Retrieval-Augmented Generation and Knowledge Graphs in Portuguese-Language Legal Documents

Vinícius Teles de Oliveira, Deivison Oliveira da Silva,
Mateus de Almeida Souza, Maurício Rodrigues Lima,
Sávio Salvarino Teles de Oliveira, Thierson Couto Rosa
Instituto de Informática - UFG
{viniteles, deivison, mateussouza, mauricio.rodrigues}@discente.ufg.br
{savioteles, thierson}@ufg.br

Abstract

This paper introduces a Graph Retrieval-Augmented Generation (GraphRAG) pipeline tailored for Question Answering (Q&A) within Portuguese legal documents. Applied to a corpus of 203 normative resolutions from Companhia Energética de Minas Gerais (CEMIG), the proposed approach addresses the structural complexity of legal texts, such as hierarchical dependencies and temporal modifications. By explicitly modeling documents as knowledge graphs with nodes representing structural units (Articles, Paragraphs, Items) and edges denoting normative relationships, the system preserves context and traceability. The retrieval mechanism reconstructs evidence paths from root to leaf, performing semantic re-ranking before generation. Evaluation using the RAGAS framework yielded a mean answer accuracy of 0.81, with a median of 1.00. Results indicate that the system performs robustly on short, focused queries, while intermediate-length questions present challenges related to semantic dispersion. The findings suggest that structurally aware retrieval significantly enhances the interpretability and precision of legal Q&A systems.

1 Introduction

The interpretation of legal documents is an inherently complex cognitive task that transcends mere text comprehension, often requiring domain-specific reasoning to resolve ambiguities and intricate cross-references (Chalkidis et al., 2021). Unlike general open-domain texts, normative acts are characterized by rigid hierarchical organizations and temporal validity constraints, where the efficacy of a provision is frequently contingent upon modifications or revocations defined in external documents (van Opijnen and Santos, 2017). Consequently, practitioners in the Brazilian legal system face significant challenges in navigating voluminous collections of unstructured text, rendering traditional keyword-based retrieval insufficient for capturing the full semantic context required for decision-making (Oliveira and Sperandio Nascimento, 2021).

To address these challenges, legal *Question Answering* (Q&A) systems require high levels of precision, traceability, and adaptability to normative changes. In collections of resolutions, ordinances, and laws, the hierarchical structure (document → article → paragraph → item) coexists with high semantic density and interdependencies such as amendments and revocations. Recently, *Retrieval-Augmented Generation* (RAG) approaches have proven effective in reducing hallucinations and anchoring responses in retrieved evidence (Benita et al., 2024; Perron et al., 2025; Packowski et al., 2024). However, when internal relationships between provisions are essential for interpretation, flat retrieval methods often fail to capture the full context. In these scenarios, graph-based representation and path-sensitive queries enhance semantic coverage and explainability, motivating variants such as *GraphRAG* (Naganawa and Hirata, 2025; Phyu et al., 2024; Carvalho et al., 2025).

This work proposes a specialized *GraphRAG* pipeline applied to the internal resolutions of Companhia Energética de Minas Gerais (CEMIG). The indexing module explicates the normative structure through a lightweight knowledge graph comprising four node types (Document, Article, Paragraph, Item) and three edge types (contain, modify, revoke). To ensure structural precision, entities are extracted using regular expressions, while semantic relationships are inferred by a Large Language Model (LLM). The retrieval mechanism is designed to reconstruct complete evidence paths from root to leaf, performing semantic re-ranking to prioritize relevance. Subsequently, the generation stage executes a single LLM call based on this structured context, preserving traceability. Thus, the proposed GraphRAG approach enhances contextualization and provides auditable answers through multiple

Proceedings of the 17th International Conference on Computational Processing of Portuguese (PROPOR 2026)
Vol. 1, pages 1–10

semantic hops (Fan et al., 2024).

The main contributions of this study are: a *GraphRAG* pipeline tailored for normative documents; a path-oriented retriever that assembles semantically complete evidence prior to generation; and an empirical evaluation on a real-world legal collection, analyzing the impact of question length on system performance.

2 Related Work

Retrieval-Augmented Generation (RAG) has emerged as a dominant paradigm for knowledge-intensive tasks, particularly in the legal domain, where hallucination minimization is critical. Some works have demonstrated that RAG systems significantly outperform standard LLMs in citing statutes and precedents. For instance, Cui et al. (2023) introduced *ChatLaw*, employing keyword-based retrieval alongside vector embedding to filter irrelevant regulations. Similarly, systematic reviews by Benita et al. (2024) highlight that while vector-based RAG captures semantic similarity, it often struggles with the precise lexical matching required for rigid legal terminologies.

To address the limitations of flat vector retrieval, approaches integrating Knowledge Graphs (KGs) have gained traction. Edge et al. (2024) formally defined *GraphRAG*, demonstrating that graph-based indices allow LLMs to answer "global" questions that require aggregating information across disparate document sections. Building on this, Sarmah et al. (2024) proposed *HybridRAG*, combining the precision of graph traversal with the scalability of vector search. In the legal context, Naganawa and Hirata (2025) showed that graph-enhanced pipelines provide superior explainability compared to pure GPT-4 baselines when navigating complex regulatory frameworks.

Recent studies have also explored hierarchical and path-oriented retrieval mechanisms that reconstruct contextual evidence through structured traversals. Approaches such as TreeRAG (Tao et al., 2025) organize documents as hierarchical trees and retrieve branches that preserve the structural dependencies between sections of a document. Similarly, PathRAG (Chen et al., 2025) retrieves relational paths from knowledge graphs and converts them into textual sequences that guide LLM reasoning through multi-hop contextual evidence. These methods share the intuition that preserving structural relationships between document segments can improve contextual coherence during generation.

A crucial distinction in current literature lies in graph construction. Most GraphRAG approaches, such as those reviewed by Pan et al. (2024), rely on LLMs to extract entities and relations from unstructured text, which can be computationally expensive and prone to extraction noise. In contrast, our work operates on *semi-structured* normative documents whose standardized drafting patterns allow the underlying hierarchy (Document $\rightarrow$ Article $\rightarrow$ Paragraph $\rightarrow$ Item) to be deterministically extracted using regular expressions. While related approaches such as TreeRAG and PathRAG also reconstruct contextual evidence through hierarchical or relational paths, they generally assume generic document structures or rely entirely on LLM-based graph construction. Our pipeline instead combines rule-based structural extraction with LLM inference only for cross-document semantic relations such as *modify* and *revoke*. This design reduces indexing cost and improves extraction reliability while enabling a retrieval process based on path linearization and semantic re-ranking, producing context sequences that preserve the complete normative hierarchy before generation.

3 Methodology

The legal documents follow a standardized organization as presented in the diagram of Figure 1. This organization allows the documents to be partitioned based on their own internal composition.

3.1 Nodes

To ensure complete information preservation, the graph nodes were designed according to the internal structure of the documents, which allows for the full utilization of the text contained in them. In turn, the graph edges were designed to represent the relationships that the different nodes have with one another.

Each node in the graph contains a descriptive text corresponding to the element it maps. For example, the descriptive text of node "article 2" is the text corresponding to "article 2" in the mapped document. These texts, in turn, carry some information. Descriptive texts with many characters tend to contain multiple pieces of information, which can dilute the conceptual focus of a node. When a text addresses several topics, RAG struggles to accurately identify the core meaning that the node is meant to convey (Hindi et al., 2025). Therefore,

it is essential that each node be described concisely and thematically, facilitating its indexing and retrieval.

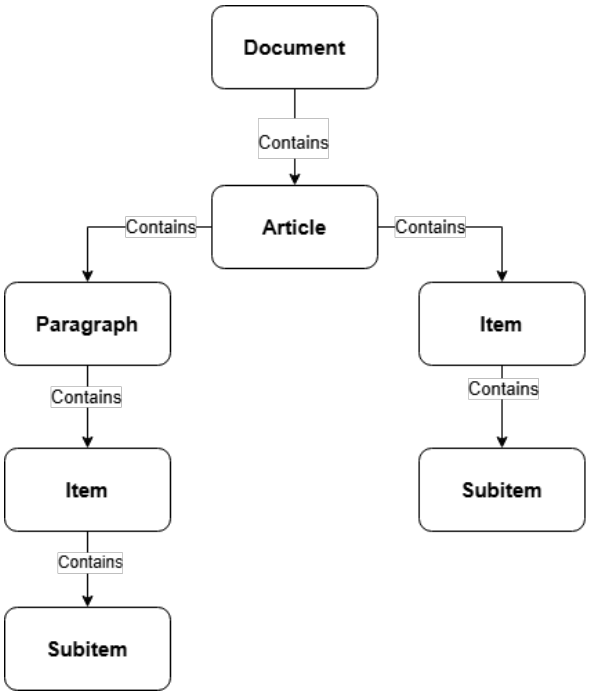

Figure 1: Document structure diagram

To ensure that each node's descriptive text remained concise and thematically cohesive, it was first necessary to define an evaluation metric. The parameter adopted was the number of edges (relations) generated by each node. A node that presents a high number of connections indicates that the information it contains is related to many other elements in the graph, reflecting an excessively broad context. This excessive scope can compromise precision in RAG-based approaches (Hindi et al., 2025). Therefore, nodes with fewer edges are beneficial.

Using the structural types shown in Figure 1 as an example, it was observed that when the graph nodes were only of the "article" type, some reached hundreds of edges. When the types included both "article" and "paragraph," there was a reduction in the density of relations, although some nodes still had hundreds of connections. The inclusion of the types "article," "paragraph," and "item" resulted in a significantly lower relation density, generally limited to just a few. However, when attempting to increase the number of node types (for example, by including "subitems"), no improvement in relation density was observed. To avoid unnecessarily increasing the graph's complexity, subitems and other smaller structures were not treated as node types, but only as parts of the descriptive text of the recognized graph node types.

To ensure that all information contained in the documents was utilized, it was necessary to create a new node type to encompass it. Finally, the graph node types were defined as: document (responsible for containing the structure preceding the first article), article, paragraph, and item.

3.2 Edges

In this case study, the edges represent the relationships between the nodes. Ideally, nodes should have few edges; therefore, the edges must capture only the essential relationships.

The edge types were divided into two groups: one responsible for guiding relationships between nodes originating from the same document, and another for those between nodes from different documents. For the first group, the "contain" relationship was selected, which derives from the internal document structure shown in Figure 1. For the second, the relationships "modify" and "revoke" were chosen, since when documents do not encapsulate the information within their own content, they alter the information contained in others. In other words, the edges were also constructed based on the structural composition of the documents.

3.3 Graph

After selecting the composition of nodes and edges, the formalization of documents through knowledge graphs followed a structure consisting of four main node types (Document, Article, Paragraph, and Item) and three edge types (Contain, Modify, and Revoke).

Each node contains, among other attributes, a unique identifier, a type, and a descriptive text that characterizes it. Similarly, each relationship has its own identifier, a source node, a target node, and a type defining the nature of the connection between graph elements.

As illustrated in Figure 2, the documents are structured so that they contain articles, which, in turn, may contain paragraphs and items. Paragraphs may also include items, reflecting the typical hierarchy of normative texts. The "contain" relationship indicates that a node is embedded within another, representing structural inclusion among elements. The "modify" relationship expresses that one node introduces changes to another, recording normative or textual modifications. Finally, the

"revoke" relationship is considered a specific case of modification but with distinct significance, as to revoke means to nullify, cancel, or render void. Thus, when this relationship occurs, it indicates that the portion of the graph it points to is "dead" and should be disregarded during answer generation.

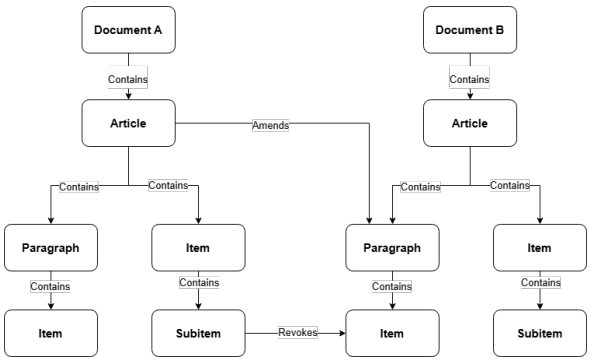

Figure 2: Graph structure

The extraction of the elements that compose the graph was carried out in two distinct processes: the first focused on identifying entities, which correspond to the graph nodes, and the second on identifying the relationships between those nodes. The way these elements were extracted was based on three main factors: the standardized structure of the documents, the format in which they were made available (all as PDF files), and the way references to other documents were made—usually through clickable links.

Initially, the text of all documents was extracted from the PDF files and converted to Markdown format to facilitate information processing and organization. Although some documents contained images, none of them carried relevant information (they were merely logos of the same company in different sizes) and were therefore discarded to maintain the simplicity of the solution—keeping it within the context of a single modality (text only). The output format, *Markdown*, was chosen so that the *hyperlinks* (links that point to another location) contained in the PDFs would be preserved, as they indicated that one document was citing another and were therefore essential for identifying relationships among documents.

3.3.1 Entities

The entities were extracted using regular expressions (Regex) (Stanley et al., 2018), which define specific search patterns. After extraction, these entities were converted into embeddings and stored in a database. This approach proved suitable due to the standardized nature of the analyzed documents.

However, using large language models (LLMs) for entity recognition was not feasible within a reasonable execution time. This occurred because some documents contained hundreds of thousands of characters, leading to incomplete entity recognition or confusion among them. Furthermore, the documents contained both entities belonging to the text itself and references to external entities. LLMs frequently failed to distinguish between these two groups.

3.4 Retriever

The retrieval module is designed to reconstruct the semantic context of legal norms by traversing the knowledge graph structure. Unlike traditional retrieval methods that treat document segments as independent units, this approach considers the hierarchical and relational dependencies between normative elements. The process formally maps a user query q to a ranked set of contextualized evidence paths. This procedure consists of two main phases: *Subgraph Induction* and *Path-Based Contextualization*.

3.4.1 Subgraph Induction and Expansion

The first phase aims to identify a relevant subgraph containing potential answers and their immediate structural neighbors. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the knowledge graph, where $\mathcal{V}$ represents the set of entities (Documents, Articles, Paragraphs, Items) and $\mathcal{E}$ represents the normative relationships.

Given a query q, we first compute its vector embedding v_q. A semantic similarity search is performed against the embeddings of all nodes in $\mathcal{V}$ to identify a set of seed nodes $\mathcal{S}$. This set contains the top-k entities with the highest cosine similarity to v_q. Formally, $\mathcal{S} \subset \mathcal{V}$ such that $|\mathcal{S}| = k$.

To capture the surrounding context, the system induces a subgraph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ derived from these seeds. The vertex set $\mathcal{V}'$ is defined as the union of the seed nodes and their direct neighbors in $\mathcal{G}$. Let $\mathcal{N}(v)$ denote the set of nodes adjacent to v. The expanded node set is defined as:

$$\mathcal{V}' = \mathcal{S} \cup \bigcup_{v \in \mathcal{S}} \mathcal{N}(v) \tag{1}$$

Consequently, $\mathcal{E}'$ comprises all edges connecting nodes within $\mathcal{V}'$. This expansion is crucial in the legal domain because a relevant paragraph (identified by similarity) often requires its parent article

or modifying clauses (identified by adjacency) to be fully understood.

3.4.2 Path Linearization and Re-ranking

The second phase transforms the induced subgraph $\mathcal{G}'$ into coherent textual sequences suitable for generation. Since legal documents follow a hierarchical structure, valid interpretation requires reading from the general scope to the specific provision. We model this as a root-to-leaf traversal within $\mathcal{G}'$.

Nodes in $\mathcal{G}'$ with an in-degree of zero are designated as local roots, while nodes with an out-degree of zero are designated as leaves. The system extracts all unique directed paths $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$ starting from a root and terminating at a leaf.

For each path p_i, a linearized textual representation $T(p_i)$ is constructed. This is achieved by concatenating the descriptive text of the nodes along the path, interleaved with tokens representing the relationship types (e.g., *contains*, *modifies*). This process aggregates fragmented information—such as an Article and its specific Item—into a single, semantically complete passage.

Finally, to ensure the generated context aligns with the user's intent, the linearized paths are re-ranked. New embeddings are computed for each text $T(p_i)$. The system calculates the similarity between v_q and these path embeddings, returning the top-k paths as the final context for the generation model. Algorithm 1 formalizes the complete retrieval pipeline.

With the entities returned by the retriever, the generation defines the entire process of communication with the LLM to generate the answer to the user's question. In a single call to the LLM, the user's question and the descriptive text of each entity obtained from retriever are sent to the model, which then returns the answer to the user.

3.5 *Dataset*

The documents used are public and were collected directly from the portal of the Companhia Energética de Minas Gerais - Brazil (CEMIG) https://www.cemig.com.br/legislacao-do-setor-eletrico/. The collection resulted in 203 normative resolutions (collegiate acts that regulate specific matters or amend internal rules) with an average of 32,458 characters per document. These resolutions were classified into two groups: base-type resolutions, which establish initial regulations, and leaf-type resolutions, which amend, complement, or revoke previous ones. For example, Resolution 846/2019 defines rules for calculating fines in the electricity sector, while Resolution 852/2019 modifies paragraphs 8 and 9 and adds paragraph 10.

Algorithm 1 Graph-Aware Context Retrieval

Require: Query q, Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, parameters k_{seed}, k_{final}
Ensure: Set of contextualized paths $\mathcal{C}_{final}$

{**Phase 1: Subgraph Induction**}
$v_q \leftarrow \text{EMBED}(q)$
$\mathcal{S} \leftarrow \text{TOPKSIMILAR}(\mathcal{V}, v_q, k_{seed})$
$\mathcal{V}' \leftarrow \mathcal{S}$
for all $u \in \mathcal{S}$ **do**
  $\mathcal{V}' \leftarrow \mathcal{V}' \cup \{v \mid (u, v) \in \mathcal{E} \vee (v, u) \in \mathcal{E}\}$
end for
$\mathcal{E}' \leftarrow \{(u, v) \in \mathcal{E} \mid u \in \mathcal{V}' \wedge v \in \mathcal{V}'\}$
Let $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$
{**Phase 2: Path Construction**}
$Roots \leftarrow \{v \in \mathcal{V}' \mid \text{InDegree}_{\mathcal{G}'}(v) = 0\}$
$\mathcal{P} \leftarrow \emptyset$
for all $r \in Roots$ **do**
  $\mathcal{P} \leftarrow \mathcal{P} \cup \text{FINDPATHS}(r, \mathcal{G}')$ {DFS to leaves}
end for
{**Phase 3: Linearization and Re-ranking**}
$\mathcal{T} \leftarrow \emptyset$
for all $path \in \mathcal{P}$ **do**
  $text \leftarrow \text{LINEARIZE}(path)$ {Concatenate node texts and edge types}
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{(path, text)\}$
end for
Sort $\mathcal{T}$ by $\text{SIMILARITY}(v_q, \text{EMBED}(text))$ descending
return Top k_{final} elements from $\mathcal{T}$

The construction of the *dataset* used only leaf-type resolutions, considered terminal elements of the graph and the most up-to-date sources. The data generation process combined automatic production by Large Language Models (LLMs) and human review in four integrated stages. Initially, question–answer pairs were automatically generated from the selected documents, aiming to capture relevant information. Next, generic or low-informative questions, such as "When was the document published?", were discarded, keeping only those with higher technical detail, such as "What criteria must a consumer meet to gain free access to the public electricity service?". The resulting

answers were manually reviewed by data scientists with different levels of experience (junior, mid-level, and senior), ensuring accuracy and fidelity to the original content. Finally, the validated set was structured into a CSV file containing the columns: document name, question, and answer. The evaluation dataset contains 31 manually validated question–answer pairs derived from the normative resolutions. Although the corpus includes 203 resolutions, constructing reliable Q&A pairs over normative documents is challenging, since legal provisions frequently modify, complement, or revoke clauses from previous resolutions. As a result, questions must be carefully designed to ensure that answers remain legally valid and grounded in the most up-to-date regulatory context. For this reason, only questions that could be unambiguously validated against the final normative state were retained in the evaluation set. Due to confidentiality requirements associated with the ongoing institutional research project conducted in collaboration with CEMIG, the curated Q&A dataset cannot be publicly released at this stage.

3.6 Experimental Setup

To ensure the reproducibility of the results and the validation of the proposed architecture, the experimental environment was instantiated with specific model configurations. The vector space for semantic similarity, essential for both the initial node selection and the path re-ranking phases, was constructed using the gemini-embedding-001 model. This model provided the embeddings for both the graph nodes (during indexing) and the user queries (during retrieval).

Regarding the retrieval hyperparameters defined in the algorithmic approach, the system was set to extract and linearize the top $k = 5$ most relevant paths to compose the final context. This context was then fed into the generative component, which utilized the gemini-2.5-flash Large Language Model. We employed the model's default configurations for temperature, top-p, and context window size to assess its baseline performance and reasoning capabilities without the influence of aggressive sampling adjustments. Finally, the automated evaluation via the RAGAS framework relied on gpt-4 acting as the judge model to compute the answer accuracy metric, ensuring a rigorous assessment of factual consistency against the ground truth dataset.

4 Results

The performance evaluation relies on the `answer_accuracy` metric from the RAGAS framework, which measures the factual consistency of the generated answer relative to the ground truth. The proposed pipeline achieved a global mean accuracy of 0.81 ($SD = 0.25$) across the valid dataset ($N = 31$). Table 1 summarizes the descriptive statistics.

Table 1: Descriptive statistics for `answer_accuracy`.

Statistic	*Value*
Valid Samples (N)	31
Mean	0.810
Standard Deviation	0.245
Minimum	0.300
Q_1 (25%)	0.500
Median (50%)	1.000
Q_3 (75%)	1.000
Maximum	1.000

A distinct characteristic of the results is the ceiling effect observed in the median (1.00) and the third quartile (1.00). This indicates a negatively skewed distribution where the majority of the responses are perfectly accurate, while the variance is driven by a subset of retrieval failures rather than a uniform degradation of quality. Consequently, the system demonstrates high reliability for successful retrieval paths, but the dispersion suggests that specific query structures may still induce hallucinations or retrieval gaps.

To investigate the impact of query complexity on performance, we stratified the samples by token count quartiles, as detailed in Table 2. This stratification tests the hypothesis that increased query length introduces noise that might degrade vector-based retrieval or, conversely, provides necessary context for disambiguation.

Table 2: Mean `answer_accuracy` stratified by question length quartiles.

Question Length (tokens)	*n*	*Mean Accuracy*
Q_1: $(0, 14.0]$	8	0.887
Q_2: $(14.0, 16.0]$	9	0.789
Q_3: $(16.0, 19.0]$	7	0.729
Q_4: $(19.0, 25.0]$	7	0.829

The data reveals a non-linear relationship be-

tween question length and accuracy. The highest performance was observed in the first quartile (short queries, ≤ 14 tokens, $\mu \approx 0.89$), followed by the fourth quartile (long queries, > 19 tokens, $\mu \approx 0.83$). A performance dip occurs in the intermediate ranges ($14 < t \leq 19$), where accuracy drops to approximately 0.73.

These findings suggest that short queries benefit from high lexical overlap with specific entities (e.g., direct lookups of fines or deadlines), ensuring precise graph entry points. Conversely, long queries provide sufficient semantic context to guide the embedding model toward the correct vector space region despite the noise. The intermediate queries, however, appear to occupy a region of "semantic ambiguity," where they lack the directness of keyword-based inquiries yet fail to provide enough contextual constraints to fully disambiguate the retrieval path, leading to the observed dispersion in the second and third quartiles.

4.1 Distributional Analysis

The visualization of the `answer_accuracy` distribution reveals significant insights into the system's reliability profile. Figures 3 and 4 depict the probability density and the cumulative distribution function (CDF), respectively.

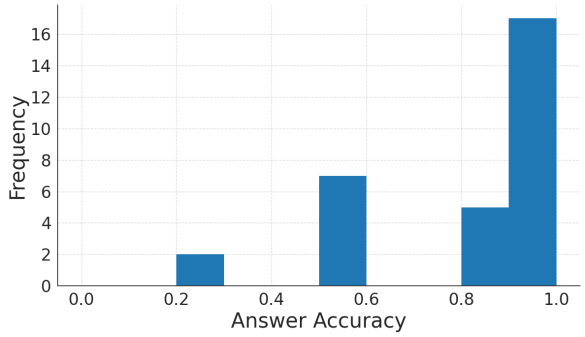

Figure 3: Frequency distribution of `answer_accuracy` showing significant negative skewness.

The histogram (Figure 3) exhibits a pronounced negative skewness, with the mode strictly located at 1.0. This distribution suggests that the retrieval pipeline operates in a "hit-or-miss" manner regarding optimal context: in the majority of cases, it retrieves the complete evidence path, leading to perfect accuracy. The tail of partial scores (< 1.0) indicates instances where the graph traversal captured relevant but incomplete nodes, resulting in answers that are factually grounded but lack total precision.

Complementing this view, the CDF in Figure 4 quantifies the system's robustness. The sharp vertical rise at $x = 1.0$ confirms that over 50% of the queries incur zero information loss. The stepwise accumulation in the lower percentiles reflects discrete degradation levels, likely corresponding to specific structural failures (e.g., retrieving an Article but missing a modifying Paragraph).

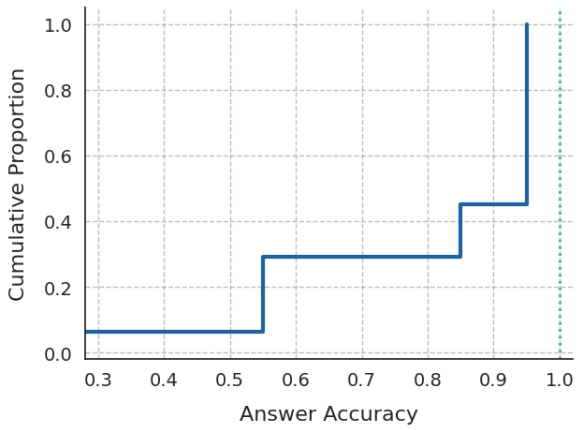

Figure 4: Cumulative Distribution Function (CDF) illustrating the ceiling effect and step-wise performance degradation.

4.2 Impact of Query Complexity

To disentangle the relationship between input complexity and performance, we analyze the variance of accuracy across different question lengths. Figure 5 stratifies the samples by token quartiles.

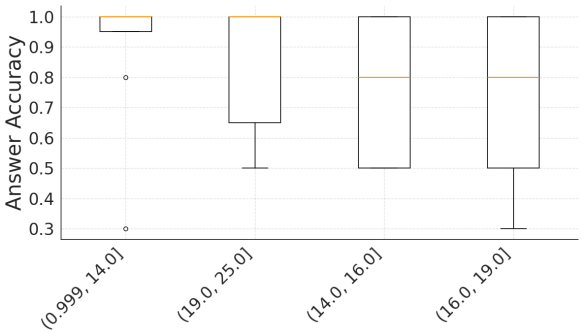

Figure 5: Accuracy variability by question length. Note the minimal variance in the first quartile compared to the dispersion in intermediate ranges.

The boxplot reveals a clear distinction in stability. The first quartile (shortest questions) presents a collapsed Interquartile Range (IQR) at the maximum score, implying a near-deterministic behavior for concise queries. These typically map directly to specific graph entities (e.g., "fine values"), facilitating precise anchor identification. In contrast, the

intermediate quartiles exhibit expanded IQRs and lower medians. This suggests that as queries become moderately complex, the semantic ambiguity increases, leading to higher variability in retrieval success.

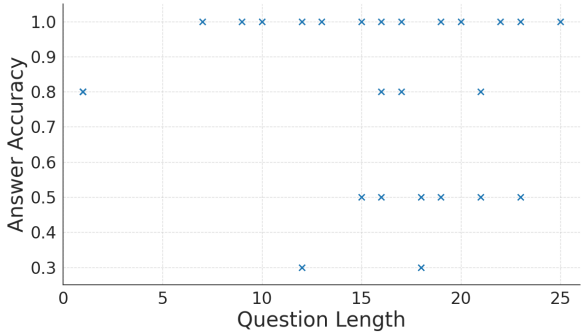

Figure 6: Scatter plot of Question Length vs. Accuracy. The distribution highlights a cluster of retrieval failures in the intermediate length range, contrasting with the ceiling effect observed at the extremes.

Finally, Figure 6 provides a granular view of this phenomenon. While the descriptive statistics suggested a simple downward trend, the scatter plot unveils a "valley of variance" in the mid-length region (approx. 15–20 tokens). In this region, the density of sub-optimal points is highest. Interestingly, the longest queries (> 20 tokens) show a tendency to recover high scores, supporting the hypothesis that while intermediate queries suffer from ambiguity, sufficiently long queries provide enough semantic context to re-align the vector retrieval, mitigating the dispersion observed in the middle range.

4.3 Discussion

The empirical results provide evidence supporting the core hypothesis of this study: that explicitly modeling the hierarchical structure of legal documents as a Knowledge Graph significantly mitigates context fragmentation. The ceiling effect observed in the median (1.00) and the negative skewness of the distribution suggest that the graph traversal mechanism acts as a "correctness guarantor." Once the retriever successfully anchors onto a relevant node (e.g., an Article), the deterministic edges allow for the reconstruction of the full normative context (Paragraphs and Items), minimizing the partial hallucinations common in flat-retrieval RAG systems.

The observed "U-shaped" performance curve indicates two distinct regimes of success regarding query complexity. The first regime, governed by short queries, relies on high lexical specificity, effectively functioning as an entity lookup operation where the embedding points directly to a graph node. The second regime, governed by long queries, relies on semantic density, where the abundance of context guides the vector search to the correct region of the latent space.

Conversely, the performance degradation in the intermediate range ($14 < t \leq 19$ tokens) exposes a "semantic ambiguity valley." In this zone, queries are often too complex for direct entity matching but lack sufficient semantic redundancy to overcome noise in the vector space. This suggests that the dispersion observed in Figure 6 is not merely stochastic but structural: intermediate queries likely land in sparse regions of the embedding space, leading to the retrieval of topologically close but legally distinct nodes (e.g., retrieving a general rule instead of its specific exception).

5 Conclusion

This study presented a Graph Retrieval-Augmented Generation (GraphRAG) pipeline tailored for the Portuguese legal domain, specifically addressing the structural complexity of normative resolutions from CEMIG. By explicitly modeling documents as knowledge graphs—connecting Articles, Paragraphs, and Items through normative edges—the system successfully mitigated the context fragmentation often observed in traditional flat retrieval methods. The overall accuracy of 0.81, combined with a median of 1.00, confirms that the path-oriented retrieval mechanism is highly effective in reconstructing complete evidence chains, provided the entry nodes are correctly identified.

The identification of a "valley of ambiguity" for intermediate-length questions constitutes a critical finding with direct implications for system design. Since the graph architecture proved robust for both concise lookups and semantically rich contexts, the performance gap in the intermediate zone suggests that optimization efforts should shift from the retrieval structure to the input processing layer. From an engineering perspective, we recommend the implementation of a query rewriting module designed to decompose intermediate questions into concise entity lookups or expand them with contextual definitions. Furthermore, integrating hybrid retrieval strategies—combining dense vectors with sparse lexical matching—could provide the necessary constraints to stabilize performance in this

specific vulnerability zone.

Future work will focus on four main axes: (i) expanding the graph schema to include temporal edges, allowing the system to reason about validity periods and revocations; (ii) refining the retrieval pipeline with dynamic top_k selection based on query uncertainty; (iii) evaluating the system on larger, multi-jurisdictional corpora to assess generalization; and (iv) incorporating controlled comparisons against conventional flat chunk-based RAG baselines and hybrid retrieval approaches in order to better quantify the specific contribution of graph-based retrieval. Ultimately, this research demonstrates that structurally aware retrieval is a prerequisite for reliable legal AI, transforming unstructured regulatory texts into auditable, high-precision decision support tools.

Limitations

This study presents certain limitations. First, the evaluation was conducted on a specific corpus of normative resolutions from a single Brazilian utility company (CEMIG). While the structural extraction relies on standard legal drafting patterns, the generalization to open-domain legal texts or different jurisdictions requires further validation. Second, the reliance on a specific embedding model and LLM (Gemini and GPT-4) may influence the performance metrics, and replacing these with open-source alternatives could yield different results. Finally, the current graph schema does not fully resolve temporal validity (e.g., automated inference of valid time-frames), which is critical for retrospective legal queries.

Acknowledgments

This work has been funded by P&D CEMIG/ANEEL PD-04950-D0677/2023. This work was supported by the National Institute of Science and Technology (INCT) in Responsible Artificial Intelligence for Computational Linguistics and Information Treatment and Dissemination (TILD-IAR) grant number 408490/2024-1.

References

J. Benita, Kosireddy Vivek Charan Tej, E. V. Kumar, G. V. Subbarao, and CH. Venkatesh. 2024. Implementation of retrieval-augmented generation (rag) in chatbot systems for enhanced real-time customer support in e-commerce. *2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS)*, pages 1381–1388.

Marco Carvalho, Fitzroy Nembhard, and Dhanish Mehta. 2025. Towards the application of graphrag to network security. *The International FLAIRS Conference Proceedings*.

Ilias Chalkidis, Abhik Jana, Dirk Dirschl, Rouvier Michel, Wilson Cameron, and Acs Gergely. 2021. Lexglue: A benchmark dataset for legal language understanding in english. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 4310–4330. Association for Computational Linguistics.

Boyu Chen, Zirui Guo, Zidan Yang, Yuluo Chen, Junze Chen, Zhenghao Liu, Chuan Shi, and Cheng Yang. 2025. Pathrag: Pruning graph-based retrieval augmented generation with relational paths. *arXiv preprint arXiv:2502.14902*.

Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *arXiv preprint arXiv:2306.16092*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 6491–6501, New York, NY, USA. Association for Computing Machinery.

Mahd Hindi, Linda Mohammed, Ommama Maaz, and Abdulmalik Alwarafy. 2025. Enhancing the precision and interpretability of retrieval-augmented generation (rag) in legal technology: A survey. *IEEE Access*, 13:46171–46189.

Hisatoshi Naganawa and Enna Hirata. 2025. Enhancing policy generation with graphrag and youtube data: A logistics case study. *Electronics*.

Elias Oliveira and Erick Sperandio Nascimento. 2021. Clustering by similarity of brazilian legal documents using natural language processing approaches. *Applied Sciences*, 11(21):10296.

Sarah Packowski, Inge Halilovic, Jenifer Schlotfeldt, and Trish Smith. 2024. Optimizing and evaluating enterprise retrieval-augmented generation (rag): A content design perspective. *Proceedings of the 2024 8th International Conference on Advances in Artificial Intelligence*.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*.

Brian E. Perron, B. Hiltz, Erin M. Khang, and Sue Ann Savas. 2025. Ai-enhanced social work: Developing and evaluating retrieval-augmented generation (rag) support systems. *Journal of Social Work Education*, 61:3 – 13.

Shoon Lei Phyu, Shuhayel Jaman, Murataly Uchkempirov, and Parag Kulkarni. 2024. Myanmar law cases and proceedings retrieval with graphrag. *2024 IEEE International Conference on Big Data (BigData)*, pages 2506–2513.

Bhaskarjit Sarmah, Dhagash Mehta, Benika Hall, Rohan Rao, Sunil Patel, and Stefano Pasquali. 2024. 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*, pages 608–616.

Simoes Stanley, P. Deepak, Munu Sairamesh, Deepak Khemani, and Sameep Mehta. 2018. Content and context: Two-pronged bootstrapped learning for regex-formatted entity extraction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Wenyu Tao, Xiaofen Xing, Yirong Chen, Linyi Huang, and Xiangmin Xu. 2025. Treerag: Unleashing the power of hierarchical storage for enhanced knowledge retrieval in long documents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 356–371.

Marc van Opijnen and Cristiana Santos. 2017. On the concept of relevance in legal information retrieval. *Artificial Intelligence and Law*, 25(1):65–87.