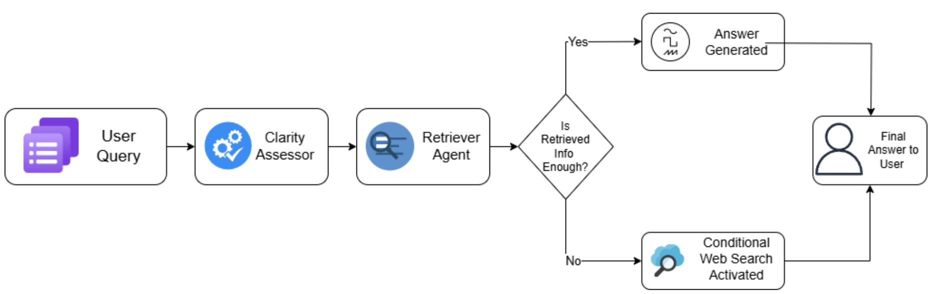

Fig. 4: Detailed Agent Workflow.

3.9 Evaluation Setup

To evaluate performance, a set of 20 diverse legal queries representative of the most common user concerns in the Bangladeshi legal domain were used. These queries include procedural legal actions, statutory interpretations, policy clarifications, and case law queries. The same query set was used across all models to ensure consistency. A comparative testing against two models are included:

- GPT-4o Mini
- Gemma3-4B

3.10 Local Deployment

To address scalability and cost concerns, the system was extended for local deployment using the Ollama framework. This enabled testing with open-source LLMs like Gemma3-4B without dependency on commercial APIs. The local setup runs on a mid to high-end GPU or CPU environment and mirrors the cloud-based agent pipeline.

4 Results

The developed multi-agent RAG system resulted in a functional chatbot capable of addressing legal queries specific to the Bangladeshi context. An example interaction showcasing the system's ability to handle a user query regarding the Right to Information (RTI) Act is presented in Figure 5.

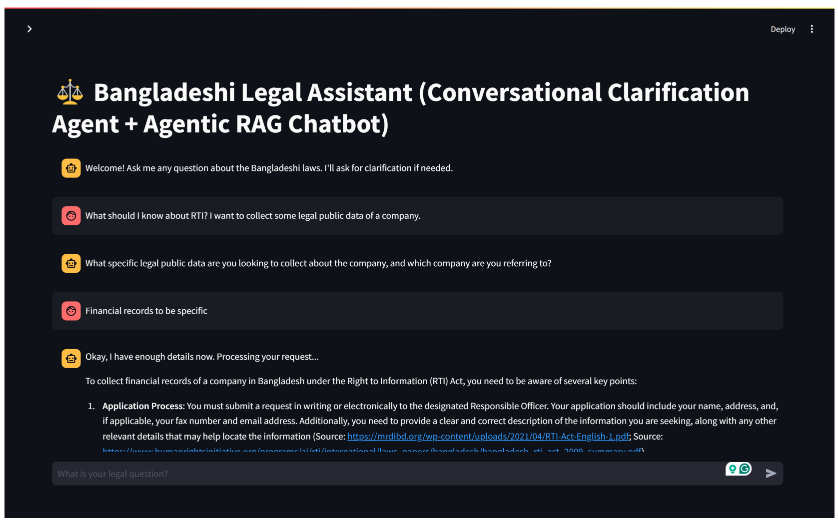

Fig. 5: Example interaction demonstrating a query ("What should I know about RTI? I want to collect some legal public data of a company.") and the corresponding detailed response generated by the multi-agent system.

To quantitatively assess the performance and reliability of the system, the Deepeval framework was employed [7] for evaluation. A curated test suite comprising 20 distinct legal questions, along with expected answers relevant to the laws and legal landscape of Bangladesh, was generated via Deepeval's 'Generate from document' feature. A separate script was used to create the test cases. The evaluation focused on a set of standard RAG metrics designed to measure different facets of the system's retrieval and generation quality. The specific metrics evaluated include:

- **Answer Relevancy:** Measures how pertinent the generated answer is to the user's query. It is calculated as the ratio of relevant statements to the total statements in the answer:

$$\text{Answer Relevancy} = \frac{\text{Number of Relevant Statements}}{\text{Total Number of Statements}}$$

- **Faithfulness:** Assesses whether the generated answer is factually consistent with the retrieved context documents. It is calculated based on the claims made in the answer versus the provided context:

$$\text{Faithfulness} = \frac{\text{Number of Truthful Claims}}{\text{Total Number of Claims}}$$

- **Contextual Precision:** Evaluates the quality of the retrieval component by measuring the proportion of relevant documents among the retrieved set, weighted by their rank:

$$\text{Contextual Precision} = \frac{1}{\text{Number of Relevant Nodes}} \sum_{k=1}^{n} \left(\frac{\text{Number of Relevant Nodes Up to Position } k}{k} \times r_k \right)$$

  where r_k indicates if the node at rank k is relevant.
- **Contextual Recall:** Measures the extent to which the retrieved context contains all the necessary information required to formulate the ideal answer (often compared against a ground truth or expected output):

$$\text{Contextual Recall} = \frac{\text{Number of Attributable Statements}}{\text{Total Number of Statements}}$$

  (Note: Deepeval calculates this based on the expected output).
- **Contextual Relevancy:** Assesses the overall relevance of the retrieved context passages to the input query, calculated as:

$$\text{Contextual Relevancy} = \frac{\text{Number of Relevant Statements}}{\text{Total Number of Statements}}$$

  (Note: Deepeval calculates this based on the input query and the retrieved context).
- **Correctness (GEval):** G-Eval is a framework that uses LLM-as-a-judge with chain-of-thoughts (CoT) to evaluate LLM outputs based on ANY custom criteria. The G-Eval metric is the most versatile type of metric deepeval has to offer, and is capable of evaluating almost any use case with human-like accuracy.
  Usually, a GEval metric will be used alongside one of the other metrics that are more system specific (such as ContextualRelevancyMetric for RAG, and TaskCompletionMetric for agents). This is because G-Eval is a custom metric best for subjective, use case specific evaluation.

Since G-Eval is a two-step algorithm that generates chain of thoughts (CoTs) for better evaluation, in deepeval this means first generating a series of evaluation-steps using CoT based on the given criteria, before using the generated steps to determine the final score using the parameters presented in an LLMTestCase.
(Note: Although GEval is great in many ways as a custom, task-specific metric, it is NOT deterministic.)

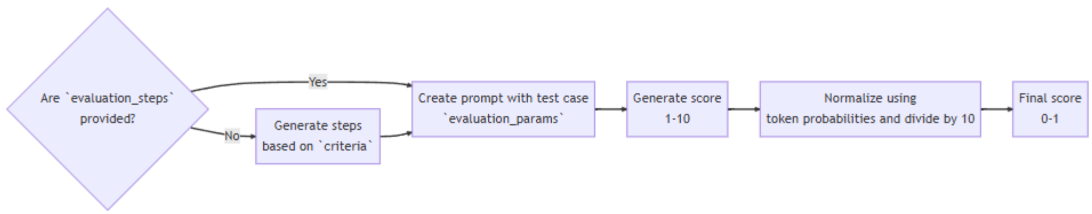

Fig. 6: A flow-chart showcasing the steps to process GEval score.

- **Hallucination:** The hallucination metric uses LLM-as-a-judge to determine whether your LLM generates factually correct information by comparing the actual output to the provided context:

$$\text{Hallucination} = \frac{\text{Number of Contradicted Contexts}}{\text{Total Number of Contexts}}$$

Deepeval utilizes an LLM-as-a-judge approach for these metrics, often providing reasoning alongside the numerical scores.

Using Deepeval, the system was conducted across 20 test case queries. A comparative testing against two models were also included:

- GPT-4o Mini
- Gemma3-4B

The comparisons between The Proposed RAG using GPT-4o-mini and Vanilla RAG using GPT-4o-mini are:

Table 1: Performance comparison of the proposed RAG system using GPT-4o-mini against the Vanilla RAG using GPT-4o-mini. Higher scores indicate better performance for all metrics except for the Hallucination Score, where a lower score is better.

Metric	**Proposed RAG performance (AVG)**	**Vanilla RAG (AVG)**	**Performance difference**	**% improvement**
Answer Relevancy	0.96	0.94	0.01	0.57
Faithfulness	0.90	0.80	0.10	10.1
Contextual Precision	0.96	0.88	0.08	8.45
Contextual Recall	0.93	0.88	0.05	4.66
Contextual Relevancy	0.61	0.54	0.06	6.59
Correctness (Geval)	0.75	0.66	0.09	8.73
Hallucination Score	0.14	0.28	-0.13	13.5

Now, the comparisons between The Proposed RAG using locally run Gemma3-4b via Ollama and Vanilla RAG using Gemma3-4b are:

Table 2: Performance comparison of the proposed RAG system using Gemma3-4b against the Vanilla RAG using Gemma3-4b.

Metric	**Proposed RAG performance (AVG)**	**Vanilla RAG (AVG)**	**Performance difference**	**% improvement**
Answer Relevancy	0.92	0.77	0.14	14.4
Faithfulness	0.70	0.67	0.02	2.53
Contextual Precision	0.89	0.85	0.03	3.44
Contextual Recall	0.96	0.91	0.05	4.69
Contextual Relevancy	0.56	0.51	0.04	4.61
Correctness (Geval)	0.62	0.61	0.01	0.57
Hallucination Score	0.22	0.27	-0.05	5.00

The proposed RAG system using **GPT-4o-mini** outperforms Vanilla RAG using **GPT-4o-mini** in 6 out of 7 metrics, with notable gains in **correctness**, **faithfulness**, and **factual grounding**, making it a more trustworthy and accurate solution.

It has proved that the proposed model has:

- **High Faithfulness (+10.1%)**: This shows the proposed model's responses are more grounded in the source material, a critical feature in factual QA systems.

- **Lower Hallucination Rate**: A significantly lower hallucination score (0.14 vs. 0.28) indicates the proposed system is more reliable and factual, with a 13.5% improvement.
- **Better Contextual Understanding**: Metrics like Contextual Precision and Recall (8.45% and 4.66% gains respectively) suggest better grasp and representation of contextually relevant information.

While the performance gap with **Gemma3-4B** is narrower in some metrics (like **Faithfulness**), the proposed RAG model using **Gemma3-4B** still consistently leads in most areas, particularly in generating more relevant and accurate responses.

It has also proved that the proposed model has:

- **Answer Relevancy (+14.4%)**: The proposed model provides much more relevant answers, which is key in user-facing applications.
- **Higher Contextual Recall and Relevancy**: It captures more meaningful context from source documents.
- **Reduced Hallucination**: A lower hallucination score again indicates greater factual reliability than **Gemma3-4B**.

Across both comparisons:

- The proposed RAG model consistently outperforms other Vanilla RAG models on critical NLP benchmarks.
- It minimizes hallucinations, enhances answer relevancy, and maintains strong contextual fidelity.
- The improvements in faithfulness, precision, and correctness demonstrate it is well-suited for real-world applications requiring accuracy, trustworthiness, and contextual understanding.

5 Discussion

The development of this multi-agent RAG system represents a significant step towards enhancing legal information accessibility in Bangladesh. Our findings, based on a comparative evaluation against baseline models, demonstrate the tangible benefits of employing a sophisticated multi-agent architecture with `LangGraph` to address the complexities of legal query processing.

The core strength of our approach is the division of labor between a Clarification Agent and a RAG Agent is validated by the quantitative results. Unlike monolithic RAG systems, our agentic pipeline consistently outperforms a vanilla RAG setup. The evaluation showed significant improvements with the GPT-4o-mini model, including a **10.1% increase in Faithfulness**, an **8.7% increase in Correctness (GEval)**, and a **13.5% reduction in Hallucination**. These gains underscore the effectiveness of the architecture's components: the Clarification Agent improves input query precision, the rewriter node optimizes retrieval, and the relevance grader filters noise from the context. This multi-step process directly mitigates the risks of generating irrelevant or factually incorrect information, a critical requirement in the legal domain and a known challenge for RAG systems as noted by Feldman et al. [10].

A key contribution of this work is the successful implementation and validation of the system using a locally deployed open-source model, Gemma3-4b, via the Ollama framework. While the absolute performance was lower than with GPT-4o-mini, the agentic structure still provided a clear advantage over a vanilla RAG using the same model, most notably a **14.4% improvement in Answer Relevancy**. This demonstrates that the architectural benefits are model-agnostic and provides a viable path for creating a cost-effective, scalable, and privacy-preserving solution suitable for low-resource settings like Bangladesh, addressing major limitations of relying solely on proprietary, API-based models.

Despite the promising results, several limitations remain. The system's effectiveness is fundamentally tied to the quality and currency of its curated dataset [3, 15, 24]. Furthermore, while our 'Deepeval' evaluation is more robust with comparative analysis, it does not replace the need for rigorous review by human legal experts, which is essential for validating nuanced legal correctness. Finally, the trade-off between the high performance of proprietary models and the accessibility of open-source models highlights the need for careful consideration during large-scale deployment.

Nonetheless, the multi-agent framework presented here offers a flexible and powerful paradigm. By breaking down the complex task into discrete, managed steps, the system achieves a demonstrable improvement in reliability and adaptability, providing a validated blueprint for leveraging advanced AI to address critical information access gaps in specialized domains like Bangladeshi law.

6 Future Work

Building upon the promising comparative results, several avenues for future work can further enhance the capabilities and reliability of this multi-agent RAG system for Bangladeshi legal information.

- **Enhanced Evaluation Framework:** While 'Deepeval' [7] provided strong comparative metrics, a more comprehensive evaluation is crucial.
  - *Multi-faceted RAG Evaluation:* Incorporating frameworks like RAGAS [9] can provide more granular diagnostics of the retrieval and generation components.
  - *Human Expert Validation:* Engaging legal experts in Bangladesh to perform a qualitative review of the system's responses against real-world queries is the ultimate test. This would identify subtle inaccuracies and provide insights beyond automated metrics.
- **Knowledge Base Expansion and Maintenance:** The system's accuracy is intrinsically tied to its knowledge base.
  - *Automated Data Ingestion Pipeline:* Developing a semi-automated pipeline to continuously identify, ingest, and index new laws, amendments, and court judgments [15, 24] is essential for maintaining currency.

  - *Diversification of Sources:* Expanding the corpus to include more case law summaries, legal FAQs, and authoritative commentaries [3] will broaden the system's contextual understanding.
- **Advanced Agent Architectures and Capabilities:** The 'LangGraph' framework [13] allows for further sophistication.
  - *Specialized Legal Analysis Agent:* An agent could be developed to perform more complex legal reasoning, such as identifying analogous case law [22] or interpreting statutes within a specific factual context.
  - *Persistent Memory for Contextual Conversations:* Integrating persistent memory [25] would enable more coherent, multi-turn dialogues, allowing the system to retain user context for more personalized assistance.
- **Model Optimization and Fine-tuning:**
  - *Optimizing Open-Source Models:* Further experimentation with efficient open-source LLMs deployed via 'Ollama' is needed to close the performance gap with proprietary models. This includes exploring quantization and other optimization techniques.
  - *Domain-Specific Fine-tuning:* Fine-tuning an open-source model like Gemma on a curated dataset of Bangladeshi legal text and question-answer pairs could significantly boost its understanding of local legal terminology and reasoning patterns, offering a powerful combination of performance and cost-effectiveness.
- **User Interface and Accessibility:**
  - *Enhanced UI/UX:* Improving the 'Streamlit' interface [23] with features like chat history, source document linking, and user feedback mechanisms will enhance trust and usability.
  - *Multilingual Support:* Developing support for Bengali would dramatically increase the system's accessibility and impact across Bangladesh.
- **Cost-Performance Analysis for Deployment:** A thorough analysis of the cost-performance trade-offs between different models (e.g., GPT-4o-mini vs. a fine-tuned Gemma) is required to develop a sustainable deployment strategy for a public-facing service.

7 Conclusion

Access to clear and accurate legal information remains a significant barrier in Bangladesh. This paper addressed this challenge by designing, implementing, and evaluating a novel Multi-Agent Retrieval-Augmented Generation (RAG) chatbot tailored for the Bangladeshi legal domain. By leveraging the `LangGraph` framework, we developed a sophisticated two-agent architecture that separates query clarification from the core retrieval and generation process.

Our quantitative evaluation demonstrated the system's superiority over standard RAG baselines. The agentic architecture delivered significant improvements in critical metrics, most notably enhancing **Faithfulness**, increasing overall **Correctness**, and reducing **Hallucinations**. Furthermore, we validated this approach on both a leading proprietary model (GPT-4o-mini) and a locally deployed open-source model (Gemma3-4b via Ollama), proving the architecture's versatility and establishing a clear path toward a cost-effective, scalable solution.

The key contribution of this work is the empirical evidence that a structured, multi-agent RAG system provides quantifiable gains in reliability and factual accuracy within a low-resource, specialized domain. This study moves beyond a theoretical proposal to offer a validated blueprint for democratizing legal knowledge in Bangladesh. It showcases an adaptable and robust approach that bridges a critical information gap, offering a promising and practical direction for future legal technology development in similar contexts worldwide.

Data Availability

The primary datasets used in this study consist of publicly available legal documents. Supreme Court judgments are available from the Supreme Court of Bangladesh website [24], Laws of Bangladesh are available from the official legislative portal [15], and legal articles were sourced from BD Law Post [3]. Specific datasets derived or compiled during the research can be made available upon reasonable request to the corresponding author, subject to any applicable licensing or privacy constraints.

References

1. A, K.R., S, K., R, R.: Legal solutions - intelligent chatbot using machine learning. In: 2023 International Conference on Disruptive Technologies (ICDT). pp. 374–378. IEEE (2023). https://doi.org/10.1109/ICDT57929.2023.10150955
2. Amato, F., Fersini, E., Gatta, M., Sciarrone, C.: An intelligent conversational agent for the legal domain. In: Proceedings of the 15th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - KDIR. pp. 236–243. SCITEPRESS – Science and Technology Publications (2023). https://doi.org/10.5220/0012194000003598
3. BD Law Post: Bd law post article. https://www.bdlawpost.com/search/label/Articles?max-results=10, accessed: 2024-07-01
4. Chroma: Chroma documentation. https://docs.trychroma.com/docs/overview/introduction, accessed: 2024-07-01
5. Cirillo, E., Fersini, E., Gatta, M., Muscio, A.: A chatbot for specialized domain. In: Intelligent Systems and Applications. IntelliSys 2023, Lecture Notes in Networks and Systems, vol. 851. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-47721-7_12
6. Cui, J., Cui, M.r., Liu, Z., Chen, B., Yuan, Y., Lin, H., Liu, B., Yao, Y., Tu, T., Yang, L.: Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model. arXiv preprint arXiv:2401.02914 (2024), https://arxiv.org/abs/2401.02914
7. DeepEval Documentation: Evaluation introduction. https://www.deepeval.com/docs/evaluation-introduction, accessed: 2024-07-01
8. Devaraj, P.N., P V, R.T., Gangrade, A., R, M.K.: Legal chatbot using langchain. arXiv preprint arXiv:2311.12719 (2023), https://arxiv.org/abs/2311.12719

9. Es, S., James, J., Espinosa-Anke, L., Schockaert, S.: RAGAS: Automated evaluation of retrieval augmented generation. arXiv preprint arXiv:2309.15217 (2023), https://arxiv.org/abs/2309.15217
10. Feldman, P., Foulds, J.R., Pan, S.: RAGged edges: The double-edged sword of retrieval-augmented chatbots. arXiv preprint arXiv:2403.01193 (2024), https://arxiv.org/abs/2403.01193
11. Isaac, O.J., Nwabueze, J.: The use of chatbots in providing free legal guidance: Benefits and limitations. ResearchGate (2025), further publication details needed for precise citation type. Original note: [Provide more specific publication details if available, e.g., journal, conference]
12. LangChain: Langchain documentation. https://python.langchain.com/docs/how_to/#document-loaders, accessed: 2024-07-01
13. LangGraph: Langgraph documentation. https://langchain-ai.github.io/langgraph/, accessed: 2024-07-01
14. LAW HELP BD: Penal codes section. https://lawhelpbd.com/category/cpc/, accessed: 2024-07-01
15. Laws of Bangladesh: Laws of bangladesh. http://bdlaws.minlaw.gov.bd/, accessed: 2024-07-01
16. Maree, M., Abu-Qauod, R., Tuffaha, B.: Transforming legal text interactions: leveraging natural language processing and large language models for legal support in palestinian cooperatives. Artificial Intelligence and Law (2023). https://doi.org/10.1007/s10506-023-09358-0
17. Nikita, Srivastav, E., Patel, A., Singh, A., Sharma, R., Rana, D.P.: LAWBOT: A smart user indian legal chatbot using machine learning framework. In: 2024 3rd International Conference on Innovative Sustainable Computational Technologies (CISCT). pp. 1–6. IEEE (2024). https://doi.org/10.1109/CISCT61344.2024.10554004
18. OpenAI: OpenAI platform: Embeddings guide. https://platform.openai.com/docs/guides/embeddings, accessed: 2024-07-01
19. PyMuPDF: Pymupdf documentation. https://pymupdf.readthedocs.io/en/latest/, accessed: 2024-07-01
20. Queudot, M., Canton, , Masse, J.J.: Improving access to justice with legal chatbots. Information **11**(5), 276 (2020). https://doi.org/10.3390/info11050276
21. Sharma, M., Trivedi-Raiyani, L., Badlani, A., Muthusamy, V.: Building a legal dialogue system: Development process, challenges and opportunities. arXiv preprint arXiv:2109.12654 (2021), https://arxiv.org/abs/2109.12654
22. Siino, M., Falco, M., Croce, D., Rosso, P.: Large language models for the legal domain: A survey. IEEE Access **13**, 19533–19554 (2025). https://doi.org/10.1109/ACCESS.2025.3533217
23. Streamlit: Streamlit documentation. https://docs.streamlit.io/, accessed: 2024-07-01
24. Supreme Court of Bangladesh: Supreme court of bangladesh. https://www.supremecourt.gov.bd/web/indexn.php, accessed: 2024-07-01
25. Weng, L.: LLM-powered Autonomous Agents. Lil'Log Blog Post. https://lilianweng.github.io/posts/2023-06-23-agent/ (Jun 2023)
26. Xu, N., Wang, K.J., Li, C.Y.: Technology acceptance model for lawyer robots with ai: A quantitative survey. In: HCI International 2022 – Late Breaking Work. HCII 2022. Lecture Notes in Computer Science, vol. 13526. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-22107-7_27