# LLM-Augmented Knowledge Graphs for Senegalese Legal RAG Systems

**Anonymous ACL submission**

## Abstract

This study improves access to Senegalese legal texts by integrating a Knowledge Graph into a Neo4j database within a Retrieval-Augmented Generation (RAG) framework. A methodology inspired by Kelsen's pyramid was developed to enrich the dataset. The augmented graph database provides significant legal cross-references for practitioners. Evaluation of the RAG system across various Large Language Models (LLMs) indicates that the latest OpenAI models produce the most relevant legal responses. Although there are some limitations, future work will aim to expand the knowledge base, refine question formulation, enhance system efficiency, and reduce hallucinations using hybrid approaches and collaborative frameworks. This research establishes foundational work for the representation and reasoning of dynamic and precise legal knowledge.

## 1 Introduction

The structuration of Senegalese legal texts is complex and multifaceted. Senegal's legal framework, influenced by French law, struggles with codification issues like rigidity and ineffectiveness (Kah, 1996). Enhancing legal pluralism requires practical applications and active citizen involvement. A thorough understanding of Senegal's legal system is crucial for its alignment with international law, aiding contributions to international criminal law (Niang, 2009). Additionally, land and labor issues highlight challenges in legal enforcement and clarity, intensifying land disputes due to legal ambiguities and statute manipulation, alongside many unregistered businesses (Lavigne Delville and Diagne, 2023).

To mitigate these challenges, the integration of Knowledge Graphs (KGs) with Large Language Models (LLMs) is advocated. This integration represents a transformative approach for efficient information processing within the legal sector. For instance, merging KGs with LLMs enhances reasoning by providing structured information, thereby minimizing hallucinations and improving interpretability in intricate legal scenarios (Wu and Tsioutsiouliklis, 2024). Furthermore, KGs in legislative contexts elucidate the intricate relationships among laws and articles, bolstering legislative processes, while their integration with LLMs promotes automated law article recommendation systems (Colombo, 2024; Chen et al., 2025). Nonetheless, an ongoing challenge persists in the timely updating of KGs to maintain their relevance and efficacy (Li et al., 2024).

Our research builds upon the framework established in Senegalese Legal Structuration Using LLM-Augmented KG (Kane et al., 2025), utilizing LLMs to derive knowledge triples from organized legal texts. Two primary aims are pursued in this study. First, we aimed to augment the existing Neo4j graph database related to land and public domain codes (Kane et al., 2025) by incorporating additional legal articles sourced from various legal frameworks. Second, employing the LLM-Augmented Knowledge Graphs methodology from prior work (Kane et al., 2025), we enhanced graph representations by generating new insights through the completion of cross-references within the KG (Ibrahim et al., 2024). Our secondary objective is to establish a Knowledge Graph-enhanced LLM framework, termed KG-enhanced LLM, through the development of a Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) system that extracts organized data from the Knowledge Graph and integrates it into LLM contexts to yield more accurate responses to user inquiries during inference (Ibrahim et al., 2024). Evaluative measures are conducted on validated ground truth data to assess the fulfillment of the outlined objectives.

Our essay is structured as follows: Initially, we examine the methodology, including dataset setup, graph database initiation, and techniques

for KG completion and KG-augmented LLM. Subsequently, we present an overview of experiments and results, comparing LLMs using various metrics. The concluding sections analyze findings, address limitations, and propose future research avenues.

## 2 Methodology

### 2.1 Dataset Configuration

To depict the Senegalese legal system globally, we utilize a dataset that represents Kelsen's pyramid hierarchy (see Figure 1), which clarifies the judicial system and normative hierarchy (Beyleveld, 1993). The hierarchy consists of four levels, with the Constitutional block at the apex, representing its paramount importance. The Senegalese constitution serves as a formal illustration of this block. The second level, the Conventionality block, encompasses legal rules originating from international conventions, exemplified by the Maputo Protocol, which aims to advance women's fundamental rights in Africa. The third level, the Legality block, consolidates organic laws, referendum laws, and ordinary laws, generally encompassing multiple laws from various legal codes. The integration of additional entities within the legal framework, including the Uniform Act on Accounting Law and Financial Information (AUDCIF), is feasible. At the lowest level is the Regulatory block, which comprises all edicts from the executive branch, including decrees and orders.

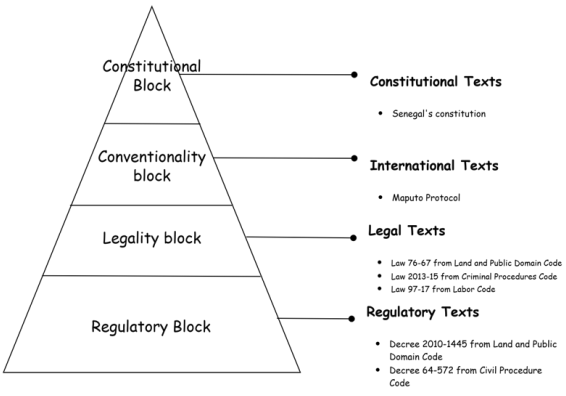


Figure 1: **Kelsen's pyramid**.

Subsequently, legal articles along with their pertinent metadata were proposed for extraction through the algorithm developed by Kane et al. (2025). A concise overview of the content encompassed in each legal document that is pertinent to Kelsen's pyramid is delineated in Table 1, which illustrates the quantity of articles extracted, as well as the statistical parameters of $\min$, $\max$, and upper fence regarding the number of paragraphs $\rho$ within the articles, in addition to those associated with the length of the paragraphs $\delta$ (measured in the number of characters). It is noteworthy that the lower fences are congruent with the $\min$ values.

| **Legal Documents** | **Number of Articles** | **N. of Paragraphs ($\rho$)** | | | **L. of Paragraphs ($\delta$)** | | |
|---|---|---|---|---|---|---|---|
| | | *Min* | *Max* | *Upper Fence* | *Min* | *Max* | *Upper Fence* |
| Land and Public Domain Code | 2040 | 1 | 186 | 8 | 1 | 1188 | 490 |
| Public Procurement Code | 154 | 1 | 59 | 12 | 16 | 1131 | 624 |
| Criminal Procedures Code | 780 | 1 | 18 | 6 | 1 | 1069 | 416 |
| Criminal Code | 557 | 1 | 37 | 8 | 4 | 888 | 482 |
| Local Authorities Code | 370 | 1 | 56 | 6 | 12 | 3172 | 460 |
| Family Code | 851 | 1 | 13 | 7 | 4 | 1259 | 444 |
| Civil Procedure Code | 920 | 1 | 22 | 6 | 1 | 1000 | 488 |
| Environment Code | 255 | 1 | 11 | 3 | 42 | 9006 | 494 |
| Labor Code | 289 | 1 | 28 | 8 | 1 | 897 | 509 |
| AUDCIF | 123 | 1 | 11 | 7 | 6 | 1435 | 582 |
| Senegalese Constitution | 108 | 1 | 17 | 8 | 17 | 617 | 349 |
| Maputo Protocol | 32 | 2 | 15 | 14 | 7 | 474 | 352 |

Table 1: Legal Documents with Corresponding Statistics

Afterward, we define the input text $X$ for processing in a graph database and for knowledge triple extraction by LLMs, as illustrated by Kane et al. (2025). $X$ comprises words from articles supplemented with contextual information for enhanced legal referencing. Specifically, $X$ is represented as $X = \{w_1, w_2, ..., w_n, c_1, c_2, ..., c_m\}$, where $w_i$ signifies the $i$th word from $n$ words in the article, and $c_j$ denotes the $j$th word from $m$ words in the supplementary context, which serves as the article's metadata.

We utilize chunks from legal documents to enhance retrieval efficiency in vector databases (Han et al., 2023). Our chunking strategy is calibrated such that each chunk does not exceed the prescribed limits of paragraph length (denoted as $\delta_f$) and paragraph count (denoted as $\rho_f$). We establish a joint probability for two events representing chunk behavior: $A = \{\rho \leq \rho_f\}$ and $B = \{\delta \leq \delta_f\}$, expressed as $P(A \cap B) = P(\rho \leq \rho_f, \delta \leq \delta_f)$. Under our assumptions, the independence of these events results in a straightforward multiplication of their probabilities:

$$P(A \cap B) = P(\rho \leq \rho_f) \times P(\delta \leq \delta_f)$$

Utilizing a joint probability framework allows for the modeling and regulation of chunking behavior based on the likelihood of conforming to defined upper limits on paragraph count and length, as established by the document's upper fences. A

chunk in a legal document is subsequently defined as:

$$\text{chunk} = \{w_{h_1}, \ldots, w_{h_K}, c_1, \ldots, c_m, c'_1, \ldots, c'_p\}$$

subject to the following conditions:

$$\begin{cases} h_1 < h_K, \\ w_{h_1} \in \{w_1, \alpha\}, \\ w_{h_K} \in \{\alpha, w_n\}, \\ h_K - h_1 < \delta_f \times \rho_f, \end{cases}$$

where $\alpha$ serves as a designated separator for text segmentation, $c'_1, \ldots, c'_p$ represent specific tokens that provide context to the chunk, and $h_K - h_1$ represents the length of the chunk.

## 2.2 Graph Initialization

We define our graph database as a labeled, direct property graph which can be formulated as a tuple: $G = (V, E)$, where $V$ is finite set of nodes (vertices), and $E \subseteq V \times V$ is a set of direct relationships (edges), with each an ordered pair $(u, v) \in V$. The graph is also associated with a node label mapping function $f : V \mapsto \mathcal{L}_V$, and a relation label mapping function $g : E \mapsto \mathcal{L}_E$. We also define a property function $p : V \mapsto \mathcal{P}_V$ which assigns key-value properties to each node. The property function is made to ensure for uniqueness of nodes regarding some properties which can be also formulated as $P_V(V_i, K) \neq P_V(V_j, K)$, where $V_i$ and $V_j$ are two distinct nodes with the same label and $K$ is the set of properties of the label. The graph is constructed from metadata of legal articles.

In Table 2, node labels and their attributes are presented. Notations are introduced to clarify unique properties denoted by *, and optional attributes are indicated by brackets. To mitigate confusion from French nomenclature, English definitions are provided in parentheses. Furthermore, only supplementary information from the original graph is included.

In Table 3, we delineate the labels of the relationships, specifying the left node (subject) and the right node (object), along with the existential conditions pertaining to the presence of entities in the metadata to establish the relationship examined. It is noteworthy that CODE may replace PROTOCOLE or CONSTITUTION based on the legal document type.

| Nodes | Attributes |
|---|---|
| ACTE (CONVENTION) | titre* (title) |
| ARRETE (ORDER) | numero* (number), multiplicatif* (multiplicative adverb) |
| ARTICLE | numero (number), [multiplicatif] (multiplicative adverb), [titre] (title), [date], contenu (content), [division_partie] (part subdivision), [division_livre] (book subdivision), [division_titre] (title subdivision), [division_sous_titre] (subtitle subdivision), [division_chapitre] (chapter subdivision), [division_section] (section subdivision), [division_sous_section] (subsection subdivision), [division_paragraphe] (paragraph subdivision), [section_application] (article type) |
| CODE | titre* (title) |
| CONSTITUTION | titre* (title) |
| DECRET (DECREE) | numero* (number), [objet] (object), [date] |
| LOI (LAW) | numero* (number), [objet] (object), [date] |
| PROTOCOLE (PROTOCOL) | titre* (title) |
| TRONCON (CHUNK) | numero (number), contenu (content) |

Table 2: Graph's Nodes and Attributes

| Relationship | Left Node | Right Node | Condition |
|---|---|---|---|
| ASSOCIER (ASSOCIATE) | LOI | DECRET | Both law and decree must be present. |
| POSSEDER (POSSESS) | CODE | LOI | Law must be present. |
| POSSEDER | CODE | DECRET | Decree must be present. |
| POSSEDER | CODE | ARRETE | Order must be present. |
| POSSEDER | LOI | ARTICLE | Law must be present, while decree may be absent. |
| POSSEDER | DECRET | ARTICLE | Decree must be present. |
| POSSEDER | ARRETE | ARTICLE | Order must be present. |
| POSSEDER | CODE | ARTICLE | Domain attribute must be present. |
| CONTIENT (CONTAINS) | ARTICLE | TRONCON | No specific condition. |
| PRECEDER (PRECEDE) | TRONCON | TRONCON | No specific condition. |

Table 3: Graph's Relationships and Conditions

## 2.3 Knowledge Graph Completion

### 2.3.1 Generating Knowledge Triples

Initially, the prompt from Kane et al. (2025) is deemed useful for generating triples. Due to the original paper's limited testing scope, a high temperature setting is necessary for enhanced creativity during inference. The temperature, a float value, moderates the likelihood of output tokens to promote variability. This method also encourages the LLM to generate inaccuracies. A temperature of 0.6 was selected, resulting in the generation of knowledge triples from 4439 articles. The input context incorporated cited references parsed according to the original methodology (Kane et al., 2025), identifying entities cited in the articles through the same LLM used for triple generation. Additionally, we evaluated an efficient LLM from Mistral AI, the mistral-medium-2505, which is closer in function to the original paper's best model, mistral-large-2411.

### 2.3.2 Evaluating Knowledge Triples Generation

From the generated knowledge triples, those identified as containing inaccuracies were utilized to formulate an enhanced prompt for the LLM. Several of these were incorporated alongside their respective inputs as exemplars in the initial prompt to emulate the desired input-output relationship (Kane et al., 2025). Certain elements, such as the identification of cited preceding or subsequent articles, as well as those featuring multiplicative adverbs, among other factors, were recognized as pivotal for enhancement.

The knowledge triples identified with inaccuracies subsequently served as a test set, from which a ground truth was established to assess the efficacy of the revised prompt. In the evaluation methodology, a reduced temperature setting of 0.4 was employed to strike a balance between creativity (allowing for more flexible reasoning) and adherence to the format exemplified by the prompt's illustrations.

Furthermore, the evaluation process was leveraged to explore inquiries regarding the significance of incorporating cited references by utilizing parsers as supplementary context within the input (Kane et al., 2025). Comparisons were conducted between: the inclusion of cited references in the prompt's exemplars and input, the exclusion of such references from the input while maintaining the examples intact, and the complete removal of all cited references from both the context of the input and the examples.

For this investigation, we assessed the equity of knowledge triples using ROUGE metrics, focusing on subjects, predicates, and objects separately. Knowledge triples are defined by the predicate "corresponds to" or "refers to," allowing for clear component differentiation. We then concatenated subjects, predicates, and objects to create distinct groups. A similar procedure was applied to the triples of ground truth knowledge before calculating ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-LSUM for sentence-level evaluation, as referenced in the article (Lin, 2004). This approach enables a more accurate comparison by considering the structural elements of knowledge triples.

### 2.3.3 Cross-Referencing

To facilitate cross-references, articles and their metadata are organized into a document $D$, allowing for the retrieval of metadata $meta$ denoted as $D_{meta}$. Algorithm 1 utilizes structured input triples for document extraction by aligning them with document metadata through mathematical and pattern recognition methods. It commences by analyzing the input string $S$, transforming it into triples $\mathcal{T}$, and refining them via a reference regular expression pattern $\Pi$.

The matching pattern $\Pi$ is a complex regular expression designed to identify triples that reference legal articles, laws, decrees, codes, constitutions, and protocols in French. It captures phrases that start with "refers to," followed by article numbers or ranges, along with optional identifiers such as law numbers ($\lambda$), decree numbers ($\delta$), legal document names ($\sigma$), multiplicative adverbs, and paragraphs.

For each relevant triple in $\mathcal{T}_r$, the algorithm applies specific regular expressions to extract components: references to raw articles ($A_{\text{raw}}$), multiplicative adverbs ($M_{\text{raw}}$), decree identifiers ($\delta$), law identifiers ($\lambda$) and legal document names ($\sigma$).

**Algorithm 1** Extract Relevant Documents from Triples

**Require:** $S$: String of input triples, $\mathcal{D} = \{D_i\}_{i=1}^{N}$: documents with metadata, $M \in \mathbb{N}$: max docs per reference, $\varphi$ : Adverbs $\rightarrow \mathbb{N}$: map from adverbs to numeric values
**Ensure:** $\mathcal{R} \subseteq \mathcal{D}$: extracted docs matching input references
Parse input triples: $\mathcal{T} \leftarrow \text{Parse}(S)$
Select matching triples: $\mathcal{T}_r \leftarrow \{t \in \mathcal{T} \mid t \sim \Pi\}$
Initialize unique keys and results: $U \leftarrow \emptyset, \mathcal{R} \leftarrow \emptyset$
Initialize count map: $C(r) \leftarrow 0, \quad \forall r$
**for all** triple $t \in \mathcal{T}_r$ **do**
  Extract content string $c$ from $t$
  Extract sets/components: $A_{\text{raw}}, M_{\text{raw}}, \delta, \lambda, \sigma \leftarrow \text{Extract}(c)$
  Expand article ranges: $A \leftarrow \text{ExpandRanges}(A_{\text{raw}})$
  Partition $A$: $A_n = \{a \in A \mid a \text{ discrete}\}, A_o = A \setminus A_n$
  Expand multiplicatif ranges:

$$M_u \leftarrow \begin{cases} \text{ExpandRanges}(M_{\text{raw}}), & M_{\text{raw}} \neq \emptyset \\ \{1\}, & \text{otherwise} \end{cases}$$

  **for all** document $D \in \mathcal{D}$ **do**
    Map multiplicatif:

$$m_D \leftarrow \begin{cases} \varphi(D_{\text{mult}}), & D_{\text{mult}} \in \text{dom}(\varphi) \\ \text{int}(D_{\text{mult}}), & \text{otherwise} \end{cases}$$

    Determine reference key $r_D \in \{\delta, \lambda, \sigma\}$ based on availability
    Define unique key $u_D \leftarrow (D_{\text{art}}, D_{\text{loi}}, D_{\text{dec}}, m_D, D_{\text{doc}})$
    **if** $u_D \notin U \wedge C(r_D) < M$ **then**
      **if** $D_{\text{art}} \in A_n \wedge m_D \in M_u$ **then**
        Add $D \rightarrow \mathcal{R}, u_D \rightarrow U, C(r_D) \leftarrow C(r_D) + 1$
      **else**
        **for all** open range $a \in A_o$ **do**
          Interpret open range: $a = \langle x \Rightarrow a' < x, \quad a = x\rangle \Rightarrow a' > x$
          **if** $((a = \langle x \wedge D_{\text{art}} < x) \vee (a = x\rangle \wedge D_{\text{art}} > x)) \wedge m_D \in M_u$ **then**
            Add $D \rightarrow \mathcal{R}, u_D \rightarrow U, C(r_D) \leftarrow C(r_D) + 1$
          **end if**
        **end for**
      **end if**
    **end if**
  **end for**
**end for**
**return** $\mathcal{R}$

The raw article references $A_{\text{raw}}$ are then transformed into distinct sets by converting number ranges and ellipses into explicit values $A_n$ and open-ended ranges $A_o$, denoting inequalities. Similarly, the multiplicative adverbs $M_{\text{raw}}$ are converted into a numeric set $M_u$, defaulting to $\{1\}$ when unspecified.

Each document $D$ in the document collection $\mathcal{D}$ is analyzed by mapping its multiplicative metadata to a numeric value $m_D$ via the function $\varphi$ or integer conversion. A unique key $u_D$ is created from the document metadata fields - article number $D_{\text{art}}$, law number $D_{\text{loi}}$, decree number $D_{\text{dec}}$, multiplicative number $m_D$, and legal document identifier $D_{\text{doc}}$ - serving as a composite identifier to avoid duplicates in the matching process.

A document $D$ is included only if its key $u_D$ remains unprocessed and the corresponding count $C(r_D)$, where $r_D \in \{\delta, \lambda, \sigma\}$ is derived from identifiers in the triple, does not exceed a predetermined limit $M$.

The inclusion condition verifies if $D_{\text{art}}$, the document's article number, exists within the discrete article set $A_n$ and its multiplicative number $m_D$ is within $M_u$. If not, the algorithm assesses whether $D_{\text{art}}$ meets the open-ended range criteria defined in $A_o$, symbolized as inequalities. If the document fulfills these range and multiplicative conditions, it is also incorporated.

Selected documents are incorporated into the result set $\mathcal{R}$, with unique keys stored in $U$ to avoid duplication and incrementing counts $C(r_D)$ to uphold maximum references. This approach ensures thorough yet regulated document extraction pertinent to input triples, achieving a balance between precision and output size via stringent matching criteria and efficient pattern matching and set expansions. The algorithm's time complexity, contingent on operational count, is approximated as $O\left(4 + T \times (5 + D \times (3 + |A_o| \times 2)\right)$, with $|A_o|$ representing the count of open-ended ranges. The complexity simplifies to $O(T \times D \times |A_o|)$, indicating reliance on the number of triples, legal articles, and the maximum count of open-ended ranges within the triples.

Subsequently, the algorithm is executed on each article's triples to extract cited documents. These documents, primarily legal articles, facilitate the establishment of a FONDER_SUR (BASE_ON) relationship between the source article and the cited documents.

## 2.4 KG-Augmented LLM

The KG-Augmented LLM operates via a Retrieval Augmented Generation system, delineated in Figure 2.

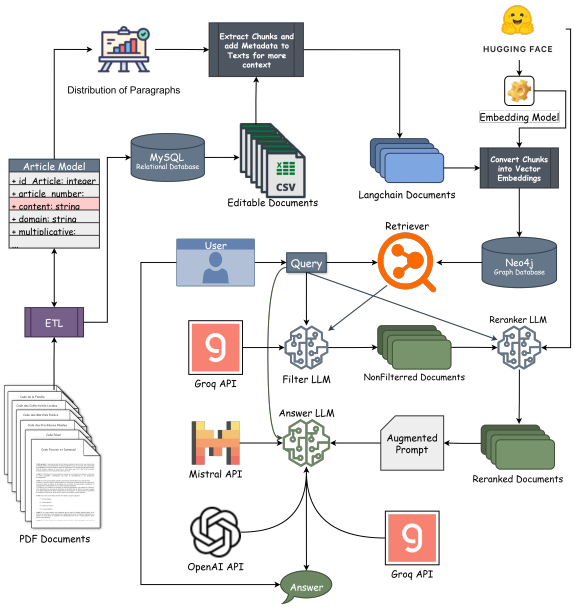


Figure 2: Retrieval Augmented Generation Framework.

### 2.4.1 Data Preparation

Initially, an Extract-Transform-Load (ETL) process was established with a standardized object model encompassing various article attributes. This ETL application on legal documents resulted in a relational database storing articles and their metadata. Subsequently, we processed the CSV files from the relational database to delineate chunks, incorporating metadata for contextual enhancement. Ultimately, these chunks were converted into documents to facilitate further analysis.

### 2.4.2 Vector Index

In the next phase, a succinct process was undertaken to create a vector index for chunks (TRONCON node) to assist in locating relevant chunks based on user inquiries. Chunks were then transformed into vector embeddings utilizing an embedding model from the Hugging Face platform. We opted for the multilingual-e5-small model by intfloat for efficient processing of French texts. The vector embeddings were subsequently incorporated as an additional attribute of the chunks within the graph database.

#### 2.4.3 Retrieval and Generation

The third critical phase commences upon user query submission. The graph database enables document retrieval. The retriever aims to produce multiple documents by aligning the query's vector embedding with those in the graph database, favoring nodes with elevated cosine scores in relation to the query embedding. Thereafter, we employed a filter-LLM from the Groq platform, namely the gpt-oss-20b, to discern and select documents relevant to the query for contextual purposes.

Another requirement is the re-ranking of documents by relevance to the query, incorporating Kelsen's hierarchy into the process. A state-of-the-art re-ranker LLM, specifically mmarco-mMiniLMv2-L12-H384-v1, was selected for processing French texts.

The re-ranked documents and query are then submitted to the Answer-LLM for the final response, which could originate from Mistral, OpenAI, or Groq platforms for comparative analysis.

In the analysis, multiple metrics were employed, including cosine similarity between vector embeddings of actual and predicted responses, ROUGE metrics (Lin, 2004), and the SacreBLEU metric (Post, 2018). However, to effectively ascertain factual accuracy and detect instances of hallucination, it is imperative to utilize task-specific metrics and frameworks that are specifically designed for the evaluation of Retrieval-Augmented Generation (RAG), which assess the components of both the retriever and the generator independently. Consequently, we employ Retrieval Augmented Generation Assessment (RAGAs) (Es et al., 2025) for a systematic assessment of the retriever's context precision and recall, as well as the generator's faithfulness, relevance, and correctness of answers, using OpenAI's gpt-4o-mini as the evaluation language model.

The retriever systematically identifies and ranks ARTICLE nodes related to salient CHUNK nodes, selecting the top $n$. It explores ARTICLE nodes to a depth of $d$ via the FONDER_SUR relation to gather related nodes. For each related node, it compiles a maximum of $l$ directly connected nodes, organized as bullet points to enrich the content. The enhanced material, along with scores and metadata, is presented in a relevance-sorted fashion, resulting in retrieved documents. The hyperparameters $n$, $d$, and $l$ optimize comprehensiveness and efficiency, leading to concise, prioritized results. The computational complexity is approximately $O(n \times d \times l)$.

## 3 Experiments and Results

The experiments utilized a Windows 11 Family 64-bit system (build 26100) on a local computer. The hardware specifications included an Intel Core i7-9750H CPU with 12 cores, 16 GB of RAM, and an NVIDIA GeForce RTX 2060 GPU featuring 6 GB of dedicated video memory.

### 3.1 Graph Completion Results

#### 3.1.1 Evaluation of Knowledge Triples Generation

The evaluation test set for knowledge triples generation included 111 articles. Table 4 presents the comparative results between the test set and the ground truth across various configurations. The metrics displayed are ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-L (R-L), and ROUGE-LSUM (R-LSum) for each triple scope, specifically subject, predicate, object, and the entire triple. The configurations are organized in rows: "Ref" indicates the inclusion of cited references in the prompt, "No-In-Ref" signifies the absence of cited references in the input, and "No-Ref" denotes the lack of cited references in both examples and input.

The findings indicate that incorporating references into the input yields superior triple generation. The second-best outcome emerges from the "No-Ref" condition. This consistency is evident as all metrics across various scopes reflect the same comparative results.

#### 3.1.2 Cross-Referencing Result

We utilized the ground-truth and LLM-generated triples, excluding the test set, to establish cross-references among articles in the graph via the algorithm in section 2.3.3. The resulting graph database consists of $13,687$ nodes and $45,961$ relationships. The counts of significant nodes and relationships are detailed in Table 5. Sub-table 5a presents the count of nodes specific to this paper's graph database. Sub-table 5b exhibits the count of primary relationships within the graph database. A substantial number of cross-references (FONDER_SUR relationship) was noted. The analysis of cross-references per article indicated a minimum of 1 and a maximum of 837, attributed to certain articles citing entire legal documents or neighboring articles. The average citation per article was found to be 15.64, with a median of 1 and an upper fence of 3.5. These findings suggest

| Configurations / Scopes | Subject | | | | Predicate | | | | Object | | | | Triple | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *R-1* | *R-2* | *R-L* | *R-LSum* | *R-1* | *R-2* | *R-L* | *R-LSum* | *R-1* | *R-2* | *R-L* | *R-LSum* | *R-1* | *R-2* | *R-L* | *R-LSum* |
| Ref | 80.8 | 77.9 | 80.5 | 81.0 | 83.9 | 78.0 | 83.9 | 84.0 | 81.6 | 78.8 | 79.9 | 81.7 | 81.9 | 80.1 | 80.3 | 82.0 |
| No-In-Ref | 90.6 | 89.7 | 90.4 | 90.6 | 93.3 | 84.2 | 93.3 | 93.3 | 92.4 | 91.3 | 91.3 | 91.8 | 92.2 | 91.6 | 91.5 | 92.2 |
| No-Ref | 86.5 | 84.6 | 86.1 | 86.5 | 90.6 | 81.5 | 90.4 | 90.6 | 88.1 | 85.9 | 87.1 | 88.1 | 88.3 | 86.9 | 87.1 | 88.3 |

Table 4: Comparison between Knowledge Triples Generation Configurations

that the majority of articles cite no more than four articles, with over 50% referencing only one. An illustrative example of the graph is provided in Figure 3, visualizing only 15 nodes for clarity. The graph depicts articles from the Criminal Code and their corresponding cross-references, including the primary law of the legal code.

| Nodes | Count |
|---|---|
| ACTE | 2 |
| ARRETE | 19 |
| ARTICLE | 6741 |
| CODE | 13 |
| CONSTITUTION | 1 |
| DECRET | 49 |
| LOI | 67 |
| PROTOCOLE | 1 |
| TRONCON | 6598 |

(a) Count of Nodes

| Left Node | Relationship | Right Node | Count |
|---|---|---|---|
| CODE | POSSEDER | ARTICLE | 4318 |
| LOI | POSSEDER | ARTICLE | 2637 |
| DECRET | POSSEDER | ARTICLE | 2358 |
| ARRETE | POSSEDER | ARTICLE | 11 |
| TRONCON | PRECEDER | TRONCON | 152 |
| ARTICLE | FONDER_SUR | ARTICLE | 26540 |

(b) Count of Relationships

Table 5: Graph Nodes and Relationships Counts

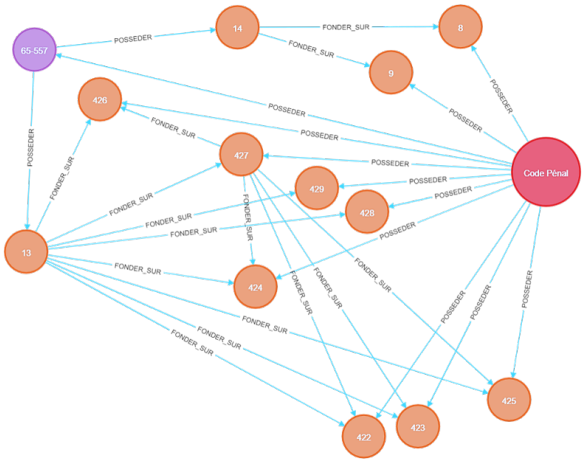


Figure 3: Graph Database Visualization : CODE, LAW, ARTICLE.

In the global context, a density of $D = 0.0002$ is derived from the equation:

$$D = \frac{|E|}{|V| \times (|V| - 1)}$$

Where $|E|$ signifies the quantity of edges and $|V|$ denotes the count of vertices. Furthermore, a relationship-to-node ratio of $R = 3.36 \simeq 3$ is computed using the formula:

$$R = \frac{|E|}{|V|}$$

### 3.2 Retrieval Augmented Generation System's Results

The RAG system's assessment utilized a test set of 41 inquiry-response pairs validated by domain experts, sourced from the analyzed legal texts. A diverse array of Large Language Models was incorporated into the benchmark from multiple artificial intelligence platforms, including:

- Mistral AI: mistral-medium-2505 (Mistral-Medium), magistral-medium-2507 (Magistral-Medium), mistral-large-2411 (Mistral-Large), pixtral-large-2411 (Pixtral-Large)
- OpenAI: o3-2025-04-16 (O3), o3-mini-2025-01-31 (O3-Mini),
- Groq: gpt-oss-120b (GPT-OSS), llama-3.3-70b-versatile (Llama-3.3)

We ensured that the benchmark exclusively comprises the most recent and high-performing models, during the experimentation phase, from OpenAI and Mistral AI, as well as the newly released open source LLM from OpenAI, available in Groq, alongside the versatile iteration of the latest Llama model, also introduced in Groq. A suitable temperature parameter of 0.4 was utilized to achieve a balance between creativity and precision. For the retrieval mechanism, we predetermined the hyperparameters: the number of documents $n$, the depth $d$, and the limit $l$ to be 10, 1, and 5, respectively, to minimize context and mitigate the context window constraints inherent in the models.

In Table 6, the statistical results are systematically outlined, with columns designated for various metrics including ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-L (R-L), ROUGE-LSUM (RLs), SacreBLEU (sBLEU), cosine similarity score (SimScore), and Estimated Inference Duration (EID) in

minutes. Additionally, the table includes a column indicating each model's number of parameters (in billions). The rows facilitate a comparative examination of distinct large language models (LLMs).

The results reveal that models developed by OpenAI yield superior outputs, with the O3-Mini model unexpectedly exceeding the O3 model in all metrics, except for sacreBLEU, where O3 excels. The Pixtral-Large model closely follows, demonstrating elevated ROUGE and similarity scores. Regarding the sacreBLEU metric, Mistral-Medium surpasses the Pixtral-Large model. Importantly, output quality is not solely determined by parameter count; factors such as architecture, data, and optimization are also significant.

Comparable EID values are observed between models from the same platform. Groq models exhibit superior inference speed relative to OpenAI models. The average recorded time for document retrieval from the graph database, including filtering and rearranging, was 7.73 minutes.

| Model | Params (B) | R1 | R2 | RL | RLs | sBLEU | Sim | EID |
|---|---|---|---|---|---|---|---|---|
| Mistral-Medium | ∼195 | 45.8 | 26.0 | 30.1 | 38.5 | 17.4 | 93.6 | 6.7 |
| Magistral-Medium | ∼195 | 44.5 | 25.2 | 28.9 | 37.8 | 16.1 | 93.4 | 7.0 |
| Mistral-Large | 123 | 45.9 | 26.3 | 30.1 | 38.4 | 16.3 | 93.3 | 6.5 |
| Pixtral-Large | 124 | 46.5 | 27.4 | 30.3 | 39.0 | 16.7 | 95.0 | 6.7 |
| O3 | ∼175 | 51.0 | 31.1 | 35.7 | 42.4 | 23.2 | 94.5 | 3.9 |
| O3-Mini | ∼200 | 52.0 | 31.4 | 36.2 | 44.0 | 23.0 | 94.6 | 4.0 |
| GPT-OSS | 120 | 30.6 | 16.9 | 20.3 | 25.2 | 7.2 | 89.2 | 2.0 |
| Llama-3.3 | 70 | 40.9 | 23.7 | 28.0 | 33.9 | 5.7 | 92.9 | 2.6 |

Table 6: RAG Results and Parameter Sizes

On the other hand, the generator evaluation is presented in Table 7 similarly to the statistical results. Answer correctness exhibits a comparable trend to ROUGE metrics, maintaining the same ranking for the most accurate models. Furthermore, answer relevance shows a pattern parallel to the SacreBLEU score, with identical configurations for the top three relevant models. The findings indicate that O3 is the most relevant model, while O3-Mini yields the highest correctness in answers. However, in terms of faithfulness, Llama-3.3 demonstrates superior performance, closely followed by Mistral-Medium and O3-Mini. The faithfulness metrics imply that GPT-OSS offers more reliable answers compared to Magistral-Medium and Pixtral-Large.

Moreover, the retriever evaluation yields a context recall of 30.6% and a context precision of 55%.

| Models / Metrics | Faithfulness | Answer Relevance | Answer Correctness |
|---|---|---|---|
| Mistral-Medium | 74.2 | 83.5 | 56.7 |
| Magistral-Medium | 59.7 | 76.4 | 57.7 |
| Mistral-Large | 70.1 | 71.2 | 56.4 |
| Pixtral-Large | 58.6 | 67.2 | 59.2 |
| O3 | 72.1 | 92.9 | 60.4 |
| O3-Mini | 73.8 | 88.2 | 68.6 |
| GPT-OSS | 62.4 | 47.6 | 52.3 |
| Llama-3.3 | 78.7 | 39.7 | 57.4 |

Table 7: Retrieval and Generation Metrics

## 4 Conclusion

The present study has significantly augmented the legal graph database initially established by Kane et al. (2025) through the incorporation of an LLM-Augmented Knowledge Graph, subsequently assessing its efficacy within a question-answer framework utilizing a Knowledge Graph-enhanced LLM architecture. We devised a methodological approach for dataset configuration informed by Kelsen's pyramid, which integrates both legal texts and pertinent contextual information, in conjunction with a chunking strategy suitable for legal and diverse textual datasets. The construction of the graph database resulted in a comprehensive structure comprising 13,687 nodes and 45,961 relationships, facilitating extensive cross-referencing opportunities for legal practitioners.

Furthermore, we generated and critically assessed knowledge triples utilizing Large Language Models, evidencing that the inclusion of cited references within the prompts significantly enhances the process of triple generation. An algorithm was developed for the extraction of relevant documents to support the completion of the graph. The efficacy of the system was evaluated through a Retrieval Augmented Generation (RAG) methodology, which integrates embeddings and retrievers to discern pertinent documents in response to user inquiries. A comparative analysis underscored the contemporary OpenAI models as the most proficient in producing relevant and accurate responses, notwithstanding certain performance trade-offs.

In summary, this research presents a robust framework aimed at enhancing legal knowledge graphs through LLM capabilities, thereby providing valuable instruments for the retrieval and analysis of legal information. Subsequent research endeavors will concentrate on mitigating existing limitations and broadening the applicability of the system.

## Limitations

In this study, we examined Kelsen's pyramid, focusing on its four primary levels. Nonetheless, an extended version exists that incorporates a fifth level related to jurisprudence. This level facilitates legal adaptability, ensuring judicial flexibility that allows the legal system to promptly address societal changes while maintaining equilibrium with superior norms (Sahlan et al., 2023). Future research should utilize juricaf.org for its vast jurisprudential data. The challenge is to integrate various attributes as nodes in the graph database effectively.

Moreover, the paper lacks a comprehensive analysis of the graph database. Guidelines should be established for legal experts on its utilization, as analyzing this legal network will elucidate law interactions and identify key nodes for legal application and interpretation (Pires et al., 2022).

Concerning the Retrieval Augmented Generation system, we assessed LLMs using a limited ground-truth of 41 question-answers. There is a need to generate more relevant questions for legal documents and meticulously label them to enhance the RAG system's generalization, potentially using topic modeling techniques like Latent Dirichlet Allocation (LDA) or BERTopic for subtopic identification and question ranking (Chali and Hasan, 2015; Grootendorst, 2022).

Additionally, the RAG system's hyperparameters were fixed to minimize memory and computational expenses. Implementing automated optimization techniques such as Grid or Random Search could significantly enhance model performance, aiding in finding an optimal balance between context window and search depth in the graph database by incorporating $n$, $d$, and $l$ as hyperparameters, while also mitigating hallucination in the RAG system (Ko et al., 2024).

The RAG system exhibits hallucination due to a low context recall of 30.6%, suggesting limited alignment with ground-truth. Enhancing context recall may involve employing graph completion techniques to rectify original cross-references from the graph database ingestion by Kane et al. (2025) related to land and public domain codes. The original ingestion process, being labor-intensive, may have introduced inaccuracies, such as misclassifying legislative articles versus regulatory ones, thus constraining context relevance. Further refinement can be achieved by rephrasing queries to better express user intent and align with the knowledge base (Shi et al., 2024). Moreover, the integration of dense vector-based retrieval with sparse methods like BM25 has been demonstrated to improve the accuracy and efficiency of question-answering systems (Hakdağlı, 2024). Lastly, combining web search functionalities with knowledge graphs could enhance the accuracy and reliability of responses generated by LLMs (Xie et al., 2024).

Future research will aim to develop a collaborative multi-agent approach, allowing our legal RAG system to integrate domain-specific data effectively, thereby alleviating performance bottlenecks and enhancing response accuracy through cooperative agents within a modular framework (Salve et al., 2024).

## Acknowledgments

We express appreciation to the Senegalese government for providing public access to essential legal codes for this research. We acknowledge our collaborators for their contributions to the code and datasets that were integral to our methodology. The authors thank the expert contributors involved in the evaluation process of the paper. Lastly, we recognize the role of AI tools, including ChatGPT, in facilitating literature identification and improving the writing's clarity and style. The authors retain full accountability for the academic integrity and conclusions presented.

## References

Deryck Beyleveld. 1993. From the 'Middle-Way' to Normative Irrationalism: Hans Kelsen's General Theory of Norms. *The Modern Law Review*, 56(1):104–119. Publisher: [Modern Law Review, Wiley].

Yllias Chali and Sadid A. Hasan. 2015. Towards Topic-to-Question Generation. *Computational Linguistics*, 41(1):1–20.

Yongming Chen, Miner Chen, Ye Zhu, Juan Pei, Siyu Chen, Yu Zhou, Yi Wang, Yifan Zhou, Hao Li, and Songan Zhang. 2025. Leverage Knowledge Graph and Large Language Model for Law Article Recommendation: A Case Study of Chinese Criminal Law. *arXiv preprint*. ArXiv:2410.04949 [cs].

Andrea Colombo. 2024. Leveraging Knowledge Graphs and LLMs to Support and Monitor Legislative Systems. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 5443–5446. ArXiv:2409.13252 [cs].

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2025. Ragas: Automated Eval-

uation of Retrieval Augmented Generation. *arXiv preprint*. ArXiv:2309.15217 [cs].

Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint*. ArXiv:2203.05794 [cs].

Özlem Hakdağlı. 2024. Hybrid Question-Answering System: A FAISS and BM25 Approach for Extracting Information from Technical Document. *Orclever Proceedings of Research and Development*, 5(1):226–237.

Yikun Han, Chunjiang Liu, and Pengfei Wang. 2023. A Comprehensive Survey on Vector Database: Storage and Retrieval Technique, Challenge. *arXiv preprint*. ArXiv:2310.11703 [cs].

Nourhan Ibrahim, Samar Aboulela, Ahmed Ibrahim, and Rasha Kashef. 2024. A survey on augmenting knowledge graphs (KGs) with large language models (LLMs): models, evaluation metrics, benchmarks, and challenges. *Discover Artificial Intelligence*, 4(1):76.

Amadou Kah. 1996. *La codification administrative au Sénégal*. These de doctorat, Paris 13.

Oumar Kane, Mouhamad M. Allaya, Dame Samb, and Mamadou Bousso. 2025. An Senegalese Legal Texts Structuration Using LLM-augmented Knowledge Graph. *arXiv preprint*. ArXiv:2510.02353 [cs].

Robin Ko, Mustafa Kağan Gürkan, and Fatoş T. Yarman Vural. 2024. ReRag: A New Architecture for Reducing the Hallucination by Retrieval- Augmented Generation. In *2024 9th International Conference on Computer Science and Engineering (UBMK)*, pages 961–965. ISSN: 2521-1641.

Philippe Lavigne Delville and D. Diagne. 2023. Dépossessions foncières, conflits de légalité et instrumentalisation du droit au Sénégal : essai de caractérisation de la conflictualité foncière rurale et péri-urbaine. *Géocarrefour*, 97(4):en ligne [20 p.].

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pages 9459–9474, Red Hook, NY, USA. Curran Associates Inc.

Yihao Li, Ru Zhang, and Jianyi Liu. 2024. An Enhanced Prompt-Based LLM Reasoning Scheme via Knowledge Graph-Integrated Collaboration. In *Artificial Neural Networks and Machine Learning – ICANN 2024*, pages 251–265, Cham. Springer Nature Switzerland.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Mandiaye Niang. 2009. The Senegalese Legal Framework for the Prosecution of International Crimes. *Journal of International Criminal Justice*, 7(5):1047–1062.

Rilder S. Pires, Henrique Santos, Ricardo Guedes, João A. Monteiro Neto, Carlos Caminha, and Vasco Furtado. 2022. Building and analyzing the brazilian legal knowledge graph. In *Proceedings of the 3rd International Workshop on Artificial Intelligence Technologies for Legal Documents (AI4Legal) co-located with the 21st International Semantic Web Conference (ISWC 2022)*, AI4Legal '22. CEUR-WS.org.

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Sahlan, Ilham Nurman, and Abdul Karim Uddin. 2023. The Role of Jurisprudence in the Development of Civil and Criminal Law. *European Journal of Law and Political Science*, 2(2):53–57.

Aniruddha Salve, Saba Attar, Mahesh Deshmukh, Sayali Shivpuje, and Arnab Mitra Utsab. 2024. A Collaborative Multi-Agent Approach to Retrieval-Augmented Generation Across Diverse Data. *arXiv preprint*. ArXiv:2412.05838 [cs].

Yunxiao Shi, Xing Zi, Zijing Shi, Haimin Zhang, Qiang Wu, and Min Xu. 2024. Enhancing Retrieval and Managing Retrieval: A Four-Module Synergy for Improved Quality and Efficiency in RAG Systems. *arXiv preprint*. ArXiv:2407.10670 [cs].

Xue Wu and Kostas Tsioutsiouliklis. 2024. Thinking with Knowledge Graphs: Enhancing LLM Reasoning Through Structured Data. *arXiv preprint*. ArXiv:2412.10654 [cs].

Weijian Xie, Xuefeng Liang, Yuhui Liu, Kaihua Ni, Hong Cheng, and Zetian Hu. 2024. WeKnow-RAG: An Adaptive Approach for Retrieval-Augmented Generation Integrating Web Search and Knowledge Graphs. *arXiv preprint*. ArXiv:2408.07611 [cs].