Reliable enterprise LLM systems for high-stakes domains: A supervised deterministic retrieval framework with knowledge graphs

Mohammad Seidpisheh[1,*], Stefan Berlik[1]

[1]Faculty of Engineering and Mathematics, Hochschule Bielefeld, University of Applied Sciences and Arts, Bielefeld, Germany

Abstract

Large Language Models (LLMs) enable fluent and contextually rich conversational interaction, but their reliance on stochastic decoding and statistical co-occurrence patterns often leads to factual hallucinations. This limits their reliability in sensitive and high-stakes domains such as healthcare, finance, and engineering. While Retrieval-Augmented Generation (RAG) systems aim to reduce this problem, many rely on embedding-based retrieval mechanisms that operate as black boxes, offering limited transparency, weak explainability, and no guarantees of schema-level correctness. To address these limitations, we propose SDR-KG-LLM, a Supervised Deterministic Retrieval Knowledge-Graph–Augmented LLM framework for reliable and explainable access to structured knowledge. The framework replaces free-form query generation and probabilistic retrieval with a supervised intent-to-template mapping layer, in which user requests are mapped to a closed set of ontology-aware query templates derived from known modeling patterns. This restriction reduces the retrieval search space, improves query quality and precision, and lowers query complexity compared to unconstrained SPARQL generation.

The architecture separates the pipeline into three stages: intent extraction, deterministic knowledge retrieval, and grounded response generation. By shifting the responsibility for factual correctness from the LLM to deterministic KG queries and ontology constraints, the system significantly reduces hallucination risk while preserving natural language fluency. Overall, the proposed framework provides a foundation for reliable LLM systems in high-stakes domains while enabling deployment with smaller and more efficient language models.

Keywords

Reliable Enterprise LLM, Knowledge Graphs, Supervised Deterministic Retrieval, Knowledge-Graph–Augmented LLM, High-Stakes Domains

1. Introduction

Many real-world applications require answering questions and supporting decisions over structured, deterministic, and high-stakes knowledge, including healthcare, compliance, finance, manufacturing, and engineering design [1]. In such settings, users must interact with complex rules, constraints, hierarchies, and dependencies. Traditional interfaces, such as forms, dashboards, and expert systems, often expose this complexity directly and become difficult to use as the underlying knowledge base grows. This creates a need for conversational interfaces that provide intuitive access to formal knowledge and rule-based logic without requiring users to understand the underlying schema or constraint language.

Modern LLMs have transformed natural language processing by generating highly fluent and contextually aware text through transformer-based architectures and large-scale parameterization [2]. However, because they rely on stochastic decoding and statistical co-occurrence patterns rather than explicit reasoning or factual verification, they remain prone to hallucinations [3]. As a result, LLMs may produce plausible-sounding but factually incorrect responses, which is especially problematic in high-stakes domains [3, 4]. Overcoming this limitation is essential for building reliable LLM systems that support precise knowledge access and dependable reasoning in expert settings [4].

Knowledge Graphs and Model-driven Systems Engineering (KGMDSE), in ISWC 2026 Workshops Joint Proceedings, October 25–26, 2026, Bari, Italy

*Corresponding author.

mohammad.seidpisheh@hsbi.de (M. Seidpisheh); stefan.berlik@hsbi.de (S. Berlik)

0000-0002-2976-9206 (M. Seidpisheh); 0009-0007-3242-4501 (S. Berlik)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-05

CEUR-WS.org/Vol-4313/kgmdse-paper5.pdf

A central challenge, therefore, is to preserve the fluency and flexibility of LLMs while reducing hallucinations through stronger factual grounding [5]. Recent work has explored three main strategies for injecting knowledge into LLM systems: (1) *knowledge-aware inference*, where knowledge is accessed during inference through retrieval or prompt grounding; (2) *knowledge-aware learning*, where knowledge is injected during pretraining or fine-tuning; and (3) *knowledge-aware validation*, where model outputs are checked and corrected against an external knowledge source [3].

Among these strategies, knowledge-aware learning is computationally expensive and does not guarantee controlled reasoning or dependable factual recall, while knowledge-aware validation typically acts only after generation and may still miss subtle errors [6, 7, 8]. In contrast, knowledge-aware inference improves accuracy by grounding the LLM in external knowledge at generation time without repeated retraining [6, 2, 7]. This has motivated increasing interest in structured external knowledge sources, especially KGs, which provide curated, semantically structured, and machine-interpretable information for grounding and verification [7, 3].

However, many existing RAG systems still rely on embedding-based retrieval, which can miss deep conceptual relations and provides only weak guarantees of structural validity [9, 4, 10, 11]. To address this limitation, this paper proposes a framework that replaces probabilistic retrieval with deterministic, schema-aware knowledge access. Rather than allowing the LLM to generate arbitrary queries over the KG, user requests are mapped to a closed set of ontology-aware query templates through a supervised intent-to-template layer. This makes retrieval deterministic, reproducible, and schema-safe, while reducing the query search space and improving precision. Unlike recent NL2SPARQL systems that still rely on generation-and-validation workflows, such as metadata retrieval, example retrieval, and query repair [12, 13], our approach replaces free-form query generation with controlled retrieval aligned with explicit ontology modeling patterns.

Buchmann et al. [14] discuss RAG and natural-language access to semantic knowledge repositories, including SPARQL generation, and note that the accuracy of such natural-language query layers does not yet meet the deterministic expectations of conventional query clients. Generative NL2SPARQL approaches such as SGPT [15] learn to generate SPARQL directly from natural-language questions, whereas SDR-KG-LLM constrains this mapping through a closed intent-to-query-structure space aligned with explicit ontology modeling patterns before execution. This shifts factual knowledge retrieval from unrestricted query generation to controlled ontology-aware retrieval followed by deterministic SPARQL execution. A further practical advantage is that SDR-KG-LLM does not require a task-specific labeled dataset of natural-language questions paired with SPARQL queries, which is particularly relevant in enterprise settings where domain ontologies may be available but such supervised data are not. The deterministic fallback mechanism further improves robustness within the supported intent space by exposing valid ontology concepts or parameters when an exact retrieval cannot be resolved.

The remainder of the paper is organized as follows. Section 2 reviews foundations of LLMs and KGs and discusses their integration for reliability. Section 3 presents the proposed SDR-KG-LLM architecture and its supervised deterministic retrieval mechanism. Section 4 concludes the paper and outlines directions for future research.

2. Background and foundations

Recent advances in LLMs have significantly improved the ability of AI systems to process, interpret, and generate natural language, making them attractive front-ends for knowledge-intensive and reasoning-oriented applications. However, despite their fluency and broad generalization capabilities, LLMs remain fundamentally probabilistic models and may produce outputs that are factually incorrect, inconsistent, or unsupported by domain knowledge. This limitation becomes especially critical in high-stakes settings, where reliability, traceability, and semantic correctness are essential. In this section, we first review the probabilistic foundations, capabilities, and limitations of LLMs. We then introduce KGs as structured and semantically grounded knowledge sources, and discuss recent research on KG-enhanced LLMs, with particular attention to their role in improving factual grounding, reducing hallucinations, and

supporting more reliable reasoning [3, 5].

2.1. Large language models: probabilistic foundations and limitations

LLMs generate text based on learned statistical distributions over language. Transformer architectures and self-attention mechanisms have enabled substantial improvements in scalability, contextual representation, and performance across a wide range of NLP tasks [16].

Despite these advances, LLMs remain probabilistic systems whose outputs are generated from learned statistical patterns rather than explicit symbolic verification. Consequently, they may produce variable, factually inaccurate, or hallucinated responses [3]. As discussed in recent surveys on LLM–KG integration and knowledge-enhanced language models [8, 17, 18], LLMs do not natively guarantee that generated statements are grounded in verifiable domain knowledge. This limitation is particularly important in factual and domain-specific settings and motivates the integration of structured external knowledge, especially KGs, to improve factual grounding and reliability.

2.2. Knowledge graphs: structured and semantically grounded knowledge

Knowledge Graphs (KGs) represent knowledge in a structured and machine-interpretable form by organizing facts as graphs of entities and relations, typically encoded as subject–predicate–object triples [19]. Unlike unstructured text, KGs make relations, entity types, and constraints explicit, which supports querying, integration, and reasoning over interconnected information.

KGs can be realized through different graph models, such as directed edge-labelled graphs, RDF graphs, and property graphs, and are used in settings ranging from open and enterprise KGs to domain-specific KGs [19]. In this work, we focus on *ontology-based* KGs built with RDF, RDFS, and OWL, because our framework requires not only factual storage but also explicit classes, properties, hierarchies, and formal restrictions. As discussed by Hogan et al., richer and inferable knowledge representations often require ontologies or rules beyond simple triples alone [19]. This ontology layer is essential in our setting because retrieval relies on modeling patterns such as subclass relations, cardinality constraints, datatype restrictions, and enumerations.

A key component of ontology-based KGs is SPARQL, the standard query language for RDF data. Although SPARQL supports expressive graph querying, writing correct queries remains difficult because it requires knowledge of both query syntax and ontology structure [20]. For this reason, our framework does not rely on unconstrained NL2SPARQL generation, but instead uses supervised ontology-aligned query templates. More broadly, KGs provide semantically grounded external knowledge that can improve factual grounding, reduce hallucinations, and support more reliable reasoning in LLM-based systems [3, 8].

2.3. KG-enhanced LLMs: integration strategies and limitations

Although LLMs provide powerful natural language interfaces, they remain probabilistic models and are therefore prone to hallucinations, generating fluent outputs that may be factually incorrect or unsupported by reliable knowledge sources [3, 8]. To mitigate this limitation, recent research has explored three main strategies for integrating KGs into LLM-based systems: *knowledge-aware learning*, *knowledge-aware inference*, and *knowledge-aware validation* [3]. In the broader integration roadmap, these approaches fall under *KG-enhanced LLMs*, in contrast to *LLM-augmented KGs* and *synergized LLMs + KGs* [8].

Knowledge-aware learning incorporates structured knowledge into model parameters through pre-training or fine-tuning, but it is computationally expensive and does not guarantee controlled reasoning or dependable factual recall [3, 8]. Knowledge-aware validation improves robustness by checking outputs against external knowledge, but it typically acts only after generation and may still miss subtle errors [3, 8]. For this reason, knowledge-aware inference is the most suitable paradigm for our setting, since it grounds the model in external knowledge at generation time without repeated retraining [3, 8].

However, many inference-time approaches still rely on probabilistic retrieval mechanisms such as embedding-based retrieval or document-oriented RAG. While these methods often improve factual grounding, they typically retrieve semantically similar text rather than schema-level, structurally precise knowledge, which weakens provenance, transparency, and structural validity [8]. This limitation is especially important when the external source is an ontology-based KG queried through SPARQL, since unconstrained NL2SPARQL generation can produce syntax errors, schema mismatches, and semantically invalid queries. Even methods that improve generation through metadata retrieval, validation, or repair still retain a generate-then-correct workflow rather than replacing it with a closed and schema-aware retrieval process [8].

In contrast, our work focuses on deterministic knowledge access directly over an ontology-based KG. Instead of injecting textual evidence into prompts or allowing the LLM to generate arbitrary queries, our approach performs controlled retrieval through ontology-aware query templates aligned with the schema and modeling patterns of the KG. This deterministic retrieval mechanism forms the core of the proposed *Supervised Deterministic Retrieval Knowledge-Graph–Augmented LLM* (SDR-KG-LLM) architecture, which aims to provide reliable, reproducible, and explainable access to structured knowledge in high-stakes domains.

3. Proposed framework for deterministic knowledge access

To address the limitations of probabilistic retrieval and unconstrained query generation discussed in the previous section, we propose a *Supervised Deterministic Retrieval Knowledge-Graph–Augmented LLM* (SDR-KG-LLM) framework for reliable access to structured domain knowledge. The framework is designed for high-stakes settings in which answers must remain consistent with explicit constraints, hierarchies, and schema-level semantics encoded in an ontology-based KG. Rather than retrieving textual fragments or allowing the LLM to generate arbitrary queries, the proposed approach maps user intents to a closed set of ontology-aware query templates aligned with recurring modeling patterns in the KG. This enables controlled, reproducible, and schema-safe retrieval over structured knowledge and provides the foundation for the architecture described in the following subsections.

3.1. Domain knowledge representation in ontology-based KGs

In the proposed framework, domain knowledge is represented through domain-specific, ontology-based KGs. Such representations include classes, object and datatype properties, subclass hierarchies, and formal restrictions that capture domain semantics in a machine-interpretable form. This ontology-based structure enables the KG to represent not only factual assertions, but also constraints and recurring modeling patterns that are essential for reliable querying and reasoning.

This representation is central to our retrieval strategy. Rather than retrieving arbitrary graph fragments or unstructured text, the framework accesses the KG through supervised query templates aligned with explicit ontology constructs. For example, component relations can be derived from subclass and object-property restriction patterns, cardinality questions from OWL cardinality constructs, numeric constraints from datatype-property range restrictions, and discrete admissible values from enumerations such as `owl:oneOf`. By grounding retrieval in these formal structures, the framework enables controlled, reproducible, and schema-safe access to structured domain knowledge, which is particularly important in high-stakes applications.

A further advantage of the framework is that it is not tied to a single ontology or KG implementation. Although retrieval is ontology-aware, it does not rely on fixed local names or manually hard-coded vocabulary specific to one KG. Instead, it operates on recurring structural modeling patterns associated with different information needs, such as hierarchies, cardinality restrictions, datatype constraints, and enumerations. During a preprocessing stage, the relevant information needs and structural modeling patterns of the target ontology or ontology family are identified and used to configure the corresponding permitted query structures. Thus, the framework is ontology-agnostic at the architectural level and schema-adaptive through preprocessing. Within a single configuration, equivalent information

needs across the supported ontologies are assumed to be represented using the same or compatible ontology modeling patterns covered by the permitted query structures. If a newly introduced ontology represents an information need through a structural pattern not covered by the current configuration, the corresponding permitted query structures must be adapted or extended. Our implementation on two different ontologies illustrates how the same overall retrieval architecture can be reused across ontologies with compatible modeling patterns without redesigning the core framework.

3.2. SDR-KG-LLM: supervised deterministic retrieval architecture

We propose a multi-stage architecture that integrates KGs into the reasoning pipeline of LLM-driven conversational systems through *supervised deterministic retrieval*. The goal of this architecture is to reduce hallucinations and improve reliability when answering questions over structured domain knowledge. Instead of treating the LLM as a self-contained reasoning engine, the proposed framework separates interpretation, retrieval, and explanation into controlled stages, where the KG acts as the authoritative knowledge source.

Unlike conventional KG–LLM pipelines that rely on document-oriented retrieval or probabilistic matching, the proposed framework performs schema-aware knowledge access through a closed set of deterministic query templates derived from ontology modeling patterns. This design avoids unconstrained query generation, ensures reproducible query behavior, and allows the system to operate reliably even with smaller language models. At the architectural level, the framework operates over recurring ontology modeling patterns rather than ontology-specific vocabulary; the concrete configuration for a target ontology family is described in Section 3.3.

The proposed framework consists of the following stages.

1. **User Input (Conversational Interface)**
   Users interact with the system through natural language queries (e.g., questions about components, constraints, properties, or relationships within a domain). The interface supports conversational interaction in which users can iteratively explore structured knowledge without needing to understand the underlying schema or query language.
2. **Intent (Semantic Interpretation)**
   In the first stage, the user query is interpreted and converted into a structured semantic representation. This step identifies the intent category (e.g., structural relations, numeric constraints, cardinality, enumerated values, or concept descriptions) together with relevant entities, slots, or parameters. The LLM acts as a lightweight semantic parser that translates natural language into a controlled intermediate representation suitable for deterministic retrieval.
3. **Supervised Permitted Query-Structure Selection**
   Instead of allowing unrestricted SPARQL generation, the framework maps the extracted intent to a permitted ontology-aware retrieval structure. Each supported intent corresponds to a known information need and a predefined structural retrieval space. For example:

   - structural relations are retrieved through subclass and OWL restriction patterns,
   - cardinality constraints through OWL cardinality and qualified cardinality expressions,
   - numeric constraints through datatype-property range restrictions represented by `owl:onDatatype` and `owl:withRestrictions` with numeric XSD facets, and
   - enumerated values through `owl:oneOf`.

   At the framework level, this stage can be implemented fully deterministically by mapping each supported intent to a dedicated retrieval function whose SPARQL structure covers the permitted ontology modeling variants for that information need. Alternatively, a constrained LLM can select and instantiate a structure while being restricted to the same predefined structural space. In either realization, the system does not permit unrestricted NL2SPARQL generation.
   We refer to this process as *supervised* because the system operates over a predefined set of intent categories, each linked to a known retrieval template. Depending on the implementation, intent

recognition may be performed in a zero-shot setting or supported with labeled example queries in a few-shot manner, but in all cases the task remains constrained template selection rather than unconstrained query generation.

4. **Deterministic Knowledge Retrieval from the KG**
   The selected template is instantiated and executed as a SPARQL query over the ontology-based KG. Since the query structure is predetermined and aligned with the ontology design, retrieval avoids the instability, syntax errors, and schema mismatches often associated with LLM-generated queries. All retrieved information therefore originates directly from the KG and remains explicitly attributable to the structured knowledge source.
5. **Grounded Natural-Language Response Generation**
   In the final stage, the LLM generates a user-friendly natural language response strictly from the retrieved query results. The model does not introduce external knowledge or speculative facts, but reformulates the structured output into an interpretable answer. This stage can also be performed in a zero-shot manner, since the LLM is used for grounded explanation rather than open-ended factual reasoning.

The proposed architecture shifts most reasoning complexity from the language model to deterministic knowledge retrieval and ontology constraints. As a result, the LLM primarily performs interpretation and explanation rather than factual reasoning. This separation significantly reduces hallucination risks and enables deployment with smaller and more efficient models compared to document-based RAG pipelines.

Figure 1 illustrates the proposed SDR-KG-LLM architecture. The pipeline begins with **Stage A** (*Intent & Slot Extraction*), where the system identifies the user intent and relevant semantic slots from the natural-language query. **Stage B** (*Supervised Deterministic Retrieval*) consists of two sub-stages. In **Stage B1**, the extracted intent is routed to a permitted ontology-aware retrieval structure. This step can be realized either through deterministic retrieval functions or through a constrained LLM operating exclusively within the predefined structural space. In **Stage B2**, the resulting SPARQL query is executed deterministically over the ontology-based KG. The retrieved KG facts are then passed to **Stage C** (*Grounded Response Generation*), which produces a natural-language response grounded in the retrieved information.

3.3. Current implementation and ontology configuration

SDR-KG-LLM is ontology-agnostic at the architectural level, while a concrete deployment is configured for a target ontology or family of ontologies. During preprocessing, the relevant information needs and structural modeling patterns of the target ontologies are identified. These determine the supervised intent space used in Stage A and the permitted ontology-aware query structures used in Stage B. The resulting configuration therefore captures how the information required by the conversational system is represented in the target ontology family, without changing the overall SDR-KG-LLM architecture.

Within one configuration, equivalent information needs across the supported ontologies are assumed to be represented using the same or compatible structural modeling patterns covered by the permitted query structures. However, ontologies may use different domain concepts, labels, local names, and admissible values. For example, an enumerated property such as color may have different local names and different admissible values in two ontologies while both represent these values through a compatible `owl:oneOf` structure. In this case, the same Stage-B query structure can be reused and the ontology-specific entities are resolved dynamically. If a newly introduced ontology represents the same information need through a structural pattern not covered by the current configuration, the corresponding permitted query structure in Stage B must be adapted or extended.

More substantial differences can also affect the semantic interpretation stage. If a new ontology family introduces information needs that cannot be represented by the existing intent space, Stage A can be configured with additional or modified intent categories together with corresponding permitted query structures in Stage B. Hence, a single configuration is not assumed to support arbitrary heterogeneous

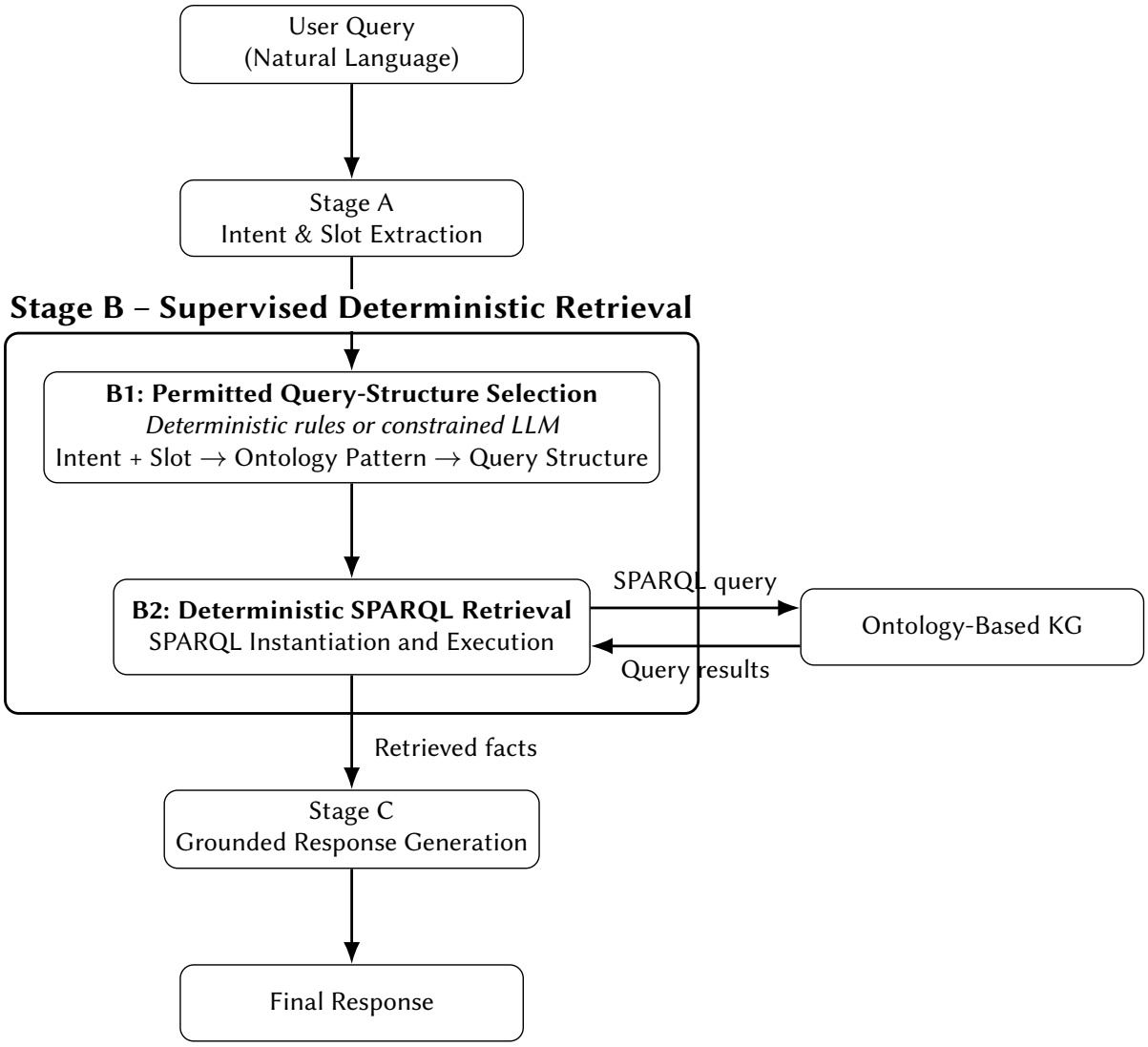


Figure 1: SDR-KG-LLM architecture. Stage A performs intent and slot extraction. Stage B selects a permitted ontology-aware query structure and executes the resulting SPARQL query deterministically over the KG. Stage C generates a natural-language response grounded in the retrieved KG facts.

ontologies without adaptation. Instead, the same SDR-KG-LLM architecture can be instantiated with different configurations for different ontology families, while ontology-specific adaptation remains localized to Stages A and B.

Based on preprocessing and analysis of the target ontologies, the current implementation defines five supervised intent categories: `components`, `cardinality`, `numeric_range`, `enum_options`, and `concept_info`. Table 1 summarizes their corresponding information needs and the main ontology structures permitted in Stage B.

At the framework level, these five intent categories can be realized as five dedicated deterministic retrieval functions, each implementing the permitted SPARQL structure for the corresponding information need. Such an implementation would remove the need for an LLM from Stage B entirely. The current prototype instead uses a constrained LLM in Stage B1 to select and instantiate a permitted SPARQL structure within the predefined structural space, without unrestricted NL2SPARQL generation.

Concrete execution trace. To demonstrate the current implementation, we trace an actual execution of the query "What is the allowed range for plate thickness?" with `Table` as the active product. Table 2 summarizes the information passed through each stage.

For `numeric_range`, the permitted structure is more specific than the presence of

Table 1
Supervised intent and query-structure configuration used in the current implementation.

Intent	Information need	Main permitted ontology structure
`components`	Component relations	OWL restrictions and object-property domain/range patterns
`cardinality`	Required or allowed component counts	OWL cardinality and qualified-cardinality restrictions
`numeric_range`	Continuous numeric constraints	Datatype-property range restrictions (`owl:onDatatype`, `owl:withRestrictions`) with numeric XSD facets
`enum_options`	Discrete admissible values	`owl:oneOf` enumerations in property ranges or restrictions
`concept_info`	Facts about ontology concepts	Type, label, comment, domain, range, and superclass relations

Table 2
Concrete execution trace for a numeric-range query.

Stage	Operation	Observed output
Input	Natural-language query	"What is the allowed range for plate thickness?"
Stage A	Intent and slot extraction	`intent=numeric_range`, `product=Table`, `slot=thickness`, `slot_role=parameter`, `confidence=high`
Stage B1	Permitted query-structure selection	The constrained LLM selects the permitted datatype-property numeric-range structure, which targets datatype-property ranges represented through OWL datatype restrictions and numeric XSD facets.
Stage B2	SPARQL instantiation and deterministic KG execution	The selected structure is instantiated for `slot=thickness`. The SPARQL query resolves `Thickness` as an `owl:DatatypeProperty`, follows its `rdfs:range` to the corresponding datatype restriction, and extracts the supported XSD facets from `owl:withRestrictions`, retrieving `minInclusive=100.0` and `maxInclusive=300.0`. The query also retrieves ontology-defined applicability information, confirming that `Thickness` applies to `Plate`. No fallback retrieval is used.
Stage C	Grounded response generation	"The allowed range for plate thickness is between 100 mm and 300 mm."

`owl:withRestrictions` alone. It first identifies a matching `owl:DatatypeProperty` and its datatype range, and then retrieves only supported numeric XSD facets (`minInclusive`, `maxInclusive`, `minExclusive`, or `maxExclusive`) from direct or equivalent-class OWL datatype restrictions.

The example therefore illustrates the central separation in SDR-KG-LLM: Stage B retrieves the relevant KG facts together with their ontology-defined applicability, while Stage C uses the original question to formulate a grounded response from these retrieved facts.

3.4. Experimental comparison

We conduct a lightweight evaluation on two ontology-based KGs developed for product-configuration scenarios: a Table ontology and a Door ontology. Both ontologies represent product knowledge using RDF, RDFS, and OWL, but contain different domain vocabularies, components, properties, admissible values, and constraints. This allows us to evaluate whether the same SDR-KG-LLM architecture can operate across different domain models while reusing compatible ontology modeling patterns.

As an initial assessment, we compare SDR-KG-LLM with a direct NL2SPARQL baseline in an ontology-only setting, where no document collection or paired natural-language–SPARQL example base is available. A document-oriented RAG baseline would therefore require transforming ontology content into a textual retrieval representation, whereas ontology-aware NL2SPARQL approaches operate directly over the structured KG but still rely on free-form query generation rather than a closed set of ontology-aware templates [12, 13, 15].

OG-RAG, for example, requires both a domain-specific ontology and a document collection: during preprocessing, domain documents are mapped onto the ontology to construct an ontology-grounded hypergraph from which relevant context is subsequently retrieved [4]. Our evaluation setting differs fundamentally because the ontology-based KG itself is the authoritative knowledge source and no accompanying document corpus is available. GraphRAG similarly constructs graph-based representations from unstructured source text and retrieves graph-derived contexts, rather than directly querying a pre-existing ontology-based KG and its OWL constraints through deterministic SPARQL structures [10]. We additionally explored embedding-based retrieval over textual representations of ontology entities for semantic candidate resolution in the same ontology setting. Although this approach was useful for synonym and typo handling, semantic similarity alone occasionally retrieved structurally inappropriate candidates. We therefore retain deterministic SPARQL-based retrieval as the authoritative mechanism for factual knowledge access.

For clarity, we report four criteria: *final correctness*, *Value F1*, *query execution success*, and *average response time*. Final correctness is manually assessed as a binary per-question criterion indicating whether the final response matches the ontology-derived reference answer. Value F1 is a coverage-oriented metric over expected ontology values. Generated answers are normalized for case, punctuation, and common written-number variants, and each expected value group is considered matched if any accepted representation occurs in the answer. Recall is the fraction of expected groups matched, while the current coverage-oriented precision is 1 when at least one expected group is matched and 0 otherwise; Value F1 is their harmonic mean. The metric does not penalize additional unsupported facts. Query execution success is manually recorded as a binary criterion indicating whether the corresponding SPARQL-based retrieval executes successfully. Average response time measures end-to-end runtime for the complete pipeline. The evaluation uses 20 questions in total, with 10 per ontology.

The 20-question core evaluation set was constructed from information explicitly represented in the two ontologies, with ten questions per ontology. The questions were designed to cover the main supported information needs and were refined through discussion with domain experts. Reference answers were derived directly from the corresponding ontology facts. Where possible, analogous information needs were included for both ontologies to assess transfer across different domain vocabularies and ontology structures.

Because the questions were designed around ontology-supported information needs, the evaluation is intentionally task-focused and may favor queries that fall within the currently supported intent space. We therefore regard the experiment as an initial validation of the proposed retrieval mechanism rather than a comprehensive benchmark. Evaluation on larger independently constructed and user-generated question sets, including unsupported and out-of-distribution information needs, remains future work.

Table 3 reports the results for both ontologies. On the Door ontology, evaluated in a zero-shot setting, SDR-KG-LLM clearly outperforms the direct NL2SPARQL baseline, achieving 0.80 final correctness, 0.77 Value F1, and 0.80 query success, while the baseline reaches only 0.10 on all three criteria. On the Table ontology, SDR-KG-LLM again performs strongly with 0.90 final correctness, 0.90 Value F1, and 0.90 query success, compared with 0.40, 0.37, and 0.40 for the baseline. This suggests that the proposed framework can transfer across different ontology structures when compatible modeling patterns are available.

We also repeated the experiments with the smaller model `GPT-4.1-nano`. Although performance decreases, SDR-KG-LLM still consistently outperforms the baseline on both ontologies. On the Door ontology, evaluated in a zero-shot setting, it reaches 0.50 final correctness, 0.43 Value F1, and 0.50 query success, compared with 0.10 on all three criteria for the baseline. On the Table ontology, it achieves 0.60 final correctness, 0.60 Value F1, and 0.60 query success, whereas the baseline reaches only 0.10,

Table 3
Comparison between the direct NL2SPARQL baseline and SDR-KG-LLM on the Door and Table ontologies. The Door ontology is additionally evaluated in a zero-shot setting, and both ontologies are also tested with a smaller LLM to assess robustness under reduced model capacity.

Ontology	Method	Final Corr.	Value F1	Query Succ.	Avg. Time
Door	Direct NL2SPARQL (GPT-4.1)	0.10	0.10	0.10	**3.17**
Door	SDR-KG-LLM (GPT-4.1)	**0.80**	**0.77**	**0.80**	10.17
Table	Direct NL2SPARQL (GPT-4.1)	0.40	0.37	0.40	**4.15**
Table	SDR-KG-LLM (GPT-4.1)	**0.90**	**0.90**	**0.90**	12.08
Table	Direct NL2SPARQL (GPT-4.1-nano)	0.10	0.07	0.10	**2.44**
Table	SDR-KG-LLM (GPT-4.1-nano)	**0.60**	**0.60**	**0.60**	8.51
Door	Direct NL2SPARQL (GPT-4.1-nano)	0.10	0.10	0.10	**2.25**
Door	SDR-KG-LLM (GPT-4.1-nano)	**0.50**	**0.43**	**0.50**	8.71

0.07, and 0.10. These results suggest that the supervised and structurally constrained retrieval design remains more robust than direct free-form NL2SPARQL generation even under reduced model capacity.

SDR-KG-LLM performs best when user queries align with ontology-supported concepts and modeling patterns. When an exact concept or property cannot be resolved, the deterministic fallback broadens the candidate space within the same intent and exposes valid ontology properties, components, or parameters to guide the user toward supported vocabulary.

The current prototype has two main failure modes. First, lexical mismatch can occur when user terminology differs from ontology labels or local names. In this case, the deterministic fallback broadens the candidate space within the detected intent and exposes valid ontology concepts, properties, or parameters. Second, intent misclassification can route a question to an inappropriate retrieval structure; this can be reduced by adding representative few-shot examples to the intent-extraction stage.

Overall, the comparison highlights the difference between free-form SPARQL generation and supervised deterministic retrieval. Whereas the baseline relies on unconstrained LLM-based query generation, SDR-KG-LLM restricts retrieval through ontology-aware templates, resulting in more reliable, grounded, and reproducible ontology-based question answering, including zero-shot transfer to a previously unseen ontology. At the same time, the results suggest that the framework is not inherently dependent on a powerful LLM at the retrieval stage, since the core retrieval process is guided by supervised deterministic mappings and could in principle be implemented fully deterministically, with the LLM reserved only for optional natural-language verbalization of retrieved KG results.

To further assess local deployment feasibility, we evaluate SDR-KG-LLM using several local open-weight LLMs. In this experiment, the retrieval pipeline and ontology configuration remain unchanged, while only the response-generation model is varied. The primary comparison uses the original 20-question evaluation set, comprising 10 questions per ontology. For the subsequent local-model deployment experiment, two additional questions were added to the Table ontology evaluation set, resulting in 22 questions in total.

Table 4 reports the local deployment feasibility of SDR-KG-LLM using open-weight LLMs. All models were deployed on an AI cluster using vLLM on a single NVIDIA A100 GPU with 80GB GPU memory. Llama 3.3 70B AWQ achieves the highest final correctness and Value F1, while Mistral Small 3.2 AWQ provides the best latency–performance trade-off. These results suggest that SDR-KG-LLM can be deployed locally with open-weight models while maintaining reliable ontology-grounded performance. Qwen3-32B exhibits substantially higher latency because its thinking mode was enabled in our deployment, increasing generation time relative to the standard generation settings used for the other local models.

Table 4
Local deployment feasibility of SDR-KG-LLM using open-weight LLMs. Results are aggregated over both ontology-based KGs using 22 ontology-grounded questions.

Model	Parameters	Final Corr.	Value F1	Query Succ.	Avg. Time
Phi-4	14B	0.82	0.917	1.00	28.78
Mistral Small 3.2 AWQ	24B	0.86	0.939	1.00	**13.07**
Qwen3-32B	32B	0.86	0.939	1.00	176.53
Llama 3.3 70B AWQ	70B	0.91	**0.962**	1.00	19.10

4. Conclusion

In this paper, we introduced SDR-KG-LLM, a framework for reliable knowledge access in conversational systems that combines LLMs with deterministic retrieval over ontology-based KGs. Instead of relying on probabilistic document retrieval or allowing LLMs to generate arbitrary queries, the proposed approach introduces a *Supervised Deterministic Retrieval* layer that maps user intents to ontology-aware query templates. These templates are executed directly over the KG using SPARQL, thereby enabling schema-safe, reproducible, and explainable access to structured knowledge.

The proposed architecture separates the conversational pipeline into three controlled stages: intent and slot extraction, deterministic KG retrieval, and grounded response generation. By shifting most of the factual reasoning burden from the LLM to explicit ontology structures and structured queries, the framework reduces hallucination risk while preserving the usability of natural language interaction. In this way, SDR-KG-LLM provides a practical alternative to black-box retrieval and unconstrained NL2SPARQL generation, particularly in high-stakes domains where correctness, traceability, and semantic consistency are essential. An additional deployment study using local open-weight LLMs demonstrated that SDR-KG-LLM remains effective across different model capacities, suggesting that reliable ontology-grounded question answering can be achieved without relying exclusively on proprietary API-based models.

Future work will extend the framework in several directions. First, we plan to evaluate SDR-KG-LLM on a broader range of ontologies and application domains in order to further assess its generalizability across different ontology designs. Second, we aim to automate the preprocessing and pattern-alignment stage so that the framework can adapt more easily to heterogeneous ontology structures. This would reduce manual effort and further strengthen the ontology-agnostic character of the architecture.

5. Limitations

This study evaluates SDR-KG-LLM on two ontology-based KGs and a limited set of configuration-oriented questions. Although the results indicate improved reliability compared with direct NL2SPARQL generation, broader experiments across additional domains, ontology modeling styles, and user populations are needed. The current implementation also assumes that relevant ontology modeling patterns can be identified during preprocessing; highly heterogeneous or weakly structured ontologies may therefore require additional alignment effort. Finally, the evaluation focuses on factual correctness, value overlap, query success, and runtime, but does not yet include a large-scale human evaluation of response usefulness or user satisfaction. The full implementation and project-specific ontology artifacts cannot be released publicly because they were developed in collaboration with industrial partners and are subject to project access restrictions. To support repeatability, this paper therefore documents the evaluation protocol, metric definitions, permitted query structures, model configurations, and a representative end-to-end execution trace.

Acknowledgments

This work was supported by the German Federal Ministry of Education and Research (BMBF) through the FH Kooperativ program [KiMaC, grant no. 13FH555KX1].

Declaration on generative AI

During the preparation of this work, the authors used generative AI tools for language editing and text refinement. The authors reviewed and edited the content as needed and assume full responsibility for the content of the publication.

References

[1] S. Berlik, M. Seidpisheh, A unified framework for intelligent product configuration using knowledge graphs and bayesian networks, in: M. Manns (Ed.), New Paradigms for Anticipated Uncertainty, Springer Nature Switzerland, Cham, 2026, pp. 131–141. doi:10.1007/978-3-032-16889-4_13.

[2] N. Ibrahim, S. Aboulela, A. Ibrahim, R. Kashef, A survey on augmenting knowledge graphs (KGs) with large language models (LLMs): models, evaluation metrics, benchmarks, and challenges, Discover Artificial Intelligence 4 (2024) 76. doi:10.1007/s44163-024-00175-8.

[3] G. Agrawal, T. Kumarage, Z. Alghamdi, H. Liu, Can knowledge graphs reduce hallucinations in LLMs? : A survey, in: K. Duh, H. Gomez, S. Bethard (Eds.), Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 3947–3960. doi:10.18653/v1/2024.naacl-long.219.

[4] K. Sharma, P. Kumar, Y. Li, OG-RAG: Ontology-grounded retrieval-augmented generation for large language models, in: C. Christodoulopoulos, T. Chakraborty, C. Rose, V. Peng (Eds.), Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Suzhou, China, 2025, pp. 32962–32981. doi:10.18653/v1/2025.emnlp-main.1674.

[5] E. Lavrinovics, R. Biswas, J. Bjerva, K. Hose, Knowledge graphs, large language models, and hallucinations: An NLP perspective, Journal of Web Semantics 85 (2025) 100844. doi:10.1016/j.websem.2024.100844.

[6] O. Ovadia, M. Brief, M. Mishaeli, O. Elisha, Fine-tuning or retrieval? comparing knowledge injection in LLMs, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 237–250. doi:10.18653/v1/2024.emnlp-main.15.

[7] J. Sequeda, D. Allemang, B. Jacob, Knowledge graphs as a source of trust for LLM-powered enterprise question answering, Journal of Web Semantics 85 (2025) 100858. doi:10.1016/j.websem.2024.100858.

[8] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, Unifying large language models and knowledge graphs: A roadmap, IEEE Transactions on Knowledge and Data Engineering 36 (2024) 3580–3599. doi:10.1109/TKDE.2024.3352100.

[9] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive NLP tasks, Advances in neural information processing systems 33 (2020) 9459–9474. URL: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

[10] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitansky, R. O. Ness, J. Larson, From local to global: A graph RAG approach to query-focused summarization, arXiv preprint arXiv:2404.16130 (2024).

[11] S. Berlik, M. Seidpisheh, Multi-stage knowledge graph-augmented LLMs for reliable product

configuration, in: Integrated Systems: AI-Augmented Engineering, Springer Nature Switzerland, 2026, pp. 46–56. doi:10.1007/978-3-032-21981-7_4.

[12] D. Allemang, J. Sequeda, Increasing the accuracy of LLM question-answering systems with ontologies, in: International Semantic Web Conference, Springer, 2024, pp. 324–339. doi:10.1007/978-3-031-77847-6_18.

[13] V. Emonet, J. Bolleman, S. Duvaud, T. Mendes de Farias, A. C. Sima, LLM-based SPARQL query generation from natural language over federated knowledge graphs, in: Proceedings of the Special Session on Harmonising Generative AI and Semantic Web Technologies (HGAIS 2024), co-located with the 23rd International Semantic Web Conference (ISWC 2024), volume 3953 of *CEUR Workshop Proceedings*, 2025. URL: https://ceur-ws.org/Vol-3953/355.pdf.

[14] R. Buchmann, J. Eder, H.-G. Fill, U. Frank, D. Karagiannis, E. Laurenzi, J. Mylopoulos, D. Plexousakis, M. Y. Santos, Large language models: Expectations for semantics-driven systems engineering, Data & Knowledge Engineering 152 (2024) 102324. doi:10.1016/j.datak.2024.102324.

[15] M. R. A. H. Rony, U. Kumar, R. Teucher, L. Kovriguina, J. Lehmann, SGPT: A generative approach for SPARQL query generation from natural language questions, IEEE Access 10 (2022) 70712–70723. doi:10.1109/ACCESS.2022.3188714.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[17] W. X. Zhao, K. Zhou, J. Li, et al., A survey of large language models, Frontiers of Computer Science 20 (2026) 2012627. doi:10.1007/s11704-026-60308-3.

[18] X. Wei, S. Wang, D. Zhang, P. Bhatia, A. Arnold, Knowledge enhanced pretrained language models: A compreshensive survey, arXiv preprint arXiv:2110.08455 (2021).

[19] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, et al., Knowledge graphs, ACM Computing Surveys (Csur) 54 (2021) 1–37. doi:10.1145/3447772.

[20] K. B. Cohen, J.-D. Kim, Evaluation of SPARQL query generation from natural language questions, in: D. Maynard, M. van Erp, B. Davis, P. Osenova, K. Simov, G. Georgiev, P. Nakov (Eds.), Proceedings of the Joint Workshop on NLP&LOD and SWAIE: Semantic Web, Linked Open Data and Information Extraction, INCOMA Ltd. Shoumen, BULGARIA, Hissar, Bulgaria, 2013, pp. 3–7. URL: https://aclanthology.org/W13-5202/.