IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 January 2025, accepted 31 January 2025, date of publication 14 February 2025, date of current version 3 March 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3542125

RESEARCH ARTICLE

Legal Query RAG

RAHMAN S. M. WAHIDUR[1], SUMIN KIM[2], HAEUNG CHOI[1], DAVID S. BHATTI[1], AND HEUNG-NO LEE[1], (Senior Member, IEEE)

[1]School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea
[2]Artificial Intelligence Graduate School, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea

Corresponding author: Heung-No Lee (heungno@gist.ac.kr)

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea Government (MSIT) (IITP-2025-RS-2021-II210118, Development of decentralized consensus composition technology for large-scale nodes) and This work was supported by the IITP (Institute of Information & Communications Technology Planning & Evaluation)-ITRC (Information Technology Research Center) grant funded by the Korea Government [Ministry of Science and Information and Communication Technology (ICT)] (IITP-2025-RS-2021-II211835).

ABSTRACT Recently, legal practice has seen a significant rise in the adoption of Artificial Intelligence (AI) for various core tasks. However, these technologies remain in their early stages and face challenges such as understanding complex legal reasoning, managing biased data, ensuring transparency, and avoiding misleading responses, commonly referred to as hallucinations. To address these limitations, this paper introduces Legal Query RAG (LQ-RAG), a novel Retrieval-Augmented Generation framework with a recursive feedback mechanism specifically designed to overcome the critical shortcomings of standard RAG implementations in legal applications. The proposed framework incorporates four key components: a custom evaluation agent, a specialized response generation model, a prompt engineering agent, and a fine-tuned legal embedding LLM. Together, these components effectively minimize hallucinations, improve domain-specific accuracy, and deliver precise, high-quality responses for complex queries. Experimental results demonstrate that the fine-tuned embedding LLM achieves a 13% improvement in Hit Rate and a 15% improvement in Mean Reciprocal Rank (MRR). Comparisons with general LLMs reveal a 24% performance gain when using the Hybrid Fine-Tuned Generative LLM (HFM), the specialized response generation model integrated into the LQ-RAG framework. Furthermore, LQ-RAG achieves a 23% improvement in relevance score over naive configurations and a 14% improvement over RAG with Fine-Tuned LLMs (FTM). These findings underscore the potential of domain-specific fine-tuned LLMs, combined with advanced RAG modules and feedback mechanisms, to significantly enhance the reliability and performance of AI in legal practice. The reliance of this study on a proprietary model as the evaluation agent, combined with the lack of feedback from human experts, highlights the need for improvement. Future efforts should focus on developing a specialized legal evaluation agent and enhancing its performance by incorporating feedback from domain experts.

INDEX TERMS Retrieval-augmented generation, legal query, LLM agent, information retrieval.

I. INTRODUCTION

Recent advancements in AI and NLP have propelled the development of powerful LLMs. These LLMs leverage advanced deep learning techniques, transformer architectures, and extensive amounts of data to provide more efficient responses to user queries. The remarkable versatility demonstrated by models like OpenAI GPT or Meta LLaMA across a wide spectrum of tasks highlights their potential. These models find applications across various fields, including law, medicine, agriculture, coding, and psychology. They often demonstrate their utility without requiring specialized prompts [1]. However, while proprietary models like BloombergGPT [2] in finance and

The associate editor coordinating the review of this manuscript and approving it for publication was Rongbo Zhu.

Med-PaLM [3] in medicine have capitalized on their distinct data accumulations to advance in their respective sectors, the legal domain has a relatively limited number of reliable LLMs. This scarcity of specialized models has hindered the digital transformation of the legal sector [4].

Law serves as a cornerstone in shaping societies, governing human interactions, and upholding justice. Accurate and up-to-date information is essential for legal professionals to make informed decisions. Legal professionals must navigate the complexities of legal language and nuanced interpretations. They also need to address the ever-evolving nature of legislation. These challenges require tailored solutions to effectively meet the unique demands of their field [5]. Current LLMs are primarily trained on general corpora, which limits their access to domain-specific resources. This restriction hinders their ability to effectively utilize comprehensive domain knowledge for practical applications [6]. LLMs also face challenges in expanding their parametric memory, which can result in generating hallucinated information [7]. This issue renders the use of such models risky in high-stakes domains, e.g., in several high-profile incidents, attorneys have been disciplined for filing court documents that referenced fabricated case law produced by AI [8]. Research indicates that general-purpose LLMs frequently hallucinate when responding to legal queries, with an average occurrence rate between 58% and 82% [9]. A promising approach to address these limitations is RAG, introduced by Lewis et al. [10], which integrates external data retrieval into the generative process. RAG aids in reducing hallucinations and facilitates continuous knowledge updates and integration of domain-specific information [11]. However, the conventional RAG method may limit LLMs' adaptability and diminish output quality by introducing irrelevant passages. This occurs because the retrieval model does not consider domain-specific relevance when retrieving passages. Additionally, the generative models lack explicit training on domain knowledge and have limited ability to follow instructions efficiently, resulting in inconsistent responses [12].

To address the above-mentioned constraints identified within the legal domain, this paper introduces a new framework named LQ-RAG. This framework employs a hybrid approach to fine-tune the two principal components of the RAG system: embedding generation module and response generation module separately. These fine-tuned modules are then integrated into the RAG ecosystem and augmented with other RAG modules, such as chunk references, document hybrid retrieval, and multi-document agents, to enhance the performance of the LQ-RAG system. Additionally, an evaluation agent powered by OpenAI GPT-4[1] with a feedback mechanism is introduced to evaluate the response generated by the generative module. If the generated response meets the preset criteria, the agent displays the response as the final output. Otherwise, the agent sends the query to a prompt agent, where prompt engineering is used to make slight adjustments that simplify the query while preserving its main idea. The modified query is then sent back to the RAG to repeat the retrieval and generation process. This recursive feedback mechanism iteratively refines the retrieved documents and generated responses by continuously evaluating answer relevance, context relevance, and groundedness, ensuring greater accuracy and alignment with the legal domain.

The key contributions of this paper are given below.

1) A pioneering RAG framework has been designed to seamlessly incorporate agent-driven recursive feedback processes, creating an innovative pathway to refine response quality and precision.
2) The framework incorporates a custom-built LLM-based evaluation agent designed to independently assess the accuracy and relevance of model-generated responses and trigger answer regeneration when necessary.
3) A fine-tuned embedding LLM and a hybrid fine-tuned generative LLM have been developed. These LLMs provide enhanced generalization, superior domain adaptation, and improved adherence to instructions.
4) Extensive evaluations were performed to assess the performance of the proposed RAG system. The results demonstrate that LQ-RAG consistently outperforms baseline models, highlighting its applicability in the legal domain.

The subsequent sections of this paper are structured as follows. Section II provides background information. Section III reviews pertinent literature. Section IV unveils the architectural framework of the proposed work. Section V presents tasks, baseline LLMs, and evaluation metrics. Section VI discusses and summarizes experimental findings. Section VII presents the conclusions. Section VIII discusses limitations and suggests potential areas for future research. Finally, section IX discusses the acknowledgment that supported this research.

II. BACKGROUND

This section explores key methodologies in NLP, with an emphasis on generative and embedding LLMs, their fine-tuning techniques, and the RAG system.

A. GENERATIVE LLMS AND EMBEDDING LLMS

The advancement of LLMs has given rise to two primary categories: generative LLMs and embedding LLMs. Generative LLMs excel in generating text by utilizing the causal language modeling approach. This technique predicts each new token based on the preceding sequence, also known as auto-regression or next-token prediction. Such a technique makes these LLMs highly effective for producing contextually coherent content. In contrast, embedding LLMs transform text into high-dimensional vector spaces, which is useful for indexing and determining semantic relationships through mathematical operations. These LLMs excel at

[1] https://platform.openai.com/docs/models

identifying semantic similarities between sentences, making them suitable for applications like search engines and recommendation systems [13].

B. LLM FINE-TUNING

Fine-tuning adapts a pre-trained language model to enhance its performance in domain-specific applications. Fine-tuning of a generative LLM employs two methods: Supervised Fine-Tuning (SFT) and Instruction Tuning (IT), each tailored to optimize LLM differently [14]. Fine-tuning offers benefits such as leveraging pre-training knowledge, reducing the need for labeled data, and enhancing model generalization. Additionally, fine-tuning an embedding LLM enriches the semantic representation of embeddings across the training data distribution, thereby enhancing retrieval performance [15]. Empirical observations indicate that the fine-tuning process commonly leads to significant improvements in retrieval evaluation metrics associated with RAG.

C. RETRIEVAL AUGMENTED GENERATION (RAG)

The RAG represents an architectural approach to enhance LLM applications by utilizing customized data sources. It marginalizes the retrieved documents to produce a distribution over the generated text. There are two methods to achieve this distribution: RAG-Sequence and RAG-Token. The RAG-Sequence model uses the same retrieved document to generate the entire response. In contrast, the RAG-Token model utilizes multiple retrieved documents to produce an answer, as shown in equations 1 and 2, respectively [10].

$$\begin{aligned} p_{\text{RAG-Sequence}}(y|x) &\approx \sum_{z \in \text{top-}K(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z) \\ &= \sum_{z \in \text{top-}K(p(\cdot|x))} p_\eta(z|x) \prod_{i=1}^{N} \\ &\quad \times p_\theta(y_i|x,z,y^{1:i-1}) \end{aligned} \tag{1}$$

$$\begin{aligned} p_{\text{RAG-Token}}(y|x) &\approx \prod_{i=1}^{N} \sum_{z \in \text{top-}K(p(\cdot|x))} \\ &\quad \times p_\eta(z|x) p_\theta(y^i|x,z,y^{1:i-1}) \end{aligned} \tag{2}$$

where x is the input sequence, y is the target sequence, and z are the retrieved documents. N denotes the target sequence length. The retriever $p_\eta(z|x)$ with parameters η provides distributions over text passages given x. The generator $p_\theta(y^i \mid x, z, y^{1:i-1})$ with parameters θ generates the current token, y^i, based on previous tokens, $y^{1:i-1}$, x, and z. top-$K(p(\cdot \mid x))$ represents the top-K truncated distribution over retrieved documents. Based on the architectural complexity, the RAG system can be categorized into three types: Naive RAG, advanced RAG, and modular RAG. Naive RAG initially follows a Retrieve-Read framework involving indexing, retrieval, and generation processes. It grapples with challenges such as low retrieval precision and hallucinations [16], [17]. Advanced RAG addresses these shortcomings through refined retrieval processes, enhancing granularity, and optimizing embedding models to improve retrieval quality [18]. Modular RAG further enhances functionality by integrating a search module for similarity retrieval, facilitating adaptable approaches for complex language tasks [19], [20].

III. RELATED WORK

Recent years have seen growing interest in leveraging LLMs for legal tasks. This section reviews several notable studies, emphasizing their key contributions and shared characteristics.

HanFei [21], a fully parameterized legal LLM with 700 million parameters, is pre-trained with large-scale legal documents. It offers features such as legal question-answering, multi-turn dialogue, article generation, and search functionalities. LawGPT_zh [22] is an open-source Chinese legal LLM built on ChatGLM-6B LoRA 16-bit instruction fine-tuning. It integrates legal Q&A datasets and high-quality legal text, enhancing the performance and professionalism of General Language Models (GLM) in the legal domain. Similarly, the LawGPT [23] series, built on Chinese-LLaMA-7B, aims to expand legal terminology and enhance semantic understanding within the legal domain. It achieves this through pre-training on extensive Chinese legal text databases. Subsequent fine-tuning on legal Q&A and judicial datasets further improves the model's effectiveness and comprehension within legal frameworks. LexiLaw [24], fine-tuned on the ChatGLM-6B architecture, aims to provide accurate and reliable legal consultation services for legal professionals, students, and general users. It achieves this by delving into particular legal matters, articles, and case analyses while also providing valuable recommendations. Lawyer LLaMA [6] engaged in continuous pre-training on Chinese-LLaMA-13B and curated multiple instructions fine-tuning datasets to enhance its capability to provide legal counsel. Additionally, it possesses the ability to generate legal articles and offer legal advice. Despite advances in pre-training and fine-tuning on domain-specific data, these models still exhibit hallucinations and biases, making them unreliable. Additionally, the knowledge cutoff date limits their ability to provide current information.

Conversely, recent assessments [25], [26] underscore the efficacy of RAG techniques in addressing question-answering tasks. DISC-LawLLM [27] is an intelligent legal system that integrates LLMs with a retrieval module, aiming to augment the models' capacity to access and utilize external legal knowledge. CBR-RAG [28], an AI-based system for legal question answering, utilizes the initial retrieval stage, indexing vocabulary, and similarity knowledge containers of the Case-Based Reasoning (CBR) cycle to enrich LLM queries with contextual relevance. LexDrafter [29] is an innovative framework tailored for drafting definition articles within legislative documents. It harnesses RAG methods and leverages existing term definitions across diverse legislative documents. This approach streamlines the drafting process

efficiently. Alotaibi et al. [30] propose Knowledge Augmented BERT2BERT (KAB), a question-answering system for Islamic jurisprudential legal questions. KAB combines retrieval-based and generative techniques. It utilizes prior knowledge sources such as previous questions, question categories, and Islamic jurisprudential reference books to provide context for its answers. Hoppe et al. [31] created an intelligent legal advisor for German documents. They demonstrated that Best Matching 25 (BM25) [32] outperforms pre-trained BERT in recall and Mean Average Precision (MAP). However, fine-tuned Dense Passage Retrieval (DPR) [33] excels on the GermanQuAD dataset.

Overall, AI-based legal work is still emerging; however, recent advancements have effectively integrated LLMs and retrieval techniques. This research extends prior work by incorporating fine-tuned LLMs with an agent-based RAG solution equipped with a feedback loop. This approach enhances the accuracy and relevance of legal responses. By advancing these technologies, this paper aims to enhance the reliability and effectiveness of AI in the legal domain.

IV. PROPOSED SYSTEM

Figure 1 illustrates the overall schematic diagram of the proposed LQ-RAG system. The proposed system is organized into two primary parts: Fine-Tuning (FT) Layer and RAG Layer. The FT Layer involves fine-tuning both the embedding LLM and the generative LLM. In contrast, the RAG Layer integrates advanced RAG modules, an evaluation agent, a prompt engineering agent, and a feedback mechanism to ensure the quality and accuracy of the generated responses.

The bottom left quadrant of the FT Layer illustrates the fine-tuning process of the embedding LLM. The top part of the diagram depicts the collection of unstructured legal domain corpora $\mathcal{C}_{\text{legal}}$, sourced from an open-access portal named Library Genesis.[2] A subset of $\mathcal{C}_{\text{legal}}$, denoted as $\mathcal{C}_{\text{sub-legal}}$, undergoes preprocessing. This subset is then utilized by the synthetic data generator, driven by OpenAI GPT-3.5-turbo,[3] to create a query-context pair-based synthetic dataset $\mathcal{D}_{\text{synthetic}}$. This data generation process involves the GPT model breaking down the unstructured text into smaller, manageable chunks and generating questions that are directly related to each chunk. A mapping function organizes the dataset by linking each generated question to a unique identifier and its corresponding text segment. The dataset is then used to fine-tune and evaluate the performance of the embedding LLM. In this research, the GIST Large Embedding v0 model was used for fine-tuning. During fine-tuning, the Multiple Negatives Ranking Loss (MNRL) [34] function is employed to minimize the distance between embeddings of similar sentences while maximizing the distance between embeddings of dissimilar sentences. This approach ensures that the embedding LLM is trained according to the objective function defined in Equation 3.

$$\mathcal{E}(x, y, \theta) = \frac{1}{B}\sum_{i=1}^{B}\left[S(x_i, y_i) - \log\sum_{j=1}^{B} e^{S(x_i, y_j)}\right] \quad (3)$$

where $x = \{x_1, x_2, \ldots, x_B\}$ is the input sequence, $y = \{y_1, y_2, \ldots, y_B\}$ is the target sequence in a training batch of size B, and θ represents the word embedding and neural network parameters used to calculate the similarity score. The similarity score $S(x_i, y_i)$ determines how x_i and y_i are positively related. On the other hand, the similarity score $S(x_i, y_j)$ determines how x_i and y_j are negatively related. In this model, the dot-product scoring introduced in [34] is utilized. The overall fine-tuning process of an embedding LLM is illustrated in Algorithm 1.

The bottom right part of the FT Layer involves the fine-tuning process of the generative LLM. LLaMA-3-8B, a general-purpose pre-trained autoregressive LLM, is utilized, where the model is represented as $p_\eta(y \mid x)$, parameterized by η with the input sequence x and target sequence y. Initially, two distinct datasets are collected and preprocessed: a domain-specific Q&A dataset $\mathcal{D}_{\text{QA}}$ and a general-purpose instruction dataset $\mathcal{D}_{\text{Instr}}$. Each dataset can be represented as input-target sequence pairs: $\mathcal{D} = \{(x_i, y_i)\}_{i=1,\ldots,N}$, where each target sequence $y_i = \{y_i^t\}_{t=1,\ldots,T_i}$ is a combination of T_i tokens. The pre-trained LLM undergoes separate fine-tuning processes with the mentioned datasets using a technique called Low-Rank Adaptation (LoRA) [35]. This technique facilitates the fine-tuning process by reducing the number of trainable parameters. Using LoRA, the weight update from pre-trained weights η_0 to $\eta' = \eta_0 + \Delta\eta$ is replaced by an update to $\eta'(\Theta) = \eta_0 + \Delta\eta(\Theta)$, where Θ is a set of parameters whose size is much smaller than $|\eta|$. Consequently, the fine-tuned LLM can acquire domain-specific knowledge and enhance its ability to follow instructions while minimizing the utilization of computational resources. This fine-tuning process aims to optimize the log-likelihood objective through gradient updates, as identified in Equation 4.

$$\mathcal{G}(\Theta) = \max_{\Theta}\sum_{(x,y)\in\mathcal{D}}\sum_{t=1}^{T}\log\left[p_{\eta'(\Theta)}(y^t \mid x, y^{1:t-1})\right] \quad (4)$$

Here, T denotes the number of tokens in y and $y^{1:t-1}$ represents the set of tokens from y^1 to y^{t-1}. Following the fine-tuning steps, the resulting models are combined together using the linear merging method to create the desired Hybrid Fine-tuned Generative LLM (HFM). Finally, the performance of the HFM is evaluated using domain-specific evaluation datasets. The complete fine-tuning process of the HFM is depicted in Algorithm 2.

The top part of the diagram, RAG-Layer, outlines the core workflow of the LQ-RAG system. The top left section of the diagram involves the utilization of the remaining unstructured legal corpora $\mathcal{C}_{\text{rem}}$, as an external knowledge source through a process known as data ingestion. This process employs parallel workers to efficiently convert the data into document

[2]https://libgen.is/

[3]https://platform.openai.com/docs/models

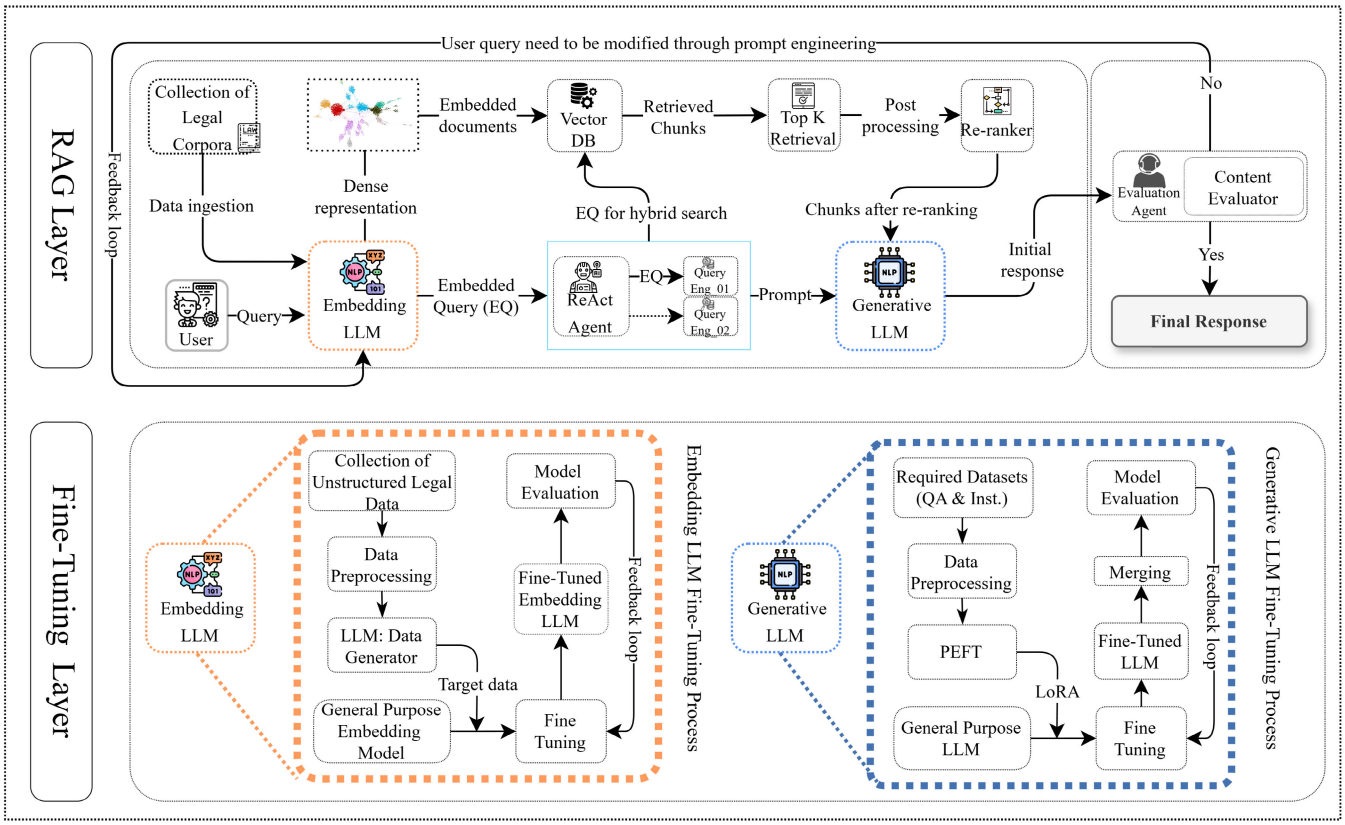

FIGURE 1. The schematic diagram of the proposed legal query RAG. The diagram is divided into two main components: Fine-tuning (FT) layer and RAG layer. The FT layer focuses on fine-tuning processes of embedding LLM and generative LLM. On the other hand, the RAG layer incorporates different RAG modules with fine-tuned LLMs, an evaluation agent, and a feedback system designed to enhance the accuracy and quality of the generated responses.

objects. Subsequently, these documents are segmented into smaller text chunks and processed through the fine-tuned embedding LLM to generate d-dimensional real-valued vectors, denoted as $E_{\text{document}} \in \mathbb{R}^{N \times d}$. An index is then built using Facebook AI Similarity Search (FAISS) [36] for all the $\mathcal{C}_{\text{rem}}$ passages that will be used in retrieval. These vectors are subsequently stored in a vector database, referred to as $\mathcal{DB}_{\text{vector}}$. When a user submits a query q, it is processed by the unified fine-tuned embedding LLM designed to handle text chunks. The embedding LLM generates vectors for the query, represented as $E_{\text{query}} \in \mathbb{R}^{N \times d}$. These query vectors are then sent to a Reasoning and Action (ReAct) [37] agent that selects an appropriate query engine tool to retrieve highly relevant text chunks from the external knowledge source through a search mechanism. The retrieval process employs a hybrid search approach integrating BM25 and DPR techniques to enhance search precision. BM25 is a fundamental non-parametric lexical method that calculates document relevance with Term Frequency (TF) and Inverse Document Frequency (IDF). On the other hand, the DPR retrieves K number of highly relevant passages $\mathcal{C}$ from the vector space. The similarity score between the q and the $\mathcal{C}$ can be defined as the dot product of their vectors. The hybrid retriever performs both retrieval processes and combines their results. It then re-ranks the findings to deliver a nuanced and comprehensive set of documents $\mathcal{C}^*$. Once the $\mathcal{C}^*$ is retrieved, it is forwarded to the post-processing unit for optionally scoring and re-ranked by the re-ranker denoted as $\mathcal{C}_{\text{re-ranked}}$. The fundamental concept focuses on prioritizing relevant document records to reduce document volume. This approach addresses the challenge of expanding context windows during retrieval. Following this, a prompt p that contains system instructions i, the user query q, and the re-ranked retrieved-context $\mathcal{C}_{\text{re-ranked}}$, is fed into the generative LLM, to synthesize the initial response r.

Finally, an evaluation agent $\mathcal{A}_{\text{evaluation}}$, powered by GPT-4, assesses answer relevance, context relevance, and groundedness for each query based on its related response. This system evaluates response quality from the HFM model, utilizing the Chain-of-thought (CoT) [38] process to ensure thorough assessment. First, the model retrieves context chunks relevant to the user's query to verify that only pertinent information is used, reducing the risk of irrelevant details causing hallucinations. For groundedness, it breaks down the response into distinct claims, searching the retrieved context for supporting evidence to ensure factual accuracy. Finally, answer relevance is checked by aligning the response directly with the user's original question to confirm it effectively addresses the intended query. This structured, sequential approach promotes accuracy, factual

grounding, and relevance in responses. In short, if r meets the criteria defined by $\mathcal{A}_{\text{evaluation}}$, it is processed as the final output. Otherwise, q enters a feedback loop, where prompt engineering is applied to modify the query using a prompt engineering agent, repeating the retrieval and generation process. This open-source, LLM-based agent is designed for seamless prompt engineering, enabling efficient query transformation optimized for complex legal question-answering tasks. The entire working process of the proposed LQ-RAG is depicted in Algorithm 3.

Algorithm 1 Embedding LLM Fine-Tuning Process

Constants: Loss function MNRL, Evaluator Eval, Learning Rate η
Input: $\mathcal{C}_{\text{sub-legal}}$
Output: Trained LLM network parameters θ_{global}

Data Collection and Preprocessing:
$\mathcal{D}_{\text{synthetic}} \leftarrow \text{LLM}(\mathcal{C}_{\text{sub-legal}})$
$\mathcal{D}_{\text{synthetic}} \in \{\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{eval}}\}$
Initialize & Fine-Tune Embedding Model:
Baseline-Model $M \leftarrow$ Pre-trained embedding LLM
$L_{\text{train\_embed}} \leftarrow \text{DataLoader}(\mathcal{D}_{\text{train}}, batch\_size)$
for each epochs **do**
  for each batch $(x^{(i)}, y^{(i)}) \in L_{\text{train\_embed}}$ **do**
    Forward Pass:
    $\hat{y}^{(i)} \leftarrow f(x^{(i)}; \theta)$
    Compute Loss:
    $L \leftarrow \text{MNRL}(\hat{y}^{(i)}, y^{(i)})$
    Backward Pass:
    Compute gradients $\nabla_\theta L$
    Update Weights:
    $\theta_{\text{local}} \leftarrow \theta - \eta \cdot \nabla_\theta L$
  end for
  $\theta_{\text{global}} \leftarrow \text{Update}(\theta_{\text{local}}, \theta_{\text{global}})$
end for
Return θ_{global}

V. TASKS, BASELINE LLMS AND EVALUATION METRICS

This section presents an overview of the study, outlining the datasets, baseline LLMs, and evaluation metrics employed for performance assessment.

A. TASKS DESCRIPTION

This paper focuses on six NLP tasks: (1) text classification, (2) multiple-choice, (3) sentence completion, (4) complex task understanding, (5) information retrieval, and (6) question answering. Each task utilizes specific datasets tailored to its requirements, as summarized in Tables 1 and Table 2. Seven datasets listed in Table 1 are employed for the information retrieval task, created in-house using book corpus data to provide a comprehensive resource for retrieval experiments. The remaining tasks: text classification, multiple-choice, sentence completion, complex task understanding, and question answering are addressed with datasets summarized in Table 2. In the question answering task, both open-domain and closed-domain retrieval scenarios are explored. An open-domain query refers to a query where relevant information is available within a broad, diverse knowledge base, whereas a closed-domain query refers to a query where relevant information is limited or absent within such a domain.

Algorithm 2 Generative LLM Fine-Tuning, & Merging Process

Constants: Loss, LoRA Parameters (rank, α), Learning Rate η
Input: Training Dataset $\in \{\mathcal{D}_{\text{QA}}, \mathcal{D}_{\text{Instr}}\}$, Eval Dataset $\mathcal{D}_{\text{eval}}$

Initialize LoRA & Fine-Tune Generative Model:
Baseline-Model $M \leftarrow$ Pre-trained generative LLM
for each trainable layer l in M **do**
  $A_l \leftarrow \text{Random Initialize}(d_{\text{in}}, \text{rank})$
  $B_l \leftarrow \text{Zeros Initialize}(\text{rank}, d_{\text{out}})$
  Integrate A_l and B_l into M as M_{LoRA}
end for
Scale LoRA layers by α: $M_{\text{LoRA}} \leftarrow \alpha \cdot (A_l \cdot B_l)$
for each epoch **do**
  for each batch$(x^{(i)}, y^{(i)}) \in$ Dataset **do**
    Forward Pass:
    $\hat{y}^{(i)} \leftarrow M_{\text{LoRA}}(x^{(i)})$
    Compute Loss:
    $L \leftarrow \text{Loss}(\hat{y}^{(i)}, y^{(i)})$
    Backward Pass:
    Compute gradients $\nabla_{A_l,B_l} L$
    Update LoRA Parameters:
    $A_l \leftarrow A_l - \eta \cdot \nabla_{A_l} L$
    $B_l \leftarrow B_l - \eta \cdot \nabla_{B_l} L$
  end for
  Update Weights
end for
$M_{\text{QA}} \leftarrow M_{\text{LoRA}}(M, \mathcal{D}_{\text{QA}})$
$M_{\text{Instr}} \leftarrow M_{\text{LoRA}}(M, \mathcal{D}_{\text{Instr}})$
Model Merging:
$M_{\text{merged}} \leftarrow \text{Linear Merging}(M_{\text{QA}}, M_{\text{Instr}})$

Algorithm 3 Response Generation Process

Constants: Embedding LLM M_{e}, Generative LLM M_{g}
Input: $\mathcal{C}_{\text{rem}}$, User Query q
Output: Final Response r

Data Ingestion:
$E_{\text{document}} \leftarrow M_{\text{e}}(\text{Data Ingest}(\mathcal{C}_{\text{rem}}))$
$\text{index} \leftarrow \text{FAISS}(E_{\text{document}})$
$\mathcal{DB}_{\text{vector}} \leftarrow \text{Store}(\text{index})$
Query Processing:
$E_{\text{query}} \leftarrow M_{\text{e}}(q)$
$\mathcal{C}^* \leftarrow \text{Hybrid Retrieval}(E_{\text{query}}, \mathcal{DB}_{\text{vector}})$
$\mathcal{C}_{\text{re-ranked}} \leftarrow \text{Re-Ranker}(\mathcal{C}^*)$
$r \leftarrow M_{\text{g}}(q, \mathcal{C}_{\text{re-ranked}})$
Evaluation and Feedback Loop:
$\text{Evaluation\_result} \leftarrow A_{\text{evaluation}}(r)$
if Evaluation_result $\subseteq$ criteria bounded **then**
  Return r
else
  $n \leftarrow 0$
  while Evaluation_result $\not\subseteq$ criteria bounded **and** $n \leq N$ **do**
    $n \leftarrow n + 1$
    $q_{\text{modified}} \leftarrow \text{ModifyQuery}(q)$
    $r \leftarrow M_{\text{g}}(q_{\text{modified}}, \mathcal{C}_{\text{re-ranked}})$
    $\text{Evaluation\_result} \leftarrow \mathcal{A}_{\text{evaluation}}(r)$
  end while
  Return r
end if

TABLE 1. Overview of datasets used for training and evaluating embedding-based large language models in the information retrieval task.

Dataset name	#of documents	#of tokens
Training dataset	1,803	1,279,145
Evaluation dataset 01	123	53,122
Evaluation dataset 02	154	102,657
Evaluation dataset 03	910	626,218
Evaluation dataset 04	145	75,774
Evaluation dataset 05	255	162,441
Evaluation dataset 06	345	219,674

B. BASELINE LLMS

This section explores the LLMs that serve as the foundation of this study. The encapsulated model configurations of all baseline LLMs are delineated in Table 3, 4 & 5. In these tables, (B) represents the number of parameters in billions.

ColBERT [48] enhances retrieval efficiency by leveraging deep language models that independently process queries and documents. LLM-Embedder [49] integrates key retrieval capabilities to boost performance across various tasks, ranging from knowledge-intensive processing to long-context modeling. BGE Embedding [50] utilizes RetroMAE, a novel retrieval-oriented pretraining paradigm based on a masked auto-encoder. GISTEmbed [51] introduces a novel approach to enhance in-batch adverse selection during contrastive training, mitigating biases and noise inherent in traditional techniques. LLaMA [52], [53] utilizes the decoder component of a transformer architecture, with attention layers accessing only preceding words in sentences at each stage. The architectural details of the selected LLaMA models are summarized in Table 4. Flan-T5 [54] leverages pre-trained T5 encoder-decoder transformer architectures and employs fine-tuning techniques to enhance performance. The architectural details of the selected models are summarized in Table 5.

C. EVALUATION METRICS

This section outlines the evaluation metrics deployed in the present study.

1) HIT RATE

Hit Rate (HR) [55] quantifies the ratio of queries in which the correct answer is present among the top-k retrieved documents. It is a common metric for evaluating retrieval-based models and search systems. The HR can be defined as:

$$\mathrm{HR} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{d_i \in D_{\mathrm{true}}(q_i)\} \tag{5}$$

where N is the total number of queries, d_i is the retrieved document, $D_{\mathrm{true}}(q_i)$ is the set of true relevant documents for query q_i, and $\mathbf{1}\{d_i \in D_{\mathrm{true}}(q_i)\}$ is the indicator function that returns 1 if d_i is in the set $D_{\mathrm{true}}(q_i)$ and 0 otherwise.

2) MEAN RECIPROCAL RANK

Mean Reciprocal Rank (MRR) [56] is particularly useful for assessing the performance of ranking algorithms in information retrieval. This metric evaluates system precision by identifying the highest-ranked relevant document for each query and calculating the mean reciprocal rank across all queries, defined as:

$$\mathrm{MRR} = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\mathrm{rank}_i} \tag{6}$$

where N is the total number of queries and rank_i is the rank position of the first relevant document for the i-th query.

3) COSINE SIMILARITY

Cosine similarity (S) [57] measures the similarity between two vectors by calculating the cosine of the angle between them and It is commonly used in natural language processing to assess the similarity between text embeddings or document vectors, defined as:

$$\mathrm{S}(\mathbf{C}, \mathbf{A}) = \frac{\mathbf{C}\cdot\mathbf{A}}{\|\mathbf{C}\|\|\mathbf{A}\|} \tag{7}$$

here, $\mathbf{C}$ and $\mathbf{A}$ are two vectors, and $\|\mathbf{C}\|$ and $\|\mathbf{A}\|$ denote their magnitudes. Cosine similarity reflects a relative between sentences and can gauge how closely related two pieces of text are in terms of their content.

4) ANSWER RELEVANCE

Answer Relevance (AR) [56] measures the degree to which the generated answer accurately addresses the given query. This metric helps evaluate the quality and pertinence of model-generated responses in language model research. AR can be defined as:

$$\mathrm{AR}(Q, A) = \frac{1}{N}\sum_{i=1}^{N} f_{\mathrm{score}}(Q_i, A_i) \tag{8}$$

where N is the total number of queries, A_i is the answer for the i-th query, Q_i is the i-th query. The function $f_{\mathrm{score}(Q_i,A_i)}$ is used to evaluate the relevance of the answer A_i with respect to the query Q_i, define as:

$$f_{\mathrm{score}(Q_i,A_i)} \in [0, 1]$$

with 0 denoting not relevant and 1 denoting highly relevant.

5) CONTEXT RELEVANCE

Context Relevance (CR) [56] measures how well the retrieved context fits the given query. This metric is crucial for evaluating the context-awareness of models, particularly in tasks requiring contextual understanding, defined as:

$$\mathrm{CR}(Q, C) = \frac{1}{N}\sum_{i=1}^{N} f_{\mathrm{score}}(Q_i, C_i) \tag{9}$$

where N is the total number of queries, Q_i is the i-th query for question, C_i is the context which model retrieved.

TABLE 2. Summary of datasets used for training and evaluation of generative LLMs.

Dataset name	Description	Task type	Count	Percentage
Legal_QA [39]	This task involves question answering using legal data from multiple domains, focusing on practical case scenarios and legal principles. The dataset primarily consists of English-language content.	Question answering	97,500	63.39%
Alpaca_cleaned [40]	This dataset is a refined version of the original Alpaca dataset. It supports instruction-tuning for language models, enhancing their ability to follow instructions accurately.	Question answering	52,800	34.33%
SQuAD_V2 [41]	This dataset is an enhanced version of SQuAD, designed for reading comprehension tasks where models must answer questions based on passages and detect when no answer is supported.	Question answering	1,000	0.65%
TruthfulQA [42]	This dataset includes two tasks: generating truthful, informative answers to questions and evaluating the truthfulness and relevance of those answers.	Question answering	817	0.53%
Law Stack Exchange (LSE) [43]	This task involves multi-label classification of legal Q&A from a forum. The dataset includes formal, theoretical, or hypothetical legal inquiries, with questions tagged by topics like "copyright" or "criminal law."	Text classification	638	0.41%
BIG-Bench Hard (BBH) [44]	This is a benchmark dataset focusing on 23 challenging tasks from the BIG-Bench suite. These tasks require multi-step reasoning and have been difficult for language models to outperform human raters.	Complex task understanding	250	0.16%
Canada Tax Court Outcomes (CTCO) [45]	This task classifies excerpts from Tax Court of Canada decisions to determine if the appeal outcome is stated, labeling it as "allowed," "dismissed," or "other" if unclear.	Text classification	250	0.16%
MMLU International Law [46]	This task involves the application of legal principles to areas of international law such as human rights, sovereignty, law of the sea, and the use of force, focusing on classifying and assessing the understanding of these concepts through multiple-choice questions.	Multiple choices	121	0.08%
Hellaswag [47]	This is a benchmark dataset for commonsense NLI, containing contexts with multiple possible endings. The task is to select the most plausible completion, which is straightforward for humans but challenging for models.	Sentence completion	100	0.07%
MMLU Professional Law [46]	This task involves the application of legal principles to areas such as torts, criminal law, contracts, property, and evidence, focusing on classifying and assessing the understanding of these concepts through multiple-choice questions.	Multiple choices	100	0.07%
Abercrombie [45]	This task involves classifying a mark's distinctiveness based on its relationship with the product or service it represents, determining whether it is generic, descriptive, suggestive, arbitrary, or fanciful.	Text classification	99	0.06%
Contract QA (CQA) [45]	This task involves binary classification to determine whether a contractual clause contains a particular type of content, such as confidentiality, arbitration, or compliance-related provisions.	Text classification	88	0.06%
Legal Reasoning Causality (LRC) [45]	The classification task involves analyzing judicial opinions in labor discrimination cases to label whether judges relied on statistical or direct evidence to determine causal links between the plaintiff's characteristics and the contested decision.	Text classification	59	0.04%

TABLE 3. Encapsulation of model configurations in baseline embedding LLMs.

Model name	Model version	Architecture	#Params(B)	Embed. dimension	Intermediate size
ColBERT	ColBERTv2	HF_ColBERT	0.11	768	3,072
LLM-Embedder	LLM-Embedder	Bert Model	0.10	1,024	3,072
BGE Embedding	Small-en-v1.5	Bert Model	0.03	384	1,536
	Base-en-v1.5	Bert Model	0.10	768	3,072
	Large-en-v1.5	Bert Model	0.33	1,024	4,096
GISTEmbed	Small-Embedding-v0	Bert Model	0.03	384	1,536
	GIST-Embedding-v0	Bert Model	0.10	768	1,536
	Large-Embedding-v0	Bert Model	0.33	1,024	4,096

TABLE 4. Concise overview and comparative analysis of LLaMA model architectures.

Model name	Architecture	#Heads	#Layers	Embed. dimension	#Params(B)	#Pretrain token(B)	Vocab size
LLaMA-2-7B	decoder only	32	32	4,096	7	2,000	32,000
LLaMA-2-13B	decoder only	40	40	5,120	13	2,000	32,000
LLaMA-3-8B	decoder only	32	32	4,096	8	15,000	128,256

TABLE 5. Summarized architecture and comparative analysis of FLAN-T5 models.

Model name	Architecture	#Heads	#Layers	Embed. dimension	#Params(B)	Model checkpoint
FLAN-T5 small	encoder-decoder	6	8	512	0.08	google/flan-t5-small
FLAN-T5 base	encoder-decoder	12	12	768	0.25	google/flan-t5-base
FLAN-T5 large	encoder-decoder	16	24	1,024	0.78	google/flan-t5-large
FLAN-T5 XL	encoder-decoder	32	24	2,048	2.85	google/flan-t5-xl

6) GROUNDEDNESS

Groundedness (G) [58] assesses the veracity of a model by evaluating its ability to differentiate between factual and hallucinatory input. This metric is used to ensure that generated responses are based on credible information. The Groundedness is defined as:

$$G(A, C) = \frac{1}{N} \sum_{i=1}^{N} f_{\text{score}}(A_i, C_i) \tag{10}$$

where N is the total number of queries, A_i is the answer, C_i is the retrieved context, and scores how well A_i is grounded in C_i.

7) ACCURACY

Accuracy (Acc) [56] evaluates whether the answer contains accurate and verified information, ensuring the reliability and validity of the generated response. This metric is generally used to assess the overall correctness of a model's predictions, as defined below expression:

$$Acc = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \times 100\% \tag{11}$$

T_P is True-Positive samples and T_N is True-Negative samples. Then, F_P is False-Positive samples and F_N is False-Negative samples.

8) EXACT MATCH

Exact Match (EM) [58] evaluates the accuracy of a model by checking if the predicted answer exactly matches the true answer. This metric is particularly relevant for tasks requiring precise answer extraction, such as question answering. The Exact Match can be defined as:

$$EM(Q, A) = \begin{cases} 1 & \text{if } A_{\text{pred}} = A_{\text{true}} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Q is the query, A_{pred} is the predicted answer, and A_{true} is the true answer.

9) BLEU SCORE

The BLEU score [59] evaluates machine-translated text quality using n-gram precision and a brevity penalty for overly short translations. This metric is commonly used in machine translation and text generation tasks, as follows:

$$\text{BLEU Score} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{13}$$

where BP is the brevity penalty, w_n is the weight for each n-gram precision p_n, and N is the maximum n-gram length.

10) ROUGE SCORE

The ROUGE score [60] measures the similarity between machine-generated long and short it and reference summaries using overlapping n-grams to calculate recall. It is generally used in summarization tasks to evaluate the quality of generated summaries, as follows:

$$\text{ROUGE-N}_R = \frac{\sum_{w \in \text{gen}} \min(\text{Count}_{\text{m-gen}}(w), \text{Count}_{\text{ref}}(w))}{\sum_{w \in \text{ref}} \text{Count}_{\text{ref}}(w)} \tag{14}$$

where *gen* represents the generated summary and *ref* represents the reference summaries.

VI. EXPERIMENT AND EVALUATION

This section details the experimental setup and provides evaluation results, categorized into three parts: (i) Embedding LLM (ii) Generative LLM and (iii) LQ-RAG system.

A. EMBEDDING LLM

The experiment described herein was designed to fine-tune the embedding LLM and compare its performance with previously established baselines, as outlined in Section V-B.

In this paper, the GIST Large Embedding v0 model from Hugging Face[4] was employed for fine-tuning. The fine-tuning process utilized the model fitting API provided by

[4]https://huggingface.co/

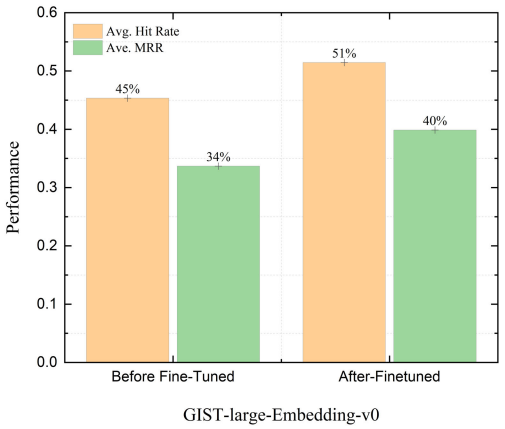

FIGURE 2. Performance evaluation of the GIST-large-embedding-v0 model before and after fine-tuning.

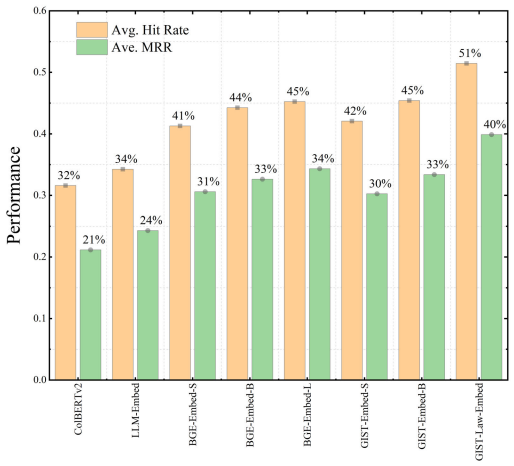

FIGURE 3. The performance evaluation avg. Hit rate & MRR for different embedding models.

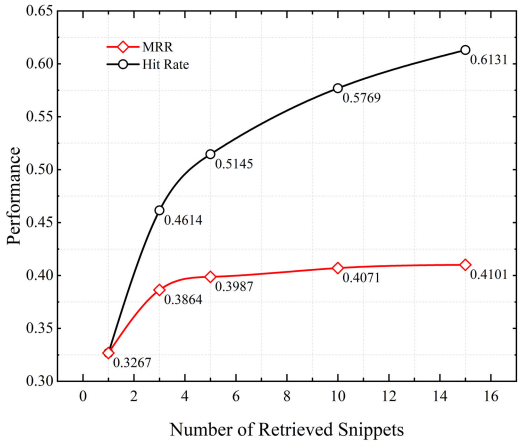

FIGURE 4. The performance evaluation of the hit rate & MRR on different K values.

LlamaIndex[5] from sentence transformers. Throughout the training phase, batch sizes of 8 and 10, along with epoch sizes ranging from 3 to 15, were iteratively tested. During the experiment, the performance of the GIST Large Embedding v0 model was evaluated both before and after fine-tuning by measuring Hit Rate and MRR. The evaluation results, shown in Figure 2, reveal that the fine-tuned model demonstrates improved performance compared to the pre-trained model. After fine-tuning, the model exhibited a 13% improvement in average Hit Rate and a 15% improvement in average MRR, indicating enhanced generalization across different corpora.

To extend this experiment, the performance of the fine-tuned model, GIST-Law-Embed, was compared with other baseline models. The experimental results, shown in Figure 3, indicated that GIST-Law-Embed significantly outperformed all other baseline LLMs in terms of average Hit Rate and average MRR scores. The GIST-Law-Embed model achieved the highest average Hit Rate of 51% and an average MRR of 40% under top K = 5. This model not only excelled in average performance but also maintained the highest scores for each document, demonstrating its robustness and consistency across various datasets. The overall performance in retrieving relevant information is summarized in Table 6 and Table 7.

Additionally, the large versions of the BGE and GIS-TEmbed series consistently outperformed their small and base counterparts, highlighting that increasing model size positively impacts retrieval capabilities. Furthermore, models with domain-specific tuning, such as GIST-Law-Embed, exhibited enhanced retrieval performance, as evidenced by their higher Hit Rate and MRR scores. This trend underscores the importance of model size and domain-specific tuning in achieving superior retrieval performance.

One key aspect of this analysis is RAG's ability to retrieve relevant information for the given context. The impact of the number of retrieved snippets (referred to as top-k) was examined and is presented in Figure 4. As documents are segmented into small chunks during the index's construction, these snippets may represent distinct sections of the original document, offering supplementary information conducive to answer generation. Given this consideration, k = 15 snippets were selected for subsequent experiments in this paper, as it successfully retrieves the original passage more than 60% of the time without significantly augmenting the input prompt size. Consequently, increasing the number of retrieved snippets consistently enhances RAG's retrieval from the original context. This observation underscores the importance of optimizing the number of snippets to balance retrieval accuracy.

B. GENERATIVE LLM

This section investigates the effectiveness of fine-tuning a generative LLM and assesses its performance against established baseline models detailed in Section V-B.

[5]https://docs.llamaindex.ai/en/stable/

TABLE 6. Summary of model performance in information retrieval: Hit rate analysis.

Model name	Hit Rate @ Top_K						
	Avg. score	Eval. data 1	Eval. data 2	Eval. data 3	Eval. data 4	Eval. data 5	Eval. data 6
ColBERTv2	0.3160	0.3185	0.4093	0.2423	0.4040	0.3004	0.2217
LLM-Embedder	0.3424	0.3552	0.4345	0.2505	0.4160	0.3124	0.2858
BGE Embedding small	0.4129	0.4208	0.4843	0.3396	0.4680	0.3905	0.3741
BGE Embedding base	0.4426	0.4431	0.5144	0.3857	0.5064	0.4121	0.3942
BGE Embedding large	0.4524	0.4500	0.5180	0.3897	0.5096	0.4208	0.4265
GISTEmbed small	0.4207	0.4158	0.4892	0.3494	0.4936	0.4089	0.3676
GISTEmbed base	0.4541	0.4468	0.5296	0.3847	0.5144	0.4469	0.4021
GISTEmbed large	0.4534	0.4634	0.5175	0.3792	0.5160	0.4382	0.4062
GIST-Law-Embed	**0.5145**	**0.5050**	**0.5741**	**0.4836**	**0.5408**	**0.5050**	**0.4788**

TABLE 7. Summary of model performance in information retrieval: MRR analysis.

Model name	MRR @ Top_K						
	Avg. score	Eval. data 1	Eval. data 2	Eval. data 3	Eval. data 4	Eval. data 5	Eval. data 6
ColBERTv2	0.2116	0.2079	0.2763	0.1546	0.2845	0.2079	0.1385
LLM-Embedder	0.2430	0.2380	0.2983	0.1715	0.3135	0.2380	0.1987
BGE Embedding small	0.3059	0.3050	0.3486	0.2412	0.3589	0.3050	0.2770
BGE Embedding base	0.3263	0.3172	0.3691	0.2739	0.3880	0.3172	0.2926
BGE Embedding large	0.3433	0.3337	0.3855	0.2790	0.4056	0.3337	0.3224
GISTEmbed small	0.3025	0.2947	0.3451	0.2418	0.3709	0.2947	0.2677
GISTEmbed base	0.3338	0.3223	0.3796	0.2744	0.4085	0.3223	0.2959
GISTEmbed large	0.3369	0.3339	0.3742	0.2705	0.4091	0.3339	0.2999
GIST-Law-Embed	**0.3987**	**0.3873**	**0.4415**	**0.3592**	**0.4411**	**0.3873**	**0.3756**

In this paper, LLaMA-3-8B is employed as the baseline model. Fine-tuning LLaMA-3-8B model requires significant memory and processing resources. To ensure compatibility and optimization, the PEFT [61], was employed. To improve inference speed and reduce the model size, 4-bit quantization approach with BitsAndBytesConfig [62] was utilized. Additionally, Key parameters for efficient training were set using the training arguments configuration. In line with these optimizations, validation performance was monitored to mitigate overfitting, early stopping was applied, and weight decay was used for regularization to improve generalization. Finally, the SFTTrainer [63] object from the trl library[6] was instantiated to manage the entire training process. To comprehensively evaluate the performance of the Hybrid Fine-tuned Generative LLM (HFM), assessments were conducted across two distinct groups. The first group focused on reasoning, commonsense sensing, language understanding, and question-answering tasks, while the second group was dedicated to legal domain-specific tasks. Figure 5 shows the comparison results between these two groups. In the first group, the performance of HFM was improved by 9%, while in the second group, the performance was improved by 38%, based on the average performance score of 11 different datasets. This result signifies that by fine-tuning and merging, the model performs better across both general and task-specific domains. The detailed evaluation results for both groups are summarized in Table 8 and Table 9.

The experimental results for reasoning and commonsense tasks demonstrated that the HFM model significantly outperformed all other models, achieving the highest scores across all metrics. The HFM model achieved an Exact Match (EM) score of 0.65 $\pm$ 0.01 in the BBH dataset, surpassing the State-of-the-Art (SOTA) model, LLaMA-3-8B, which scored 0.62 $\pm$ 0.01. Additionally, in the Hellaswag dataset, the HFM model led with an accuracy of 0.62 $\pm$ 0.01, compared to LLaMA-3-8B's 0.60 $\pm$ 0.01.

The experimental results for the language understanding and question-answering tasks demonstrated that the HFM model substantially outperformed the other models across most metrics. For the TruthfulQA dataset, HFM demonstrated superior performance with a blue score of 0.52 $\pm$ 0.02, Rouge 1 of 0.58 $\pm$ 0.02, and Rouge L of 0.58 $\pm$ 0.02.

[6]https://huggingface.co/docs/trl/en/index

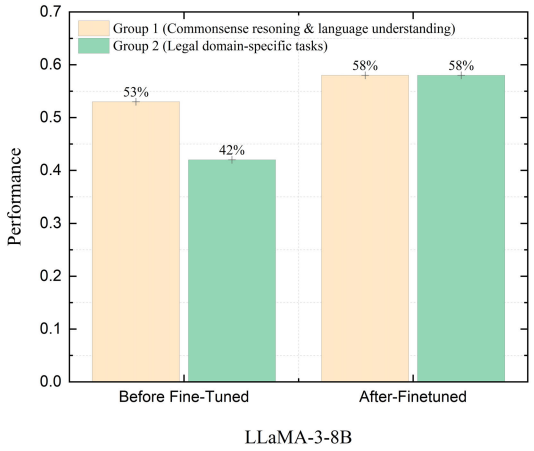

FIGURE 5. Performance evaluation of fine-tuned model across multiple tasks.

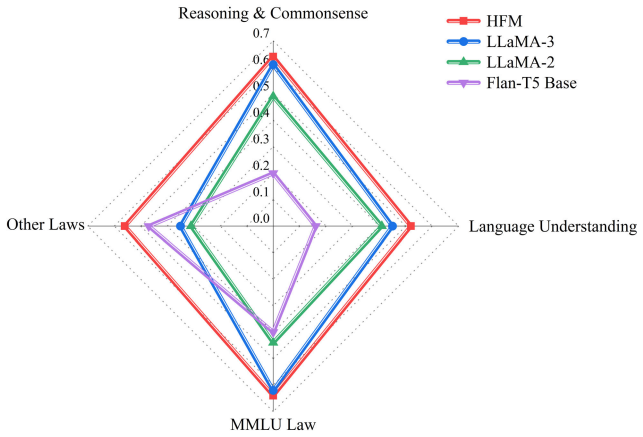

FIGURE 6. Performance evaluation of diverse models across multiple tasks.

These scores were higher than those of LLaMA-3-8B, which scored 0.44 ± 0.02, 0.42 ± 0.02, and 0.40 ± 0.02, respectively. On the other hand, in the SQuAD_v2 dataset, HFM achieved an accuracy of 0.45 ± 0.02, which is slightly less than the baseline model LLaMA-3-8B, which scored 0.50 ± 0.02. Despite this, the overall performance of the HFM model in the first group tasks showcases its enhanced capabilities.

The experimental findings for the second group of tasks, which are specific to the legal domain, demonstrated that the HFM model consistently surpassed the performance of other models. For the MMLU International Law dataset, HFM achieved a score of 0.81 ± 0.03, significantly higher than LLaMA-3-8B's score of 0.77 ± 0.04. Similarly, in the MMLU Professional Law dataset, HFM led with a score of 0.47 ± 0.01, while LLaMA-3-8B scored 0.46 ± 0.01. For the Abercrombie classification dataset, HFM achieved a score of 0.54 ± 0.04, compared to LLaMA-3-8B's 0.45 ± 0.05. In the Legal Reasoning Causality (LRC) dataset, HFM demonstrated superior performance with a score of 0.75 ± 0.01, outperforming LLaMA-3-8B's score of 0.52 ± 0.01. In the Law Stack Exchange (LSE) dataset, the Flan-T5 large model led with a score of 0.63 ± 0.01, while HFM scored 0.28 ± 0.01. For the Canada Tax Court Outcomes (CTCO) dataset, Flan-T5 large also performed best with a score of 0.68 ± 0.01, while HFM scored 0.66 ± 0.01. Lastly, in the Contract QA (CQA) dataset, HFM achieved a score of 0.56 ± 0.01, surpassing LLaMA-3-8B's score of 0.19 ± 0.01.

Overall, the HFM model achieved the highest performance across both evaluation groups, as depicted in Figure 6, demonstrating its robustness and effectiveness in handling both general language understanding and domain-specific queries. The results highlight the effectiveness of fine-tuning and merging techniques in enhancing model performance across diverse tasks, demonstrating their value in improving generalization and downstream task performance.

C. LQ-RAG SYSTEM

This section investigates the LQ-RAG system, comparing it with Naive RAG and RAG with FTM. The goal is to identify the strengths and the weaknesses of the proposed system in legal contexts through empirical evaluation.

For open-domain question answering, the test questions encompass a diverse range of types, including constitutional provisions, explanations of amendments, and hypothetical scenarios designed to simulate real-world legal inquiries. Table 10 presents a subset of these questions, illustrating the variety and specificity of the queries used for evaluation. Additionally, Table 11 provides detailed experimental results, offering insights into the system's performance across these diverse question types. Figure 7 presents the average relevance scores for three RAG configurations. The Naive RAG configuration achieved an average score of 65%, indicating basic performance without specialized tuning. The RAG with FTM improved to 70%, reflecting a 7% increase over Naive RAG, which suggests that fine-tuning enhances the model's ability to retrieve and generate relevant information. The proposed LQ-RAG system attained the highest score of 80%, showing a 23% improvement over Naive RAG and a 14% improvement over RAG with FTM. This substantial improvement is attributed to the advanced integration and fine-tuning techniques in LQ-RAG, which enhance its ability to understand and retrieve contextually relevant information, resulting in more coherent answers. The evaluation results as illustrated in Figure 8, further reinforce the effectiveness of the proposed system. For the same evaluation dataset, the system achieved scores of 88% in answer relevance, 70% in context relevance, and 82% in groundedness. In contrast, the Naive RAG model struggled to meet the threshold levels in these metrics, especially in context relevance. Although the RAG with FTM performed better than the Naive RAG, the score was still not satisfactory enough to be considered an acceptable answer.

In contrast, for closed-domain question answering, the evaluation results are summarized in Table 12. The evaluation

TABLE 8. Experimental results of reasoning and language tasks.

Model name	Reasoning & Commonsense		Language Understanding & Question Answering			
	BBH	Hellaswag	TruthfulQA			Squad_v2
	EM (3-shot)	Accuracy	Blue Score	Rouge_1	Rouge_L	Accuracy
Flan-T5 small	0.12 ± 0.01	0.28 ± 0.02	0.30 ± 0.02	0.31 ± 0.02	0.31 ± 0.02	0.04 ± 0.02
Flan-T5 base	0.24 ± 0.01	0.40 ± 0.05	0.28 ± 0.01	0.30 ± 0.01	0.29 ± 0.01	0.03 ± 0.02
Flan-T5 large	0.29 ± 0.01	0.39 ± 0.01	0.28 ± 0.02	0.29 ± 0.02	0.29 ± 0.02	0.04 ± 0.02
Flan-T5 xl	0.35 ± 0.01	0.47 ± 0.01	0.48 ± 0.02	0.48 ± 0.02	0.49 ± 0.02	0.03 ± 0.02
LLaMA-2-7B	0.40 ± 0.01	0.57 ± 0.01	0.34 ± 0.02	0.32 ± 0.02	0.32 ± 0.02	**0.50 ± 0.02**
LLaMA-2-13B-4bit	0.45 ± 0.01	0.59 ± 0.01	0.32 ± 0.02	0.31 ± 0.02	0.30 ± 0.02	0.45 ± 0.02
LLaMA-3-8B	0.62 ± 0.01	0.60 ± 0.01	0.44 ± 0.02	0.42 ± 0.02	0.40 ± 0.02	**0.50 ± 0.02**
HFM	**0.65 ± 0.01**	**0.62 ± 0.01**	**0.52 ± 0.02**	**0.58 ± 0.02**	**0.58 ± 0.02**	0.45 ± 0.02

TABLE 9. Experimental results of legal domain-specific tasks.

Model name	MMLU Law		Others Law				
	Int.	Prof.	Abercrombie	LRC	Stock Ex.	CTCO	CQA
Flan-T5 small	0.44 ± 0.05	0.31 ± 0.01	0.06 ± 0.03	0.50 ± 0.01	0.47 ± 0.01	0.01 ± 0.04	0.76 ± 0.01
Flan-T5 base	0.47 ± 0.05	0.32 ± 0.01	0.25 ± 0.03	0.48 ± 0.01	0.56 ± 0.01	0.34 ± 0.01	0.74 ± 0.01
Flan-T5 large	0.56 ± 0.05	0.34 ± 0.01	0.36 ± 0.04	0.61 ± 0.01	**0.63 ± 0.01**	**0.68 ± 0.01**	0.77 ± 0.01
Flan-T5 xl	0.69 ± 0.04	0.38 ± 0.01	0.33 ± 0.04	0.66 ± 0.01	0.58 ± 0.05	0.63 ± 0.01	**0.95 ± 0.01**
LLaMA-2-7B	0.56 ± 0.05	0.31 ± 0.01	0.23 ± 0.04	0.64 ± 0.01	0.07 ± 0.01	0.53 ± 0.01	0.10 ± 0.01
LLaMA-2-13B-4bit	0.67 ± 0.04	0.39 ± 0.01	0.39 ± 0.04	0.66 ± 0.01	0.03 ± 0.01	0.39 ± 0.01	0.46 ± 0.01
LLaMA-3-8B	0.77 ± 0.04	0.46 ± 0.01	0.45 ± 0.05	0.52 ± 0.01	0.17 ± 0.01	0.41 ± 0.01	0.19 ± 0.01
HFM	**0.81 ± 0.03**	**0.47 ± 0.01**	**0.54 ± 0.04**	**0.75 ± 0.01**	0.28 ± 0.01	0.66 ± 0.01	0.56 ± 0.01

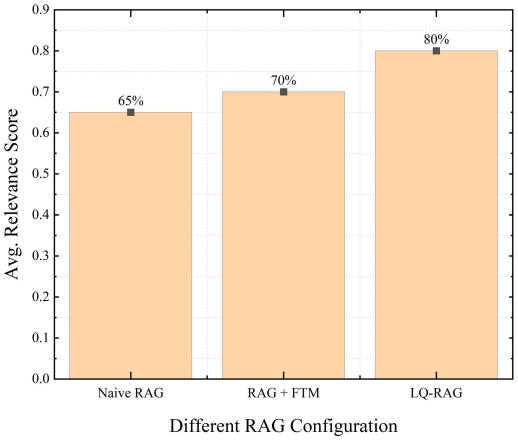

FIGURE 7. The average relevance score of the RAG system across various network architectures.

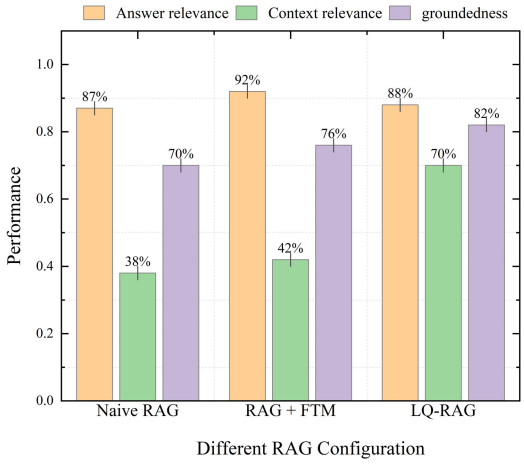

FIGURE 8. The evaluation of the RAG triad across diverse network architectures.

encompassed posing five distinct sets of queries. Both the Naive RAG and RAG with FTM consistently achieved an answer relevancy score of 88%. However, with the LQ-RAG system, the answer relevancy score decreased to 72%, indicating a discernible difference in answer relevance performance across these RAG system configurations. Throughout the assessment, the context relevancy score and the groundedness score both remained less than 50% across all RAG system

TABLE 10. Sample questions for evaluating the performance of LQ-RAG system.

Sr. no.	Sample question
Question: 1	When was the U.S. Constitution signed, and who was the convention president?
Question: 2	What can happen to someone after impeachment, according to constitutional law?
Question: 3	What does the 25th Amendment say about the President being unable to perform duties in relation to the 14th Amendment's provisions on citizenship and representation?
Question: 4	For my exam, can you explain what the 25th Amendment says about a President unable to perform duties?
Question: 5	What does the 25th Amendment say about presidential succession and the 26th Amendment about voting rights?
Question: 6	On which date was the Constitution of the United States signed and who was the president of the convention?
Question: 7	According to the constitutional law, what are the potential repercussions that an individual may face following impeachment?
Question: 8	According to the 25th Amendment, what is the protocol when the President of the United States is incapacitated and unable to fulfill his duties, particularly in the context of the 14th Amendment's provisions on citizenship and representation?
Question: 9	As a law student studying for an exam on American constitutional amendments, I need to clarify something. Can you explain what happens when a President is unable to discharge the powers and duties of his office according to the 25th Amendment?
Question: 10	What does the 25th Amendment of the US Constitution state about the presidential succession and what does the 26th Amendment state about the voting rights?
Question: 11	How does the 25th Amendment address the issue of presidential incapacity and what steps are taken to ensure a smooth transition of power?
Question: 12	What specific circumstances trigger the activation of the 25th Amendment and how does it impact the President's role and responsibilities?
Question: 13	How does the 25th Amendment relate to the 14th Amendment's provisions on citizenship and representation in cases of presidential incapacity?
Question: 14	What measures are put in place to ensure the President's health and well-being, and how do they relate to the 25th Amendment?
Question: 15	How does the 25th Amendment impact the Vice President's role and responsibilities during presidential incapacity, and what steps are taken to ensure a smooth transition of power?

TABLE 11. Performance evaluation of the LQ-RAG system across different user queries.

Sample	Naive RAG				RAG + FTM				LQ-RAG			
	AR	CR	G	Latency (s)	AR	CR	G	Latency (s)	AR	CR	G	Latency (s)
Question: 1	0.90	0.44	0.92	3.1	0.93	0.45	0.80	9.8	0.89	0.72	0.85	13.2
Question: 2	0.80	0.30	0.65	6.8	0.90	0.40	0.78	12.1	0.87	0.68	0.80	15.4
Question: 3	0.88	0.45	0.52	9.3	0.91	0.38	0.74	10.3	0.90	0.73	0.84	12.5
Question: 4	0.95	0.45	0.90	5.1	0.95	0.46	0.81	13.0	0.88	0.69	0.81	16.2
Question: 5	0.85	0.35	0.60	8.3	0.92	0.44	0.79	10.2	0.86	0.71	0.82	14.8
Question: 6	0.90	0.40	0.75	6.5	0.90	0.39	0.73	11.8	0.89	0.72	0.83	13.6
Question: 7	0.80	0.25	0.55	9.7	0.93	0.41	0.75	10.4	0.88	0.70	0.81	15.1
Question: 8	0.92	0.54	0.80	4.7	0.91	0.43	0.76	10.2	0.87	0.68	0.79	14.7
Question: 9	0.87	0.32	0.65	7.8	0.92	0.40	0.72	12.3	0.89	0.74	0.84	14.3
Question: 10	0.83	0.38	0.70	7.2	0.94	0.47	0.79	10.9	0.88	0.69	0.82	15.7
Question: 11	0.88	0.27	0.62	8.5	0.91	0.39	0.74	11.5	0.90	0.71	0.83	14.6
Question: 12	0.85	0.42	0.78	6.4	0.90	0.38	0.77	10.8	0.86	0.72	0.80	15.2
Question: 13	0.89	0.33	0.68	7.5	0.93	0.42	0.78	11.6	0.87	0.70	0.81	14.5
Question: 14	0.91	0.36	0.72	7.2	0.92	0.40	0.75	12.2	0.89	0.73	0.85	13.2
Question: 15	0.86	0.40	0.73	9.8	0.94	0.43	0.80	11.4	0.88	0.68	0.79	15.4

setups. This outcome was anticipated as the system was not provided with any relevant context information during the response generation. Based on these findings, the answers generated by the RAG systems are potentially incorrect because the retrieved information lacks the required context. This casts doubt on the answer relevance scores reported by all configurations of the evaluated RAG systems. Based on the experimental results, it is evident that evaluating answer generation by RAG systems requires considering all three criteria: answer relevance, context relevance, and groundedness to ensure confidence in the accuracy and reliability of the generated responses.

One of the key concerns in RAG implementation is time complexity. To address this issue, we conducted an experiment to measure the average response generation times for different configurations. The results, depicted in Figure 9, indicate that the Naive system exhibits the lowest latency, taking only 7.2 seconds to complete five sets of questions.

TABLE 12. RAG performance evaluation under closed-domain question and answering.

Modality	Answer relevance	Context relevance	Groundedness
Naive RAG	0.88	0.31	0.26
RAG + FTM	0.88	0.24	0.19
LQ-RAG	0.72	0.48	0.35

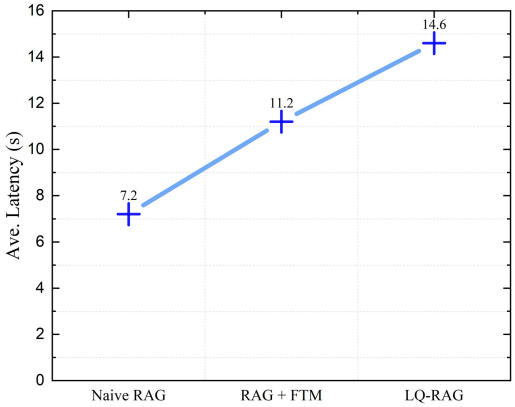

FIGURE 9. The average time complexity of RAG system across diversenetwork architectures.

In contrast, the RAG with FTM takes 11.2 seconds, and the proposed LQ-RAG system requires 14.6 seconds, which is double the time of the Naive case. From this, it can be inferred that while the Naive RAG system responds faster, incorporating advanced modules increases the system's time complexity.

VII. CONCLUSION

This paper addresses domain-specific challenges in the legal field, where traditional RAG systems often fail in information extraction and response generation. To address these issues, the LQ-RAG framework integrates RAG with a recursive feedback mechanism, combining specialized LLMs and an agent-driven approach for response evaluation and query engineering. This multi-layered system reduces hallucinations and ensures precise, contextually relevant responses. Fine-tuning a general-purpose LLM with legal corpora resulted in a 15% improvement over baseline models, while a hybrid fine-tuned generative LLM achieved up to 24% better performance across various tasks compared to general domain LLMs. The LQ-RAG architecture outperformed all baseline models, with a 23% improvement in average relevance score over the naive configuration and a 14% improvement over RAG with fine-tuned LLMs. Its adaptable design facilitates adoption across other specialized domains with minimal adjustments, enabling professionals to make high-quality, informed decisions.

VIII. LIMITATIONS & FUTURE WORK

While the current work demonstrates significant advancements, a few limitations remain, including reliance on GPT-4 as the evaluation agent, high response generation time, and the absence of feedback from domain experts. Future efforts will focus on addressing these issues by optimizing time complexity, developing a specialized legal evaluation agent with domain-specific expertise, and incorporating feedback from legal practitioners to ensure the model's practical utility and alignment with legal reasoning and context. Additionally, benchmark datasets specifically designed for the legal domain will be incorporated to further validate the approach. State-of-the-art optimization techniques will also be applied to enhance hit rate and MRR, improving the system's practical viability in legal applications. Empirical experiments will be conducted in real-world legal scenarios to demonstrate the system's effectiveness.

ACKNOWLEDGMENT

The authors gratefully acknowledge the high-performance GPU computing support provided by HPC-AI Open Infrastructure through GIST SCENT, which was instrumental in enabling this research.[7]

REFERENCES

[1] Q. Lang, S. Tian, M. Wang, and J. Wang, "Exploring the answering capability of large language models in addressing complex knowledge in entrepreneurship education," *IEEE Trans. Learn. Technol.*, vol. 17, pp. 2053–2062, 2024.
[2] G. B. Mohan, R. P. Kumar, P. V. Krishh, A. Keerthinathan, G. Lavanya, M. K. U. Meghana, S. Sulthana, and S. Doss, "An analysis of large language models: Their impact and potential applications," *Knowl. Inf. Syst.*, vol. 66, no. 9, pp. 5047–5070, Sep. 2024.
[3] B. Meskó and E. J. Topol, "The imperative for regulatory oversight of large language models (or generative AI) in healthcare," *npj Digit. Med.*, vol. 6, no. 1, p. 120, Jul. 2023.
[4] J. Lai, W. Gan, J. Wu, Z. Qi, and P. S. Yu, "Large language models in law: A survey," *AI Open*, vol. 5, pp. 181–196, 2024.
[5] D. M. Katz, M. J. Bommarito, S. Gao, and P. Arredondo, "GPT-4 passes the bar exam," *Phil. Trans. Roy. Soc. A*, vol. 382, Mar. 2023, Art. no. 20230254.
[6] Q. Huang, M. Tao, C. Zhang, Z. An, C. Jiang, Z. Chen, Z. Wu, and Y. Feng, "Lawyer LLaMA technical report," 2023, *arXiv:2305.15062*.
[7] V. Magesh, F. Surani, M. Dahl, M. Suzgun, C. D. Manning, and D. E. Ho, "Hallucination-free? Assessing the reliability of leading AI legal research tools," 2024, *arXiv:2405.20362*.
[8] W. Benjamin, "Here's what happens when your lawyer uses ChatGPT," New York Times, New York, NY, USA, Tech. Rep., 2023. [Online]. Available: https://www.nytimes.com/2023/05/27/nyregion/avianca-airline-lawsuit-chatgpt.html
[9] M. Dahl, V. Magesh, M. Suzgun, and D. E. Ho, "Large legal fictions: Profiling legal hallucinations in large language models," *J. Legal Anal.*, vol. 16, no. 1, pp. 64–93, Jan. 2024.
[10] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-T. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 9459–9474.
[11] J. Chen, H. Lin, X. Han, and L. Sun, "Benchmarking large language models in retrieval-augmented generation," in *Proc. AAAI Conf. Artif. Intell.*, Mar. 2024, vol. 38, no. 16, pp. 17754–17762.
[12] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to retrieve, generate, and critique through self-reflection," in *Proc. 12th Int. Conf. Learn. Represent.*, Jan. 2023, pp. 1–30.
[13] Y. Xia, Z. Xiao, N. Jazdi, and M. Weyrich, "Generation of asset administration shell with large language model agents: Toward semantic interoperability in digital twins in the context of industry 4.0," *IEEE Access*, vol. 12, pp. 84863–84877, 2024.

[7] https://openhpc.kr/

[14] R. S. M. Wahidur, I. Tashdeed, M. Kaur, and H.-N. Lee, "Enhancing zero-shot crypto sentiment with fine-tuned language model and prompt engineering," *IEEE Access*, vol. 12, pp. 10146–10159, 2024.

[15] J. Bednár, J. Náplava, P. Barančíková, and O. Lisický, "Some like it small: Czech semantic embedding models for industry applications," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, Mar. 2024, pp. 22734–22742.

[16] X. Ma, Y. Gong, P. He, H. Zhao, and N. Duan, "Query rewriting in retrieval-augmented large language models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2023, pp. 5303–5315.

[17] R. Sharma. (2024). *Exploring Advanced RAG Techniques for AI*. [Online]. Available: https://markovate.com/blog/advanced-rag-techniques/

[18] ILIN. (2023). *Advanced RAG Techniques: An Illustrated Overview*. [Online]. Available: https://pub.towardsai.net/advanced-rag-techniques-an-illustrated-overview-04d193d8fec6

[19] Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan, and W. Chen, "Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy," in *Proc. Findings Assoc. Comput. Linguistics, EMNLP*, Stroudsburg, PA, USA, 2023, pp. 9248–9274.

[20] W. Yu, D. Iter, S. Wang, X. Yi-hong, M. Ju, S. Sanyal, C. Zhu, M. Zeng, and M. Jiang, "Generate rather than retrieve: Large language models are strong context generators," in *Proc. 11th Int. Conf. Learn. Represent.*, 2022, pp. 1–27. [Online]. Available: https://openreview.net/forum?id=fB0hRu9GZUS

[21] J. Wen and W. He. (2023). *HanFei-1.0*. [Online]. Available: https://github.com/siat-nlp/HanFei

[22] H. Liu, Y. Liao, and Y. Meng. (2023). *Chinese Law Large Language Model*. [Online]. Available: https://github.com/LiuHC0428/LAW_GPT

[23] H.-T. Nguyen, "A brief report on LawGPT 1.0: A virtual legal assistant based on GPT-3," 2023, *arXiv:2302.05729*.

[24] H. Li. (2023). *LexiLaw*. [Online]. Available: https://github.com/CSHaitao/LexiLaw

[25] D. Soong, S. Sridhar, H. Si, J.-S. Wagner, A. C. C. Sá, C. Y. Yu, K. Karagoz, M. Guan, S. Kumar, H. Hamadeh, and B. W. Higgs, "Improving accuracy of GPT-3/4 results on biomedical data using a retrieval-augmented language model," *PLOS Digit. Health*, vol. 3, no. 8, Aug. 2024, Art. no. e0000568.

[26] C. Zakka, R. Shad, A. Chaurasia, A. R. Dalal, J. L. Kim, M. Moor, R. Fong, C. Phillips, K. Alexander, E. Ashley, J. Boyd, K. Boyd, K. Hirsch, C. Langlotz, R. Lee, J. Melia, J. Nelson, K. Sallam, S. Tullis, M. A. Vogelsong, J. P. Cunningham, and W. Hiesinger, "Almanac—Retrieval—Augmented language models for clinical medicine," *NEJM AI*, vol. 1, no. 2, pp. 1–45, 2024.

[27] S. Yue, W. Chen, S. Wang, B. Li, C. Shen, S. Liu, Y. Zhou, Y. Xiao, S. Yun, X. Huang, and Z. Wei, "DISC-LawLLM: Fine-tuning large language models for intelligent legal services," 2023, *arXiv:2309.11325*.

[28] N. Wiratunga, R. Abeyratne, and L. Jayawardena, "CBR-RAG: Case-based reasoning for retrieval augmented generation in LLMs for legal question answering," in *Proc. Case-Based Reasoning Res. Development. ICCBR*, vol. 14775, J. A. Recio-Garcia, M. G. Orozco-del-Castillo, and D. Bridge, Eds., Springer, 2024, pp. 445–460.

[29] A. Chouhan and M. Gertz, "LexDrafter: Terminology drafting for legislative documents using retrieval augmented generation," in *Proc. Int. Conf. Comput. Linguistics, Lang. Resour. Eval. (LREC-COLING)*, 2024, pp. 10448–10458.

[30] S. S. Alotaibi, A. A. Munshi, and A. T. Arag, "KAB: Knowledge augmented BERT2BERT automated questions-answering system for jurisprudential legal opinions," *Int. J. Comput. Sci. Netw. Security, IJCSNS*, vol. 22, pp. 346–356, Jun. 2022.

[31] C. Hoppe, D. Pelkmann, N. Migenda, D. Hötte, and W. Schenck, "Towards intelligent legal advisors for document retrieval and question-answering in German legal documents," in *Proc. IEEE 4th Int. Conf. Artif. Intell. Knowl. Eng. (AIKE)*, Dec. 2021, pp. 29–32.

[32] S. Robertson, H. Zaragoza, and M. Taylor, "Simple BM25 extension to multiple weighted fields," in *Proc. 13th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, Nov. 2004, pp. 42–49.

[33] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-T. Yih, "Dense passage retrieval for open-domain question answering," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Stroudsburg, PA, USA, 2020, pp. 6769–6781.

[34] M. Henderson, R. Al-Rfou, B. Strope, Y.-H. Sung, L. Lukacs, R. Guo, S. Kumar, B. Miklos, and R. Kurzweil, "Efficient natural language response suggestion for smart reply," 2017, *arXiv:1705.00652*.

[35] J. E. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2021, pp. 1–53. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

[36] M. Douze, A. Guzhva, and C. Deng. (2024). *The Faiss Library*. [Online]. Available: https://github.com/facebookresearch/faiss

[37] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing reasoning and acting in language models," 2022, *arXiv:2210.03629*.

[38] J. Lee, X. Wang, D. Schuurmans, M. Bosma, E. H. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 24824–24837.

[39] I. Bunescu. (2023). *QA Legal Dataset Train*. [Online]. Available: https://huggingface.co/datasets/ibunescu/qa_legal_dataset_train

[40] T. Rohan and G. Ishaan. (2023). *Stanford Alpaca: An Instruction-following LLaMA Model*. [Online]. Available: https://github.com/tatsu-lab/stanford_alpaca

[41] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for SQuAD," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, Stroudsburg, PA, USA, 2018, pp. 784–789.

[42] S. Lin, J. Hilton, and O. Evans, "TruthfulQA: Measuring how models mimic human falsehoods," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, Stroudsburg, PA, USA, 2022, pp. 3214–3252.

[43] J. Li, R. Bhambhoria, and X. Zhu, "Parameter-efficient legal domain adaptation," in *Proc. Natural Legal Lang. Process. Workshop*, 2022, pp. 119–129. [Online]. Available: https://aclanthology.org/2022.nllp-1.10

[44] M. Suzgun, N. Scales, N. Schärli, S. Gehrmann, Y. Tay, H. W. Chung, A. Chowdhery, Q. Le, E. Chi, D. Zhou, and J. Wei, "Challenging BIG-bench tasks and whether chain-of-thought can solve them," in *Proc. Findings Assoc. Comput. Linguistics: ACL*. Stroudsburg, PA, USA, 2023, pp. 13003–13051.

[45] N. Guha et al., "Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 44123–44279.

[46] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," in *Proc. Int. Conf. Learn. Represent.*, May 2021, pp. 1–6. [Online]. Available: https://openreview.net/forum?id=d7KBjmI3GmQ

[47] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi, "HellaSwag: Can a machine really finish your sentence?" in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Florence, Italy, 2019, pp. 4791–4800.

[48] O. Khattab and M. Zaharia, "ColBERT," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, New York, NY, USA, Jul. 2020, pp. 39–48.

[49] P. Zhang, S. Xiao, Z. Liu, Z. Dou, and J.-Y. Nie, "Retrieve anything to augment large language models," 2023, *arXiv:2310.07554*.

[50] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, D. Lian, and J.-Y. Nie, "C-pack: Packed resources for general Chinese embeddings," in *Proc. 47th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, New York, NY, USA, Jul. 2024, pp. 641–649.

[51] A. V. Solatorio, "GISTEmbed: Guided in-sample selection of training negatives for text embedding fine-tuning," 2024, *arXiv:2402.16829*.

[52] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," 2023, *arXiv:2307.09288*.

[53] AI@Meta. (2024). *Llama 3 Model Card*. [Online]. Available: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

[54] H. W. Chung et al., "Scaling instruction-finetuned language models," *J. Mach. Learn. Res.*, vol. 25, pp. 1–53, 2024.

[55] X. Zhang, X. Zhou, Z. Zhang, L. Wang, and P. Wang, "A novel method to improve hit rate for big data quick reading," in *Proc. Int. Conf. Artif. Intell. Adv. Manuf. (AIAM)*, Oct. 2019, pp. 39–43.

[56] S. Roychowdhury, S. Soman, H. G. Ranjani, N. Gunda, V. Chhabra, and S. K. Bala, "Evaluation of RAG metrics for question answering in the telecom domain," 2024, *arXiv:2407.12873*.

[57] P. Xia, L. Zhang, and F. Li, "Learning similarity with cosine similarity ensemble," *Inf. Sci.*, vol. 307, pp. 39–52, Jun. 2015.

[58] A. Stolfo, "Groundedness in retrieval-augmented long-form generation: An empirical study," in *Proc. Findings Assoc. Comput. Linguistics, NAACL*. Stroudsburg, PA, USA, 2024, pp. 1537–1552.

[59] K. Papineni, S. Roukos, T. J. Ward, and W.-J. Zhu, "BLEU," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics (ACL)*. Morristown, NJ, USA, 2001, p. 311.

[60] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. Text Summarization Branches Out*, Jul. 2004, pp. 74–81.

[61] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. D. Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2790–2799.
[62] T. Dettmers, M. Lewis, and Y. Belkada, "Gpt3. Int8 (): 8-bit matrix multiplication for transformers at scale," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 30318–30332.
[63] W. Leandro, B. Younes, T. Lewis, and B. Edward. (2020). *TRL: Transformer Reinforcement Learning*. [Online]. Available: https://github.com/huggingface/trl

RAHMAN S. M. WAHIDUR received the B.Sc. degree in electrical and electronics engineering from the Ahsanullah University of Science and Technology, Dhaka, Bangladesh, in 2009. He is currently pursuing the combined M.S. and Ph.D. degree with Gwangju Institute of Science and Technology, Gwangju, South Korea. He is also a Research Assistant with the INFOrmation Processing, Controlling, and NETwork Laboratory (INFONET LAB). Prior to his current academic endeavors, he held the position of Telecommunication Engineer at various multinational corporations, from 2010 to 2019. His research interests include natural language processing, deep learning, blockchain price modeling, and generative AI.

SUMIN KIM received the B.S. degree in communications and convergence software from Kwangwoon University, Seoul, South Korea, in 2021. She is currently pursuing the Ph.D. degree with the Artificial Intelligence Graduate School, Gwangju Institute of Science and Technology, Gwangju, South Korea. Her research interests include continual learning, reinforcement learning, natural language processing, and financial price modeling.

HAEUNG CHOI received the B.S. degree in electrical, electronics, and computer engineering from Kyungpook National University, in 2013, and the M.S. degree in electrical, electronics, and computer engineering from Gwangju Institute of Science and Technology, in 2015, where he is currently pursuing the Ph.D. degree. He is also a Researcher at LiberVance Company. His research interests include blockchain and cybersecurity.

DAVID S. BHATTI received the Ph.D. degree in computer science from the School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2020. He is currently a Postdoctoral Researcher at the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju, South Korea. He is also working on augmenting HSI with AI and retrieval-augmented generation (RAG). His research interests include network security, deep learning, and hyperspectral imaging.

HEUNG-NO LEE (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of California at Los Angeles, Los Angeles, CA, USA, in 1993, 1994, and 1999, respectively. He was a Research Staff Member with HRL Laboratories, LLC, Malibu, CA, USA, from 1999 to 2002. From 2002 to 2008, he was an Assistant Professor with the University of Pittsburgh, Pittsburgh, PA, USA. In 2009, he joined the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju, South Korea. His research interests include information theory, signal processing theory, blockchain, communications/networking theory, and their applications to wireless communications and networking, compressive sensing, future internet, and brain–computer interface. He was a recipient of several prestigious national awards, including the Top 100 National Research and Development Award, in 2012, the Top 50 Achievements of Fundamental Research Award, in 2013, and the Science/Engineer of the Month in January 2014.