Benchmarking KG-based RAG Systems: A Case Study of Legal Documents

Jaycent G. Ongris[1,*], Fariz Darari[1,*], Berty C. L. Tobing[2], Douglas R. Faisal[2] and On Lee[2]

[1]Faculty of Computer Science, Universitas Indonesia, Depok 16424, Indonesia
[2]GDP Labs, Jakarta 12950, Indonesia

Abstract

Retrieval-augmented generation (RAG) systems enhance language model outputs by incorporating external knowledge, typically in the form of unstructured text. Recent advancements have introduced structured sources such as knowledge graphs (KGs) to improve retrieval precision and interpretability. This study benchmarks several KG-based and hybrid RAG frameworks, including HippoRAG 2, Nano GraphRAG, LightRAG, and LlamaIndex, to be compared with a naïve RAG baseline, in the context of legal question answering (QA). The evaluation is performed on a multilingual legal corpus comprising EU Directives and Indonesian Government Regulations. A semi-automated pipeline, combining language models and human refinement, is used to generate high-quality QA datasets. We assess system performance using Ragas answer accuracy metric and identify the trade-offs between efficiency, interpretability, and accuracy. Our findings demonstrate the superior performance of hybrid approaches, particularly LightRAG Mix and LlamaIndex Hybrid, in terms of accuracy. Conversely, KG-only systems often underperform due to their inability to fully capture the semantics of the text. This work provides actionable insights for the development of reliable and multilingual legal QA systems.

Keywords

Retrieval-augmented generation, knowledge graph, legal question answering

1. Introduction

In recent years, retrieval-augmented generation (RAG) has emerged as a promising solution to address limitations of large language models (LLMs), including hallucinations, lack of transparency, and limited knowledge updating capability [1]. By augmenting the prompt with knowledge from external memory, RAG enables LLMs to generate more grounded and contextually relevant outputs. This external memory is typically constructed from unstructured text corpora, from which documents are retrieved at inference time based on the input query.

Building on this foundation, recent advances have explored the integration of graph-structured data, such as knowledge graphs (KGs), to provide a more semantically rich and structured source of knowledge. The use of graph data offers an abstraction of lengthy textual content while preserving the relational knowledge of the underlying information [2], thereby enabling more precise and interpretable retrieval. Additionally, retrieving relevant graph communities enables more effective handling of query-focused summarization (QFS) task by capturing broader contextual information embedded within the graph structure [2].

Moreover, recent research has investigated hybrid RAG systems that combine unstructured text with structured KGs [3]. This approach seeks to harness the strengths of both modalities: unstructured text offers rich natural language content, while structured KGs provide contextual information, semantic relationships, and explicit links. By leveraging these complementary sources, hybrid RAG systems can improve factual accuracy, enhance retrieval precision, and increase interpretability, thereby achieving the best of both worlds.

RAGE-KG 2025: The Second International Workshop on Retrieval-Augmented Generation Enabled by Knowledge Graphs, co-located with ISWC 2025, November 2–6, 2025, Nara, Japan

*Corresponding authors.
jaycent.gunawan@ui.ac.id (J. G. Ongris); fariz@ui.ac.id (F. Darari); berty.c.l.tobing@gdplabs.id (B. C. L. Tobing); douglas.r.faisal@gdplabs.id (D. R. Faisal); onlee@gdplabs.id (O. Lee)

In the legal domain, the application of RAG is particularly significant due to the high stakes involved in legal question answering (QA), where accuracy, traceability, and interpretability are paramount. Legal documents are often extensive, intricate, and characterized by hierarchical and referential structures. RAG systems designed specifically for legal contexts, often termed legal RAG, have the potential to support legal practitioners in efficiently navigating complex legal texts and responding to legal queries with transparent and well-grounded justifications. Furthermore, the integration of legal knowledge graphs enables the system to reason over legal content in a structured and coherent manner.

In this work, we tailor and benchmark several KG-based RAG approaches for the legal domain, evaluating their effectiveness in addressing domain-specific challenges. Our experiments are conducted on real-world legal documents from both the European Union and Indonesia, highlighting the multilingual aspect of our approach. We also present key insights and lessons learned from our experimental evaluations.

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 describes the research methodology. Section 4 presents the experimental results and analysis. Section 5 discusses key insights and lessons learned. Finally, Section 6 concludes the paper.

2. Related Work

In this section, we review relevant literature that informs and contextualizes our work.

2.1. RAG

Retrieval-augmented generation, or RAG [1], is a method in which language models retrieve external textual information during inference time to augment their responses. Unlike conventional language models that rely solely on their training data, RAG enables models to access up-to-date or domain-specific information from a separate retrieval component, often built on top of dense or sparse vector indices.

The benefits of RAG include improved factual accuracy, reduced hallucination, and enhanced interpretability. RAG has proven especially useful in domains where precision and source traceability are critical. For instance, in the legal domain, a RAG-based assistant can retrieve and summarize relevant sections of regulations, judicial decisions, or contracts in response to a user query such as "What factors should be considered when assessing the defectiveness of a product under Directive (EU) 2024/2853?"

2.2. GraphRAG

GraphRAG [2] extends the idea of RAG by using structured data sources, such as knowledge graphs, as the retrieval backbone instead of unstructured text. In GraphRAG, the language model queries and retrieves facts represented in the form of triples or subgraphs, which are then used to generate answers grounded in curated knowledge.

Unlike text-based RAG, GraphRAG leverages explicit entity and relationship representations in graph data to provide more precise information. In the legal context, GraphRAG can be used to query a legal knowledge graph for relationships between laws, institutions, and processes. For example, it can answer questions like "Are compensation schemes like national health systems covered by Directive (EU) 2024/2853?" by directly navigating through relationships such as regulated by, amended by, or enforced by.

A hybrid approach combines both RAG and GraphRAG to balance the breadth of natural language coverage with the precision of structured data. For example, a hybrid system can retrieve both textual descriptions from case law and structured metadata about the involved legal articles.

The term *GraphRAG* in this paper refers to the general concept of RAG using graph-structured knowledge bases, as previously described. This should not be confused with Microsoft's *GraphRAG* [4], a complex graph-based architecture that we do not implement in this work due to its impracticable system and infrastructure requirements.

2.3. HippoRAG

HippoRAG [5] is a framework designed to provide LLMs with long-term memory, drawing inspiration from human neurobiology. It comprises three main parts: the LLM (acting as the artificial neocortex), the PHR encoder (as parahippocampal region), and an open KG (as the artificial hippocampus). These components work together to mimic human memory processes. The original HippoRAG utilized a two-stage process of offline indexing (converting text to KG triples, with the PHR identifying synonyms) and online retrieval (LLM extracting entities, PHR linking to KG, and Personalized PageRank for context).

HippoRAG 2[1] [6] builds on this foundation, maintaining the two-stage process while introducing key refinements for better human memory alignment. It integrates conceptual and contextual information within the KG for a more comprehensive index, enables more context-aware retrieval by leveraging the broader KG structure, and incorporates recognition memory to improve graph search. These advancements aim to overcome the original HippoRAG's limitations, creating a more robust and effective long-term memory system for LLMs.

2.4. Nano GraphRAG

Nano GraphRAG[2] [7] is a lightweight toolkit for GraphRAG. In essence, Nano GraphRAG uses LLMs to extract structured information (entities and relationships) from unstructured text, organizes this information into a knowledge graph, and then uses this graph along with embeddings and an LLM to answer queries in a more informed and contextual way than traditional RAG. As described on its GitHub page, Nano GraphRAG is presented as "a simple, easy-to-hack GraphRAG implementation." It aims to provide a "smaller, faster, cleaner GraphRAG[3]" while retaining core functionality, distinguishing itself from potentially more complex official implementations. Notably, Nano GraphRAG is characterized as "small yet portable," supporting various technologies like Faiss, Neo4j, and Ollama, and is designed to be asynchronous and fully typed.

2.5. LightRAG

LightRAG[4] [8] introduces a novel approach by integrating graph structures into how it indexes and retrieves text. This framework features a dual-level retrieval system: one for precise information concerning individual entities and their connections (low-level retrieval), and another for broader topics and themes (high-level retrieval). By merging graph structures with vector representations, LightRAG effectively finds related entities and their connections, leading to faster responses without sacrificing relevant context. An incremental update algorithm also strengthens the system, allowing it to integrate new data promptly and remain effective in dynamic data landscapes.

In its implementation, LightRAG supports five retrieval modes: `local`, `global`, `hybrid`, `naive`, `mix`. The `local` mode focuses on low-level context such as individual entities and their immediate relationships, leveraging fine-grained graph connections to retrieve context-specific information. The `global` mode, in contrast, retrieves from the entire knowledge base and emphasizes broader, high-level information that may be relevant beyond the immediate context. The `hybrid` mode combines both local and global strategies to balance fine-grained relevance with comprehensive coverage. The `naive` mode applies a straightforward dense retrieval approach without incorporating graph-based reasoning or structure-aware techniques (cf. Subsection 2.1). Lastly, the `mix` mode integrates both dense vector search and knowledge graph-based retrieval, aiming to exploit the complementary strengths of unstructured and structured knowledge sources.

[1] https://github.com/OSU-NLP-Group/HippoRAG
[2] https://github.com/gusye1234/nano-graphrag
[3] This refers to Microsoft's GraphRAG [4].
[4] https://github.com/HKUDS/LightRAG

2.6. LlamaIndex

LlamaIndex[5] [9], formerly known as GPT Index, is a comprehensive framework for building RAG pipelines. This open-source data orchestration framework, available in Python and TypeScript, aims to simplify context augmentation for generative AI applications. It addresses the need to equip LLMs, which are pre-trained on public data, with private or domain-specific information for more accurate and relevant responses. LlamaIndex provides tools for data ingestion through various connectors (APIs, PDFs, SQL, etc.), offers methods to structure data into indices and graphs for LLM compatibility, and presents an advanced retrieval and query interface to enrich LLM outputs with relevant context. It also facilitates seamless integration with other application frameworks and provides both high-level APIs for rapid prototyping and lower-level APIs for detailed customization.

LlamaIndex also supports a property graph abstraction[6], enabling users to represent data as nodes and edges enriched with custom attributes. This is particularly useful for modeling structured knowledge bases such as KGs, where entities and relationships can be explicitly encoded. The framework includes a built-in property graph retriever that allows subgraph retrieval based on query relevance, leveraging both semantic similarity and graph structure. Furthermore, LlamaIndex provides extensibility by allowing users to define custom retrievers, making it possible to combine different retrieval strategies. This flexibility is instrumental in building hybrid retrievers that integrate both KG-based and text-based retrieval pipelines to balance the precision of structured graph queries with the broader coverage of unstructured text search.

3. Methodology

This section outlines the methodology employed in our study. Specifically, we describe the legal documents utilized, detail the preprocessing procedures applied to the data, and present the evaluation metrics used to assess the performance of the systems.

We compare the results of HippoRAG 2 (`v2.0.0a3`) [6], Nano GraphRAG in `local` mode (`v0.0.8.2`) [7], LightRAG (`v1.2.3`) in three modes—`mix`, `local`, and `hybrid`—[8], and LlamaIndex (`v0.12.45`) using both the *Property Graph* and *Hybrid* retrievers [9]. Additionally, we include the results of Naïve RAG, which relies solely on dense vector retrieval. For the implementation of naïve RAG, we utilized the version provided by the LlamaIndex library.

3.1. Legal Documents

We utilize two types of legal documents in this study: the European Union (EU)'s Directives and Indonesian Government Regulations (*Peraturan Pemerintah*, commonly abbreviated as PP). EU Directives are legislative acts that set out goals all EU member states must achieve, but they allow individual countries flexibility in how to implement them within their national legal systems. In contrast, Indonesian Government Regulations are binding national regulations issued by the central government to operationalize laws passed by the Indonesian parliament, often providing technical or administrative details.

We chose these two sources to reflect both supranational and national legislative contexts, enabling our system to handle diverse legal structures and linguistic nuances. The EU Directives provide a formalized legal corpus with clearly defined structures, while the Indonesian Government Regulations offer a distinct legal language and cultural context that is important for testing the adaptability of the experimented systems in low-resource, non-Western settings. For each category, we selected three documents covering the domains of labor, taxation, and digital commerce. The specific documents used are described in the following subsections, with each document segmented and analyzed at the article level.

[5] https://github.com/run-llama/llama_index

[6] https://docs.llamaindex.ai/en/stable/examples/property_graph/property_graph_basic/

Table 1
List of EU'S Directives Used.

Number	Title	Articles Count
Directive 2014/55/EU	Electronic Invoicing in Public Procurement	14
Directive (EU) 2019/770	Certain Aspects Concerning Contracts for the Supply of Digital Content and Digital Services	27
Directive (EU) 2024/2853	Liability for Defective Products and Repealing Council Directive 85/374/EEC	24

3.1.1. EU's Directives

We begin our analysis with the EU's Directives, which serve as primary references for legal structure and content. These directives were obtained from EUR-Lex[7], the official platform providing access to EU law. The documents are available in English in both PDF and HTML formats; we used the HTML versions to avoid the need for optical character recognition (OCR) and reduce manual refinement efforts, as we can parse the document directly using BeautifulSoup[8] in Python. Details about the selected directives are provided in Table 1.

Each directive was segmented at the article level, and non-article content such as preambles and annexes was stored in separate metadata files. We excluded the annex section of Directive (EU) 2024/2853 from our analysis, as it contains a correlation table linking provisions of the new directive to those of the previous one. While useful for legislative cross-referencing, this table does not provide substantive legal content relevant to the question answering or reasoning tasks considered in this study.

3.1.2. Indonesian Government Regulations

To assess the multilingual capabilities of our approach, we also incorporated Indonesian Government Regulations. These documents were retrieved from the official Regulations Database of the Audit Board of Indonesia[9] and are presented in the Indonesian language in PDF format. Table 2 presents the selected regulations.

Table 2
List of Indonesian Government Regulations (PP) used.

Number	Title	Articles Count
PP 58/2023	*Tarif Pemotongan Pajak Penghasilan Pasal 21 atas Penghasilan Sehubungan dengan Pekerjaan, Jasa, atau Kegiatan Wajib Pajak Orang Pribadi* (Income Tax Article 21 Withholding Rates on Income Related to Employment, Services, or Activities of Individual Taxpayers)	5
PP 36/2021	*Pengupahan* (Wages)	86
PP 7/2025	*Penyesuaian Iuran Jaminan Kecelakaan Kerja bagi Perusahaan Industri Padat Karya Tertentu Tahun 2025* (Adjustment of Work Accident Insurance Premiums for Certain Labor-Intensive Industry Companies in 2025)	12

We applied OCR to extract text from the PDFs, followed by manual refinement to address recognition errors and formatting issues. Each document was segmented at the article level, with non-article content such as preambles and closing statements stored in separate metadata files. The annex section of PP 36/2021 was retained due to its importance in determining tax rates based on income levels. Tabular

[7] https://eur-lex.europa.eu/homepage.html
[8] https://pypi.org/project/beautifulsoup4/
[9] https://peraturan.bpk.go.id/

Table 3
QA Dataset Entry Counts for Each Regulation.

Regulation	Dataset Entries
Directive 2014/55/EU	29
Directive (EU) 2019/770	53
Directive (EU) 2024/2853	46
PP 58/2023	18
PP 36/2021	87
PP 7/2025	18

entries were converted into sentences for consistency, and the annex content was stored separately. Explanation sections (*penjelasan*) were omitted across all documents, as they primarily serve to clarify legislative intent rather than introduce new legal obligations.

3.2. QA Dataset

We construct the QA dataset through a semi-automatic pipeline that combines the scalability of LLMs with careful human oversight. Initially, QA pairs are automatically generated from the parsed documents using a propietary LLM, providing a base set of questions and answers. In this case, we are using OpenAI's GPT-4o, but this can be replaced by other models. This is followed by a manual refinement phase, where human annotators review the generated pairs to correct errors, improve clarity, and ensure alignment with the source material. In addition to refining existing QA pairs, annotators also contribute by creating new questions to enrich the dataset. This process is also aided by a propietary LLM, which assists annotators in revising question phrasing and generating new potential QA pairs. The result is a high-quality QA dataset suitable for downstream applications.

Our dataset construction process is modular and model-agnostic. While we initially employed a proprietary LLM to generate and refine QA pairs, the pipeline is designed to be adaptable, allowing the use of alternative proprietary or open-source models without disrupting the overall workflow. This flexibility ensures that others can replicate or extend our approach using different LLMs depending on availability or preference.

The dataset is exported as a JSON file, with each entry comprising three keys: `question`, `answer`, and `context`. The first two keys contain the question and its respective answer, whereas the last key provides the content of the document referenced in generating the question and answer. The number of entries generated for each legal document varies depending on both the number of articles and the richness of their content. Table 3 presents a breakdown of the dataset entries for each document. The dataset is available on Google Drive[10].

3.3. Evaluation Pipeline

The overall pipeline of our work is illustrated in Figure 1. It is divided into two main stages: **Indexing** and **Querying & Evaluation**. In the indexing stage, legal documents in HTML or PDF format are ingested and parsed into unstructured text, as described in Subsection 3.1. These processed text documents serve as the foundation for creating the QA dataset, with the assistance of LLMs and human annotators (cf. Subsection 3.2). Finally, the text documents are indexed using the framework adopted for our experiments.

In the querying and evaluation stage, each question from the QA dataset is submitted to the system, which has been pre-indexed with the corresponding text documents. The generated answers are then collected and evaluated using the Ragas framework to assess their quality (cf. Subsection 3.4). Do note that our evaluation is conducted independently for each legal document; that is, the retrieval scope for

[10]QA Dataset Link: https://drive.google.com/drive/folders/1lCR6LZW_7aj16FQ7W7Gl_7DxYRXK-sS6

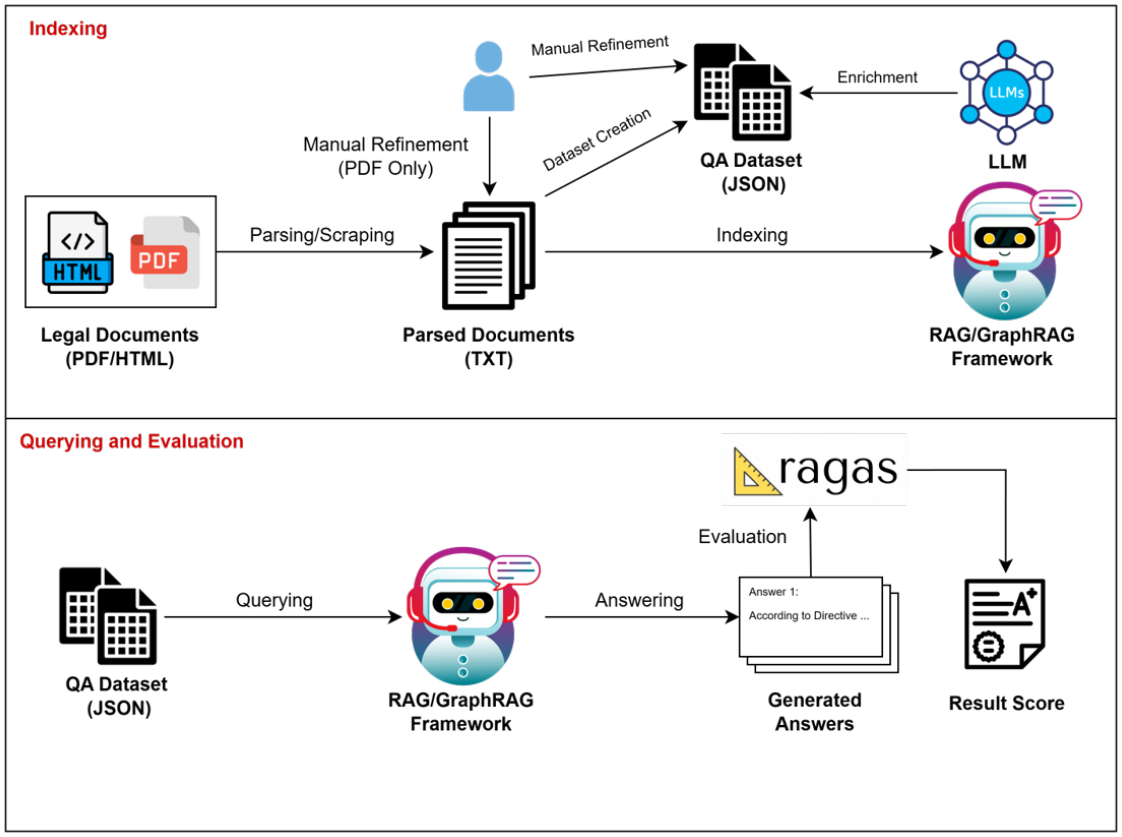

Figure 1: Overall Pipeline of Our Work.

each question is limited to a single regulation, the one from which the question was originally derived, rather than across all six documents. This setup ensures a focused and fair assessment of grounding and relevance within the context of the source regulation.

3.4. Evaluation Metrics

For evaluation, we use Ragas[11] [10], a framework designed to assess the outputs of RAG pipelines automatically. Since our dataset includes verified ground truth answers, we focus on evaluating the alignment between these ground truth answers and the model-generated responses. To do this, we employ the **answer accuracy**[12] metric, which measures the degree of agreement between the model's answer and the reference answer automatically using LLMs (in this case, OpenAI's GPT-4o). The evaluation involves two distinct prompts, each yielding a rating on a 0, 2, or 4 scale. These ratings are then normalized to a [0, 1] range and averaged to produce the final score, with higher values indicating better performance. We prioritize answer accuracy as it offers a direct and interpretable measure of factual correctness, which is particularly critical in legal QA settings.

Furthermore, to facilitate qualitative analysis, we classify the Ragas' resulting scores into three categories: GOOD, AVERAGE, and BAD. A score is labeled as GOOD if it is greater than 0.5, indicating strong agreement between the generated and ground truth answers, suggesting a high-quality response. Scores exactly equal to 0.5 are considered AVERAGE, reflecting a moderate alignment that may benefit from further refinement. Lastly, scores below 0.5 fall into the BAD category, signaling poor alignment and a need for significant adjustment.

[11] https://docs.ragas.io/en/stable/
[12] https://docs.ragas.io/en/stable/concepts/metrics/available_metrics/nvidia_metrics/#answer-accuracy

Table 4
Language Model Configurations for Each Framework.

Framework	Embedding Model	Generative LLM
HippoRAG 2	`mContriever` [11]	`GPT-4o`[13]
Nano GraphRAG	`text-embedding-3-small`[14]	`GPT-4o`[13] and `GPT-4o mini`[15]
LightRAG	`text-embedding-3-small`[14]	`GPT-4o`[13]
LlamaIndex	`text-embedding-3-small`[14]	`GPT-4o`[13]
Naïve RAG	`text-embedding-3-small`[14]	`GPT-4o`[13]

3.5. System Requirements and Reproducibility Note

To facilitate reproducibility, we emphasize that running our benchmark does not require high-performance computing infrastructure. Since the majority of the pipeline relies on external API calls, particularly for OpenAI's embeddings and LLMs, the experiment can be conducted on widely accessible platforms like Google Colab or a standard computer. However, if working with massive datasets with opting to compute embedding locally, systems equipped with more powerful GPUs and ample memory may speed up the process, especially the indexing phase.

4. Results and Analysis

This section presents the experimental results based on the previously described documents and QA datasets. A detailed explanation of these results is also provided.

4.1. Experimental Results

All experiments were conducted using the default configurations of each framework. Table 4 lists the embedding models and generative LLMs employed in the experiments. Table 5 presents the evaluation results for each framework across the different documents. In this table, the average answer accuracy scores are reported in the AA column, while the percentage of GOOD answers is shown in the PG column. Finally, the elapsed indexing and querying time for each framework which can be used to assess practicality, is reported in Table 6. The distribution of querying time for each framework is also displayed in Figure 2. We recorded the time data on the experiments involving Directive (EU) 2024/2853.

Table 5
Experimental Results by Framework. AA = Answer Accuracy, PG = Percentage of GOOD Answers (in %).

Framework	EU 2014/55		EU 2019/770		EU 2024/2853		PP 58/2023		PP 36/2021		PP 7/2025		Average	
	AA	PG	AA	PG	AA	PG	AA	PG	AA	PG	AA	PG	AA	PG
HippoRAG 2	0.72	72.41	0.75	67.92	0.77	71.74	0.94	94.44	0.82	80.46	0.69	72.22	0.78	76.53
Nano GraphRAG	0.56	58.62	0.65	54.72	0.62	45.65	0.71	66.67	0.64	63.22	0.71	77.78	0.65	61.11
LightRAG Mix	**0.84**	**96.55**	0.84	**88.68**	0.80	73.91	**0.96**	**100.00**	0.86	87.36	0.83	**88.89**	**0.86**	**89.23**
LightRAG Local	0.71	68.97	0.60	50.94	0.57	52.17	0.54	50.00	0.68	60.92	0.82	77.78	0.65	60.13
LightRAG Hybrid	0.78	82.76	0.73	71.70	0.62	60.87	0.82	83.33	0.77	74.71	**0.86**	**88.89**	0.76	77.04
LlamaIndex PG[16]	0.51	44.83	0.57	45.28	0.60	50.00	0.51	50.00	0.58	50.57	0.65	66.67	0.57	51.22
LlamaIndex Hybrid	0.78	82.76	**0.87**	86.79	**0.84**	**80.43**	0.90	88.89	0.91	89.66	0.82	83.33	0.85	85.31
Naïve RAG	0.83	**96.55**	0.84	**88.68**	0.72	71.74	0.82	88.89	**0.92**	**95.40**	0.82	77.78	0.82	86.51

[13] https://platform.openai.com/docs/models/gpt-4o
[14] https://platform.openai.com/docs/models/text-embedding-3-small
[15] https://platform.openai.com/docs/models/gpt-4o-mini
[16] Short for Property Graph

Table 6
Summary Statistics of Querying Time (QT) and Indexing Time (IT). Information is presented in seconds (s).

Framework	Avg QT	Min QT	Max QT	Sum QT	IT
GraphRAG-based Frameworks					
HippoRAG 2	4.18	**0.04**	10.01	192.24	120.00
Nano GraphRAG	7.38	3.66	12.62	339.43	103.00
LightRAG Mix	9.31	5.64	14.50	428.35	397.00
LightRAG Local	6.44	3.34	20.48	296.38	397.00
LightRAG Hybrid	7.40	3.82	14.29	340.31	397.00
LlamaIndex PG	3.29	2.21	**5.98**	151.50	**29.40**
LlamaIndex Hybrid	**3.05**	1.72	6.96	140.26	103.00
Non-GraphRAG-based Framework					
Naïve RAG	1.76	0.78	4.99	80.74	5.90

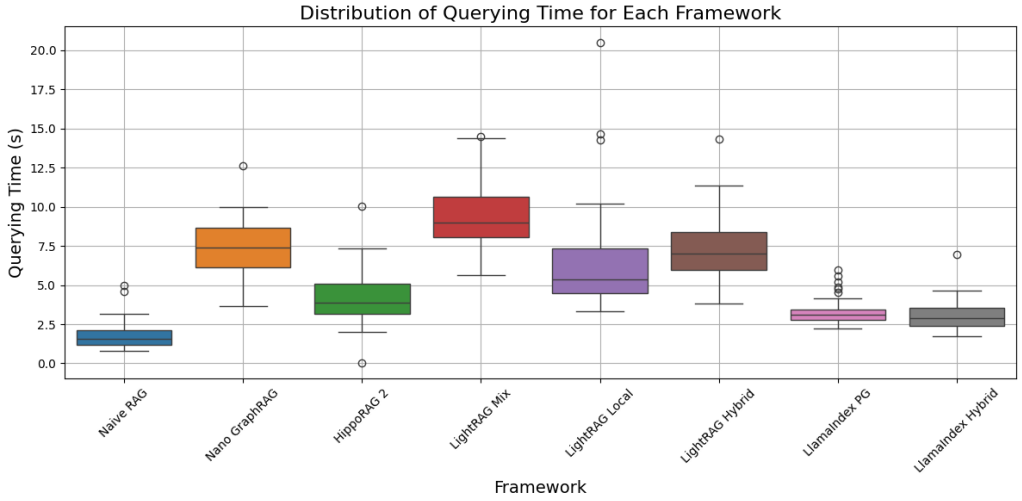

Figure 2: Distributions of Querying Time.

4.2. Explanation

Overall, LightRAG Mix demonstrates consistently strong performance across all six legal documents, achieving an average AA of 0.86 and an average PG of 89.23%. It attained the highest PG scores in four out of six cases, including a perfect 100% on PP 58/2023. Notably, it ranks within the top-3 for both AA and PG across all documents, underscoring its robustness and reliability. These high scores suggest that LightRAG Mix provides accurate and stable predictions across a diverse range of regulatory and legal texts. However, its longer indexing and querying times may pose a limitation in time-sensitive scenarios, suggesting a trade-off between effectiveness and efficiency.

LlamaIndex Hybrid also performs competitively, attaining the highest AA scores on EU 2019/770 and EU 2024/2853. Similar to LightRAG Mix, it consistently ranks in the top-3 across all documents for both AA and PG, highlighting the effectiveness of the hybrid retrieval strategy which combines text and KG retrieval. Coupled with its relatively fast querying and indexing time compared to other GraphRAG-based frameworks, LlamaIndex Hybrid stands out as one of the most balanced approaches, offering strong performance while maintaining practical efficiency.

On the other hand, LlamaIndex PG, which exclusively relies on a property graph constructed from the text documents, consistently lags behind in performance. While it benefits from highly efficient querying and indexing, the results suggest that property graph-based retrieval alone may be insufficient, especially when handling nuanced or implicitly expressed information that is common in legal texts

but not explicitly encoded in the KG. This highlights the limitations of relying solely on structured representations without complementary retrieval mechanisms that capture semantic context.

Interestingly, the naïve RAG baseline, despite its simplicity, performs comparably to more advanced methods on certain documents, achieving notably high PG scores on EU 2014/55, EU 2019/770, and PP 36/2021. This highlights that for certain document types, a straightforward RAG without incorporating a KG can still yield high-quality responses. It is also significantly faster than GraphRAG-based frameworks, further reinforcing its practicality and reliability as a lightweight retrieval option.

The Nano GraphRAG and LightRAG Local variants demonstrate mixed results. Their lower scores suggest that limiting retrieval to local neighborhoods within the KG can hinder performance. By focusing solely on low-level retrieval, i.e., emphasizing entities and their immediate neighbors [8], these models struggle to capture the broader context required to accurately interpret complex legal documents. In terms of querying and indexing time, this localized approach does not necessarily translate into greater efficiency. For instance, LightRAG Local records the longest maximum querying time (20.48s) and the average querying time of both Nano GraphRAG (7.38s) and LightRAG Local (6.44s) rank as the third and fourth highest among all tested frameworks, respectively. This reflects the overhead incurred by handling numerous fine-grained local queries without the benefits of higher-order contextualization.

LightRAG Hybrid, the full version of LightRAG combines both low-level (local) and high-level (global) retrieval by retrieving a broad set of relationships while simultaneously conducting in-depth exploration of specific entities [8]. Empirically, this results in balanced performance across multiple documents and metrics, with LightRAG frequently ranking within the top-4 in both AA and PG. Although its querying time is generally higher than the local variant, the added retrieval depth proves worthwhile, offering a strong trade-off between effectiveness and efficiency for comprehensive legal QA.

Lastly, HippoRAG 2 also delivers solid performance across the board, where it frequently places within the top-4 of PG across multiple documents. While it does not outperform the top models in terms of accuracy, it maintains competitive efficiency, with relatively fast querying and indexing times.

From a temporal efficiency perspective, the results reveal a clear performance hierarchy. Naïve RAG significantly outperforms all GraphRAG-based frameworks in both querying time (1.76s average) and indexing time (5.90s). Among the GraphRAG-based frameworks, LlamaIndex Hybrid achieves the fastest average querying time (3.05s), while LlamaIndex PG demonstrates the most efficient indexing (29.40s). However, even the fastest GraphRAG approach requires nearly twice the querying time of naïve RAG, highlighting the computational overhead inherent in graph-based retrieval mechanisms.

This temporal analysis highlights a fundamental trade-off between accuracy and efficiency. While GraphRAG-based frameworks like LightRAG Mix and LlamaIndex Hybrid deliver superior accuracy and answer quality, they come at the cost of increased computational complexity. The choice between approaches should therefore be guided by application requirements, naïve RAG for quick responses, and GraphRAG-based methods for applications where answer quality and accuracy justifies the additional querying and indexing time and the associated computational resources.

5. Discussion

This chapter analyzes and interprets the results of the evaluation as in Section 4. The goal is to better understand the relative strengths and limitations of each system in answering regulatory and legal questions.

5.1. Answer Length

One of the key observations from our evaluation is that longer answers tend to reduce overall accuracy. While detailed responses can be beneficial for broader questions, they often introduce unnecessary or hallucinated content that detracts from the correctness of the answer. This issue was especially apparent in Nano GraphRAG, which in its default setting, frequently generates overly elaborate responses. These answers may contain information not supported by the retrieved context, which undermines their reliability.

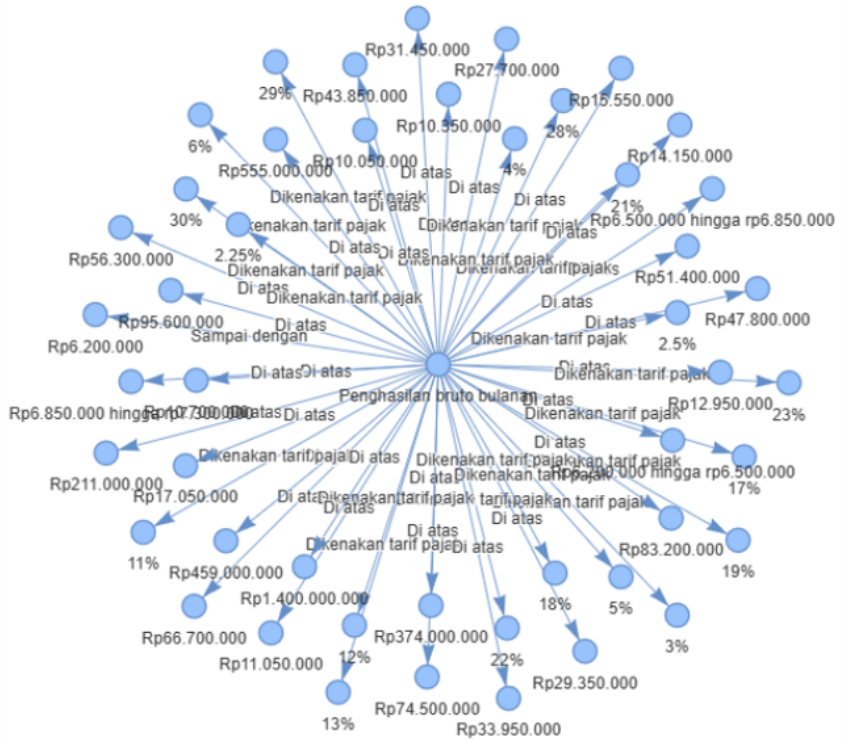

Figure 3: Constructed KG for PP 58/2023 using LlamaIndex's Property Graph. "*Dikenakan tarif pajak*" = subject to tax rate, "*Sampai dengan*" = up to, or "*Di atas*" = above.

This phenomenon highlights a trade-off between completeness and precision. While longer, more detailed answers may seem thorough, they also carry a higher risk of drifting away from the retrieved source material. On the other hand, more concise answers tend to stay closer to the retrieved content, making them more accurate and trustworthy. However, overly brief responses may lack sufficient detail to fully address the question. This was evident in Nano GraphRAG's performance, where excessively long answers often led to decreased grounding quality. These observations suggest that finding an optimal response length is crucial to balancing informativeness and faithfulness.

We further validated this in an experiment using LightRAG Mix on the Directive (EU) 2019/770 document. By switching the response type from a `Multiple Paragraphs` setup to a `Single Paragraph` configuration, the model's accuracy improved from 0.844 to 0.887. Overall, controlling answer length is a critical factor in improving the effectiveness of RAG-based systems. Further experiments might include observing the trade-off between the accuracy and conciseness[17] metrics in order to find the optimal response length.

5.2. KG-Only Systems

KG-only systems often struggle because they introduce an abstraction layer that may fail to preserve the full meaning or detail of the original text. In our evaluation, LlamaIndex's Property Graph, which uses only KG representations, frequently generated incorrect or incomplete answers. This suggests that important contextual or relational nuances from the source documents were lost or not fully captured during the graph construction process.

For instance, in PP 58/2023, the limitations of a KG-only approach are clearly illustrated through its representation of income tax brackets. The KG, as shown in Figure 3, connects various income thresholds (e.g., Rp74.500.000) with tax rates (e.g., 5%, 10%) via overly generic edges like "*Dikenakan tarif pajak*" (subject to tax rate), "*Sampai dengan*" (up to), or "*Di atas*" (above) without establishing explicit, interpretable links between each income category and its respective tax percentage. This leads

[17] https://github.com/langchain-ai/openevals?tab=readme-ov-file#conciseness

to ambiguity and poor semantic clarity, making it difficult for the system to accurately answer questions about tax categorization. As a result, only one out of several tax-related questions was answered correctly, with an overall average accuracy of 0.51. In contrast, when the system was complemented with the original source text, the accuracy significantly improved to 0.90. However, the indexing and querying time using this hybrid system is slightly higher due to the additional overhead of retrieving and processing both structured graph data and unstructured textual content simultaneously.

5.3. Multilingual Capabilities

Most RAG systems are primarily optimized for English by default. This assumption poses significant challenges when applying these systems to non-English corpora, such as Indonesian Government Regulations. One critical bottleneck lies in the use of language-specific embedding models. For instance, our initial evaluation on PP 58/2023 using HippoRAG 2 with Contriever [11], which was trained predominantly on English data, resulted in poor retrieval performance, reflected by a lower average answer accuracy of just 0.69. In contrast, switching to mContriever, a multilingual variant trained on a diverse corpus that includes Indonesian, significantly improved the performance to an average answer accuracy of 0.94, as in Table 5. This highlights the importance of selecting embedding models that are compatible with the target language.

In addition to embedding selection, prompt design also plays a key role in multilingual performance. Our further experiment with HippoRAG 2 on PP 7/2025 showed that explicitly stating in the prompt that the documents were in Indonesian resulted in a noticeable boost in answer accuracy from 0.69 to 0.78. This suggests that LLMs may implicitly rely on language priors unless clearly instructed otherwise. Without such cues, even a strong model might attempt to interpret documents through an English-biased lens, potentially leading to misinterpretations.

6. Conclusion

In this research, we performed benchmarking over several popular KG-based and hybrid RAG in the context of legal QA. We evaluated the performance of four frameworks, including HippoRAG 2, Nano GraphRAG, LightRAG (in three configurations: `mix`, `local`, and `hybrid`), LlamaIndex (in two configurations: property graph and hybrid), and naïve RAG. These systems were assessed using multilingual legal documents from both the European Union and Indonesia, encompassing various domains such as taxation, labor regulation, and digital commerce.

To support our evaluation, we constructed a high-quality QA dataset through a semi-automated pipeline combining LLMs and human annotators. This approach enabled us to generate diverse and accurate QA pairs grounded in legal documents, with refinements ensuring clarity and faithfulness to the source texts. For evaluation, we adopted one of the metrics provided by Ragas, the answer accuracy, which quantifies answer quality based on alignment with ground truth. By focusing on this normalized answer accuracy score and the derived percentage of high-quality (GOOD) responses, we were able to draw meaningful comparisons across different frameworks.

Our results indicate that hybrid systems, especially LightRAG Mix and LlamaIndex Hybrid, consistently outperform KG-only baseline in both accuracy and reliability. The fusion of structured and unstructured retrieval proves effective in handling complex legal content. In contrast, KG-only approaches often struggle due to the abstraction and loss of contextual nuance, while verbose answers tend to introduce hallucinated content. The observed performance drops in these cases underscore the importance of grounding, controlled verbosity, and semantic coverage in RAG-based legal systems.

Overall, this study provides critical empirical insights into the trade-offs between interpretability, efficiency, and accuracy in legal QA systems. It not only highlights the promise of hybrid RAG architectures but also underscores the value of careful dataset construction and targeted evaluation. Our benchmark offers a foundation for future research and development of more trustworthy, multilingual, and domain-sensitive RAG solutions for legal and other high-stakes applications.

Future Work. Future work may enhance the adaptability of legal RAG systems by expanding the dataset to include case law, cross-jurisdictional regulations, and real-world legal queries. Developing interactive interfaces and incorporating rule-based reasoning could support complex tasks such as compliance checking. We also plan to incorporate additional Ragas metrics, such as faithfulness and context relevance, for a more holistic evaluation beyond answer accuracy, which we currently prioritize for its direct interpretability. Furthermore, exploring cross-document retrieval is a promising direction, though it introduces a larger search space and a higher risk of retrieving irrelevant content. Moreover, adaptive response length mechanisms, such as dynamic truncation or expansion based on query complexity, may further improve answer precision, as supported by granularity-aware legal QA research [12]. Lastly, our future direction is to incorporate open-source LLMs to benchmark their performance against proprietary models, thus creating a more comprehensive evaluation that addresses the critical aspects of accessibility and reproducibility. Since our pipeline is mostly built using LangChain, substituting to an open-source model is a straightforward task.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT and Claude for grammar and spelling checking and paraphrasing. The author(s) also used Perplexity to obtain an initial set of related work for this paper. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Curran Associates Inc., Red Hook, NY, USA, 2020.

[2] B. Peng, Y. Zhu, Y. Liu, X. Bo, H. Shi, C. Hong, Y. Zhang, S. Tang, Graph retrieval-augmented generation: A survey, 2024. URL: https://arxiv.org/abs/2408.08921. arXiv:2408.08921.

[3] B. Sarmah, D. Mehta, B. Hall, R. Rao, S. Patel, S. Pasquali, Hybridrag: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction, in: Proceedings of the 5th ACM International Conference on AI in Finance, ICAIF '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 608–616. URL: https://doi.org/10.1145/3677052.3698671. doi:10.1145/3677052.3698671.

[4] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitansky, R. O. Ness, J. Larson, From local to global: A graph rag approach to query-focused summarization, 2025. URL: https://arxiv.org/abs/2404.16130. arXiv:2404.16130.

[5] B. J. Gutierrez, Y. Shu, Y. Gu, M. Yasunaga, Y. Su, HippoRAG: Neurobiologically inspired long-term memory for large language models, in: The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024. URL: https://openreview.net/forum?id=hkujvAPVsg.

[6] B. J. Gutiérrez, Y. Shu, W. Qi, S. Zhou, Y. Su, From rag to memory: Non-parametric continual learning for large language models, 2025. URL: https://arxiv.org/abs/2502.14802. arXiv:2502.14802.

[7] G. Ye, nano-graphrag: A simple, easy-to-hack graphrag implementation, https://github.com/gusye1234/nano-graphrag, 2025.

[8] Z. Guo, L. Xia, Y. Yu, T. Ao, C. Huang, Lightrag: Simple and fast retrieval-augmented generation, 2025. URL: https://arxiv.org/abs/2410.05779. arXiv:2410.05779.

[9] J. Liu, LlamaIndex, 2022. URL: https://github.com/jerryjliu/llama_index. doi:10.5281/zenodo.1234.

[10] S. Es, J. James, L. Espinosa Anke, S. Schockaert, RAGAs: Automated evaluation of retrieval augmented generation, in: N. Aletras, O. De Clercq (Eds.), Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations,

Association for Computational Linguistics, St. Julians, Malta, 2024, pp. 150–158. URL: https://aclanthology.org/2024.eacl-demo.16/.

[11] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, E. Grave, Unsupervised dense information retrieval with contrastive learning, 2021. URL: https://arxiv.org/abs/2112.09118. doi:10.48550/ARXIV.2112.09118.

[12] D. Faisal, F. Darari, R. Ryanda, Granularity-aware legal question answering: a case study of indonesian government regulations, International Journal of Advances in Intelligent Informatics 10 (2024) 359–378. URL: https://ijain.org/index.php/IJAIN/article/view/1105. doi:10.26555/ijain.v10i3.1105.