

Article

The Chunking Crisis in Hierarchical Structural Law: Bridging the Ontological Gap in Legal RAG via Deterministic Ontology-Driven Ingestion

Youssef Al Mouatamid ^{1,2,*}, Marie Bonnin ¹  and Jihad Zahir ^{2,3}¹ Laboratoire des Sciences de l'Environnement Marin (LEMAR), University of Brest, F-29280 Plouzané, France² Laboratoire d'Informatique et de Systèmes Intelligents (LISI), Cadi Ayyad University, Marrakesh 40000, Morocco³ Unité de Modélisation Mathématique et Informatique des Systèmes Complexes (UMMISCO), Institut de Recherche pour le Développement (IRD) France Nord, F-93143 Bondy, France* Correspondence: youssef.almouatamid@univ-brest.fr

Abstract

Standard Retrieval-Augmented Generation (RAG) architectures struggle with hierarchical legislation because chunking separates listed entities, such as protected species in annexes, from their governing articles. To bridge this ontological gap, we propose Semantic Breadcrumbs, a deterministic retrieval framework whose ontology-driven ingestion prefixes each chunk with its structural hierarchy and legal category. At inference time, a two-hop retriever anchors the questioned entity under a phrase-level identity gate, fetches the governing rules through citation-priority indexing, and routes a lightweight generative model into route-specific prompt contracts ending in a typed legal status. We evaluated the architecture on maritime and environmental legislation from nine Francophone African jurisdictions, using 104 citation-verified list-to-rule questions, control arms, and exact paired tests with Holm correction. Semantic Breadcrumbs reached 91.35% accuracy and a 91.43% macro F1-score, against 55.77% and 54.59% for standard chunking, a significant difference after correction. It was more accurate than exhaustive annex expansion with 2.6 times less context, a difference that is not significant after correction. Four context-aware chunking baselines at a matched budget trail it significantly after correction. It converted 86.21% of hallucination traps into typed abstentions. The evidence covers list-to-rule questions with one embedding model; a further generator replicates the ordering.

Keywords: generative AI; information retrieval; legal artificial intelligence; retrieval-augmented generation (RAG); legal informatics; document chunking; hallucination



Academic Editor: Vladimir Stantchev

Received: 21 September 2026

Revised: 18 October 2026

Accepted: 23 October 2026

Published: 27 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The integration of large language models (LLMs) into professional legal workflows has fundamentally transformed computational law [1–3]. However, the adoption of generative AI for critical legal tasks remains limited by the persistent risk of hallucinations [4]. Retrieval-Augmented Generation (RAG) has established itself as a key method for reducing this risk by grounding responses in retrieved documents from external sources [5,6], yet it does not preserve the structural relationships that legal reasoning depends on.

This limitation arises because legal documents are not simple linear texts. Rules, conditions, and penalties appear in articles, while the entities subject to those rules, such as protected species or regulated equipment, often appear in separate annexes. To answer

legal questions correctly, an AI system must connect these related parts of the document. Standard RAG pipelines work well for unstructured text, but they struggle when the relevant information is distributed across different sections of a hierarchical document [7].

The problem begins during document processing. Standard natural language processing (NLP) text splitters divide the document into independent chunks, and embedding models represent each chunk as a vector in semantic space. This process breaks the structural links between an entity in an appendix and its related rule in an article [7,8]. We refer to this structural failure as the guillotine effect. Once these links are lost, the retriever cannot bridge the ontological gap because the governing article does not explicitly mention the entity. The LLM, therefore, receives incomplete context and must rely on its pre-trained knowledge, which increases the risk of legally incorrect hallucinations through parametric leakage [9].

Recent literature has proposed several architectural approaches to improve RAG in structured domains, but these approaches still have important limitations when applied to hierarchical legal documents. Many studies focus on improving chunking strategies, using GraphRAG, extending context windows, or adopting multi-agent RAG systems [8,10]. Traditional RAG methods attempt to preserve context by detecting semantic boundaries or simply increasing the size of the chunks. When the relevant information spans different parts of a document, such as an article that refers to an appendix many pages away, these systems often insert large sections of the document into the LLM's context window. Although this approach can retrieve the required information, it also introduces a large amount of unrelated text, and the resulting increase in context size can reduce answer quality through the "Lost in the Middle" effect [11–13].

GraphRAG addresses this problem by representing legal knowledge as a graph, allowing the system to model relationships between entities and rules directly. However, building accurate legal knowledge graphs from unstructured PDF documents remains a major challenge. The process often requires manual annotation or intermediate LLM-based extraction, both of which increase complexity and introduce risks of parsing errors [14,15]. Multi-agent RAG systems attempt to automate this process by assigning an intermediate LLM to identify entities and generate structured outputs before retrieval. While this approach reduces manual effort, it replaces deterministic document processing with probabilistic generation. Errors made during this intermediate step can carry through the pipeline, especially when the model processes complex legal tables. These architectures also require more computation, which limits their suitability for resource-constrained deployments and privacy-sensitive legal environments.

These limitations suggest that improving legal RAG requires more than larger context windows or additional generative components. A reliable solution must preserve document structure through deterministic processing while avoiding unnecessary computational complexity.

To this end, this paper proposes Semantic Breadcrumbs, a RAG architecture designed for deeply nested laws. The core idea is that, instead of asking a model to reconstruct the legal hierarchy at inference time, every atomic chunk is annotated at ingestion time with an explicit textual link to its parent structure and a label carrying its legal category. Concretely, the system shifts the computational load from the inference stage to a highly structured ingestion stage. A declarative YAML ontology, executed by a Python 3.11-based stateful memory stack, tracks the connections between articles, annexes, and legal entities, and translates these topological links into explicit textual markers. These "semantic breadcrumbs" are injected directly into each atomic text chunk before vectorization, so that the topology is stated in the text that the embedding model reads. Because the structural links are embedded into the text itself, a standard dense retriever can locate related rules and entities

without any hierarchy reconstruction during inference, which removes a probabilistic error source and reduces the amount of context needed for generation. The same discipline governs inference: anchoring, identity verification, rule fetching, routing, and abstention are all decided by explicit rules, so the generative model is confined to the final deductive step. By grounding the architecture in deterministic data preparation rather than heavy parameter tuning, we establish a frugal AI pipeline that maximizes retrieval completeness, and thereby end-to-end answer accuracy, while minimizing the contextual footprint and computational costs.

The closest prior work is context-aware chunking, which keeps chunks small and re-stores their context before indexing. Contextual Retrieval prepends to each chunk a short description that an LLM generates from the whole document [16], late chunking pools each chunk vector from a long-context encoding of the full document [17], and summary-augmented chunking prefixes every chunk with a document-level summary, an approach evaluated on legal datasets [7]. Deterministic variants take the added context from the document itself: the article title in Dense Passage Retrieval [18], cascading metadata such as titles and section paths in HiQA [19], document metadata prefixed to chunks of regulatory filings [20], and the headings that the chunkers of the Docling toolkit [21] add to a chunk before embedding [22]. In the legal domain, generic chunkers have been compared on the General Data Protection Regulation [23], chunks that follow the structure of the German Civil Code have been compared with their contextual variants [24], and the structure of legislation has been encoded with a graph neural network to improve statutory article retrieval [25]. Semantic Breadcrumbs belongs to the deterministic family and extends it in three ways: a legal-category token, which is the term that the governing article shares with a list row; anchors taken from legal lists, including lists embedded in articles and an inference stage that reads the prefix for identity-gated two-hop retrieval and typed abstention routing.

The main contributions of this paper are as follows:

- A deterministic ingestion pipeline, driven by a declarative YAML ontology, that extracts topological relationships and legal categories during the ingestion stage and injects them into text chunks before vectorization. Three normalization passes (instrument-level namespacing, in-article species-list detection, and category inheritance) extend the anchor index beyond annexes, so that standard dense retrieval preserves structural relationships across every list-bearing topology of the corpus.
- An identity-gated deterministic two-hop retriever: a phrase-group identity gate routes each question to a Deep, Verify, or NoAnchor route; rule fetching follows a citation-priority index interleaved with dense retrieval and a typed verdict contract with route-specific prompts turns abstention into a deterministic routing outcome instead of a probabilistic model behavior.
- A retrieval evaluation measure, the *Topological Completeness Score (TCS)*, which verifies with word-boundary matching whether both required nodes of a question entered the generation context, making the evidence fed to each strategy auditable.
- An evaluation protocol designed to exclude alternative explanations of the results: 104 citation-verified questions across nine jurisdictions and three failure-mode subsets, budget-matched and decomposition control arms, and exact paired significance tests with Holm correction.

To evaluate these contributions, we conducted an ablation study on hierarchical maritime and environmental legislation from Francophone African countries. Semantic Breadcrumbs reached 91.35% accuracy against 55.77% for standard chunking, a difference that remains significant after correction for multiple comparisons. It was also more accurate than exhaustive annex expansion with 2.6 times less context. The comparison with four context-aware chunking baselines at a matched budget remains significant after correction, and a further

generator reproduces the ordering of the strategies. The remainder of this paper is organized as follows. Section 2 formalizes the ontological gap and presents the ingestion pipeline, the identity-gated retriever, and the typed verdict interface. Section 3 details the corpus, the strategies and control arms, the test bank, and the metrics. Section 4 reports the results, Section 5 discusses attribution, generalization, and limitations, and Section 6 concludes.

2. Methodology

To address the topological failures of standard RAG models in the legal domain, we propose Semantic Breadcrumbs, whose ingestion stage follows the medallion layering pattern from data engineering. This section formalizes the ontological gap and details our deterministic pipeline.

2.1. Mathematical Formulation of the Ontological Gap

Written law does not follow a linear sequence of text. Instead, it is structured as a directed acyclic graph (DAG) $G = (V, E)$. The set of vertices V consists of legal provisions, which are divided structurally into two subsets:

1. Abstract rules V_{rule} located in Articles (e.g., conditions, penalties).
2. Specific entities V_{entity} located in Annexes (e.g., listed taxonomic species).

The edges E represent the implicit and explicit hierarchical dependencies that link an entity to its legal rule.

Figure 1 instantiates G on the Guinean wildlife protection order. Article 2 declares the species of category I integrally protected, and Article 3 attaches the operative prohibitions (capture, detention, and killing) to the same category; neither article names a single species. The species appear only in ANNEXE I, a taxonomic table whose rows, such as the three sea turtles shown, carry no legal verb. Every edge of G , therefore, passes through the shared legal category *Espèces Intégralement Protégées*: it is the only material that links an annex row to the rules governing it. A chunking boundary placed between the articles and the annex severs all membership edges at once, so no annex-row vector retains any lexical trace of Article 3.

In a standard NLP pipeline, an embedding function $\phi : V \rightarrow \mathbb{R}^d$ maps text segments into a high-dimensional Euclidean space. The limitation of standard RAG that we call the ontological gap arises because the edge dependencies E lie outside the text of each segment, so the embedding function has no access to them. We introduce the term as an operational notion specific to this paper, and we do not present it as an established formal property of embedding models. If an entity v_{entity} appears only in an appendix and the corresponding rule v_{rule} refers only to the broader category, the two segments share little vocabulary, and the cosine similarity of their vectors, $\phi(v_{entity})$ and $\phi(v_{rule})$, tends to fall below the retrieval threshold τ :

$$\cos(\phi(v_{entity}), \phi(v_{rule})) < \tau. \quad (1)$$

Operationally, a question exhibits the ontological gap when single-pass retrieval does not return both v_{entity} and v_{rule} . Section 3.5 defines the corresponding measure, and Section 4 reports it.

This disruption of the legal hierarchy (the guillotine effect) breaks the links between related legal rules and causes the retriever to lose access to relevant context. When large language models (LLMs) attempt to reconstruct these missing links during inference, they introduce a probabilistic bottleneck that can lead to parametric leakage, including the generation of unsupported information.

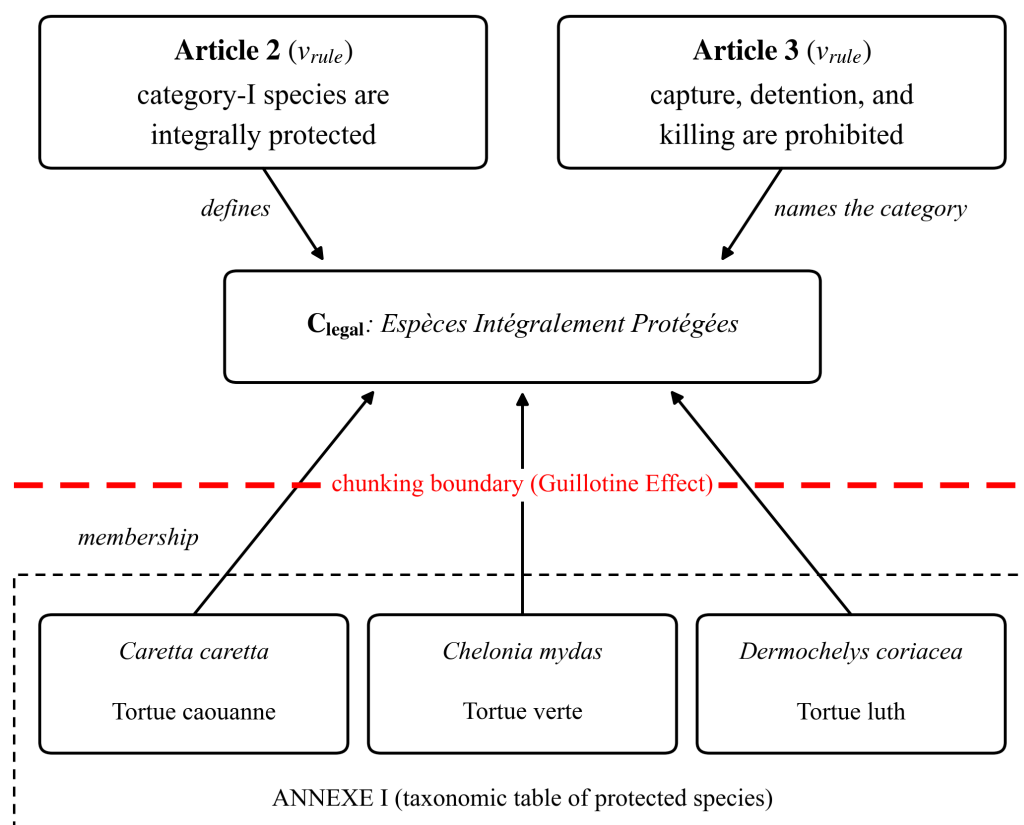


Figure 1. The legal DAG $G = (V, E)$ instantiated on the Guinean wildlife protection order. The rule nodes (Articles 2 and 3) never name a species, the entity nodes (rows of the ANNEXE I table) carry no legal verb, and every edge passes through the shared legal category C_{legal} . A single chunking boundary severs all membership edges at once, leaving each annex row lexically orphaned from its governing articles.

2.2. The Ontology-Driven Ingestion Pipeline

To keep the dependencies E recoverable by a dense retriever, we developed a stateful, deterministic data pipeline organized according to the medallion pattern from data engineering, which refines data through successive Bronze, Silver, and Gold layers.

1. **Bronze Layer (Raw State):** Ingests raw documents into a structured Markdown format, preserving tabular boundaries and structural integrity.
2. **Silver Layer (Structural Mapping):** Parses text blocks by tracking their hierarchy with a stateful Python memory stack. It uses a declarative YAML ontology to define the structure. The ontology sets clear structural levels, including BOOK, TITLE, CHAPTER, SECTION, and SUBSECTION, as well as special umbrella markers for Articles and Annexes.
3. **Gold Layer (Topological Injection):** Prior to vectorization into the persistent ChromaDB environment, the system executes a topology preservation function $B : V \rightarrow V'$. We artificially prepend to the raw text node v its hierarchical dependency path and its legal category to create a “Semantic Breadcrumb”:

$$v' = v \oplus E_{path} \oplus C_{legal}. \quad (2)$$

This structural metadata injection gives the enriched chunk v' explicit lexical overlap with the legal rule. The overlap is intended to raise the similarity between $\phi(v')$ and $\phi(v_{rule})$, and it does not guarantee their proximity.

Three deterministic normalization passes complete the Gold layer. First, every umbrella identifier is namespaced by its source instrument, which removes the cross-instrument collisions in which an umbrella expansion would otherwise mix articles from different laws. Second, species lists embedded inside articles are detected structurally: pipe-delimited table rows, or dash-prefixed rows carrying a parenthesized Latin binomial or a *Famille*: header, qualify when at least three such rows cover at least 30% of the umbrella’s content. This pass matters because three jurisdictions of the corpus (Cameroon, Senegal, and Tunisia) attach their species lists to articles instead of annexes, and an annex-only index would leave the retriever blind in exactly those jurisdictions. The resulting anchor flag (*is_anchor*), therefore, generalizes annex membership to list membership and covers 2183 chunks: 1807 annex rows and 376 in-article list rows. Third, each list row inherits the legal category of its list header, and the header text overrides any conflicting label produced upstream, so that C_{legal} is always the category the document itself asserts.

Since the operands of B are central to the rest of the paper, Table 1 defines each of them on the same green-turtle example as Figure 1. Two of these deserve emphasis. E_{path} is the *structural* coordinate of the chunk: the ordered path of hierarchy levels and umbrella markers (e.g., a title, chapter, or annex number) produced by the Silver layer. C_{legal} is not a structural level and not the entity itself: it is the *legal status category* that the document assigns to the entity, extracted from the annex (e.g., *Espèces Intégralement Protégées*). It is precisely this category that the governing article names, which is why injecting it into the entity chunk restores the severed edge.

Table 1. Notation of the topology preservation function B , instantiated on the green-turtle example of Figure 1.

Symbol	Definition	Example (Guinea)
v	Raw atomic text node	Cheloniidae Chelonia mydas Tortue verte I
E_{path}	Structural hierarchy path (levels and umbrella marker from the YAML ontology)	Guinée > ANNEXE I
C_{legal}	Legal status category assigned to the entity by the document	Espèces Intégralement Protégées
v'	Enriched chunk (Semantic Breadcrumb)	[Contexte: [Guinée ANNEXE I (Espèces Intégralement Protégées)]] Cheloniidae Chelonia mydas Tortue verte I

2.3. Identity-Gated Two-Hop Retrieval and Deterministic Routing

At inference time, legal queries frequently require crossing the ontological gap, and standard single-pass retrieval is ineffective: reconstructing the hierarchy during generation forces the LLM into exactly the probabilistic reasoning that produces parametric leakage. We, therefore, enforce a strict separation of tasks: every control-flow decision, from anchoring to abstention, follows deterministic rules, and the LLM performs only the last deductive step.

Hop 1 (entity anchoring under an identity gate). The system runs a dense search restricted to the anchor index (*is_anchor*: True) of the question’s jurisdiction and submits the top candidates to a deterministic identity gate. The gate extracts the species-designation groups of the question as contiguous content tokens: the connectors *de*, *du*, and *des* may appear inside a group, parentheses and commas close a group, attribute words such as *mâle* and *femelle* are stopworded, and token comparison is accent-normalized and tolerant of plural inflection. Each candidate anchor row is split into designation cells at pipes,

parentheses, commas, and dashes, with adjacent cells merged when a Latin binomial is split across them. A cell that contains a complete designation group is a *full match*. A cell that shares tokens with a group without containing it is a *partial match*, which is definitionally the near-species confusion case: a shared head noun or epithet, as between *Panthère* and *Panthère nébuleuse*. No overlap leaves the question unanchored; before concluding that, the system attempts one deterministic lexical scan of the full anchor index, which rescues exact designations that dense retrieval ranked poorly.

The gate's outcome selects one of three routes, so abstention becomes a routing decision before it is ever a generation decision:

- **Deep** (full match): the anchored breadcrumb enters the context as identity-verified evidence, and the prompt treats its category as authoritative.
- **Verify** (partial match): the anchor is kept, and a shared <Avertissement_Identite> block warns that the retrieved entity matches the questioned designation only partially; the prompt frames the identity question adversarially.
- **NoAnchor** (no match): the system falls back to capped flat retrieval under an absence-tolerant prompt, in which ABSENTE is the expected outcome whenever the questioned entity does not appear in the context.

Hop 2 (citation-priority rule fetching). For anchored questions, the system extracts C_{legal} deterministically from the breadcrumb and assembles the governing rules from two complementary sources. A reverse citation index, built at ingestion, maps every explicit reference to an annex or table ("annexe II", "tableau 2", Roman or Arabic numerals) to the articles that utter it within the same instrument, and follows one further step of explicit article cross-references ("l'article 7 ci-dessus"). Umbrellas found through this index take priority and are interleaved with the results of a question-aware dense query over the same jurisdiction's articles, under a cap of three umbrellas of at most 3500 characters each. The citation index covers the two reference patterns that defeat purely semantic rule retrieval: articles that cite their annex by number without naming the legal category, and articles that delegate their conditions to a neighboring article. The retrieved context is then handed to the generator under a route-specific prompt contract, described next.

2.4. Route Contracts and the Typed Verdict Interface

Each route carries a prompt contract, and these contracts are the proposed strategy's only prompt-level component: the Full Annex and Modular Agentic strategies of Section 3.3 share every retrieval mechanism above and answer under a standard prompt. PROMPT_DEEP declares the breadcrumb authoritative and imposes a deduction rule: rules attached to a legal category apply to every listed member of that category, so the model may not abstain merely because no article names the animal. PROMPT_VERIFY imposes the opposite discipline: it instructs the model to compare designations as names alone and to abstain whenever the questioned designation is not literally the retrieved one. PROMPT_FLAT makes ABSENTE the default for entities absent from the context and adds a decision rule for prohibition lists. All prompts share a status vocabulary that defines the six typed statuses below and three shared rules: a scope rule (the status types the act the question asks about, performed with the permit the question names, and the breadcrumb prevails over class or table terminology), a specialization rule (the rule of the entity's own legal régime prevails over a general clause of another régime), and an anti-parametric guard (world knowledge and international instruments such as the Convention on International Trade in Endangered Species (CITES) are excluded grounds).

Every answer must end with a single typed line, Statut Juridique : X, with X drawn from {INTERDIT, AUTORISE_SOUS_CONDITION, AUTORISE, OBLIGATOIRE, NON_OBLIGATOIRE, ABSENTE}. The status is question-independent: it types the legal situation

of the act, and a deterministic mapper (Section 3.5) converts it into the graded answer, so question-polarity resolution leaves the model entirely. The typed interface also enables a final deterministic safeguard, *abstention repair*: a full-match Deep anchor together with an ABSENTE status contradicts the gate's own verification, so the system issues exactly one re-ask with ABSENTE excluded from the vocabulary. Appendix B prints the four prompts and the repair note verbatim. The next section describes the corpus, the test bank, and the protocol under which this pipeline was evaluated.

3. Experimental Setup

This section describes the corpus, the inference environment, the four retrieval strategies and three control arms, the test bank and its construction protocol, and the metrics. Throughout, we separate two layers of the system: harness components (parsing, grading, context caps) apply to every strategy identically, and method components (anchor index, identity gate, citation-indexed rule fetching, abstention repair) are shared by the three structured strategies, which, therefore, differ only in their ablated dimension.

3.1. Corpus and Dual-Track Vectorization

The evaluation corpus consists of 14 maritime and environmental legislative instruments from nine Francophone African jurisdictions, yielding 3397 atomic chunks after ingestion. This field was chosen for its deeply hierarchical structure and its heavy use of tabular appendices, particularly taxonomic lists of protected marine and terrestrial species. Two declarative YAML profiles parse the corpus: *Profile_Deep_Hierarchy* for nested codes (titles, chapters, sections, annexes) and *Profile_Flat_Articles* for flat orders. Appendix C prints both profiles verbatim. The instruments were retrieved as official PDFs and converted to structured Markdown by optical character recognition (OCR). We verified the converted text manually, instrument by instrument, while authoring these profiles and the normalization passes. The validation of the reference answers (Section 3.4) rests on verbatim extracts of the ingested corpus checked against the official texts, so OCR errors do not affect the evidence of any evaluated question.

To separate ingestion compute from inference compute, we initialized a persistent ChromaDB environment containing two distinct vector spaces. The baseline space holds 3105 chunks produced by standard Recursive Character Text Splitting (chunk size $c = 250$, overlap $o = 20$); the Gold Layer space holds the 3397 breadcrumb-enriched chunks of Section 2.2, of which 2183 carry the anchor flag. All retrieval operations are filtered by jurisdiction metadata, so that each question is answered strictly within the legal corpus of its own country.

3.2. Inference Environment and Frugal AI Deployment

To test the architecture with a small open-weight model, we opted for a frugal AI approach. Embeddings were generated using the bge-m3 model [26], while generation was handled by mistral-nemo, a 12-billion-parameter (12B) open-weight LLM [27], with the temperature strictly fixed to $T = 0.0$, a 16,384-token context window, and a 1200-token generation limit, so that the evaluation is reproducible from the corpus, the bank, and the engine. Contexts are capped at 9000 characters for all strategies except Full Annex, whose main arm may grow to 45,000 characters; Section 3.3 adds a 12,000-character budget arm for that strategy. Appendix A gives the complete configuration. Every run of this study, including the baseline conditions and the robustness run, used locally served open-weight models: no legal text or question left the authors' machines, and the released dataset carries the ingested corpus, the bank, and the per-question outputs (Data Availability Statement).

No model was trained or fine-tuned in this study. The retrieval design and the route prompts encode properties of the legal texts themselves: the annex-article topology, the category-based application of rules, the citation conventions of legislative drafting, and the precedence of a special régime over a general clause. The prompts contain no species names and no jurisdiction-specific rules.

3.3. Retrieval Strategies and Control Arms

For each query Q , we define the set of required semantic nodes $V_{req} = \{v_{entity}, v_{rule}\}$ and evaluate how each strategy forms the context window W provided to the LLM:

- **Strategy 1 (Naive Baseline):** Standard flat chunking with single dense retrieval, $W = \text{Top-}k(\phi(v), \phi(Q)), k = 3$.
- **Strategy 2 (Full Annex—Bloat):** Retrieves the specific entity chunk, fully integrates the parent appendix into the context, and fetches the corresponding articles: $W = \{v_{entity}\} \cup V_{annex} \cup V_{rule}$, capped at 45,000 characters.
- **Strategy 3 (Semantic Breadcrumbs—Proposed):** Two-hop retrieval with $W = \{v'\} \cup V_{rule}$, where $v' = v_{entity} \oplus E_{path} \oplus C_{legal}$ is the enriched chunk of Section 2.2, answered under the route contracts of Section 2.4.
- **Strategy 4 (Modular Agentive):** A generative approach in which an intermediate LLM call parses the anchored row into a JSON digest $d(v')$ before rule retrieval: $W = \{d(v')\} \cup V_{rule}$, answered under the standard prompt.

Strategies 2–4 share Hop 1, the identity gate, and Hop 2, so their evidence coverage is identical by construction and they differ in the representation handed to the generator: raw annex expansion, the breadcrumb, or a generative digest. Strategy 3 additionally differs in its route contracts. Because that second difference confounds the attribution of Strategy 3's advantage, the evaluation includes three control arms, each of which isolates one alternative explanation of the results:

- **Arm 1b (budget-matched Naive):** Strategy 1 with $k = 12$, which raises the naive context budget to the proposed strategy's footprint. It tests whether the baseline is merely context-starved.
- **Arm 2b (Full Annex @12k):** Strategy 2 capped at 12,000 characters. It tests whether a smaller context budget changes the accuracy of Full Annex.
- **Arm 3b (decomposition):** Strategy 3's retrieval and routing combined with the standard prompt of Strategies 2 and 4. It separates the contribution of the representation from that of the route contracts.

Table 2 summarizes which components each of the seven configurations contains. The breadcrumb-enriched index serves Hop 1 in every structured configuration, so the comparisons among Strategies 2–4 vary the representation given to the generator and the prompt.

Four context-aware chunking baselines complete the comparison. They share the Gold-layer chunk boundaries, the embedding model, the jurisdiction filter, the standard prompt, and the typed grading, and they retrieve in a single dense pass under a context budget of 3300 characters, matched to the proposed strategy's realized average of 3313. Only the indexed prefix varies: no prefix; the heading path alone, that is, the breadcrumb without its legal category; the full breadcrumb and a document-level summary of about 150 characters, generated once per instrument by the same model. A converse ablation keeps the full two-hop retrieval, routing, and abstention repair, and removes the breadcrumb: the chunks are embedded and shown without their prefix, and the Hop-2 query carries no legal category. Section 4.4 reports these conditions and the ablation.

Table 2. Components of the seven evaluated configurations. Typed verdicts and deterministic grading apply to all of them; abstention repair applies to every configuration except the two naive ones. The four context-aware chunking baselines (single-pass, standard prompt) are defined at the end of this section and reported in Section 4.4.

Configuration	Chunks and Index	Hop 1 and Identity Gate	Hop 2 and Citation Index	Context Given to the Generator	Prompt
Naive, $k = 3$ (Strategy 1)	Recursive, 250/20, no prefix	No	No	Top-3 chunks	Standard
Naive, $k = 12$ (Arm 1b)	Recursive, 250/20, no prefix	No	No	Top-12 chunks	Standard
Full Annex @45k (Strategy 2)	Structural, breadcrumb	Yes	Yes	Whole list of the anchor and rules, up to 45,000 characters	Standard
Full Annex @12k (Arm 2b)	Structural, breadcrumb	Yes	Yes	Whole list of the anchor and rules, up to 12,000 characters	Standard
Breadcr., std. prompt (Arm 3b)	Structural, breadcrumb	Yes	Yes	Anchored breadcrumb chunk and rules	Standard
Semantic Breadcrumbs (Strategy 3)	Structural, breadcrumb	Yes	Yes	Anchored breadcrumb chunk and rules	Route contracts
Modular Agentic (Strategy 4)	Structural, breadcrumb	Yes	Yes	LLM-generated digest of the anchor and rules	Standard

3.4. Test Bank Design and Construction Protocol

The test bank contains 104 questions with citation-verified ground truth across nine jurisdictions: 58 GAP, 17 FLAT, and 29 TRAP questions (per-jurisdiction counts appear with the per-jurisdiction results in Section 4). Every question asks for the legal status of an act that concerns one named species, so the bank is a targeted test of list-to-rule reasoning and does not sample legal questions in general. The three subsets target known RAG failure modes:

1. **GAP** evaluates the guillotine effect, where v_{entity} lies in an annex or species list and v_{rule} in a separate article. A question qualifies only under a strict criterion: its answer must be underdetermined without both the list-membership hop and the governing-article hop.
2. **FLAT** evaluates standard retrieval, where no category-specific rule has to be selected: the species and its rule share one passage, or the species belongs to a single-régime list governed by one article.
3. **TRAP** tests parametric leakage by querying entities that the legal texts of the jurisdiction do not name (e.g., kangaroos in Guinea), forcing the LLM into a hallucination state if the control flow fails.

The bank design preceded the evaluation: the three subsets formalize recurring cases of legal logic that we identified in seminar discussions with legal experts, and the construction protocol was fixed before the system was evaluated. Question drafting was human-in-the-loop: a large language model proposed candidate questions from the legal texts under a single structural constraint, that each question instantiates the GAP, FLAT, or TRAP case. Every candidate entered the bank only after we checked it against the legal texts of its jurisdiction. Ground truth was authored from the legal texts alone: every question carries its source instrument and a verbatim citation locating its required nodes in the corpus, and

every expected answer was validated against those citations. Species were chosen across the protection categories of each jurisdiction and across answer polarities, and the absence of each TRAP entity was verified by a search of the corpus of its jurisdiction. A legal expert in environmental law then checked each of the 104 reference answers against the official texts and confirmed them. Three integrity rules governed construction and maintenance:

1. No question is ever added or dropped because of the outcome any strategy produces on it.
2. Ground-truth fields are never visible to retrieval, which receives query text only.
3. The per-question polarity annotations of Section 3.5 were authored and audited without reference to any system output.

3.5. Metrics, Grading, and Statistical Tests

Typed grading. The model's terminal status line is parsed and mapped deterministically: ABSENTE grades as ABSENTE; any other status grades as OUI if it belongs to the question's oui_si set and as NON otherwise. The oui_si annotation lists the statuses that answer the question affirmatively *as phrased*: for “is hunting prohibited?” it is {INTERDIT}, and for “can a permit be obtained?” it is {AUTORISE, AUTORISE_SOUS_CONDITION}. The annotation encodes question polarity, never the expected answer, so grading is exact string mapping with no heuristic matching.

Accuracy, recall, and F1. We report accuracy with 95% Wilson score confidence intervals (CIs) [28] and the macro-averaged recall and F1-score over the three graded classes (OUI, NON, ABSENTE); per-subset recall and F1 are macro-averaged over the classes present in that subset's ground truth (only ABSENTE for TRAP).

TCS. The Topological Completeness Score measures the retriever's ability to place both required nodes inside the context window W assembled for the model:

$$TCS = \left(\frac{|V_{req} \cap W|}{|V_{req}|} \right) \times 100. \quad (3)$$

A node counts as present only if its text matches inside W at word boundaries (case-insensitive), so that “Article 3” is not satisfied by “Article 31”; 12 of the 150 required node strings are prefix-ambiguous in this sense, and word-boundary matching removes the ambiguity. A per-node hit log accompanies every answer, which makes the evidence fed to each strategy auditable after the fact. TCS is undefined for TRAP questions, where $V_{req} = \emptyset$ by construction; the overall TCS is, therefore, averaged over the 75 GAP and FLAT questions.

Relation to standard retrieval metrics. TCS is an evidence-recall measure: every GAP and FLAT question has two required nodes, so the average TCS equals the share of the 150 required node strings found in the assembled contexts. It differs from Recall@ k in two ways. Its unit is the required evidence string instead of the ranked chunk, and its cutoff is the context that each strategy assembles, whereas Recall@ k fixes k . We report two refinements alongside it. *Recall by node type* separates the 59 required nodes that name an article from the 91 that name a species or an operative term. The *complete-evidence rate* is the share of questions whose context contains every required node, which is the condition that the ontological gap of Section 2.1 violates. The macro recall reported in Section 4 is a property of the graded answers; TCS and its refinements are properties of the assembled evidence. Reporting both separates retrieval completeness from the quality of the final answer.

Paired significance. All strategies answer the same 104 questions, so we compare them with exact McNemar tests [29], computed as a two-sided binomial test on the discordant pairs and reported as $b:c$, the questions the first strategy answers correctly and the second does not, against the converse. The evaluation comprises twelve paired comparisons, and we control the family-wise error rate over this family with the Holm procedure [30]. We report

both the unadjusted and the Holm-adjusted p -value of each test, and we call a difference significant only when its adjusted value is below 0.05. **Context length** is the character count of the assembled context. The next section reports the results obtained under this setup.

4. Results

4.1. Overall Performance

Table 3 summarizes the overall results. Semantic Breadcrumbs reached 91.35% accuracy and a 91.43% macro F1-score, against 55.77% and 54.59% for the naive baseline, at 2.6 times less context than Full Annex (3313 characters against 8737). Macro recall follows the same ordering: 91.22% against 55.74% (Table 3). The advantage over Naive is significant after Holm correction (44:7 discordant pairs, adjusted $p < 0.0001$). The 12.50-point advantage over Full Annex (18:5, $p = 0.0106$) does not remain significant after correction (adjusted $p = 0.0637$); the twelve paired tests are listed in Section 4.3. Modular Agentic trails by 4.81 points, a difference that is not significant (9:4, $p = 0.2668$), and Section 5.2 resolves that comparison on abstention, evidence auditability, and inference cost.

Table 3. Overall performance over the 104-question test bank. CIs are 95% Wilson score intervals. Recall (Rec.) and F1 are macro-averaged over the three graded classes. TCS is averaged over the 75 GAP and FLAT questions, since it is undefined on the TRAP subset.

Strategy	Acc. (%)	95% CI	Rec. (%)	F1 (%)	TCS (%)	Ctx.
Naive (Baseline)	55.77	[46.2, 64.9]	55.74	54.59	23.33	649
Full Annex (Bloat)	78.85	[70.0, 85.6]	77.53	80.22	94.67	8737
Semantic Breadcr.	91.35	[84.4, 95.4]	91.22	91.43	95.33	3313
Modular Agentic	86.54	[78.7, 91.8]	84.96	86.24	79.33	3173

Two structural observations frame the subset analysis. First, evidence coverage and answer quality are different quantities: Full Annex attains near-parity TCS (94.67 against 95.33) yet trails by 12.50 accuracy points, so the gap between the two strategies does not come from what was retrieved. The parity itself is a design invariant, since both strategies share the anchor and the rule fetch; the only per-question divergence is one Madagascar question whose Full Annex context was truncated at the 45,000-character cap, a mechanism that penalizes Full Annex alone. Second, the naive baseline's TCS of 23.33% confirms that standard chunking rarely delivers both required nodes, and Section 4.3 shows that raising its retrieval budget does not repair this.

Table 4 decomposes this coverage with the evidence-recall measures of Section 3.5, which gives the ontological gap a measured value. The naive baseline retrieves 26.37% of the species and term nodes and 18.64% of the article nodes, and it assembles complete evidence for three of the 75 questions (4.00%); at $k = 12$, the complete-evidence rate rises to 28.00%. Full Annex and the proposed strategy place every species and term node in context, and their residual misses are article nodes (88.14% recall for the proposed strategy). Modular Agentic shares the same rule fetch, and its lower recall of species and term nodes (73.63%) reflects its digest, which paraphrases the anchored row. Table 5 breaks the results down by failure mode.

Table 4. Evidence recall over the 75 GAP and FLAT questions (150 required nodes), computed from the per-node hit logs. Complete evidence is the share of questions whose context contains every required node. The proposed strategy and Arm 3b share one retrieval and, therefore, one row.

Configuration	Species/Term Nodes ($n = 91$)	Article Nodes ($n = 59$)	All Nodes (TCS)	Complete Evidence
Naive, $k = 3$	26.37	18.64	23.33	4.00
Naive, $k = 12$ (Arm 1b)	53.85	47.46	51.33	28.00
Full Annex @45k	100.00	86.44	94.67	89.33
Full Annex @12k (Arm 2b)	95.60	74.58	87.33	80.00
Semantic Breadcrumbs, Arm 3b	100.00	88.14	95.33	90.67
Modular Agentic	73.63	88.14	79.33	66.67

Table 5. Evaluation by failure mode ($n = 104$: GAP 58, FLAT 17, TRAP 29). Per-subset recall and F1 are macro-averaged over the classes present in the subset's ground truth (only ABSENTE for TRAP, where recall equals accuracy). TCS is undefined (n/a) on the TRAP subset, where $V_{req} = \emptyset$.

Failure Mode	Metric	Naive (Baseline)	Full Annex (Bloat)	Semantic Breadcr. (Proposed)	Modular Agentic
GAP	Accuracy	53.45%	82.76%	91.38%	89.66%
	Recall	51.68%	81.49%	91.83%	88.82%
	F1 Score	60.35%	83.77%	91.36%	89.34%
	TCS	16.38%	93.10%	93.97%	74.14%
	Context Size (Chars)	624	11,482	3327	3090
FLAT	Accuracy	41.18%	76.47%	100.00%	100.00%
	Recall	41.43%	75.71%	100.00%	100.00%
	F1 Score	51.67%	75.71%	100.00%	100.00%
	TCS	47.06%	100.00%	100.00%	97.06%
	Context Size (Chars)	732	4644	2280	2243
TRAP	Accuracy	68.97%	72.41%	86.21%	72.41%
	Recall	68.97%	72.41%	86.21%	72.41%
	F1 Score	81.63%	84.00%	92.59%	84.00%
	TCS	n/a	n/a	n/a	n/a
	Context Size (Chars)	650	5647	3891	3884

4.2. Failure-Mode Analysis

GAP: bridging the guillotine effect. On the 58 questions whose entity and rule live in separate structures, the naive baseline's TCS of 16.38% measures the guillotine effect at scale: standard chunking almost never reunites an annex row with its governing article. Semantic Breadcrumbs restores coverage (TCS 93.97%) and converts it into 91.38% accuracy. Full Annex reaches essentially the same coverage (93.10%) yet stops at 82.76% accuracy, from contexts that average 11,482 characters. We report the 8.62-point difference at similar coverage as an observed difference: the paired test against Full Annex on the full bank does not remain significant after correction (Section 4.3).

FLAT: standard retrieval. This subset requires no category-specific rule, and the structured strategies solve it (100.00% for Semantic Breadcrumbs and Modular Agentic). The two baselines still fail it in opposite ways. The naive retriever misses roughly half the required evidence (TCS 47.06%), retrieving nearby fragments that lack the operative clause. Full Annex holds perfect coverage yet answers only 76.47%, from contexts that average 4644 characters. With 17 questions, 16 of them from one Moroccan order, we report this difference without attributing it to context length.

TRAP: deterministic abstention. The 29 trap questions separate the strategies most sharply. The identity gate produced no full match on any trap, so every trap was answered under either the NoAnchor contract or the adversarial Verify contract, and Semantic Breadcrumbs converted 25 of 29 into correct typed abstentions (86.21%). Every other strategy stalls at or below 72.41%: with the same gate but the standard prompt, Full Annex and Modular Agentic still assert statuses for absent species, which points to the route contracts

as the source of the difference; Section 4.3 examines this reading with the decomposition arm. The four failed traps are analyzed in Section 4.9.

4.3. Control Arms and the Context-Budget Ablation

Table 6 reports the three control arms next to their reference strategies, and Table 7 lists the twelve paired comparisons with their unadjusted and Holm-adjusted p -values.

Table 6. Control arms next to their reference strategies ($n = 104$ each). Each arm isolates one alternative explanation of the headline result.

Configuration	Acc. (%)	95% CI	TCS (%)	Ctx.
Naive, $k = 3$	55.77	[46.2, 64.9]	23.33	649
Naive, $k = 12$ (Arm 1b)	54.81	[45.2, 64.0]	51.33	2621
Full Annex @45k	78.85	[70.0, 85.6]	94.67	8737
Full Annex @12k (Arm 2b)	81.73	[73.2, 88.0]	87.33	6155
Breadcr., std. prompt (Arm 3b)	83.65	[75.4, 89.5]	95.33	3313
Semantic Breadcrumbs (full)	91.35	[84.4, 95.4]	95.33	3313

Table 7. The twelve paired comparisons (exact McNemar tests, $n = 104$). $b:c$ counts the questions that only the first configuration answers correctly against those that only the second does. Adjusted p -values follow the Holm procedure over the twelve tests; the comparisons with naive chunking and with the four context-aware chunking baselines (Section 4.4) remain significant.

Comparison	$b:c$	p	Holm-Adjusted p
Semantic Breadcrumbs vs. Naive, $k = 3$	44:7	<0.0001	<0.0001
Semantic Breadcrumbs vs. Naive, $k = 12$ (Arm 1b)	43:5	<0.0001	<0.0001
Semantic Breadcrumbs vs. Full Annex @45k	18:5	0.0106	0.0637
Semantic Breadcrumbs vs. Full Annex @12k (Arm 2b)	15:5	0.0414	0.2069
Semantic Breadcrumbs vs. Breadcr., std. prompt (Arm 3b)	12:4	0.0768	0.3073
Semantic Breadcrumbs vs. Modular Agentic	9:4	0.2668	0.8005
Breadcr., std. prompt (Arm 3b) vs. Full Annex @45k	9:4	0.2668	0.8005
Naive, $k = 12$ (Arm 1b) vs. Naive, $k = 3$	15:16	1.0000	1.0000
Semantic Breadcrumbs vs. No prefix	40:3	<0.0001	<0.0001
Semantic Breadcrumbs vs. Heading path	23:4	0.0003	0.0025
Semantic Breadcrumbs vs. Full breadcrumb, single-pass	21:5	0.0025	0.0175
Semantic Breadcrumbs vs. Document summary	24:4	0.0002	0.0016

Budget does not bridge the gap. Quadrupling the naive retrieval budget (Arm 1b, $k = 12$) doubles the baseline's TCS (23.33 to 51.33) and leaves its accuracy statistically identical: 54.81% against 55.77% (15:16, $p = 1.0$). The additional chunks arrive without their statutory links, so the recovered evidence cannot be composed into a verdict; Semantic Breadcrumbs beats the budget-matched baseline 43:5 (Holm-adjusted $p < 0.0001$). The context-starved-baseline explanation of Table 3 is thereby excluded.

Two budgets for Full Annex. Full Annex at a 12,000-character cap (Arm 2b) scores 81.73%, against 78.85% for its 45,000-character arm, with lower evidence coverage (TCS 87.33 against 94.67). The Wilson intervals of the two arms overlap widely (Table 6), and context length was not varied independently of jurisdiction and annex size, so this comparison does not establish an effect of context length. Semantic Breadcrumbs reaches a numerically higher accuracy than Full Annex at both budgets from 2.6 times less context, and Table 7 reports both paired comparisons.

Route contracts: a suggestive effect. Arm 3b answers with the proposed retrieval and routing but under the baselines' standard prompt, and reaches 83.65%: on par with Full Annex at 45k (78.85%; 9:4, $p = 0.2668$) from 2.6 times less context, with FLAT at 100.00% and

GAP at 86.21%. What the arm loses is hallucination resistance: its TRAP accuracy is 68.97%. With the route contracts, TRAP accuracy is 86.21% and overall accuracy is 7.70 points higher (12:4, $p = 0.0768$; Holm-adjusted $p = 0.3073$). This difference is not statistically significant. We, therefore, report the contribution of the route contracts as an observed effect, concentrated in the TRAP subset, which a larger bank would have to confirm.

4.4. Context-Aware Chunking Baselines and Component Ablations

Table 8 reports the four single-pass conditions of Section 3.3 and the converse ablation next to the two-hop arms. Prefixes help single-pass retrieval: every prefixed condition is above the unprefixed one, which equals the naive baseline's accuracy (55.77%), and the full breadcrumb reaches 75.96%. None approaches the two-hop arms. The best condition assembles complete evidence for 45.33% of the GAP and FLAT questions, against 90.67% behind the two-hop retriever, and the proposed strategy's advantage over each condition is significant after Holm correction over the enlarged family of twelve tests (adjusted $p \leq 0.0175$, Table 7). The comparison, therefore, locates the architecture's contribution in the retrieval stage: what the single-pass conditions lack is not the prefix but the identity-gated second hop that assembles the governing rules.

Two component readings follow. First, the full-breadcrumb condition is the breadcrumb without the two-hop retrieval: under the same prompt, the two-hop arm is 7.69 points higher (83.65% against 75.96%, 17:9 discordant pairs), with complete evidence rising from 26.67% to 90.67%. Second, the heading-path condition isolates the legal-category token at the index level: 73.08% against 75.96% for the full breadcrumb (10:7), a difference this bank cannot distinguish, so the category's measured contribution appears with the second hop, which queries with it. The document summary attains the best single-pass coverage (TCS 70.00%) and does not convert it into the best accuracy.

The converse ablation keeps the two-hop retrieval and removes the breadcrumb: the chunks are embedded and shown without their prefix, and the second hop receives no legal category. The identity gate still anchors, with the same 71 Deep routes as the full retrieval, although eleven of them need the lexical rescue instead of two. Evidence coverage stays high (TCS 86.67%, complete evidence 73.33%), yet accuracy falls to 67.31% (CI [57.8, 75.6]): 16.34 points below the two-hop arm with the breadcrumb (22:5 discordant pairs) and below the breadcrumb-only condition. The abstention repair, which never fired for the full system, fired seven times. Neither component alone approaches their combination, and the two directions separate the mechanisms: the two-hop retrieval completes the evidence, and the breadcrumb lets the generator type a status from it.

4.5. Per-Jurisdiction Results

Table 9 reports accuracy by jurisdiction. Semantic Breadcrumbs reaches 100.00% in four jurisdictions and falls below 70.00% in none, a floor that only Modular Agentic approaches (69.57% in Guinea); the naive baseline and Full Annex both drop to 33.33% (in Gabon and Madagascar, respectively). Each jurisdiction contributes between seven and 23 questions, so these per-jurisdiction figures are indicative. The proposed strategy's own minimum is Congo (70.00%), and its three Congolese misses all belong to the failure taxonomy of Section 4.9.

Table 8. Context-aware chunking baselines (single-pass dense retrieval over the Gold chunks under a 3300-character budget) and the converse ablation (two-hop retrieval over the unprefixed chunks, no category in the second hop, under the full method’s caps), all with the standard prompt. Recall and F1 are macro-averaged over the graded classes; TCS and complete evidence (Compl.) are over the 75 GAP and FLAT questions. The two-hop arms are repeated for reference.

Configuration	Acc. (%)	95% CI	Rec. (%)	F1 (%)	TCS (%)	Compl. (%)	Ctx.
No prefix	55.77	[46.2, 64.9]	56.02	54.38	54.00	26.67	1896
Heading path	73.08	[63.8, 80.7]	72.04	71.75	62.67	37.33	2536
Full breadcrumb	75.96	[66.9, 83.2]	75.15	75.51	56.67	26.67	2733
Document summary	72.12	[62.8, 79.8]	70.75	71.34	70.00	45.33	2843
Two-hop, no breadcrumb (ablation)	67.31	[57.8, 75.6]	66.57	67.39	86.67	73.33	2787
Breadcr., std. prompt (Arm 3b)	83.65	[75.4, 89.5]	81.79	83.35	95.33	90.67	3313
Semantic Breadcrumbs (full)	91.35	[84.4, 95.4]	91.22	91.43	95.33	90.67	3313

Table 9. Accuracy by jurisdiction (DRC = Democratic Republic of the Congo). No jurisdiction falls below 70% under the proposed strategy.

Jurisdiction	Naive	Full Annex	Sem. Breadcr.	Mod. Agentic	<i>n</i>
Cameroon	62.50	75.00	100.00	100.00	8
Congo	60.00	90.00	70.00	90.00	10
Gabon	33.33	100.00	88.89	88.89	9
Guinea	56.52	86.96	91.30	69.57	23
Madagascar	55.56	33.33	100.00	77.78	9
Morocco	52.38	76.19	100.00	100.00	21
DRC	87.50	87.50	75.00	100.00	8
Senegal	55.56	66.67	100.00	77.78	9
Tunisia	42.86	85.71	85.71	85.71	7

4.6. Robustness Across Generators

To separate the architecture from the generator, we reran the four strategies with a further open-weight generator of the same size class, gemma4 (12B), under the unmodified engine, prompts, indexes, and grading, at temperature 0, in a single run. Table 10 reports it next to the main run. The ordering replicates: Semantic Breadcrumbs is the most accurate configuration under both generators, reaching 98.08% (CI [93.3, 99.5]), and converting 96.55% of the TRAP questions into typed abstentions. Naive chunking falls further behind (42.31%, at its unchanged TCS of 23.33%), so the retrieval gap does not close with a stronger generator. Absolute levels rise across the structured strategies and the per-strategy differences narrow; this run varies the generator only, and the embedding model is unchanged (Section 5.4).

Table 10. Robustness across generators: accuracy, macro recall, macro F1, and TCS of the four strategies under the main generator and a further 12B-class open-weight generator (unmodified engine, prompts, and indexes; temperature 0; single run). Retrieval is deterministic and shared, so TCS differs between the generators only for Modular Agentic, whose digest the generator writes.

Strategy	Mistral-Nemo				Gemma4 12B			
	Acc.	Rec.	F1	TCS	Acc.	Rec.	F1	TCS
Naive (Baseline)	55.77	55.74	54.59	23.33	42.31	45.69	39.86	23.33
Full Annex (Bloat)	78.85	77.53	80.22	94.67	97.12	96.91	97.14	94.67
Semantic Breadcr.	91.35	91.22	91.43	95.33	98.08	98.06	98.12	95.33
Modular Agentic	86.54	84.96	86.24	79.33	94.23	93.95	94.28	90.67

4.7. Routing and Repair Diagnostics

The deterministic switch routed the 104 questions as follows: 71 to the Deep route (69 dense-anchored, answered at 94.2% accuracy, plus two lexical rescues), 28 to NoAnchor (92.9%), and five to Verify (60.0%). Four of the five Verify routes are TRAP questions whose designation partially matches a listed species; the two Verify failures are the parametric identity errors of Section 4.9. All 104 proposed-strategy outputs were parsed strictly, with no fallback extraction, and the abstention repair never fired for the proposed strategy; across the two baseline strategies that share the mechanism, it fired five times and corrected three answers. The seven Deep-routed questions below TCS 100 share a corpus pattern: their governing article is referenced in the fetched rules under a category phrase, so the article-number node fails the word-boundary check. Six of the seven were nevertheless answered correctly by direct breadcrumb deduction; the seventh belongs to the override cluster below.

4.8. Qualitative Behavior

Table 11 contrasts the naive baseline and the proposed architecture on one question per failure mode; all answers were generated at $T = 0.0$, and excerpts are translated from the original French outputs. The GAP example shows the guillotine effect end to end: with orphaned fragments of the permit-procedure regime (TCS = 0%), the baseline assimilates an integrally protected species to the authorizable category and grants the permit in principle, while the proposed pipeline anchors the species in its annex and applies the prohibition article. In the TRAP example, the baseline asserts an authorization for a species that does not exist in Guinean law, with no supporting passage in its context; the identity gate never anchors, and the model types ABSENTE. The FLAT example shows the complementary error: the baseline retrieves a logbook-declaration clause and concludes that no return obligation exists, while the two-hop retriever assembles the annex entry with the operative article.

4.9. Failure Taxonomy

Nine questions were answered incorrectly, and all nine failures are generation-side: the anchor decision was correct in every case, and four of the five non-TRAP failures had both required nodes in the context (TCS = 100). We group the nine into four classes.

(a) Breadcrumb override (four questions). On four permit questions from Congo and the Democratic Republic of the Congo, the model asserts that the species is *intégralement protégée* against an Annexe II or Tableau II breadcrumb, in three cases while quoting the correct permit article inside the same answer. On the Sitatunga question, for instance, the answer states “the Sitatunga is an integrally protected class A species” immediately after citing the class B permit rule of Article 6. The typed status contradicts the quoted evidence, a pure assertion error of the 12B generator; the fourth member of the cluster is the question with the missing word-boundary node noted above. Modular Agentic answers exactly these four questions correctly, and Section 5.2 interprets that asymmetry.

(b) Parametric identity (three questions). Two Verify-routed traps fail through invented taxonomy: for the clouded leopard, the model reasons that the species “is also known as *Panthera pardus*” and equates the two; for the eastern gorilla, it names *Gorilla beringei* and then declares it the same species as *Gorilla gorilla*. A third trap fails on the flat route by importing the gorilla into a context that does not contain it.

(c) Double negation (one question). One Tunisian question is misread through its embedded negation (“species *not* listed in Article 1”), inverting the scope of a prohibition.

(d) Rule misuse (one question). One Congolese trap misapplies the specialization rule of Section 2.4 to rationalize a parametric claim that the okapi belongs to class A: the model invokes the rule although the okapi appears nowhere in its context.

No retrieval mechanism can remove these classes: each failure occurs after correct routing, at the assertion step. The next section discusses what they imply for the architecture and its limits.

Table 11. Qualitative comparison on one representative question per failure mode. GT = ground truth; statuses are the typed verdicts of Section 2.4. Answer excerpts are translated from French.

Mode	Question (Original French)	Naive (Baseline)	Semantic Breadcrumbs (Proposed)
GAP	«Un chasseur a-t-il le droit d’obtenir un permis pour abattre une Panthère (Léopard)?» <i>Can a hunter obtain a permit to shoot a panther (leopard)?</i> (Guinea; GT: NON)	TCS = 0%. Retrieves permit-procedure fragments of the category-II regime (Articles 12 and 14) and misapplies them to an integrally protected species: “the hunter can obtain a permit to shoot a Panther (Leopard) if they satisfy the conditions of Article 14”. AUTORISE_SOUS_CONDITION → OUI (incorrect).	TCS = 100%. Hop 1 anchors the panther in ANNEXE V (<i>Espèces Intégralement Protégées</i>); Hop 2 fetches Article 3: “Article 3 expressly prohibits the capture, detention, and killing of specimens of integrally protected species”. INTERDIT → NON (correct).
FLAT	«En cas de capture accidentelle d’une Tortue luth, le pêcheur doit-il obligatoirement la remettre à la mer?» <i>If a leatherback turtle is caught accidentally, must the fisherman return it to the sea?</i> (Morocco; GT: OUI)	Retrieves the logbook-declaration clause and concludes: “there is no explicit obligation to return accidentally caught specimens to the sea”. NON_OBLIGATOIRE → NON (incorrect).	Retrieves the ANNEXE I entry and Article premier: accidentally caught specimens “must be immediately returned to the sea”. OBLIGATOIRE → OUI (correct).
TRAP	«L’abattage du Kangourou roux est-il autorisé avec un permis de petite chasse?» <i>Is shooting the red kangaroo authorized with a small-game permit?</i> (Guinea; GT: ABSENTE ; the species does not exist in Guinean law)	Asserts, with no supporting passage in its context, that “the killing of the red kangaroo is authorized with a small-game permit” under Article 14, assigning a legal status to a species absent from Guinean law. AUTORISE_SOUS_CONDITION (incorrect).	The identity gate finds no anchor (NoAnchor route); the model reports that the red kangaroo is not named in the provided rules. ABSENTE (correct).

5. Discussion

5.1. What the Control Arms Support

The 35.58-point advantage over standard chunking is the result that the evidence establishes, and the control arms indicate where it comes from with unequal strength. The first element is retrieval completeness of *linked* evidence. The budget-matched arm doubles the baseline’s TCS and leaves its accuracy unchanged (15:16), and it still assembles complete evidence for only 28.00% of the questions, against 90.67% for the breadcrumb pipeline; the comparison with this arm remains significant after Holm correction (43:5, adjusted $p < 0.0001$). The context-aware chunking baselines sharpen the point: single-pass retrieval over prefixed chunks assembles complete evidence for at most 45.33% of the questions (Section 4.4), so the prefix alone does not restore the linked evidence that the two-hop retriever assembles.

The second element is the representation handed to the generator. At similar evidence coverage, the breadcrumb pipeline was numerically more accurate than raw annex expansion at both tested budgets, from less context than either arm (3313 characters against 8737 and 6155). Neither difference remains significant after correction, so the arms support an efficiency claim and leave an accuracy advantage unconfirmed. The third element is

deterministic routing under route contracts, which the representation enables: because the breadcrumb states the category explicitly, the gate can verify identity lexically and select a contract. The decomposition arm associates the contracts with 7.70 accuracy points overall and 17.24 points on TRAP questions, a difference that is suggestive at this sample size (12:4, $p = 0.0768$ before correction). The three elements are not equally supported: the first is statistically established, and the other two are observed effects that a larger bank would have to confirm.

5.2. Deterministic Pipelines and the Agentic Alternative

With every shared retrieval upgrade in place, Modular Agentic reaches 86.54%, and the paired comparison with the proposed strategy is not significant (9:4, $p = 0.2668$). The comparison is, therefore, decided on the axes where the strategies do separate, and each favors the deterministic pipeline. Semantic Breadcrumbs abstains better (TRAP 86.21% against 72.41%, 25 against 21 of the 29 traps). It keeps its evidence auditable: the digest paraphrases the anchored row, so the per-node audit of what the generator saw fails on a fifth of the required nodes (TCS 79.33 against 95.33), which matters in a domain where verifying the evidence trail is part of the deliverable. It also costs half the inference: one LLM call per question against two, with no probabilistic parsing step in the control flow.

The override cluster adds one instructive nuance in the other direction. The four questions Modular Agentic wins over the proposed strategy are exactly the four breadcrumb-override failures, and we read the effect as a commitment device: the digest step commits the extracted category to text before the generator reads the rules, which appears to inoculate the final call against re-deriving the species' status from world knowledge. This observation suggests a deterministic hybrid: a post-generation consistency check comparing the asserted status with the breadcrumb category, in the spirit of the abstention repair, would target the same cluster without paying the agentic strategy's cost. We leave this mechanism to future work.

5.3. Topology Coverage

The evaluation supports a claim about list-to-rule reasoning in hierarchical legislation, and it does not support a claim of general superiority for legal RAG. The bank is a targeted stress test: 87 of its 104 questions are GAP or TRAP cases, which probe the two failure modes that the architecture addresses, and the 17 FLAT questions are a control. Within that scope, the nine jurisdictions span deeply nested codes, flat orders, in-article species lists, and one régime whose annex and permit rule live in different instruments, and the proposed strategy never falls below 70.00% in any of them. The per-jurisdiction samples are small, so we read this floor as an indication of stability across topologies and not as an estimate of performance in each jurisdiction. Legal questions that do not hinge on list membership, such as procedural or interpretive questions, lie outside the evaluation.

5.4. Scope and Limitations

Three boundaries delimit the claims. First, the architecture governs evidence preparation and routing, and by design it leaves the final deductive step to the generator. Section 4.9 locates all nine residual failures at that step, after correct routing and, in four of the five non-TRAP cases, with complete evidence in context. The results, therefore, measure how far deterministic data architecture alone carries a 12B open-weight generator; what remains is a property of the generator, not of retrieval. Second, topology and jurisdiction are entangled in the FLAT subset: 16 of its 17 questions come from Morocco, because flat orders are where that topology occurs in this corpus. Within the evaluation, FLAT is accordingly a control showing that the structured pipeline does not degrade retrieval where no category-specific

rule has to be selected, and the reading across topologies rests on the per-jurisdiction results of Table 9. Third, the breadcrumb encodes the membership edge that chunking severs, a minimality choice that delegates distinctions internal to an annex, such as big-game and small-game sub-lists, to the fetched rule text. No residual failure traces to this delegation, although corpora that lean more heavily on sub-list structure would stress it further. Beyond these, the usual scope conditions apply: the corpus is French-language legislation, and each question declares its jurisdiction.

Three further limitations concern the evaluation. First, it relies on a single bank of 104 questions, with no separate development or held-out split, and the bank is small at the level of a subset or a jurisdiction (17 FLAT questions, and seven to 23 questions per jurisdiction). An independently constructed test set would be needed to estimate performance beyond this bank. To our knowledge, no public benchmark covers list-to-rule questions on this legislation, and constructing one was not an aim of this study: each question requires locating the species and its governing provisions in the official texts and validating a legal reading, so extending the bank requires time-consuming and labor-intensive legal expertise. Second, every run uses one embedding model (bge-m3), the main evaluation uses one generator (mistral-nemo, 12B) at temperature 0, and the robustness run of Section 4.6 is a single run with one further generator. It replicates the ordering of the strategies; other embedding models remain to be tested. Third, the errors of the ingestion stages were not quantified separately. On the bank, the designation of the questioned species reached the context for all 75 GAP and FLAT questions (Table 4), and no residual failure traces to a parsing error; the effect of parsing errors on answers beyond the bank remains to be measured.

6. Conclusions

Standard RAG architectures are not well-suited for hierarchical legal documents, and this paper's evidence locates the failure at ingestion: chunking severs the edges that connect listed entities to their governing rules, and a fourfold retrieval budget did not re-link them. We presented Semantic Breadcrumbs, a deterministic framework that preserves those edges by injecting each chunk's hierarchy path and legal category into the text before vectorization, and that keeps inference deterministic through an identity-gated two-hop retriever, citation-priority rule fetching, route-specific prompt contracts, and a typed verdict interface.

We evaluated these claims on 104 citation-verified list-to-rule questions across nine jurisdictions, under budget-matched and decomposition control arms and exact paired tests with Holm correction. Semantic Breadcrumbs reached 91.35% accuracy and a 91.43% macro F1-score, against 55.77% and 54.59% for standard chunking, a difference that remains significant after correction. It was also more accurate than exhaustive annex expansion while consuming 2.6 times less context, a difference that does not remain significant after correction, and converted 86.21% of hallucination traps into typed abstentions. It was significantly more accurate than four context-aware chunking baselines at a matched context budget, and the ordering of the strategies held under a further generator.

The control arms qualify the architectural lesson. Retrieval completeness of linked evidence is the established factor, whereas the compact representation and the route contracts are associated with further gains that this bank is too small to confirm. Within list-to-rule questions on hierarchical legislation, and for the embedding model and the generators tested, these findings are consistent with the central reading of this study: legal faithfulness is first a data architecture problem, since the most accurate configuration of the evaluation was a lightweight, locally served model over deterministically prepared context, ahead of the same model over more context or behind an additional generative stage.

Author Contributions: Conceptualization, Y.A.M.; methodology, Y.A.M.; software, Y.A.M.; validation, Y.A.M., M.B. and J.Z.; formal analysis, Y.A.M.; investigation, Y.A.M.; data curation, Y.A.M.; writing—original draft preparation, Y.A.M.; writing—review and editing, Y.A.M., M.B. and J.Z.; visualization, Y.A.M.; supervision, M.B. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the AIME project (Artificial Intelligence for Marine Ecosystems) and by the ISblue project, Interdisciplinary Graduate School for the Blue Planet (ANR-17-EURE-0015) and co-funded by a grant from the French government under the program “*Investissements d’Avenir*” embedded in France 2030.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in Zenodo at <https://doi.org/10.5281/zenodo.23441282>: the 104-question test bank with its citation-verified ground truth and the verbatim source extracts of each reference answer, the per-question evaluation outputs of every evaluated configuration, including the context-aware chunking baselines, the component ablation, and the robustness run, the ingested corpus chunks of the nine evaluated jurisdictions, and the aggregated result tables. The complete experimental configuration is given in Appendix A, the prompts are printed verbatim in Appendix B, and the ontology profiles in Appendix C. The legislative texts of the corpus are public documents of their respective jurisdictions, retrieved from FAOLEX.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RAG	Retrieval-Augmented Generation
LLM	Large language model
NLP	Natural language processing
OCR	Optical character recognition
DAG	Directed acyclic graph
TCS	Topological Completeness Score
CI	Confidence interval
CITES	Convention on International Trade in Endangered Species
DRC	Democratic Republic of the Congo
YAML	YAML Ain’t Markup Language

Appendix A. Experimental Configuration

Table A1 gathers the complete configuration of the reported runs.

Table A1. Configuration of the reported runs. Character counts refer to the assembled context; “umbrella” denotes an article or an annex with all of its chunks.

Component	Setting
Source documents	Official PDFs of 14 instruments from nine jurisdictions, retrieved from FAOLEX [31]
Bronze layer	MinerU document parser [32], output in structured Markdown
Silver layer	Two YAML profiles (Profile_Deep_Hierarchy, Profile_Flat_Articles) executed by a stateful memory stack; levels BOOK, TITLE, CHAPTER, SECTION, SUBSECTION; umbrella markers for articles and annexes
Gold layer	Breadcrumb prefix [Contexte: [jurisdiction hierarchy levels > umbrella (legal category)]]; instrument-level namespacing; in-article list detection (at least three list rows covering at least 30% of the umbrella); category inheritance from the list header; 3397 chunks, 2183 anchors

Table A1. Cont.

Component	Setting
Naive chunks	Recursive character splitting, chunk size 250, overlap 20 (langchain-text-splitters 0.3.11); 3105 chunks
Embedding model	bge-m3, 16-bit floating-point weights, 567M parameters, served by Ollama 0.34.0
Vector store	ChromaDB 1.0.20, one collection per track, every query filtered by jurisdiction
Generator	mistral-nemo (Mistral-Nemo-Instruct-2407), 12B parameters, 4-bit quantization (Q4_0), served by Ollama 0.34.0; temperature 0, generation limit 1200 tokens
Hardware	NVIDIA RTX A2000 laptop GPU, 8 GB
Naive retrieval	Dense top-3 chunks (top-12 for Arm 1b)
Hop 1	Dense top-3 over the anchors of the jurisdiction; identity gate with full, partial, or no match; one lexical scan of all anchors of the jurisdiction when no dense candidate matches
Hop 2	Citation index: at most two umbrellas, with one further step of article cross-reference; dense query formed by the question and the legal category of the anchor, top-5 chunks outside annexes, grouped by umbrella; citation and dense results interleaved; at most three umbrellas of at most 3500 characters each
NoAnchor fallback	Dense top-3 chunks outside annexes and one lexical umbrella match, at most three umbrellas
Context caps	9000 characters (Naive, Semantic Breadcrumbs, Modular Agentic, fallback); 45,000 characters (Full Annex); 12,000 characters (Arm 2b)
Modular Agentic	One additional LLM call that extracts the anchored row into a JSON digest
Grading	Deterministic mapping of the typed status through the oui_si annotation of each question
Baseline conditions	Four single-pass conditions over the Gold chunks (Section 4.4): prefix none, heading path, full breadcrumb, or a document-level summary of about 150 characters generated once per instrument; dense top-20 under a 3300-character budget; standard prompt; served by Ollama
Converse ablation	Two-hop retrieval, routing, and abstention repair of the full method over the unprefixed chunks; no legal category in the Hop-2 query; standard prompt; served by Ollama
Robustness run	gemma4:12b under the unmodified engine, prompts, and indexes; temperature 0

Appendix B. Prompts

This appendix prints the four prompts of the evaluation verbatim, in French, as they appear in the evaluation engine. The prompts share a status vocabulary, described in Section 2.4, and the final-line instruction below, inserted at the markers {STATUS_VOCAB} and {FINAL_LINE}; at inference, {context} and {query} receive the assembled context and the question. The standard prompt serves the naive configurations, Full Annex, Modular Agentic, Arm 3b, and the baseline conditions; the three route prompts serve the proposed strategy, and the repair note is appended on the single re-ask of the abstention repair.

STATUS_VOCAB (shared status vocabulary)

- INTERDIT : l'acte visé est interdit.
- AUTORISE_SOUS_CONDITION : le texte prévoit une voie ORDINAIRE pour cet acte : permis , autorisation préalable.
- AUTORISE : l'acte est libre, sans autorisation requise.
- OBLIGATOIRE / NON_OBLIGATOIRE : si une action DOIT être accomplie , réponds par OBLIGATOIRE ou NON_OBLIGATOIRE.
- ABSENTE : l'espèce ou l'objet exact de la question n'apparaît pas dans le contexte fourni.

FINAL_LINE (shared final-line instruction)

Termine IMPÉRATIVEMENT ta réponse par une ligne exactement de la forme :
Statut Juridique : INTERDIT (ou AUTORISE_SOUS_CONDITION, AUTORISE, OBLIGATOIRE, NON_OBLIGATOIRE, ou ABSENTE)

PROMPT_DEEP (Deep route)

Tu es un expert juridique. Utilise UNIQUEMENT le contexte fourni.

RÈGLE ABSOLUE : La catégorie indiquée dans la balise [Contexte: ...] de l'<Entite_Definie> est la vérité juridique absolue.

RÈGLE DE DÉDUCTION : Les règles des <Regles> s'appliquent par CATÉGORIE. Une règle visant une catégorie s'applique à CHAQUE espèce listé dans cette catégorie, même si la règle ne nomme pas cet espèce individuellement. L'espèce de la question figure dans l'<Entite_Definie> : tu DOIS donc conclure sur un statut à partir de sa catégorie et des règles de cette catégorie

{STATUS_VOCAB}

Structure OBLIGATOIREMENT ta réponse selon ce format exact :

- Analyse du statut : [Énonce le statut exact de l'animal d'après sa catégorie]
- Règle applicable : [Cite les règles pertinentes dans les <Regles>]
- Conclusion : [Explique le statut de l'acte visé par la question]
- Statut Juridique : [UN SEUL mot du vocabulaire ci-dessus]

{FINAL_LINE}

Contexte:

{context}

Question: {query}

PROMPT_VERIFY (Verify route)

Tu es un expert juridique. Utilise UNIQUEMENT le contexte fourni.

RÈGLE D'IDENTITÉ (PRIORITAIRE) : L'<Avertissement_Identite> signale que l'entité retrouvée ne correspond que partiellement au nom demandé. Compare rigoureusement le nom complet de l'animal de la question avec l'animal nommé dans l'<Entite_Definie>

Cette vérification se fait UNIQUEMENT sur les noms présents dans le contexte ; jamais d'après tes connaissances générales (CITES, listes mondiales, statut de l'espèce ailleurs dans le monde). Si l'espèce est DIFFÉRENTE, le statut est ABSENTE même si tu crois savoir qu'elle est protégée ailleurs.

RÈGLE DE DÉDUCTION : Si et seulement si l'identité est confirmée, les règles s'appliquent par catégorie à chaque espèce listé, même non nommé individuellement.

{STATUS_VOCAB}

Structure OBLIGATOIREMENT ta réponse selon ce format exact :

- Vérification d'identité : [même espèce ou espèce différente, et pourquoi]

- Règle applicable : [Cite les règles pertinentes, ou indique qu'aucune ne s'applique]
 - Conclusion : [Explique le statut de l'acte visé par la question]
 - Statut Juridique : [UN SEUL mot du vocabulaire ci-dessus]
- {FINAL_LINE}

Contexte:

{context}

Question: {query}

PROMPT_FLAT (NoAnchor route)

Tu es un expert juridique. Utilise UNIQUEMENT le contexte fourni dans les <Règles>.

RÈGLE D'IDENTITÉ : Si l'espèce exact de la question n'apparaît nulle part dans le contexte, le Statut Juridique est ABSENTE .

RÈGLE DE DÉCISION : Si l'espèce de la question APPARAÎT dans le contexte ; tu DOIS conclure sur un statut d'après les règles fournies ; ABSENTE n'est pas permis dans ce cas .

{STATUS_VOCAB}

Structure OBLIGATOIREMENT ta réponse selon ce format exact :

- Analyse : [Résume le cadre légal exposé]
 - Règle applicable : [Cite la ou les règles exactes]
 - Conclusion : [Explique le statut de l'acte visé par la question]
 - Statut Juridique : [UN SEUL mot du vocabulaire ci-dessus]
- {FINAL_LINE}

Contexte:

{context}

Question: {query}

PROMPT_STANDARD (all other configurations)

Tu es un expert juridique. Utilise UNIQUEMENT le contexte fourni.

{STATUS_VOCAB}

Structure OBLIGATOIREMENT ta réponse ainsi :

- Analyse du statut : [statut]
- Règle applicable : [article]
- Conclusion : [réponse]
- Statut Juridique : [UN SEUL mot du vocabulaire ci-dessus]

Si l'information sur l'animal exact de la question est introuvable dans le contexte, le Statut Juridique doit être ABSENTE.

{FINAL_LINE}

Contexte:

{context}

Question: {query}

REPAIR_NOTE (abstention repair)

CONTRAINTE SUPPLÉMENTAIRE : L'espèce de la question est PRÉSENTE dans le contexte (ancrage vérifiée). Le statut ABSENTE est donc EXCLU. Réponds avec l'un des statuts : INTERDIT, AUTORISE_SOUS_CONDITION, AUTORISE, OBLIGATOIRE, NON_OBLIGATOIRE.

Appendix C. Ontology Profiles

This appendix prints the two YAML profiles of the ingestion ontology verbatim, as the Silver layer executes them: `Profile_Flat_Articles` for flat orders and `Profile_Deep_Hierarchy` for nested codes. Each entry is a regular expression: `hierarchy_levels` defines the structural levels that the memory stack tracks, and `umbrella_markers` delimits the article, annex, and end-of-document segments. A configuration header, not printed here, maps each source document to one of the two profiles (Table A1). The profiles contain no species names and no question-specific patterns: their expressions name the structural conventions of legislative drafting and the signature formulas of the instruments. Both profiles and the normalization passes of Section 2.2 were settled from the corpus structure and fixed before the evaluation; none was changed in response to the outcome of any bank question.

profiles:

```
Profile_Flat_Articles:
  hierarchy_levels: []
  umbrella_markers:
    - '(?:Article|Art\.\?)\s+(?:premier|PREMIER| [0-9]+) '
    - 'ANNEXE\s+[IVX0-9]+'
    - 'Annexe\s+[IVX0-9]+'
    - 'Fait\s+(?:à|le) '

Profile_Deep_Hierarchy:
  hierarchy_levels:
    - 'LIVRE(?:\s+(?: [IVX0-9]+|PREMIER))?\s|Livres(?:\s+(?: [IVX0-9]+|premier))?'
    - 'TITRE\s+(?: [IVX0-9]+|PREMIER)\s|Titre\s+(?: [IVX0-9]+|premier) '
    - 'CHAPITRE\s+(?: [IVX0-9]+|PREMIER)\s|Chapitre\s+(?: [IVX0-9]+|premier) '
    - 'SECTION\s+(?: [IVX0-9]+|PREMIER[EÈ]RE)\s|Section\s+(?: [IVX0-9]+|premi[eè]re) '
    - 'SOUS-SECTION(?:\s+(?: [IVX0-9]+|PREMIER[EÈ]RE))?\s|Sous-section(?:\s+(?: [IVX0-9]+|premi[eè]re))?'
    - 'PARAGRAPHE(?:\s+(?: [IVX0-9]+|PREMIER))?\s|Paragraphe(?:\s+(?: [IVX0-9]+|premier))?'
  umbrella_markers:
    - '(?:Article|Art\.\?)\s+(?:premier|PREMIER| [0-9]+) '
    - 'ANNEXE\s+[IVX0-9]+'
    - 'Annexe\s+[IVX0-9]+'
    - 'Fait\s+(?:à|le) '
```

References

1. Gupta, S.; Agrawal, V.; Negi, Y.; Karunanithi, S.; Balakrishnan, A. Reducing Hallucinations in Legal AI: A Retrieval Augmented Generation-Based Model for Accurate Legal Guidance. *IEEE Access* **2026**, *14*, 44451–44463. [\[CrossRef\]](#)
2. Guha, N.; Nyarko, J.; Ho, D.E.; Ré, C.; Chilton, A.; Narayana, A.; Chohlas-Wood, A.; Peters, A.; Waldon, B.; Rockmore, D.N.; et al. LEGALBENCH: A collaboratively built benchmark for measuring legal reasoning in large language models. In Proceedings of the 37th International Conference on Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023.
3. Al Mouatamid, Y.; Bonnin, M.; Zahir, J. Elevating Legal Understanding: A Dedicated Instruction Dataset and an Optimized LLM for Legal Information Extraction to Support the Juridical Interpretation of Marine Environmental Law. In *2025 International Conference on Intelligent Systems: Theories and Applications (SITA)*; IEEE: New York, NY, USA, 2025; pp. 1–6. [\[CrossRef\]](#)
4. Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Trans. Inf. Syst.* **2025**, *43*, 1–55. [\[CrossRef\]](#)

5. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Proceedings of the 34th International Conference on Neural Information Processing Systems, Virtual, 6–12 December 2020.
6. Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, M.; Wang, H. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv* **2024**, arXiv:2312.10997.
7. Reuter, M.; Lingenberg, T.; Liepina, R.; Lagioia, F.; Lippi, M.; Sartor, G.; Passerini, A.; Sayin, B. Towards Reliable Retrieval in RAG Systems for Large Legal Datasets. In *Proceedings of the Natural Legal Language Processing Workshop 2025, Suzhou, China, 8 November 2025*; Aletras, N., Chalkidis, I., Barrett, L., Goanță, C., Preoțiuc-Pietro, D., Spanakis, G., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 17–30. [[CrossRef](#)]
8. Edge, D.; Trinh, H.; Cheng, N.; Bradley, J.; Chao, A.; Mody, A.; Truitt, S.; Metropolitansky, D.; Ness, R.O.; Larson, J. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. *arXiv* **2024**, arXiv:2404.16130.
9. Dassen, M.; Kotula, R.; Murray, K.; Yates, A.; Lawrie, D.; Kayi, E.; Mayfield, J.; Duh, K. FACTUM: Mechanistic Detection of Citation Hallucination in Long-Form RAG. In *Proceedings of the Advances in Information Retrieval: 48th European Conference on Information Retrieval, ECIR 2026, Delft, The Netherlands, 29 March–2 April 2026*; Proceedings, Part I; Springer Nature: Cham, Switzerland, 2026; pp. 272–288. [[CrossRef](#)]
10. Al Mouatamid, Y.; Bonnin, M.; Zahir, J. From Norms to Indicators (N2I-RAG): An Agentic Retrieval-Augmented Generation Framework for Legal Indicator Computation. *arXiv* **2026**, arXiv:2605.26926.
11. Liu, N.F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; Liang, P. Lost in the Middle: How Language Models Use Long Contexts. *Trans. Assoc. Comput. Linguist.* **2024**, *12*, 157–173. [[CrossRef](#)]
12. Xu, P.; Ping, W.; Wu, X.; McAfee, L.; Zhu, C.; Liu, Z.; Subramanian, S.; Bakhturina, E.; Shoeybi, M.; Catanzaro, B. Retrieval meets Long Context Large Language Models. *arXiv* **2024**, arXiv:2310.03025.
13. Jin, B.; Yoon, J.; Han, J.; Arik, S.O. Long-Context LLMs Meet RAG: Overcoming Challenges for Long Inputs in RAG. *arXiv* **2024**, arXiv:2410.05983.
14. Chen, Z.; Zhang, Q.; Xiang, Z.; Wei, Z.; Gao, L.; Huang, X.; Zhang, Z.; Su, J. LegalGraphRAG: Multi-Agent Graph Retrieval-Augmented Generation for Reliable Legal Reasoning. *arXiv* **2026**, arXiv:2605.28120.
15. Oliveira, V.T.d.; Silva, D.O.d.; Souza, M.d.A.; Lima, M.R.; Oliveira, S.S.T.d.; Rosa, T.C. Retrieval-Augmented Generation and Knowledge Graphs in Portuguese-Language Legal Documents. In *Proceedings of the 17th International Conference on Computational Processing of Portuguese (PROPOR 2026)-Volume 1, Salvador, Brazil, 13–16 April 2026*; Souza, M., de Dios-Flores, I., Santos, D., Freitas, L., Souza, J.W.d.C., Ribeiro, E., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2026; pp. 1–10.
16. Anthropic. Introducing Contextual Retrieval. Published 19 September 2024. 2024. Available online: <https://www.anthropic.com/engineering/contextual-retrieval> (accessed on 11 October 2026).
17. Günther, M.; Mohr, I.; Williams, D.J.; Wang, B.; Xiao, H. Late Chunking: Contextual Chunk Embeddings Using Long-Context Embedding Models. *arXiv* **2024**, arXiv:2409.04701. [[CrossRef](#)]
18. Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; Yih, W.t. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020*; Webber, B., Cohn, T., He, Y., Liu, Y., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 6769–6781. [[CrossRef](#)]
19. Chen, X.; Gao, P.; Song, J.; Chen, X.; Tan, X. HiQA: A Hierarchical Contextual Augmentation RAG for Multi-Documents QA. *arXiv* **2024**, arXiv:2402.01767. [[CrossRef](#)]
20. Yousuf, R.B.; Xu, S.; Sharma, M.; Neeser, A.; Latimer, C.; Ramakrishnan, N. Utilizing Metadata for Better Retrieval-Augmented Generation. In *Proceedings of the Advances in Information Retrieval: 48th European Conference on Information Retrieval, ECIR 2026, Delft, The Netherlands, 29 March–2 April 2026*; Lecture Notes in Computer Science; Proceedings, Part I; Springer Nature: Cham, Switzerland, 2026; Volume 16483, pp. 305–319. [[CrossRef](#)]
21. Livathinos, N.; Auer, C.; Lysak, M.; Nassar, A.; Dolfi, M.; Vagenas, P.; Berrospi Ramis, C.; Omenetti, M.; Dinkla, K.; Kim, Y.; et al. Docling: An Efficient Open-Source Toolkit for AI-driven Document Conversion. *arXiv* **2025**, arXiv:2501.17887. [[CrossRef](#)]
22. Docling Project. Hybrid Chunking. Docling Documentation. Available online: https://docling-project.github.io/docling/_generated/examples/hybrid_chunking/ (accessed on 11 October 2026).
23. Ferraris, A.F.; Audrito, D.; Siragusa, G.; Piovano, A. Legal Chunking: Evaluating Methods for Effective Legal Text Retrieval. In *Legal Knowledge and Information Systems (JURIX 2024)*; Frontiers in Artificial Intelligence and Applications; IOS Press: Amsterdam, The Netherlands, 2024; Volume 395, pp. 275–281. [[CrossRef](#)]
24. Prior, M.; Milanova, N.; Schultz, A. Chunking German Legal Code. *arXiv* **2026**, arXiv:2605.19806. [[CrossRef](#)]
25. Louis, A.; van Dijck, G.; Spanakis, G. Finding the Law: Enhancing Statutory Article Retrieval via Graph Neural Networks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, Dubrovnik, Croatia, 2–6 May 2023*; Vlachos, A., Augenstein, I., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 2761–2776. [[CrossRef](#)]

26. Chen, J.; Xiao, S.; Zhang, P.; Luo, K.; Lian, D.; Liu, Z. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. *arXiv* **2024**, arXiv:2402.03216.
27. Mistral AI. *Mistral NeMo*; 12B open-weight model, released July 2024; Mistral AI: Paris, France, 2024.
28. Wilson, E.B. Probable Inference, the Law of Succession, and Statistical Inference. *J. Am. Stat. Assoc.* **1927**, *22*, 209–212. [[CrossRef](#)]
29. McNemar, Q. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika* **1947**, *12*, 153–157. [[CrossRef](#)] [[PubMed](#)]
30. Holm, S. A Simple Sequentially Rejective Multiple Test Procedure. *Scand. J. Stat.* **1979**, *6*, 65–70.
31. Food and Agriculture Organization of the United Nations. FAOLEX Database. Available online: <https://www.fao.org/faolex/en/> (accessed on 15 July 2026).
32. Wang, B.; Xu, C.; Zhao, X.; Ouyang, L.; Wu, F.; Zhao, Z.; Xu, R.; Liu, K.; Qu, Y.; Shang, F.; et al. MinerU: An Open-Source Solution for Precise Document Content Extraction. *arXiv* **2024**, arXiv:2409.18839. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.