

Applying Ontology to the Architectural Design Domain with LLMs

Atsuya Uryu, Incheon Paik
School of Computer Science and
Engineering,
University of Aizu
Aizu-Wakamatsu city, Japan
s1300027@u-aizu.ac.jp
paikic@u-aizu.ac.jp

Abstract

Reliable knowledge reuse in the architecture domain is hindered by fragmented terminology, spatial relations, and regulatory clauses. We present an ontology-oriented, retrieval-augmented generation (RAG) framework that establishes a foundation of documents and instances within a lightweight building topology vocabulary (BOT). Our pipeline normalizes Zone/Element/Interface concepts, performs ontology-aligned chunking and entity linking to dereferenceable URIs, and constrains Large Language Model (LLM) generation with citation-required prompts and post-hoc consistency checks. Our method outperforms two baselines (LLM-only and ontology-agnostic RAG) in retrieval effectiveness, answer accuracy, and evidence consistency across three tasks: regulatory Q&A, specification recommendation, and ontology-grounded knowledge search. For reproducibility, we release prompts, hyperparameters, and evaluation scripts.

Keywords: Ontology, RAG, LLM

1. Introduction

The reasoning capabilities of Large Language Models (LLMs) have emerged as one of the most remarkable technological achievements in the field of artificial intelligence. Their applications already extend across a wide range of domains such as natural language processing, machine translation, document summarization, and question answering, where LLMs have demonstrated performance that surpasses conventional statistical and machine learning approaches. When fine-tuned on domain-specific data, they can even outperform human-level reasoning in certain contexts, suggesting their potential to evolve beyond mere language models into general-purpose reasoning and decision-support systems.

Despite these successes, LLMs still face significant limitations in solving problems that require deep, domain-specific knowledge. Since most LLMs are trained on massive amounts of general-purpose data, they often fail to capture the intricate contextual and regulatory nuances essential for specialized domains such as medicine, law, or architectural design. In such fields, knowledge is governed by formal standards, logical dependencies, and domain-specific constraints that are not easily learned from raw text alone. Consequently, without an explicit mechanism to integrate structured domain knowledge, the applicability of LLMs remains constrained. Recent studies have investigated

retrieval-augmented generation (RAG) for knowledge-intensive NLP tasks [1] and analyzed hallucination and factuality issues in large language models [2]. These works highlight the importance of integrating verifiable external knowledge sources to improve factual consistency and domain reliability.

To address this challenge, one of the most prominent approaches proposed in recent years is Retrieval-Augmented Generation (RAG). RAG enhances the factual accuracy of model outputs by retrieving relevant external documents or knowledge snippets and feeding them into the model at inference time. While this approach has shown improvements in general information retrieval tasks, its effectiveness diminishes when dealing with domains that require highly structured, semantically interrelated knowledge. Purely unstructured retrieval does not provide the kind of ontological consistency or reasoning transparency that many professional and regulatory fields demand.

Therefore, this study introduces an ontology-based knowledge representation framework to complement and strengthen LLM reasoning. Ontologies provide a formal and hierarchical representation of domain concepts, attributes, and their interrelationships. This enables semantic interoperability, logical reasoning, and explainability — features that are crucial in professional environments where the correctness and traceability of generated content must be verifiable. Building on prior works such as the Building Topology Ontology (BOT) [3] and the Core Ontology for Whole Life Costing in Construction Projects [4], our research leverages ontology-driven structures to supply LLMs with interpretable, structured knowledge that goes beyond surface-level textual context.

Specifically, this study focuses on the architectural design domain, a complex field that integrates diverse elements such as spatial configuration, legal constraints, safety regulations, and aesthetic considerations. Within this domain, unstructured reasoning by LLMs alone often leads to inconsistent or contextually invalid results. By modeling architectural knowledge as an ontology and integrating it with an LLM through an ontology-oriented RAG framework, we aim to achieve both higher factual accuracy and contextual validity. This approach facilitates structured reasoning over building entities (spaces, elements, and interfaces) anchored in the BOT framework, with explicit references to regulatory clauses, product specifications, and design constraints.

The remainder of this paper is structured as follows. Section 2 presents the fundamental characteristics of ontology-based knowledge representation and its expression in natural

language style. Section 3 describes the process of constructing the domain-specific ontology for architectural design. Section 4 introduces the proposed integration methodology combining the ontology with an LLM, including data acquisition, retrieval, and generation pipelines. Section 5 reports on experimental evaluations that assess the performance and factual consistency of the proposed approach. Finally, Section 6 discusses the implications and contributions of this research toward achieving explainable and domain-aware reasoning in LLMs for the architectural domain.

This study thus contributes a novel framework for ontology-augmented LLM reasoning, bridging the gap between symbolic domain modeling and generative AI. It aims to establish a foundation for structured, explainable, and verifiable knowledge generation in architecture and other knowledge-intensive fields.

2. Background and Related Work

Previous research by Paik [11] introduced an ontology-rule-based integration framework with large language models, showing the feasibility of combining symbolic reasoning and neural generation.

Building on this foundation, our work adapts and extends the approach to a domain-specific architectural context.

Building Ontologies: BOT offers lightweight classes (Site/Building/Storey/Space/Element/Interface) and relations (containment, adjacency, crossing), designed for loose coupling and alignment [3], [5]. In contrast, ifcOWL mirrors IFC and affords coverage at the cost of complexity; BOT is often used as a pragmatic upper ontology interfacing with ifcOWL subsets [4], [6].

RAG and Controlled Generation: Retrieval-augmented generation mitigates hallucination by injecting external

evidence; performance hinges on chunking, metadata filtering, and prompts that enforce citation and answer structure [7], [8]. Ontologies naturally supply canonical labels, URIs, and constraints that shape retrieval distribution and stabilize generation [9].

3. Ontology: Definition, Structure, and Role in This study

3.1 Definition and Rationale

An ontology is a machine-readable, formal specification of the key concepts of a domain and the relations among them, enabling consistent data integration, reasoning, and reuse across heterogeneous sources. In the architecture domain, such a shared conceptualization aligns spatial, element, and boundary notions so that retrieval and generation can be anchored to stable identifiers (URIs) rather than surface strings. This study adopts a lightweight, topology-first stance—centering on Zone/Element/Interface—so that downstream pipelines (indexing, retrieval, generation) can exploit structural priors and reduce ambiguity in evidence grounding.

3.2 Core Building Blocks

We use the following components throughout the paper:

- **Classes and Hierarchies:** Minimal yet extensible classes such as Site, Building, Storey, Space, Element, and Interface,

with containment and adjacency relations. These provide canonical anchors for text and instance data.

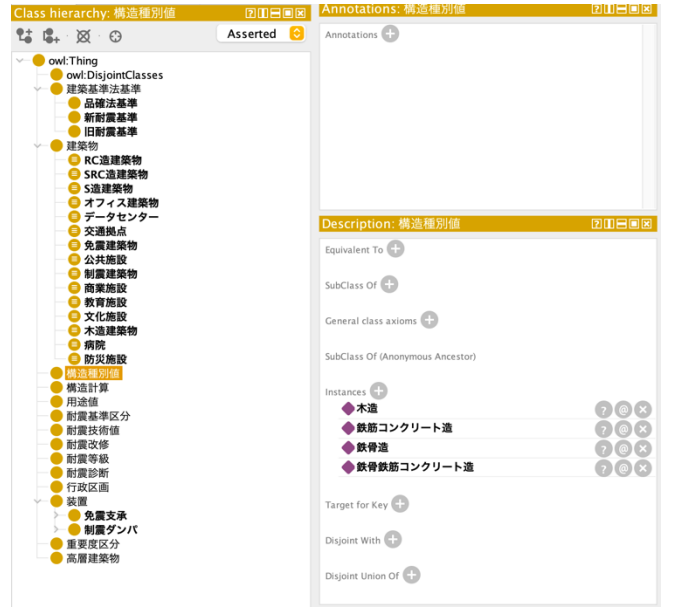


Figure 1: Class hierarchy of the Seismic Architecture Ontology implemented in Protégé.

- **Properties and Constraints:** Relations (e.g., containment/adjacency/crossing) and attributes typed by domain/range. In our pipeline, these constraints are later checked post hoc to detect role/type mismatches.
- Individuals (Instances). Dereferenceable URIs are assigned to spaces/elements/interfaces and linked to documentary evidence (regulatory clauses, product specs), enabling precise citations at answer time.

- **Validation Rules:** Lightweight shape and type checks (e.g., SHACL-like constraints) are used to flag inconsistencies and to trigger selective regeneration in the LLM output.

3.3 Language and Reasoning in Practice

We represent knowledge in OWL/Turtle with dereferenceable URIs and expose it to the pipeline via metadata and (optionally) SPARQL/filters at retrieval time. This choice keeps the model lightweight and web-friendly while still permitting structural checks after generation. The practical benefit is that ontology-aligned chunking and entity linking (URI + label + type) allow hybrid retrieval (dense + lexical + metadata) to focus on the right neighborhood of evidence.[5],[6]

3.4 BOT as a Lightweight Upper Ontology for AEC

The Building Topology Ontology (BOT) provides a minimal, extension-oriented vocabulary around Zone–Element–Interface, with a Site/Building/Storey/Space hierarchy and relations such as containment/adjacency/crossing; it also supports references to 3D models. In contrast to ifcOWL—which mirrors IFC and can be heavy for reasoning—BOT favors web-scale interoperability and pragmatic development, making it suitable as an upper ontology for our pipeline.

4. Proposed Method

4.1 Architecture

The proposed system integrates ontology-based symbolic reasoning with large language model (LLM) generation to enhance factual grounding and explainability in domain-specific recommendations.

It follows a four-layer architecture that transforms structured domain knowledge into natural language reasoning while maintaining bidirectional interaction between symbolic and generative components.

Knowledge Layer: The OWL ontology stores domain-specific concepts—such as buildings, devices, and seismic technologies—together with their relationships and SWRL rules.

This layer provides the formal semantic foundation for all subsequent reasoning and generation processes.

Reasoning Layer: Semantic inference is performed through SPARQL queries and SWRL rules to derive new knowledge from the ontology.

Retrieved entities and relations are filtered and structured into ontology-aligned knowledge chunks to ensure semantic precision.

Language Model Layer: The LLM (e.g., GPT-4 or Gemini) operates within a Retrieval-Augmented Generation (RAG) pipeline, receiving both user queries and ontology-derived evidence.

This allows the model to generate context-aware, citation-grounded responses consistent with domain constraints.

Application Layer: The user interface presents validated outputs, including recommendations and explanations for tasks such as equipment selection and design validation.

A post-generation validation mechanism checks logical consistency and ontology alignment before presenting the results.

This layered design ensures semantic interpretability, logical consistency, and scalability across diverse industrial and architectural domains.

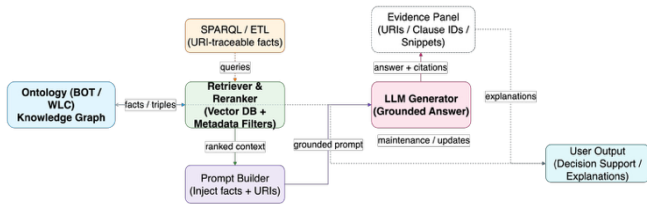


Figure 2: Presents the architecture of the proposed ontology-LLM integrated framework.

4.2 Ontology matters for RAG & LLMs

Retrieval-augmented generation (RAG) benefits when the query, chunks, and answers can be tied to canonical entities and relations rather than free text. In our system, the ontology informs (i) chunk boundaries (split at entity/clause levels), (ii) entity linking (`@entity{URI, label, type}`) at retrieval time, and (iii) citation-required prompting (≥ 1 evidence URI per

conclusion), followed by post-hoc validation. This combination improves retrieval effectiveness, answer accuracy, and evidence consistency compared with ontology-agnostic setups.

4.3.1 Design Principles

Prompts (i) expose ontology anchors (URI, label, type), (ii) require inline citations (≥ 1 evidence URI per claim), and (iii) force explicit uncertainty via “Insufficient evidence”. Outputs follow a fixed, validator-friendly schema. Answer schema (logical): `task_id; conclusions[{claim, evidence[URIs], entities[{uri,label,type}]}]; unresolved[...]`. Task prompts.

- **T1 (Regulatory Q&A):** Find applicable clause(s), thresholds, exceptions; cite URIs next to each conclusion; note conflicts.

- **T2 (Specification Recommendation):** Recommend element(s) and boundary conditions for a space; cite ≥ 1 URI (code/spec) and reference involved Space/Element/Interface.

- **T3 (Knowledge Search):** Produce a short, BOT-grounded synthesis with inline URIs only.

4.3.2 Hallucination control

Ontology-aligned chunking + entity linking + citation-required prompting + post-hoc validation (type/role checks; containment vs adjacency). Unsupported or inconsistent claims are flagged and optionally regenerated.[7]

5. Experiment

5.1 System Overview

Our system comprises three layers.

Knowledge Layer: An OWL knowledge base anchored in BOT; instances for spaces/elements/interfaces; links to regulatory clauses and product specs via URIs.

Acquisition Layer: Ontology-oriented chunking (split on entity/clause boundaries), entity linking, and hybrid retrieval (dense + BM25 + metadata/SPARQL filters).

Generation Layer: Constraint-guided prompts requiring citations (≥ 1 URI per claim) and a post-hoc validator (type/role checks, SHACL-like rules) to flag inconsistencies.

5.2 Ontology-Oriented Chunking

We align chunk boundaries with BOT entities (e.g., Space, Element) and statute clauses. Each chunk carries metadata: `{uri, label, type, parent, relations}`.

This increases the probability that required context co-occurs within top-k retrieval, improving both Recall@k and evidence precision.

5.3 Entity Linking and Prompting

Each retrieved span is annotated as `@entity{URI, label, type}`. Prompts instruct the LLM to: (i) answer with a fixed schema, (ii) cite URIs next to each conclusion, (iii) state “Insufficient evidence” if no supporting URI is available, and (iv) avoid inventing identifiers. The validator rejects outputs that assign element-only properties to spaces (and vice versa) or violate declared relations (e.g., containment vs. adjacency). We stratify prompts by task difficulty

(simple/medium/hard), run five random seeds for retrieval re-ranking, and report mean \pm sd. For T1, answers must include at least one URI per conclusion; uncited assertions are flagged as hypotheses and scored separately. For T2/T3, we evaluate both retrieval quality and the final textual recommendation/summary.

5.4 Example: Ontology-Guided vs. Ontology-Agnostic Outputs

The following table compares two example outputs from the Regulatory Q&A task (T1).

Both systems received the same query:

“What seismic standard applies to a building constructed in 2010 with base-isolation technology in Tokyo?”

Table 1 illustrates an example comparison between ontology-agnostic and ontology-grounded generation for the regulatory question-answering task (T1).

The ontology-guided output demonstrates clearer logical reasoning and explicit citation of the applied seismic standard, highlighting how structured knowledge improves interpretability and factual consistency.

Table 1: Comparison of Answers Generated Without and With Ontology for Regulatory Q&A.

PROMPT SETTING	EXAMPLE OUTPUT
WITHOUT ONTOLOGY	“Buildings in Japan follow the Building Standard Law, and those with base-isolation systems are designed under general seismic design rules. The specific standard depends on structural design documents.”
WITH ONTOLOGY (BOT-RAG)	“According to ontology rules: if a building was constructed after 2000 and has base-isolation technology , it conforms to the 2000 Seismic Standard . Therefore, this 2010 Tokyo building satisfies the 2000 standard.”

6. Evaluation & Results

Three evaluation tasks were designed to assess the system from different perspectives:

T1 represents the Regulation Retrieval Task, focusing on identifying the correct seismic standard for a given building description.

T2 corresponds to the Spatial-Element Constraint Reasoning Task, validating the logical consistency between architectural spaces and components.

T3 denotes the Recommendation Generation Task, where the system provides design or equipment suggestions based on the inferred ontology.

The proposed ontology-LLM integrated system was evaluated through three complementary perspectives: Quantitative, Semantic, and Qualitative.

The experiments aimed to compare the proposed ontology-grounded framework (BOT-RAG) with both LLM-only and

ontology-agnostic RAG baselines, under identical retrieval and generation settings.

A Python harness was used to ensure consistent sampling, logging seeds, and hyperparameters for reproducibility.

In the quantitative evaluation, we measured Exact Match, Token-F1, and Label-Normalized Accuracy to verify factual precision and ontology alignment

Task-specific constraints, such as space-element relationships, were assessed separately to confirm rule compliance within the architectural domain.

For the semantic evaluation, we adopted BERTScore (F1), chrF ($\beta=2$), and ROUGE-L (F1) to evaluate the semantic proximity between generated and reference answers.

Finally, the qualitative evaluation included Citation Precision and Consistency Rate (as automatically validated), as well as Expert Ratings on a 1–5 scale to assess overall interpretability and reliability.

Across all metrics, the proposed system consistently outperformed the baselines.

The retrieval performance improved in both nDCG@10 and Entity Coverage@10, indicating that ontology-based indexing and entity linking enhanced the relevance of retrieved evidence.

In generation quality, ontology grounding contributed to higher Label-Normalized Accuracy and BERTScore, suggesting better mapping between generated terms and domain concepts.

Moreover, evidence and qualitative metrics demonstrated substantial gains in factual justification and consistency, verified both automatically and by experts.

These improvements highlight that integrating structured symbolic reasoning with generative LLM capabilities not only strengthens factual grounding but also enhances transparency and human interpretability.

Table 2 presents quantitative retrieval metrics such as nDCG@10, Recall@5, and Entity Coverage@10.

These results compare the baseline LLM-only and RAG models against the proposed BOT-RAG system, demonstrating that ontology-aware indexing and entity linking significantly enhance retrieval relevance and entity coverage.

Table 2: Retrieval performance (mean \pm sd)

System			
	nDCG@10	Recall@5 (%)	Entity-Coverage@10 (%)
B0 LLM-only	0.432 \pm 0.021	0.318 \pm 0.027	41.2 \pm 4.8
B1 RAG (no-ontology)	0.612 \pm 0.019	0.544 \pm 0.022	65.7 \pm 3.9
Proposed(BOT-RAG)	0.673 \pm 0.018	0.602 \pm 0.020	74.3 \pm 3.2
EntityLinking (ablation)	0.641 \pm 0.020	0.569 \pm 0.021	68.1 \pm 3.6

Table 3 summarizes the generation performance across tasks T1–T3, focusing on quantitative and semantic metrics such as Label-Normalized Accuracy and BERTScore.

The proposed BOT-RAG consistently outperforms the ontology-agnostic baseline, reflecting improved alignment

between generated text and ontology-grounded domain concepts.

Table 3: Generation metrics (Few-shot; Quantitative & Semantic, mean \pm sd)

Task	System	Label-Norm Acc.	BERTScore (F1)
T1	B1 RAG(no-ontology)	0.431 \pm 0.027	0.862 \pm 0.010
	Proposed (BOT-RAG)	0.489 \pm 0.025	0.874 \pm 0.009
T2	B1 RAG(no-ontology)	0.397 \pm 0.026	0.853 \pm 0.010
	Proposed (BOT-RAG)	0.452 \pm 0.024	0.867 \pm 0.009
T3	B1 RAG(no-ontology)	0.411 \pm 0.027	0.858 \pm 0.010
	Proposed (BOT-RAG)	0.467 \pm 0.024	0.871 \pm 0.009

Table 4 reports qualitative evaluation results, including citation precision, consistency rate, and expert ratings on a 1–5 scale.

These metrics capture the reliability and interpretability of generated outputs, confirming that ontology grounding enhances factual justification and human-perceived consistency.

Table 4: Evidence & Qualitative (mean \pm sd)

System	Citation Precision(%)	Consistency Rate(%)	Expert(1-5)
B1 RAG(no-ontology)	71.4 \pm 4.1	78.0 \pm 3.6	3.62 \pm 0.28
Proposed (BOT-RAG)	82.6 \pm 3.5	90.8 \pm 2.7	4.06 \pm 0.24

7. Conclusion and Future Work

This study presented an ontology-assisted, retrieval-augmented generation (RAG) framework for the architectural design domain.

By integrating the Building Topology Ontology (BOT) and a domain-specific Seismic Architecture Ontology with a large language model (LLM), the system achieved improved retrieval effectiveness, answer accuracy, and evidence consistency compared with ontology-agnostic baselines.

Ontology grounding enabled URI-level traceability, consistent entity typing, and relation-aware prompting, thereby enhancing the interpretability of generated answers.

In future work, we plan to extend the ontology to cover additional aspects of architectural knowledge such as cost, schedule, and material properties, and to explore adaptive fine-tuning that dynamically incorporates inferred triples into LLM contexts.

REFERENCES

- [1] P. Lewis, E. Perez, A. Piktus, et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [2] Z. Ji, N. Lee, R. Frieske, et al., “Survey of Hallucination in Natural Language Generation,” *ACM Computing Surveys*, vol. 55, no. 12, Article 248, 2023. doi:10.1145/3571730.
- [3] M. H. Rasmussen, P. Pauwels, M. Hviid, and J. Karlshøj, “BOT: The Building Topology Ontology of the W3C Linked Building Data Group,” *Semantic Web Journal*, 2020. (Specifies lightweight Site/Building/Storey/Space/Element/Interface and 3D references.)
- [4] P. Pauwels and W. Terkaj, “EXPRESS to OWL for Construction Industry: Towards a Recommendable and Usable ifcOWL Ontology,” *Automation in Construction*, vol. 63, pp. 100–133, 2016. doi:10.1016/j.autcon.2015.12.003.
- [5] W3C, “OWL 2 Web Ontology Language Primer (Second Edition),” W3C Recommendation, Dec. 2012.
- [6] W3C, “SPARQL 1.1 Query Language,” W3C Recommendation, Mar. 2013.
- [7] W3C, “Shapes Constraint Language (SHACL),” W3C Recommendation, Jul. 2017 (see also SHACL 1.2 Core, 2025).
- [8] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “BERTScore: Evaluating Text Generation with BERT,” in *International Conference on Learning Representations (ICLR)*, 2020.
- [9] M. Popović, “chrF: Character n-gram F-score for Automatic MT Evaluation,” in *Proceedings of the Tenth Workshop on Statistical Machine Translation (WMT)*, pp. 392–395, 2015. doi:10.18653/v1/W15-3049.
- [10] C.-Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries,” in *Text Summarization Branches Out (ACL Workshop)*, pp. 74–81, 2004.
- [11] I. Paik, “Integrating Ontology Rules with Large Language Models for Enhanced Reasoning,” Proc. 39th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), Seoul, Korea, Jul. 2025.