A hybrid RAG approach for generating ontology-based information containers from building permit application submissions using text-based LLMs

Philipp Hagedorn[1], Jayesh Sudhakar Adlinge[2], Judith Fauth[2] and André Borrmann[2]

[1]Ruhr University Bochum, Universitätsstraße 150, 44801 Bochum, Germany
[2]Technical University of Munich, Arcisstr. 21, 80333 Munich, Germany

philipp.hagedorn-n6v@rub.de, jayesh.adlinge@tum.de, judith.fauth@tum.de andre.borrmann@tum.de

Abstract -

Building permit applications usually consist of multiple heterogeneous, unstructured, and non-machine-readable data. To efficiently check and process applications, relevant information must be extracted, and documents must be classified and linked to the application and process steps in the review. To address this, this study presents an end-to-end method for generating ontology-structured information containers from PDF-based building permit applications by combining Large Language Models with a hybrid Retrieval Augmented Generation (RAG) layer. The proof-of-concept demonstrates that the proposed pipeline can populate OntoBPR entities while producing standard-compliant container structures in accordance with ISO 21597. Remaining challenges relate to scalability, handling of ambiguous regulatory language, and quantitative evaluation of the pipeline. Overall, the study demonstrates the potential of retrieval-augmented LLM pipelines for advancing automation and semantic integration in digital building permit processes.

Keywords -

Architecture, Engineering and Construction (AEC); Digital Building Permitting (DBP); Knowledge Graphs (KG); Retrieval Augmented Generation (RAG); Neuro-symbolic AI

1 Introduction

Building permit authorities typically receive application submissions as heterogeneous collections of documents: (scanned) PDF application forms, text-based reports, drawings, exports from specialist software, and occasionally models from Building Information Modeling (BIM) serialized in the Industry Foundation Classes (IFC) [1]. Even when these materials are available in digital form, they are usually not structured to support automated reasoning, traceability, or even code-compliance checking [2]. In particular, the formal review is identified as an essential step in building permit review, because missing documents and information delay the permit procedure, as it is depicted in the study of Fauth et al. [3] reporting that 65 % of building applications were delayed because of missing or erroneous application documents. Moreover, previous research has identified the relevant taxonomy for the building permit process [4], which has been incorporated into an ontology representing processes, stakeholders, and documents in the building permit review, denominated as the OntoBPR framework. The OntoBPR framework initially included a set of ontologies as the formal semantic representation of permit-related terms [2]. In recent research, this approach was integrated with the Information Container for linked Document Delivery (ICDD) to model not only the documents but also the interconnections between them [5], and here, in particular, between the documents submitted in an application package. While ICDD containers in the OntoBPR framework are generated manually, the objective of this article is to automatically generate information containers and ontology-based structures from the submitted documents and to classify the submitted documents with metadata to facilitate formal checks of the application's completeness.

Recent advances in Large Language Models (LLMs) and multimodal AI systems enable automatic extraction, structuring, and semantic classification of information from heterogeneous, unstructured documents [6]. These capabilities allow for the generation of metadata and knowledge representations that support automated reasoning, traceability, and completeness checks [7, 8]. The capabilities to generate and summarize text, as well as to adapt to additional given context, also known as few-shot learning, are the significant benefits of LLMs that are identified in the review of Ploennings et al. [9], who analyzed the potentials, challenges, and research directions for applying LLMs and Large Vision Models (LVM) in the Architecture, Engineering, and Construction (AEC) domain. Moreover, document generation and compliance checks were identified as one of the main use cases for Large Language Models (LLMs) in the construction domain in a literature review by Kampelopoulos et al. [10]. Wu et al. [11] argue that LLMs are trained on general knowledge rather than AEC-specific knowledge. While

training an LLM on AEC knowledge is an option, it is impractical due to the enormous costs and training time [11]. A solution to this issue is Retrieval Augmented Generation (RAG), in which information or documents are segmented into smaller chunks, vectorized, and injected into LLMs [11]. Based on the literature, this research aims to employ LLMs and customize them for the use case of analyzing building permit documents by a state-of-the-art RAG implementation. Following up on this aim, the research question formulated in this article is the following: *How can unstructured building permit applications be converted into standard-compliant ICDD containers and OntoBPR-aligned RDF metadata?*

To answer this research question, the remainder of the paper is structured as follows: First, related work on customizing LLMs for specific document processing is provided. Second, the methodology and concept of this approach are described. Third, results from implementing the conceptualized pipeline are presented, and, lastly, the results are discussed and conclusions drawn.

2 Related Work

2.1 OntoBPR and Information Containers

Structuring building permit applications in information containers was introduced with OntoBPR in [2] and offers advantages, including facilitating formal checks on the application, increasing transparency in the permit review by sharing progress updates, and supporting decision-making through rule-based checks of machine-readable reports. The OntoBPR schema is implemented as an ontology, which is available at *https://w3id.org/ontobpr#*. For structuring the submission of a building application, the ontology is instantiated in an ICDD container, which itself is compliant with ISO 21597-1 [12]. Fig. 1 depicts an exemplary OntoBPR ICDD container of a submitted building application in the file system

These ICDD containers have a predefined structure defined through ontologies, thus semantically describing documents and their interconnections. OntoBPR is a semantic layer on top of the generic description of documents, adding context to the meaning of documents and to the purpose of the information container as a building application container. Main classes that are relevant to this study are the `ct:Document` class and its subclasses (c.f. [12]), as well as the classes `obpr:BuildingApplication` and `obpr:Building` from OntoBPR. The OntoBPR framework and the ICDD use a shared technology stack comprising the Resource Description Framework (RDF) and the Web Ontology Language (OWL) for the formal description of the knowledge graph, and Shapes and Constraint Language (SHACL) for validation [2].

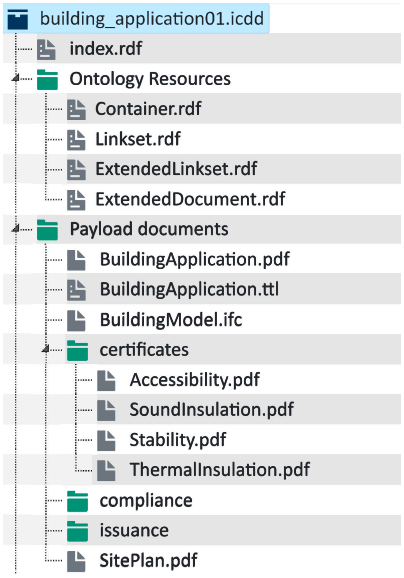

Figure 1. OntoBPR ICDD Container example

2.2 Extraction from Documents

Recent methodological developments have shifted information extraction in the AEC domain from rule-based systems to approaches driven by LLMs. This transition is primarily motivated by the need to handle heterogeneous, complex, and weakly structured data sources that are common in construction projects. The extraction of metadata from engineering drawings remains a longstanding challenge due to limited standardization across documents. Recent research introduces multimodal processing pipelines that combine lightweight Convolutional Neural Networks with visual language models to detect and interpret title blocks and embedded annotations. Empirical evaluations demonstrate clear performance gains over classical optical character recognition techniques, particularly in terms of robustness to layout variability [13].

A further emerging direction is the use of LLMs for knowledge graph construction. This line of work focuses on the automated extraction of entities and relations from unstructured text to populate graph-based representations. Recent studies demonstrate the direct generation of RDF triples from natural language inputs and the population of domain-specific ontologies through zero-shot and few-shot prompting strategies. Applications include knowledge models, e.g., for building materials, indicating the growing maturity of generative approaches for semantic structuring in the AEC domain [14, 15, 16].

2.3 Retrieval Augmented Generation (RAG)

Conventional RAG systems often exhibit limited performance when operating on flat data representations (i.e., lacking explicit structural, relational, or semantic mod-

eling). Therefore, recent research focuses on hybrid retrieval-augmented generation to address the structural complexity of heterogeneous documents that combine text, tables, and code. Structure-aware RAG extends classical pipelines by explicitly modeling structural relations. Approaches such as *GraphRAG*, *StructRAG*, and *LightRAG* integrate knowledge graphs into the retrieval stage. Context retrieval is no longer based on isolated text segments but on linked entities and relations, such as document hierarchies or citation networks. This strategy yields a coherent, semantically connected context. The explicit coupling of neural language models with symbolic knowledge graph structures also aligns these approaches with contemporary interpretations of neuro-symbolic artificial intelligence. Empirical studies report improved answer diversity and superior performance on multi-hop reasoning tasks compared to standard RAG [17, 18].

Ontology-guided retrieval augmented generation addresses domain-specific requirements by embedding formal ontologies into the retrieval and generation pipeline. Systems such as *GrOWL RAG* and *OntologyRAG* use structured domain knowledge to constrain and guide language model outputs, which is particularly relevant for tasks that require strict schema compliance, including document-level relation extraction and code-to-concept mapping. Reported results indicate reduced hallucination rates and increased precision in specialized application scenarios [19, 20]. Hybrid and collaborative RAG aims to overcome the limitations of single retrieval strategies. State-of-the-art systems combine sparse retrieval methods, such as BM25 for lexical matching, with dense retrieval based on vector embeddings for semantic similarity. This combination improves recall and robustness across query types [21, 22]. Recent work introduces a concept called *retrieval augmented retrieval*, in which LLMs generate candidate answers or query expansions prior to retrieval. This mechanism expands the effective query space and improves zero-shot performance in low-resource settings [23].

3 Methodology and Concept

This article presents a proof-of-concept pipeline that uses a hybrid RAG approach to ingest building application documents and generate a compliant ICDD container populated with the OntoBPR knowledge graph. Therefore, a pipeline is implemented and evaluated using a use case as a proof-of-concept. The hybrid RAG approach was chosen after studying several methods of RAG that are referred to in the literature in the previous section.

3.1 Ingestion and Preprocessing

The implemented pipeline comprises four main steps, as shown in Fig. 2, which are described in detail. In the first step, the ingestion component stages and cleans a ZIP file containing a building application submission. The included files are copied into an ICDD-like layout under a case-specific output directory `Payload documents/<CASE_ID>/`. During this process, the system constructs the ICDD container description (index.rdf) using *rdflib*[1]. It creates a `ct:ContainerDescription` individual for the container. Each payload file is registered as a `ct:InternalDocument` with a relative path and MIME type derived from the file extension and heuristics.

The text extraction focuses on PDF documents identified in the first step of the MIME type analysis. The system first attempts to use *pdfminer.six*[2] to read any text layer directly. If this yields insufficient output, an OCR fallback is triggered: pages are rasterized using *pdf2image*[3], and *pytesseract*[4] is applied to recognize text in the resulting images. The extracted text is segmented into overlapping chunks of configurable size. Each chunk is represented by a *TextChunk* record that carries the case identifier, a document identifier, page information, a sequence index, and a generalized role specification inferred from the source filename using a small set of patterns. These chunks, stored in a JSON Lines[5] file, form the basis for retrieval.

3.2 Hybrid Indexing and Retrieval

The retrieval layer combines vector-embedded semantic similarity search with a simple notion of document structure in a temporary knowledge graph. On the vector side, each text chunk is encoded using a SentenceTransformers model (*all-MiniLM-L6-v2*)[6]. The resulting embeddings are stored and indexed for fast cosine-similarity search. Given a textual query describing an information need (for instance, ”application reference” or ”site address”), this index returns chunks whose content is semantically similar, even when the exact query phrase does not appear. On the graph side, a Neo4j[7] database temporarily stores relationships between cases, documents, and chunks. This graph records which chunks belong to which document and preserves their sequence. When answering a query, the system can follow these edges to retrieve contextually neighbouring chunks within the same document or section, thereby complementing vector search with structural context. A dual retrieval function draws from both the vector index and the graph, merges the results, removes duplicates, and orders them into a single context window. This combined context is provided to the LLM for extraction in the next step.

[1] rdflib, v.7.1.4, https://github.com/RDFLib/rdflib
[2] pdfminer.six, v.20250506, https://github.com/pdfminer/pdfminer.six
[3] pdf2image, v.1.17.0, https://github.com/Belval/pdf2image
[4] pytesseract, v.0.3.13, https://github.com/madmaze/pytesseract
[5] see https://jsonlines.org/
[6] https://github.com/huggingface/sentence-transformers
[7] Neo4j, v.5.25.0, https://github.com/neo4j/neo4j

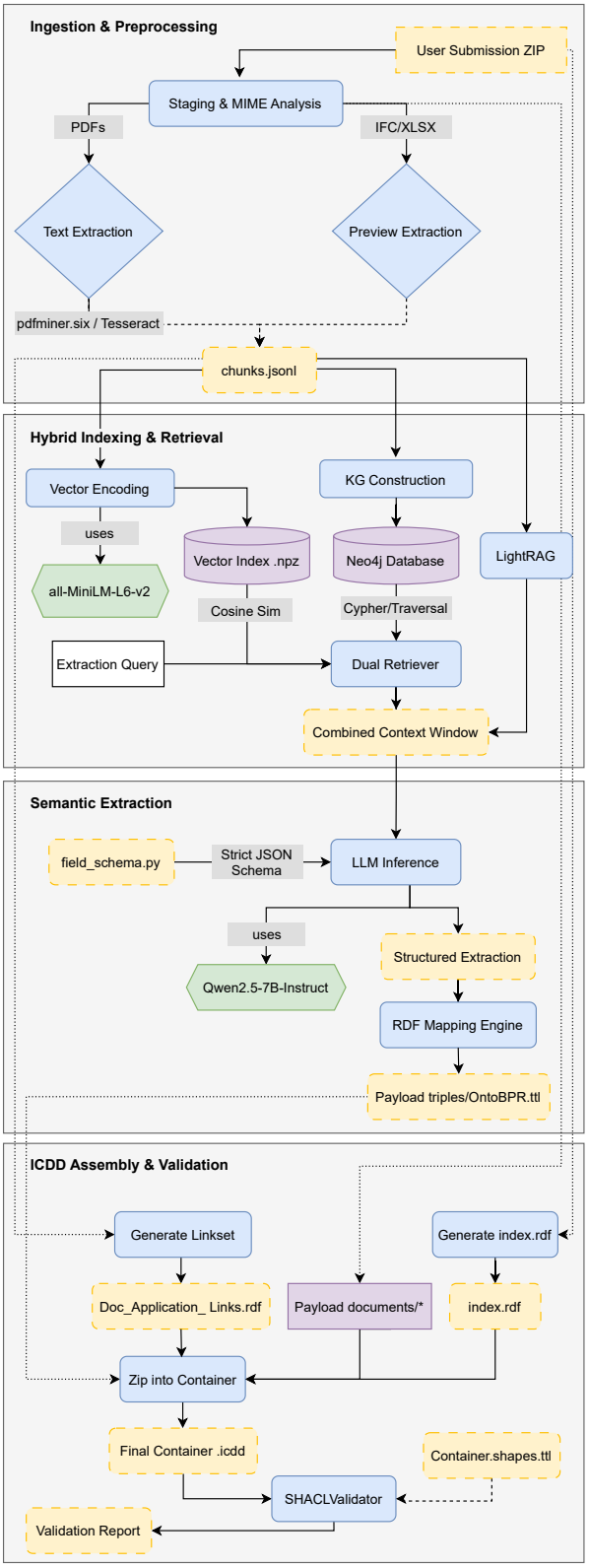

Figure 2. Implementation overview

3.3 Semantic Extraction

In Step 3, the utilized *Qwen 2.5-7B-Instruct*[8] LLM is not required to generate novel RDF structures or arbitrary JSON formats (c.f. also in Fig. 2). Instead, the pipeline relies on a predefined schema, which delineates all relevant data fields. At the current development stage, this schema covers application-level information, such as an internal identifier, application type, authority name, and site address, as well as building-level information, including the building's intended use and, where available, simple quantitative descriptors. For each case, the prompt provided to the LLM combines both the dual retrieval context and the JSON schema. The accompanying instructions emphasize that the model must rely exclusively on the supplied context, leave fields blank when no information is available, and produce syntactically valid JSON that is schema-compliant. The result is a structured JSON document containing the predicted values for all fields, organized under entities such as `building_application` and `building`. These JSON files are stored alongside the other case-specific outputs and subsequently serve as inputs to the OntoBPR mapping and evaluation script.

The JSON extraction in Step 5 is then transformed into an RDF graph aligned with the OntoBPR ontology. The mapping logic implemented using *rdflib* creates individuals representing the building application and, where applicable, the building itself. JSON fields are converted into datatype properties, e.g., an extracted application reference is mapped to `ontobpr:applicationIdentifier` or a site address to `ontobpr:siteAddress`, while the building use type is mapped to `ontobpr:buildingUseType`.

Moreover, for employing also the capability of the LLM to classify the documents based on the extracted terms semantically, OntoBPR does include the class `ontobpr:DocumentClassification` with a set of named individuals with German and English labels, such as `ontobpr:ApplicationForm`, `ontobpr:EngineeringProofs`, `ontobpr:PlotRelatedPlans`, or `ontobpr:Unclassified`.

The mapping uses a fixed set of OntoBPR classes and properties defined in code. The LLM therefore contributes the values, but the ontology vocabulary and URI patterns remain under deterministic control. This choice avoids the common problem of LLMs inventing nonexistent ontology terms (hallucinating) and keeps the RDF graph compliant with OntoBPR for the implemented subset. The resulting triples are written to a Turtle file in the output directory and later included as part of the container's semantic payload.

[8] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

3.4 ICDD Assembly and Validation

To maintain traceability between semantic data and source documents, a simple linkset is generated using the ICDD linkset vocabulary. This file expresses relationships between documents and, where appropriate, between documents and semantic entities. Before packaging, the combined RDF graph, including the container description, internal documents, linksets, and OntoBPR triples, is validated using SHACL. The prototype employs *pySHACL*[9] together with SHACL shapes `Container.shapes.ttl` derived from the ICDD standard validation resources. Finally, the container description, payload documents, ontology resources, linksets, and generated triples are assembled into a directory and compressed as a *.icdd* file. The overall pipeline and the assembly of generated artifacts are displayed in detail in Figure 2.

4 Proof of Concept and Results

The current prototype has been tested on a representative real-world permit application (`Apl_25-04543-FULL`). The corresponding submission archive is approximately 25 MB in size and consists of twelve distinct files, including text-based PDF forms, scanned or image-based PDFs, and several additional attachments such as images and spreadsheets (see Tab. 1). The documents exhibit typical characteristics of administrative practice: overlapping content across versions, varying terminology for similar concepts, and partial mixing of languages. Some information is distributed across different sources, for example, addresses that appear only in headers or footers, or quantitative data that appears only in drawings.

The system was executed in an environment with a single NVIDIA A100 GPU. *Qwen-2.5-7B-Instruct* was used as the extraction model, and *all-MiniLM-L6-v2* served as the sentence embedding model for the vector index. The hybrid retrieval layer combines dense vector search with a small *Neo4j*-based document graph. The precise configuration details (such as quantization settings or batch sizes) are less critical for the analysis presented here than the overall architectural pattern: retrieval-supported extraction followed by deterministic mapping to RDF.

For the examined case, the pipeline successfully produced an ISO 21597-1 container. The resulting `index.rdf` file lists all payload documents as internal documents, assigns the expected MIME types based on file extensions, and attaches a container description node with the required core metadata. SHACL validation against the available container shapes confirmed that the basic structural constraints of Part 1 are satisfied for this example, after the usual relaxation of strictly closed shapes to allow additional metadata. A simple linkset was generated to demonstrate the integration of Part 2. The linkset associates documents with each other and, where applicable, connects them to semantic entities. In the tested container, these links were successfully resolved to internal paths, and the linkset passed the structural checks applied in this prototype.

[9] pySHACL, v.0.30.1,https://github.com/RDFLib/pySHACL/

Table 1. Composition of the submission archive

Type	Count	Description
Administrative documents	3	Application form, receipt of application, and cover letter.
Planning documents	1	Planning Statement providing policy-based and contextual justification.
Technical drawings and plans	3	Architectural and technical drawings submitted as part of the application.
Supporting documents	6	Consultation, notification, and objection documents, including schedules, photographs, and spreadsheets.
Total	13	**6.53 MB zipped archive**

The OntoBPR payload produced by the mapping step is intentionally modest in scope. For the examined case, the pipeline created a `ontobpr:BuildingApplication` individual and associated it with a small number of datatype properties, such as an application identifier, an authority name, and a site address (see Fig. 3, lines 6–16). From a purely syntactic and ontological point of view, the graph is consistent in that it uses declared OntoBPR properties and datatypes and does not introduce ad hoc vocabulary (hallucination). However, this does not imply that all values are correct or complete; it only confirms that the mapping logic respects the chosen subset of the ontology. The behavior of the hybrid retrieval layer is best understood through concrete examples. For fields whose mentions are relatively explicit in the text, such as the application identifier or the name of the authority, vector-based retrieval generally retrieves appropriate chunks from application forms or covering letters. In such cases, the LLM can fill the JSON schema with plausible values, and the subsequent mapping to OntoBPR is straightforward.

For more distributed or abstract fields, such as building use type (see Fig. 3, line 20), the benefit of combining vector and graph retrieval becomes more apparent. When the relevant information is spread across a design statement and other supporting documents, vector search alone may return passages that are topically related but do not contain a clear, machine-readable statement of the use type. In these situations, extending the context with structurally neighboring chunks from the same document, identified via the graph, can provide the model with enough continuity to extract a meaningful value. This observation is

```
@prefix bp-app: <https://example.org/Apl_25-04543-FULL#> .
@prefix obpa: <https://w3id.org/obpa#> .
@prefix ontobpr: <https://w3id.org/ontobpr#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

bp-app:BA_Apl_25-04543-FULL
  rdf:type obpa:BuildingApplication ;
  rdf:type ontobpr:BuildingApplication ;
  obpa:includes bp-app:Project_Apl_25-04543-FULL ;
  ontobpr:applicationIdentifier "PP-13765429" ;
  ontobpr:applicationType "Variation of condition" ;
  ontobpr:buildingAuthority "centralplanningteam@westminster.gov.uk" ;
  ontobpr:hasApplicationDocument bp-app:Doc_5 ;
  ontobpr:hasBuildingApplicationContainer bp-app:Container_Apl_25-04543-FULL ;
  ontobpr:siteAddress "3 Anonymous Street, London, W1U 3QG" ;
.
bp-app:Building_Apl_25-04543-FULL
  rdf:type obpa:Building ;
  rdf:type ontobpr:Building ;
  ontobpr:buildingUseType "Mixed-use" ;
  ontobpr:hasBuildingDocument bp-app:Doc_1 ;
  ontobpr:hasBuildingDocument bp-app:Doc_11 ;
  ontobpr:hasBuildingDocument bp-app:Doc_2 ;
  ontobpr:hasBuildingDocument bp-app:Doc_3 ;
.
bp-app:Project_Apl_25-04543-FULL
  rdf:type obpa:ConstructionProject ;
  obpa:hasBuilding bp-app:Building_Apl_25-04543-FULL ;
.
bp-app:Doc_1
  ontobpr:hasDocumentClassification ontobpr:TechnicalDrawing ;
.
bp-app:Doc_10
  ontobpr:hasDocumentClassification ontobpr:Unclassified ;
.
bp-app:Doc_11
  ontobpr:hasDocumentClassification ontobpr:BuildingDescription ;
.
bp-app:Doc_12
  ontobpr:hasDocumentClassification ontobpr:Unclassified ;
.
```

Figure 3. Excerpt of the generated OntoBPR-KG

qualitative but suggests that structural information helps guide the model toward relevant parts of long documents.

The classification of documents (see Fig. 3, lines 30ff.) according to OntoBPR classified ~39 % of the submitted documents, while eight documents remain unclassified after the procedure. While this classification is not the main achievement of this research, it requires a dedicated evaluation on a large-scale dataset and with variations in the classification descriptions in future work.

The complete pipeline during the case study execution required approximately 170 seconds on the described hardware, with retrieval preparation accounting for the majority of the overall runtime (see Tab. 2). Preprocessing of PDF documents, particularly OCR for scanned pages, accounts for a substantial portion of the overall runtime. Building the vector index and populating the *Neo4j* graph adds additional overhead but remains manageable for a single case. LLM inference for the relatively small schema used here contributes comparatively little to total runtime.

Table 2. Reported runtime for the test case

Pipeline Component	**Runtime (s)**
Ingestion and staging	0.12
Text extraction and chunking	28.00
Retrieval preparation	135.50
Retrieval query execution	3.84
LLM extraction and OntoBPR mapping	1.85
Container generation and validation	0.42
Total Runtime	**169.73**

In terms of container generation, the prototype was able to create basic link relations from the semantic layer back to the primary application document, and SHACL validation showed no structural violations for these links. This demonstrates that the principle of traceability from semantic statements to source documents can be implemented within the ICDD framework.

5 Discussion and Conclusions

OntoBPR was introduced in prior work as an ontology-based framework to support digital building permit review processes. This study extends that work by operationalising OntoBPR through standardized semantic information delivery using the Information Container for linked Document Delivery (ICDD) and by integrating a retrieval-augmented generation (RAG) approach to support information access during the building permit phase. Rather than proposing a new ontology, the study demonstrates how OntoBPR can be embedded into an interoperable, AI-supported submission and review mechanism that addresses fragmentation in digital building permitting.

Many digital permitting approaches remain limited to isolated functions, particularly automated code checking, and rely on document-centric information exchange. Integrating OntoBPR with ICDD bundles heterogeneous documents, datasets, and models into a semantically enriched container, where relationships between regulations, submitted evidence, and procedural steps are explicitly represented using RDF. This shifts the building application from a static document collection to a machine-interpretable information package that supports validation, traceability, and reuse. The integration of a RAG approach further extends this capability. By using the structured semantic content of ICDD containers as a trusted knowledge base, RAG enables context-aware retrieval and synthesis of relevant regulatory provisions and decision-relevant information, supporting building officials while maintaining transparency through semantic grounding.

The pipeline can generate an ISO 21597-1 container with an embedded OntoBPR application graph for a realistic building-permit submission, passing all SHACL checks. The results of this study must be interpreted in light of several intentional design constraints and limitations of the prototype implementation. OntoBPR is employed in a deliberately restricted manner. Only a subset of classes and properties related to the building application and core building attributes is instantiated. More expressive components of the ontology, such as workflow stages, involved actors, decision-making processes, and explicit representations of building-code knowledge graphs, are intentionally excluded in the first step of this research. Con-

sequently, the findings should be understood as a proof-of-concept for semantic extraction and representation using LLMs with the presented hybrid RAG approach, rather than a comprehensive OntoBPR instantiation.

The evaluation dataset is small, and the quantitative metrics produced by the evaluation script should therefore be considered indicative rather than statistically conclusive. The reported values primarily demonstrate overall feasibility and expose typical error patterns in document restructuring, attribute extraction, and semantic mapping. Although the ingestion pipeline is conceptually capable of handling heterogeneous document types, the current experimental focus is predominantly on PDF-based submissions. Other common artifacts in building permit applications include IFC models, which are central to the future development of digital building permitting. These artifacts are not yet analyzed with comparable semantic depth. Extending the pipeline towards multimodal extraction and tighter integration of model-based information remains an important avenue for future research, particularly given IFC's role in digital building permitting.

While this study is a first approach towards the automatic processing of building applications in practice, larger and more heterogeneous datasets in a real-world scale will be required to enable statistically meaningful evaluations and comparative benchmarking, which will be provided in future work. Finally, although the system can operate in a fully automated manner once configured, the generated ICDD containers and RDF graphs still require expert review for critical or legally relevant use. Integrating systematic human-in-the-loop refinement mechanisms for assessing the quality of the generated container and annotations would not only increase trust and adoption but also enable iterative quality improvement through expert feedback loops. Future research must therefore focus on empirical validation through pilot deployments with building authorities, assessing the scalability, usability, and acceptance of OntoBPR-enabled ICDD submissions and RAG-based decision support.

Acknowledgments

This research uses publicly available and research-oriented data sources. Annotated building permit documents were obtained from the ACCORD dataset, supplemented by real-world permits from public application portals (e.g., the City of Westminster). All documents were used solely for research, processed locally, and any identifying information was anonymized or excluded. JF received funding through the R&D project "Development of a process model and identification of innovative approaches to the building permit procedure in the building life cycle" funded by brain-SCC GmbH.

References

[1] Jeroen Werbrouck, Philipp Hagedorn, Donkers, Alex, and Judith Fauth. Semantic Web Technologies for Building Permitting. In Fauth, Guler et al., editor, *Digital Transformation of Building Permitting*. Taylor & Francis, 2026. URL `https://www.routledge.com/9781041091752`.

[2] Philipp Hagedorn, Judith Fauth, Sven Zentgraf, Sebastian Seiß, Markus König, and Ioannis Brilakis. OntoBPR: An ontology-based framework for performing building permit reviews using standardized information containers. *Advanced Engineering Informatics*, July 2025(66), 2025. ISSN 14740346. doi:10.1016/j.aei.2025.103369.

[3] Judith Fauth, Aurica Pötz, Michael Brenner, and Hannes Heitzhausen. Das Baugenehmigungsverfahren und dessen Digitalisierung: eine Bestandserhebung. *Bautechnik*, 102(10):571–580, 2025. ISSN 0932-8351. doi:10.1002/bate.202500011.

[4] Judith Fauth, Tanya Bloch, Francesca Noardo, Nicholas Nisbet, Stefanie-Brigitte Kaiser, Peter Nørkjær Gade, and Jernej Tekavec. Taxonomy for building permit system - organizing knowledge for building permit digitalization. *Advanced Engineering Informatics*, 59:102312, 2024. ISSN 14740346. doi:10.1016/j.aei.2023.102312.

[5] Anne Göbels and Jakob Beetz. Using ICDD Containers for documentation data archives: Semi-automated creation of linked documentation data archives using an Information Container for linked Document Delivery. *Proceedings of the 27th International Conference on Cultural Heritage and New Technologies (CHNT), Vienna, Austria*, 2022.

[6] Chen Han, Yuanyuan Li, and Xijin Tang. DocPolicyKG: A Lightweight LLM-Based Framework for Knowledge Graph Construction from Chinese Policy Documents. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 4753–4757, New York, NY, USA, 2025. ACM. doi:10.1145/3746252.3760904.

[7] Kiara M. A. Arevalo, Shruti Ambre, and Rene Dorsch. AutOnto: Towards A Semi-Automated Ontology Engineering Methodology. In *Knowledge Graphs and Semantic Web*, pages 225–241, Cham, 2025. Springer Nature. doi:10.1007/978-3-031-81221-7_16.

[8] Lea Höltgen, Sven Zentgraf, Philipp Hagedorn, and Markus König. Utilizing Large Language Models for Semantic Enrichment of Infrastructure Con-

dition Data: A Comparative Study of GPT and Llama Models. *AI in Civil Engineering*, 4(1), 2025. doi:10.1007/s43503-025-00055-9.

[9] Joern Ploennigs, Markus Berger, Thomas Wortmann, Jakob Kirchner, Jakob Beetz, Alina Roitberg, Karsten Menzel, and Björn Ommer. Building Foundation Models - Potentials, Challenges and Research Directions for Using LLM and LVM in AEC. In *Proceedings of the 2025 European Conference on Computing in Construction*, 2025. doi:10.35490/EC3.2025.268.

[10] Dimitrios Kampelopoulos, Athina Tsanousa, Stefanos Vrochidis, and Ioannis Kompatsiaris. A review of LLMs and their applications in the architecture, engineering and construction industry. *Artificial Intelligence Review*, 58(8), 2025. ISSN 1573-7462. doi:10.1007/s10462-025-11241-7.

[11] Chengke Wu, Wenjun Ding, Qisen Jin, Junjie Jiang, Rui Jiang, Qinge Xiao, Longhui Liao, and Xiao Li. Retrieval augmented generation-driven information retrieval and question answering in construction management. *Advanced Engineering Informatics*, 65:103158, 2025. ISSN 14740346. doi:10.1016/j.aei.2025.103158.

[12] ISO 21597-1. *Information container for linked document delivery: Exchange specification - Part 1: Container*. International Organization for Standardization, Geneva, Switzerland, 1 edition, 2020. URL `https://www.iso.org/standard/74389.html`.

[13] Alessio Lombardi, Li Duan, Ahmed Elnagar, Ahmed Zaalouk, Khalid Ismail, and Edlira Vakaj. Title Block Detection and Information Extraction for Enhanced Building Drawings Search. In *Proceedings of the 2025 European Conference on Computing in Construction*, 2025. doi:10.35490/EC3.2025.344.

[14] Soichi Onozuka and Takaaki Ohnishi. Analysis of LLMs for RDF Triple Generation: Semantic and Syntactic Evaluation Using WebNLG. In *2025 19th International Conference on Semantic Computing (ICSC)*, pages 136–143, 2025. doi:10.1109/ICSC64641.2025.00025.

[15] Teeradaj Racharak, Tongyu Wang, and Watanee Jearanaiwongkul. An automated medical rdf knowledge graph construction from text using in-context learning. In *2024 16th International Conference on Knowledge and System Engineering (KSE)*, pages 465–471, 2024. doi:10.1109/KSE63888.2024.11063495.

[16] Sarah Ayad and Lydia Khelifa Chibout. Ontology Population With Large Language Models (LLMs): A Case Study on Asbestos Ontology. *Applied Ontology*, 20(1):89–106, 2025. doi:10.1177/15705838251334528.

[17] Runsong Jia, Bowen Zhang, Sergio José Rodríguez Méndez, and Pouya G. Omran. StructRAG: Structure-Aware RAG Framework with Scholarly Knowledge Graph for Diverse Question Answering. In *Companion Proceedings of the ACM on Web Conference 2025*, page 2567–2573, New York, NY, USA, 2025. ACM. doi:10.1145/3701716.3717819.

[18] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation, 2025. URL `https://arxiv.org/abs/2410.05779`.

[19] Wilma Johanna Schmidt, Diego Rincon-Yanez, Irlan Grangel-González, Adrian Paschke, and Evgeny Kharlamov. Document-Level Relation Extraction with Ontology-Guided RAG. In *Linking Meaning: Semantic Technologies Shaping the Future of AI*, Studies on the Semantic Web. IOS Press, 2025. doi:10.3233/SSW250006.

[20] Hui Feng, Yuntzu Yin, Emiliano Reynares, and Jay Nanavati. OntologyRAG: Better and Faster Biomedical Code Mapping with Retrieval-Augmented Generation (RAG) Leveraging Ontology Knowledge Graphs and Large Language Models, 2025. URL `https://arxiv.org/abs/2502.18992`.

[21] Ye Yuan, Chengwu Liu, Jingyang Yuan, Gongbo Sun, Siqi Li, and Ming Zhang. A Hybrid RAG System with Comprehensive Enhancement on Complex Reasoning, 2024. URL `https://arxiv.org/abs/2408.05141`.

[22] Jaedong Lee, Hyosoung Cha, Yul Hwangbo, and Wonjoong Cheon. Enhancing Large Language Model Reliability: Minimizing Hallucinations with Dual Retrieval-Augmented Generation Based on the Latest Diabetes Guidelines. *Journal of Personalized Medicine*, 14(12), 2024. ISSN 2075-4426. doi:10.3390/jpm14121131.

[23] Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Yibin Lei, Tianyi Zhou, Michael Blumenstein, and Daxin Jiang. Retrieval-augmented retrieval: Large language models are strong zero-shot retriever. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15933–15946, Bangkok, Thailand, 2024. doi:10.18653/v1/2024.findings-acl.943.