J-KGRAG: A Hybrid Retrieval-Augmented Generation Architecture for Legal Norm Understanding with Knowledge Graphs

Vinícius Teles Oliveira
Instituto de Informática
UFG
Goiânia, Brasil
viniciustelesdeoliveira@gmail.com

Maurício Rodrigues Lima
Instituto de Informática
UFG
Goiânia, Brasil
k20x@outlook.com

Sávio Teles
Instituto de Informática
UFG
Goiânia, Brasil
savioteles@ufg.br

Elisângela Silva Dias
Instituto de Informática
UFG
Goiânia, Brasil
elisangelasd@ufg.br

***Abstract*—This paper presents Juridic KGRAG (J-KGRAG), a hybrid architecture that enhances Retrieval-Augmented Generation (RAG) by integrating structured legal knowledge through a domain-specific knowledge graph. The system is designed to address the challenge of retrieving up-to-date legal information in highly interdependent normative documents, a frequent scenario in the Brazilian public sector. The method is applied to a corpus of 42 normative acts from Court of Accounts of the State of Goiás, Brazil, where legal articles are frequently updated, repealed, or referenced by newer documents. J-KGRAG enriches standard dense retrieval with a graph-based expansion step that identifies and retrieves updated entities omitted in the initial search. Experimental results indicate a significant improvement in factual accuracy (+75%) and overall answer correctness (+16%) compared to a naive RAG baseline. In addition, a manually curated benchmark of 53 legal question–answer pairs is released, and a qualitative analysis is performed to highlight the advantages of structured retrieval. The results demonstrate that combining symbolic legal representations with LLM-based generation improves both the consistency and the reliability of answers in legal domains.**

***Keywords*—Legal Question Answering; Knowledge Graphs; Retrieval-Augmented Generation (RAG).**

I. Introduction

The legal domain is characterized by vast repositories of complex documents and highly specialized language. In Brazil, as of 2023, there were approximately 84 million cases distributed across 91 courts, over 80% under the jurisdiction of State Courts, handled by 18,000 judges and 275,000 public servants, with 35 million new cases filed (↑ 9.5% vs. 2022) [1]. This overload makes manual processes for searching, summarizing, and answering legal queries unfeasible.

Natural Language Processing (NLP), and particularly Large Language Models (LLMs), have shown effectiveness in interpreting complex text and handling large-scale queries [2]–[4]. However, the continuous emergence and evolution of regulations renders frequent retraining of these models impractical, both due to computational cost and time.

The Retrieval-Augmented Generation (RAG) approach addresses this limitation by pairing a search engine with the LLM, retrieving relevant passages from a corpus without requiring retraining [5]. Still, traditional RAG fails to capture structural relationships between documents (e.g., citations and hierarchies), suffers from the "lost in the middle" effect in long contexts [6], and retrieves only textual subsets, lacking a global view of the normative structure.

This work proposes Juridic KGRAG (J-KGRAG), a hybrid architecture that integrates knowledge graphs into the RAG pipeline. Knowledge graphs represent information as structured entities and relations, enabling the explicit modeling of dependencies, revocations, and updates among legal norms, relations that are typically invisible to traditional RAG approaches. The objectives are to: represent the temporal and relational dynamics of legal norms in a graph; incorporate this graph into the RAG retrieval mechanism; evaluate how such integration affects the accuracy of responses to complex queries; and build a new question-answering dataset focused on legal regulations. The goal is to enhance the system's ability to handle normative evolution and significantly improve legal information retrieval.

II. Background

This section introduces the key concepts necessary to understand our method, describing the structure and interdependencies of legal norms, emphasizing the challenges they pose for information retrieval.

A. Legal Norms and Their Structure

Legal norms are official documents, such as laws, resolutions, and decrees, that define or modify rights and obligations within a legal system. In Brazilian public institutions like the Courts of Accounts, these documents are frequently revised or revoked by newer ones, creating a dense network of temporal and semantic dependencies. Table II-A, for instance, shows a regulation that explicitly amends the content of a prior norm.

Document	Provision	Content
Resolution N. 10/2019	Art. 3	Public tenders must be published at least 15 business days in advance.
Resolution N. 25/2023	Art. 1	Article 3 of Resolution No. 10/2019 shall henceforth read: "Public tenders must be published at least 10 business days in advance".

Table I
EXAMPLE OF NORMATIVE RELATION: ARTICLE MODIFIED BY A SUBSEQUENT REGULATION

Such changes are obvious to legal experts but often opaque to NLP systems relying purely on textual similarity. Moreover, legal texts tend to be lengthy and written in formal language, requiring temporal reasoning and explicit context tracking, tasks that traditional retrieval and generation models struggle with.

B. Knowledge Graphs and Retrieval-Augmented Generation

Knowledge graphs (KGs) represent information as triples (h, r, t), where entities are connected through semantic relations. In the legal domain, KGs can model revocations, amendments, and hierarchical references among norms, providing a structured representation of the law's evolution over time. These graphs can be constructed from unstructured text using named entity recognition and relation extraction methods, often supported by large language models (LLMs).

Recent advances in Retrieval-Augmented Generation (RAG) have shown that combining LLMs with external sources of information improves factual accuracy and contextual relevance [5]. When these sources include structured knowledge, such as legal KGs, the retrieval component can be guided not only by text similarity but also by normative logic and document dependencies. Prior work has demonstrated the benefits of this hybrid approach in question answering and fact verification [7].

This work embeds a legal knowledge graph into the retrieval phase of a RAG pipeline, capturing dependencies between norms and enabling inference over updates and revocations.

III. Related Work

Recent studies have explored the integration of knowledge graphs (KGs) into Retrieval-Augmented Generation (RAG) pipelines to enhance question answering (QA) systems. This section reviews the literature on combining structured information from KGs with the contextual retrieval capabilities of RAG models, aiming to improve the factual accuracy and interpretability of generated answers.

A. Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) integrates external knowledge sources with large language models (LLMs) to improve factuality and domain specificity in text generation. Recent surveys provide comprehensive overviews of RAG approaches across retrieval, augmentation, and generation dimensions [8]–[12]. For instance, Fan et al. [8] and Gao et al. [9] categorize RAG methods based on architectural components, while Zhao et al. [12] examine multimodal retrieval settings, and Yu et al. [13] propose evaluation strategies for RAG-based systems.

While some prior work has explored integrating RAG with knowledge graphs, most approaches focus on unstructured or textual data. In contrast, our work emphasizes indexing, retrieval, and generation grounded in structured graph representations, offering a distinct path from purely text-based augmentation.

B. Knowledge Graphs in NLP Pipelines

Incorporating knowledge graphs (KGs) into NLP pipelines involves three key stages: indexing, retrieval, and generation. Various indexing strategies have been proposed, including structural (graph-native), textual, and vector-based methods. Graph-based indexing preserves full topological structure for traversal using algorithms like BFS or shortest paths [14], [15]. Text-based indexing converts triples into natural language using templates or LLMs [16], [17], while vector-based indexing employs graph embeddings or neural encoders to facilitate efficient similarity search [7], [18].

Retrieval approaches can be non-parametric, model-based, or GNN-based. Non-parametric methods rely on graph algorithms and entity linking to extract subgraphs [19], [20], while LLMs such as StructGPT [21] use prompt-based interfaces to iteratively gather relevant paths.

In the generation stage, both GNNs and LLMs are used to synthesize answers. GNNs encode graph structure and pass node embeddings to MLPs or decoders [22], while encoder-decoder models like GPT-4 [2] generate answers based on transformed graph inputs. Hybrid approaches concatenate GNN outputs as prompts or prefix embeddings to LLMs [7], [23], leveraging both structural awareness and linguistic fluency.

While many works have explored these components independently or in combinations, few focus on structured retrieval pipelines grounded in normative legal documents, which is the focus of this work.

C. Knowledge Graphs in Legal Applications

Knowledge graphs have been increasingly applied to legal question answering (QA) tasks due to their ability to enhance interpretability and structured reasoning [24]. Prior work has explored enriching legal knowledge bases with ontologies and semantic structures to support retrieval and QA [25], [26]. For instance, systems like AILA [27] incorporate domain-specific graphs to improve legal query understanding and ranking of candidate answers.

Efforts have also emerged to construct national or transnational legal KGs, such as the transformation of Austrian legislation into a graph structure to support European legal harmonization [28]. Other systems target specific domains, like legal auditing or case law analysis [29], using tools like Neo4J and Cypher to support structured legal queries.

However, most existing approaches focus either on static QA datasets, ontology modeling, or legal search tools. Few works propose dynamic integration of normative relations, such as revocations and updates, into retrieval pipelines. Our work addresses this gap by embedding legal norms into a KG structure and using it directly within a retrieval-augmented generation framework.

In contrast to prior studies that primarily rely on unstructured legal texts or static question–answer pairs, this study introduces a structured and dynamic knowledge graph tailored to the evolution of legal norms, capturing temporal relations such as updates, revocations, and dependencies between regulations. By integrating this graph directly into a RAG pipeline, the approach aims to improve the factual consistency and domain relevance of generated answers in legal QA tasks. This is the first reported approach that combines retrieval-augmented generation with a graph of normative legal relationships for question answering in the context of Brazilian public law.

IV. Proposed Method

This section presents J-KGRAG, a hybrid retrieval and generation architecture designed to enhance question answering over legal documents by integrating a knowledge graph into the Retrieval-Augmented Generation (RAG) pipeline.

Legal norms often contain temporal and logical dependencies, such as amendments, revocations, or updates across different documents.Standard RAG pipelines, which rely solely on vector-based retrieval, may fail to capture these relations, returning outdated or incomplete answers.

To address this issue, J-KGRAG introduces a knowledge graph that models explicit normative relationships such as amendments and revocations. Figure 1 shows an example from Table II-A where one resolution modifies a specific article from a previous regulation, illustrating how such relations are encoded in the graph.

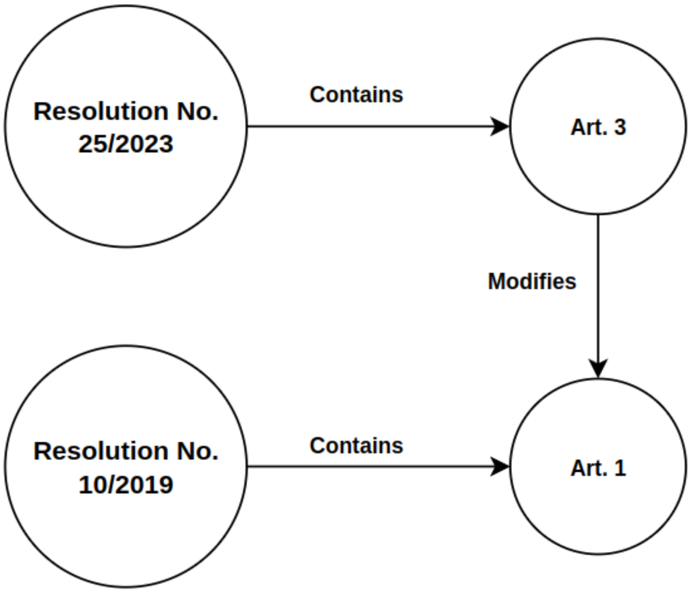

Figure 1. Example of a knowledge graph modeling a legal modification: *Resolution No. 25/2023* modifies *Article 1* of *Resolution No. 10/2019* via its own *Article 3*.

The method combines dense retrieval, entity mapping, and graph-based expansion to enrich the context fed into the language model. This section details the architecture, graph construction, retrieval mechanism, and inference process used in our implementation.

A. System Architecture

The architecture of J-KGRAG is composed of two main stages: indexing and inference. In the indexing stage, both a vector database and a knowledge graph are built from the same corpus of legal documents. For the vector database, documents are split into chunks, each of which is embedded using a pre-trained language model and stored for later retrieval. In parallel, a legal-specific knowledge graph is constructed by defining two types of entities: documents and articles. Each document is summarized by an LLM to form a document node with a descriptive field. Chunks are then processed to extract article entities using LLM-based entity recognition. When duplicate articles are detected, based on name similarity, their descriptions are merged and summarized into a single representation to avoid redundancy.

Once the entities are defined, a second pass over the chunks is performed to extract relational triples using LLMs. Only three types of relations are allowed: *contains*, linking a document to the articles it includes; *amends*, when one article modifies another; and *revokes*, when an article invalidates a previous one. These relationships form a structured knowledge graph capturing the temporal and hierarchical dependencies present in the legal domain.

During inference, a user query is first passed through a dense retriever that selects relevant text chunks from the vector store. From the selected chunks, the system identifies linked entities in the knowledge graph. These entities serve as anchors for a second and critical retrieval step, which is the core of our method. In this stage, the system explores neighboring entities that are directly connected (i.e., first-order neighbors) to the ones initially retrieved. This graph-based expansion enables the discovery of updated or complementary information that was not captured in the initial dense retrieval. Finally, the user question and the descriptions of all retrieved entities are composed into a prompt that is sent to a large language model. This enriched context increases the likelihood that the model will generate accurate and up-to-date answers, especially in cases where legal norms have been modified or revoked.

V. Results

A. Experimental Setup

This section describes the experimental design used to evaluate the proposed architecture, including dataset preparation, compared baselines, evaluation metrics, and implementation details.

Dataset. The dataset consists of normative documents issued by the Court of Accounts of the State of Goiás (TCE-GO). A total of 42 documents were selected, including 32 published in 2024 and 10 from 2023. Notably, all ten documents from 2023 were amended, revoked, or corrected by the 2024 documents, forming a network of interdependencies. This characteristic was central to the problem addressed in this work, where tracking up-to-date legal information is essential. To facilitate entity extraction, a preprocessing step was performed to disambiguate abbreviated references commonly found in legal texts. For example, expressions like “Art. 2º” were automatically converted to “Article 2 of DOCUMENT XX/YYYY” using the GPT-4o-mini model, ensuring that each reference could be correctly linked to its respective document. In addition to the primary dataset, an evaluation set was constructed to assess the effectiveness of the system. Initially, five questions and five answers were generated for each of the 42 documents using GPT-4o, resulting in 210 question-answer (QA) pairs. After manual filtering and refinement, only 53 QA pairs remained, representing well-formed, relevant, and high-quality items used as ground truth for evaluation.

Compared Models. Three models were evaluated in this study. The first, *Naive RAG*, is a standard retrieval-augmented generation pipeline that performs dense vector search over document chunks, without incorporating any knowledge graph. The second, *KGRAG*, extends the *Naive RAG* approach by integrating a knowledge graph into the retrieval process. During inference, it combines both the retrieved chunks and the entities and relations obtained from the graph to enrich the context. The third model, *KGRAG without chunks*, is a variant of *KGRAG* in which only the structured information, namely, entities and relations extracted from the knowledge graph, is passed to the language model. In this configuration, chunks are excluded from the final prompt, allowing the analysis to isolate the specific contribution of the graph-based context.

Evaluation Metrics. To measure answer quality, the RAGAS framework [30] was employed, leveraging LLM-based judgments to assess factual accuracy and semantic similarity. Three key metrics were adopted. FactualCorrectness assesses whether the generated answer is factually consistent with the ground truth by using claim decomposition and natural language inference, with scores ranging from 0 to 1. SemanticSimilarity measures the semantic proximity between the generated and reference answers using a cross-encoder to compute cosine similarity. Finally, AnswerCorrectness is a composite metric that combines FactualCorrectness and SemanticSimilarity to provide a holistic view of the answer’s quality.

Evaluation Environment. All components of the architecture were implemented using LangChain. GPT-4o-mini was employed for all tasks: answer generation, entity/triple extraction, and automatic evaluation. Documents were segmented into chunks of 600 tokens with an overlap of 200 tokens. For each query, the top-6 chunks were retrieved, followed by the top-10 related entities and top-10 graph relations. The system was evaluated entirely via automated LLM-based scoring using RAGAS.

B. Quantitative Evaluation

Table II summarizes the results obtained by the three evaluated models, *Naive RAG*, *J-KGRAG*, and *J-KGRAG without chunks*, across three RAGAS metrics: *Semantic Similarity*, *Answer Correctness*, and *Factual Correctness*. The results highlight the benefits of integrating a knowledge graph into the RAG pipeline, especially when used as a standalone context source, without relying on retrieved document chunks.

Model	Semantic Similarity	Answer Correctness	Factual Correctness
J-KGRAG No Chunks	**0.918**	**0.497**	**0.240**
J-KGRAG	0.916	0.435	0.202
Naive RAG	0.912	0.430	0.137

Table II
QUANTITATIVE EVALUATION METRICS FOR THE THREE COMPARED MODELS.

Model	Semantic Similarity	Answer Correctness	Factual Correctness
dataset-crewai-k20	**0.963**	**0.699**	**0.517**
dataset-crewai-k10	0.962	0.670	0.513
Naive RAG	0.912	0.430	0.137

Table III
QUANTITATIVE EVALUATION METRICS FOR THE CREWAI

J-KGRAG without chunks achieved the highest scores in all three metrics. It reached a *Semantic Similarity* of 0.918, slightly outperforming *J-KGRAG* (0.916) and *Naive RAG (0.912)*. The largest gain, however, was observed in *Factual Correctness*, where *J-KGRAG without chunks* scored 0.240, significantly higher than *J-KGRAG* (0.202) and *Naive RAG* (0.137). This indicates that the knowledge graph was particularly effective in providing up-to-date and accurate information that would otherwise be missing in traditional retrieval-based responses.

Interestingly, removing the document chunks from the inference prompt resulted in improved performance across all metrics. This suggests that excessive or redundant context may overwhelm the language model, and that targeted, structured information from the knowledge graph offers a more efficient and precise source of truth for answering legal queries.

These findings support the hypothesis that a graph-based expansion mechanism can significantly enhance the reasoning capabilities and factual alignment of RAG-based systems, especially in domains with highly interdependent and evolving documents such as law.

C. Discussion

The quantitative results from the evaluation metrics highlight the superiority of the *KGRAG model without chunks* in terms of factual accuracy and overall correctness. To complement this analysis, this section presents a qualitative comparison of answers generated for four selected questions from the evaluation set. These questions were chosen due to the notable differences in model performance, providing illustrative cases where the chunk-free *KGRAG* model demonstrated clear advantages or, conversely, revealed specific limitations.

In Table IV, Question 8 asks about the supervisor responsible for an audit. While the *Naive RAG* model incorrectly identifies Gustavo Henrique de Faria as the supervisor, confusing him with the coordinator, the *KGRAG* models correctly retrieve the updated entity, André Pinheiro de Magalhães, based on Article 3 of Ordinance No. 6/2024. This article was not retrieved through dense vector search but was accessible through graph-based entity expansion. The graph's ability to link normative updates proved essential to correcting outdated information and providing accurate answers, particularly in the chunk-free version, which maintained clarity and avoided redundancy.

	Content
Question 8	Who is the supervisor of the team responsible for conducting the Financial Audit of the State Secretariat for Social Development?
Expected Answer	The supervisor of the audit team is André Pinheiro Magalhães.
Naive RAG	The supervisor of the team responsible for conducting the Financial Audit of the State Secretariat for Social Development is Gustavo Henrique de Faria.
KGRAG	The supervisor of the team responsible for conducting the Financial Audit of the State Secretariat for Social Development is André Pinheiro de Magalhães. This information is based on the description of Ordinance No. 6/2024, which appoints the new audit team and mentions André Pinheiro de Magalhães as the supervisor.
KGRAG without chunks	The supervisor of the team responsible for conducting the Financial Audit of the State Secretariat for Social Development (SEDS) is André Pinheiro de Magalhães.

Table IV
COMPARISON OF MODEL-GENERATED ANSWERS TO THE QUESTION ABOUT THE AUDIT TEAM SUPERVISOR.

Table V presents Question 12, which seeks the objective of a specific inspection task. The *Naive RAG* model returns a vague answer, citing only that an inspection would take place. In contrast, both *KGRAG* variants retrieve a more complete summary of the document, including details about the scope of the inspection, the contract involved, and the deadline for the final report. These results demonstrate how knowledge graph entities, specifically, document-level summaries, enrich the prompt with broader context, leading to more comprehensive and aligned answers.

Otherwise, the Table VI focuses on Question 11, which requires identifying the supervisor for a particular contract. While both the *Naive RAG* and *KGRAG* with chunks incorrectly assign the role to Raul Suzuki Pinto Rabelo, the KGRAG without chunks correctly identifies Ricardo Souza Lobo, as stated in Article 3 of Ordinance No. 15/2024. This distinction highlights a key benefit of the chunk-free approach: by relying solely on entities and relations, it reduces noise and focuses on structured information, thus avoiding hallucinations introduced by irrelevant or overlapping chunk content.

	Content
Question 12	What is the objective of the audit team designated by Resolution No. 8/2024 of *Court of Accounts of the State of Goiás, Brazil*?
Expected Answer	The objective of the audit team is to carry out an inspection, together with *Goiás State Agency for Infrastructure and Transport – GOINFRA*, to assess the quality and quantity of maintenance services performed on a sample to be defined during the planning phase of the inspection, concerning the state road network, both paved and unpaved, including ferries, runways, and fences of the aerodromes that are part of Lot 04 – Contract No. 14/2023 – *GOINFRA*.
Naive RAG	The objective of the audit team designated by Resolution No. 8/2024 of *Court of Accounts of the State of Goiás, Brazil* is to carry out an inspection with *Goiás State Agency for Infrastructure and Transport – GOINFRA*.
KGRAG	The objective of the audit team designated by Resolution No. 8/2024 of *Court of Accounts of the State of Goiás, Brazil* is to carry out an inspection with *GOINFRA* to evaluate maintenance services on the state road network. The final report must be submitted by 05/31/2024.
KGRAG without chunks	The objective of the audit team designated by Resolution No. 8/2024 of *Court of Accounts of the State of Goiás, Brazil* is to evaluate maintenance services on the state road network of *GOINFRA*. The final report must be submitted by 05/31/2024.

Table V
COMPARISON OF ANSWERS GENERATED BY THE MODELS FOR THE QUESTION REGARDING THE PURPOSE OF THE AUDIT MENTIONED IN RESOLUTION NO. 8/2024.

	Content
Question 11	Who will supervise the audit of Lot 15 from Contract No. 026/2023?
Expected Answer	The audit will be supervised by civil servant Ricardo Souza Lobo, with assistance from Iasmin David Guimarães.
Naive RAG	The audit of Lot 15 from Contract No. 026/2023 will be supervised by civil servant Raul Suzuki Pinto Rabelo, under the coordination of Celso Hiroki Sakuma.
KGRAG	The audit of Lot 15 from Contract No. 026/2023 will be supervised by civil servant Raul Suzuki Pinto Rabelo, who is assigned to the audit team under the coordination of Celso Hiroki Sakuma, as indicated in the texts and the list of entities.
KGRAG without chunks	The audit supervision of Lot 15 from Contract No. 026/2023 will be carried out by Ricardo Souza Lobo, as established in Article 3 of Resolution 15/2024.

Table VI
COMPARISON OF ANSWERS GENERATED BY THE MODELS FOR THE QUESTION ABOUT THE SUPERVISOR OF THE AUDIT FOR LOT 15 FROM CONTRACT NO. 026/2023.

Finally, Table VII presents Question 46, a case in which the chunk-free *KGRAG* model underperforms. The question asks for a list of holidays and optional leave days in May 2024. While both the *Naive RAG* and *KGRAG* provide detailed answers listing the relevant dates and events, even if with minor inconsistencies, the *KGRAG without chunks* fails to deliver a complete response, citing insufficient data. This limitation stems from the fact that the required list was not explicitly encoded in the graph structure as entities or relations, exposing a scenario where chunk-based retrieval has a clear advantage.

	Content
Question 46	How many holidays or optional workdays will there be in May 2024? What are they?
Expected Answer	There will be 3 holidays and 1 optional workday in May: Labor Day, Our Lady's Day, Corpus Christi, and one optional workday.
Naive RAG	In May 2024, there will be three holidays or optional workdays: 01/05/2024 - Wednesday - Labor Day; 24/05/2024 - Friday - Our Lady Help of Christians Day (Optional workday); 31/05/2024 - Friday - Optional workday. Therefore, the answer is that there will be three holidays or optional workdays in May 2024.
KGRAG	In May 2024, there will be 3 holidays or optional workdays: 01/05/2024 (Wednesday) - Labor Day; 24/05/2024 (Friday) - Our Lady Help of Christians Day (Optional workday); 31/05/2024 (Friday) - Optional workday. Therefore, the answer is that there will be 3 holidays or optional workdays in May 2024.
KGRAG without chunks	According to Resolution No. 32/2024, which discloses the calendar of holidays and optional workdays for 2024, the specific information about holidays or optional workdays in May is not included in the provided lists. Therefore, it is not possible to determine how many holidays or optional workdays there will be in May 2024 or what they are based on the available data.

Table VII
COMPARISON OF MODEL-GENERATED ANSWERS TO THE QUESTION ABOUT HOLIDAYS/OPTIONAL WORKDAYS IN MAY 2024.

These case studies illustrate that while the *KGRAG without chunks* architecture excels in accuracy and precision, especially in handling normative updates, it may lack the granularity needed for detailed list-based questions. This suggests that an ideal system may require adaptive strategies that combine structured knowledge with selectively retrieved textual chunks, depending on the nature and complexity of the user query.

VI. CONCLUSION

This work introduced *J-KGRAG*, a hybrid architecture that integrates Retrieval-Augmented Generation (RAG) with a specialized knowledge graph to mitigate hallucinations in large language models within the Portuguese legal domain. Quantitative results showed improvements of *+75%* in *Factual Correctness* and *+16%* in *Answer Correctness* compared to Naive RAG, demonstrating that explicitly modeling relationships such as amendments and revocations enables more accurate retrieval of current legal norms. The qualitative analysis reinforced that entity-based expansion effectively bridges gaps typically found in dense retrieval, such as the "*lost in the middle*" issue [6], reducing contradictory outputs and hallucinations without compromising semantic similarity.

Beyond its methodological contribution, this work implemented an indexing pipeline that normalizes and structures the Court of Accounts of the State of Goiás, Brazil, regulations into entity, triple, and chunks; released an open benchmark with 53 carefully reviewed question–answer pairs; and conducted a comparative study that highlights the role of contextual granularity, chunks versus entities, in final answer generation.

Nonetheless, there are limitations. The corpus is still narrow (42 regulations), the knowledge graph assumes instantaneous updates (without retroactive validity), and the entire evaluation process relies on LLM-based heuristics (RAGAS), lacking human validation from legal professionals. From a systems perspective, indexing is performed in batch mode; in real-world deployments, documents would arrive continuously and require incremental updates. Moreover, retrieval currently uses simple breadth-first algorithms, without the integration of GNNs or supervised re-rankers.

Future work includes developing streaming ingestion pipelines for near real-time updates of entities and relations, as well as exploring temporal knowledge graphs. Graph-based models such as GraphSAGE or Graph Transformers could be used to score contextual subgraphs and combine those scores with dense retrievers [7], [20]. Another key direction is incorporating human-in-the-loop evaluation through blind testing with prosecutors and auditors to assess practical utility and trust. Additional enhancements may include techniques like *self-consistency* or *chain-of-verification* for greater explainability. To evaluate robustness, the method should be replicated across other courts (e.g., TJ-SP, STF) and in legal systems in English or Spanish, in order to examine the impact of terminological and structural variations. Incorporating non-textual annexes such as scanned tables or documents, via vision–language models, with metadata stored in the graph would also enable queries involving maps or budget spreadsheets. Finally, it will be important to measure the energy consumption of the full pipeline [31] and explore compact vector-database indexing and parameter pruning techniques to reduce computational costs without sacrificing accuracy.

In summary, KGRAG demonstrates that combining explicit semantic representations with LLMs increases the reliability of AI-based legal systems. By releasing code, data, and evaluation protocols, this work aims to catalyze further research into legal assistants capable of keeping up with Brazil's rapidly evolving regulations, with transparency, traceability, and technical rigor.

References

[1] S. T. Federal, "Justiça em números: presidente do stf divulga dados do judiciário brasileiro," 2024, accessed: 2025-02-26. [Online]. Available: https://portal.stf.jus.br/noticias/verNoticiaDetalhe.asp?idConteudo=542620

[2] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[3] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[4] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[6] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 157–173, 2024.

[7] X. He, Y. Tian, Y. Sun, N. V. Chawla, T. Laurent, Y. LeCun, X. Bresson, and B. Hooi, "G-retriever: Retrieval-augmented generation for textual graph understanding and question answering," *arXiv preprint arXiv:2402.07630*, 2024.

[8] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua, and Q. Li, "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*, 2024, pp. 6491–6501.

[9] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and H. Wang, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, 2023.

[10] Y. Huang and J. Huang, "A survey on retrieval-augmented text generation for large language models," *arXiv preprint arXiv:2404.10981*, 2024.

[11] S. Wu, Y. Xiong, Y. Cui, H. Wu, C. Chen, Y. Yuan, L. Huang, X. Liu, T.-W. Kuo, N. Guan *et al.*, "Retrieval-augmented generation for natural language processing: A survey," *arXiv preprint arXiv:2407.13193*, 2024.

[12] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang, and B. Cui, "Retrieval-augmented generation for ai-generated content: A survey," *arXiv preprint arXiv:2402.19473*, 2024.

[13] H. Yu, A. Gan, K. Zhang, S. Tong, Q. Liu, and Z. Liu, "Evaluation of retrieval-augmented generation: A survey," *arXiv preprint arXiv:2405.07437*, 2024.

[14] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec, "Qa-gnn: Reasoning with language models and knowledge graphs for question answering," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 535–546.

[15] D. Taunk, L. Khanna, S. V. P. K. Kandru, V. Varma, C. Sharma, and M. Tapaswi, "Grapeqa: Graph augmentation and pruning to enhance question-answering," in *Companion Proceedings of the ACM Web Conference 2023*, 2023, pp. 1138–1144.

[16] S. Li, Y. Gao, H. Jiang, Q. Yin, Z. Li, X. Yan, C. Zhang, and B. Yin, "Graph reasoning for question answering with triplet retrieval," *arXiv preprint arXiv:2305.18742*, 2023.

[17] Y. Huang, Y. Li, Y. Xu, L. Zhang, R. Gan, J. Zhang, and L. Wang, "Mvp-tuning: Multi-view knowledge retrieval with prompt tuning for commonsense reasoning," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 13 417–13 432.

[18] Y. Hu, Z. Lei, Z. Zhang, B. Pan, C. Ling, and L. Zhao, "Grag: Graph retrieval-augmented generation," *arXiv preprint arXiv:2405.16506*, 2024.

[19] J. Delile, S. Mukherjee, A. Van Pamel, and L. Zhukov, "Graph-based retriever captures the long tail of biomedical knowledge," *arXiv preprint arXiv:2402.12352*, 2024.

[20] C. Mavromatis and G. Karypis, "Gnn-rag: Graph neural retrieval for large language model reasoning," *arXiv preprint arXiv:2405.20139*, 2024.

[21] J. Jiang, K. Zhou, Z. Dong, K. Ye, W. X. Zhao, and J.-R. Wen, “Structgpt: A general framework for large language model to reason over structured data,” *arXiv preprint arXiv:2305.09645*, 2023.

[22] J. Dong, Q. Zhang, X. Huang, K. Duan, Q. Tan, and Z. Jiang, “Hierarchy-aware multi-hop question answering over knowledge graphs,” in *Proceedings of the ACM Web Conference 2023*, ser. WWW ’23. New York, NY, USA: Association for Computing Machinery, 2023, p. 2519–2527. [Online]. Available: https://doi.org/10.1145/3543507.3583376

[23] M. Zhang, M. Sun, P. Wang, S. Fan, Y. Mo, X. Xu, H. Liu, C. Yang, and C. Shi, “Graphtranslator: Aligning graph model to large language model for open-ended tasks,” *Proceedings of the ACM on Web Conference 2024*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:267627926

[24] J. Martinez-Gil, “A survey on legal question–answering systems,” *Computer Science Review*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:238856773

[25] F. Dai, Z. Zhao, C. Sun, and B. Li, “Intelligent audit question answering system based on knowledge graph and semantic similarity,” *2022 11th International Conference of Information and Communication Technology (ICTech))*, pp. 125–132, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251762977

[26] F. Sovrano, M. Palmirani, and F. Vitali, “Legal knowledge extraction for knowledge graph based question-answering,” in *International Conference on Legal Knowledge and Information Systems*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:229377581

[27] W. Huang, J. Jiang, Q. Qu, and M. Yang, “Aila: A question answering system in the legal domain,” in *International Joint Conference on Artificial Intelligence*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:220480990

[28] E. Filtz, S. Kirrane, and A. Polleres, “The linked legal data landscape: linking legal data across different countries,” *Artificial Intelligence and Law*, vol. 29, pp. 485 – 539, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:233293743

[29] A. Thomas and S. Sangeetha, “Knowledge graph based question-answering system for effective case law analysis,” in *International Conference on Frontiers in Intelligent Computing: Theory and Applications*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:265467619

[30] S. Es, J. James, L. Espinosa Anke, and S. Schockaert, “RAGAS: Automated evaluation of retrieval augmented generation,” in *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, N. Aletras and O. De Clercq, Eds. St. Julians, Malta: Association for Computational Linguistics, Mar. 2024, pp. 150–158. [Online]. Available: https://aclanthology.org/2024.eacl-demo.16

[31] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean, “Carbon emissions and large neural network training,” *arXiv preprint arXiv:2104.10350*, 2021.