A Provenance-Aware Agentic Framework for Knowledge Graph Augmentation Using RDF 1.2 and SHACL

Alex Vigani[1], Samuel Bustamante Larriet[2], Nina Hosseini-Kivanani[3], José Emilio Labra Gayo[2] and Anisa Rula[1]

[1]University of Brescia, Dept. of Information Engineering, Brescia, Italy
[2]University of Oviedo, School of Computer Science, Calle Valdés Salas, S/N, Oviedo, Spain
[3]University of Luxembourg & Radio Télévisioun Lëtzebuerg (RTL), Esch-sur-Alzette, Luxembourg

Abstract

Knowledge Graph augmentation from unstructured text requires extracted assertions to be aligned with an ontology, semantically checked, and linked to their source. Large Language Models (LLMs) can automate extraction, but may generate unsupported, duplicated, or structurally inconsistent knowledge. We propose an agentic framework in which an LLM interacts with a semantic backend through the Model Context Protocol (MCP) to explore an existing graph, generate RDF 1.2 annotated statements, and use SHACL validation to iteratively correct candidate assertions before integration. Statement annotations record the generating model, the source document, and the validation method, while structural and provenance constraints are enforced through rudof. A pilot prototype implements the core LLM–MCP–SHACL workflow using a fragment of the FoCOSA ontology and `deepseek-chat`, evaluated on a short corpus describing six foods across ten dietary categories. The system produced typed and labelled individuals for all six foods, linked them only to the four diet classes declared in the ontology, correctly omitted the undeclared categories, and emitted a graph that independently passed SHACL validation.

Keywords

Knowledge Graph augmentation, Large Language Models, agentic systems, data provenance, RDF 1.2, SHACL validation, Model Context Protocol, semantic data integration

CEUR-WS.org/Vol-4313/glow-paper9.pdf

1. Introduction

Knowledge Graphs (KGs) support the representation, integration, and reasoning of structured knowledge [1]. However, KG augmentation remains complex because new information must be extracted from heterogeneous sources, aligned with an ontology (a formal schema of concepts and relations), validated, and integrated without breaking semantic consistency. This process is traditionally formulated as a multi-stage pipeline involving knowledge extraction, ontology mapping, entity resolution, entity fusion, data cleaning, and knowledge completion [2, 3], whose stages are usually optimised independently, leading to error propagation and limited interaction across the pipeline [2, 4].

Large Language Models (LLMs) can support this process by extracting entities, relations, and candidate triples from unstructured documents, and recent work has explored agentic workflows that coordinate retrieval, extraction, ontology mapping, and graph population [5, 6, 7]. However, because LLMs are probabilistic, they may generate unsupported, duplicated, or semantically inconsistent assertions [8]. Comparatively little attention has been devoted to integrating provenance management and semantic validation as active components of the construction workflow, leaving the trustworthy integration of LLM-generated knowledge an open challenge.

In this paper we address this gap. Rather than treating provenance as a post-hoc artifact and validation as an *a posteriori* verification step, we model KG augmentation as a single orchestrated

Graph-enhanced LLMs for trustwOrthy Web data management (GLOW), in ISWC 2026 Workshops Joint Proceedings, October 25–26, 2026, Bari, Italy

alex.vigani@unibs.it (A. Vigani); bustamantesamuel@uniovi.es (S. B. Larriet); nina.hosseinikivanani@ext.uni.lu (N. Hosseini-Kivanani); jelabra@gmail.com (J. E. L. Gayo); anisa.rula@unibs.it (A. Rula)
0009-0006-6771-1028 (A. Vigani); 0009-0005-8631-2682 (S. B. Larriet); 0000-0002-0821-9125 (N. Hosseini-Kivanani); 0000-0001-8907-5348 (J. E. L. Gayo); 0000-0002-8046-7502 (A. Rula)



© 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

workflow that integrates ontology exploration, knowledge extraction, provenance management, and SHACL validation [9] within one iterative, LLM-driven process, orchestrated through the Model Context Protocol (MCP) [10]. This design makes provenance and validation active components of the construction process, based on provenance-aware RDF 1.2 assertions and an iterative SHACL-guided generate–repair loop. The main contributions of this paper are:

- A provenance-aware agentic framework for KG augmentation that orchestrates ontology exploration, knowledge extraction, semantic validation, and graph integration through the MCP.
- A provenance-driven construction workflow based on RDF 1.2 annotated statements, in which fine-grained metadata accompany each generated assertion throughout its lifecycle, enabling complete traceability and supporting subsequent construction tasks.
- An iterative SHACL-guided generate–repair loop, implemented through rudof, that validates both generated assertions and their provenance metadata and uses validation feedback to refine candidate triples before integration into the KG.

The remainder of this paper is organised as follows. Section 2 reviews related work on KG construction pipelines, LLM- and agent-based construction, and provenance representation. Section 3 presents the proposed provenance-aware agentic framework, detailing the rudof semantic backend, the provenance model and SHACL shapes, the explore and generate–repair phases, and the use of non-asserted triples for ontology evolution. Section 4 describes a pilot implementation and a preliminary evaluation on a food-and-diet corpus. Section 5 concludes and outlines directions for future work.

2. Related Work

Our contribution intersects three lines of research: the formalisation of KG construction as a multi-stage pipeline, the use of LLMs and agentic architectures to automate it, and the representation of provenance and statement-level metadata.

Knowledge Graph construction pipelines. KG construction is commonly formulated as a multi-stage pipeline comprising knowledge extraction, ontology mapping, entity resolution, entity fusion, data cleaning, and knowledge completion. Zhong et al. [2] survey these tasks and observe that traditional task-specific pipelines, in which stages are optimised independently, are prone to error propagation, motivating more integrated architectures. Hofer et al. [4] similarly argue that, while the individual components of KG construction have benefited from decades of research, their interactions remain largely unexplored, and highlight metadata management as an underdeveloped concern despite its importance for reproducibility and data quality. More recently, the same line of work has been cast as a data-integration problem: Hofer and Rahm [3] introduce a generic pipeline model and the KGI-Bench benchmark, evaluating integration pipelines along coverage, correctness, and consistency. These contributions establish the stages and evaluation criteria of KG construction, but leave the orchestration of those stages—and the role of provenance and validation within it—largely open.

LLM- and agent-based construction. The adoption of LLMs has accelerated KG construction from heterogeneous sources. Recent approaches employ LLMs and agentic architectures to automate entity and relation extraction, ontology mapping, and graph population [5, 6], and roadmaps such as Pan et al. [7] frame the mutual reinforcement of LLMs and KGs as a broader research programme. Many of these systems adopt the reasoning–acting paradigm [11], in which an LLM interleaves reasoning with tool invocations. However, such approaches concentrate on extraction and generation: provenance, when present, is recorded as an output artifact rather than exploited during construction, and validation—if performed at all—is applied only after candidate assertions have been produced. As a consequence, the probabilistic errors characteristic of LLMs [8] are not systematically prevented from entering the graph.

Provenance and statement-level metadata. Provenance is a long-standing concern in KG management, with PROV-O [12] widely adopted to represent the origin, transformation, and reliability of data [13]. At the level of individual assertions, statement-level metadata has been expressed through RDF reification and its successors—the RDF-star model [14], singleton properties [15], and the nanopublication format [16]—now consolidated in the RDF 1.2 statement-annotation mechanism [17]. These works provide the representational foundation on which our framework builds, but they focus on *representing* provenance rather than integrating it into an LLM-driven augmentation process.

Positioning. Our work draws these threads together. We treat KG augmentation as a single orchestrated workflow in which an LLM coordinates exploration, generation, provenance annotation, and validation through the MCP. Provenance, expressed as RDF 1.2 annotations, and SHACL validation are not post-hoc additions but active components of an iterative generate–repair loop, so that information produced at each step becomes contextual input for the next.

3. Approach

Figure 1 illustrates the architecture of the proposed provenance-aware agentic framework. The framework takes three inputs, a target ontology (a T-Box together with its existing A-Box), a set of unstructured source documents, and a catalogue of SHACL shapes [9] — and produces an extended A-Box in which every newly generated statement is enriched with RDF 1.2 provenance annotations, accompanied by a log assembled from the information emitted by the MCP server. This log mitigates the opacity of the LLM by exposing the tools it invoked to produce each new statement (together with their inputs and outputs), the exploration steps performed over the ontology, the validation executions, and the resulting conformance reports. At its core, the framework is an *integration loop* orchestrated by an LLM that interacts with the semantic backend through the Model Context Protocol (MCP) [10]. This loop comprises two nested phases, an *explore* phase and a *generate–repair* phase, whose roles are detailed in Section 3.3.

3.1. rudofMCP as the Semantic Backend

The semantic backend of the framework is rudof,[1] an open-source Rust toolkit for working with knowledge graphs. rudof is exposed to the LLM through rudof_mcp, an MCP server that publishes the library's functionality as tools invocable by any MCP-compatible agent. The framework relies on four groups of tools:

- **Data management**: `load_rdf_data_from_sources` and `export_rdf_data` allow the LLM to load the ontology from files, URLs, or SPARQL endpoints, and to serialise the augmented graph back into standard RDF formats.
- **Node inspection**: `node_info` exposes the outgoing and incoming arcs of a given RDF node, enabling the LLM to navigate the ontology without materialising the entire graph in its context window.
- **Query**: `execute_sparql_query` supports SELECT, CONSTRUCT, ASK, and DESCRIBE queries over the graph, complementing `node_info` as an additional means of exploration.
- **Validation**: `validate_shacl` drives the generate–repair phase; candidate A-Box statements are validated against the SHACL shapes, and the resulting conformance report is returned to the LLM as structured feedback.

By interacting with the graph exclusively through this MCP surface, the LLM cannot manipulate RDF data outside a controlled semantic layer.

[1] https://github.com/rudof-project/rudof

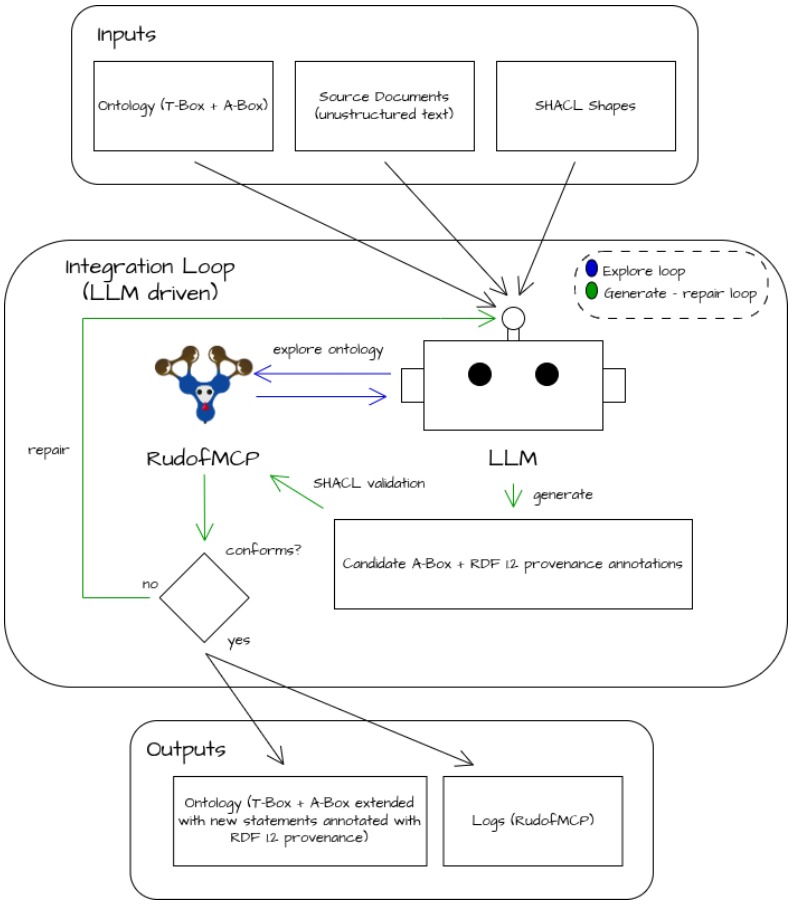


Figure 1: Provenance-aware KG augmentation framework: the LLM drives an integration loop against rudof via MCP, alternating exploration and generation with SHACL-based repair.

3.2. Provenance Model: Vocabulary and SHACL Shapes

To describe how each new assertion was produced, the framework attaches provenance metadata to individual assertions using the RDF 1.2 statement-annotation mechanism [17]: each candidate A-Box triple is emitted together with a set of annotation triples that qualify it with the relevant provenance information. In the general design, this metadata reuses established provenance vocabularies such as PROV-O [12] and the nanopublication model [16], and introduces new terms only where no existing vocabulary captures a required aspect of the generation process; the vocabulary adopted by the current prototype is described in Section 4.

The SHACL shape catalogue supplied to the framework is organised along two complementary dimensions. *Structure shapes* constrain the domain content of the A-Box in the usual sense (cardinalities, datatypes, admissible classes, and value ranges), ensuring that newly generated assertions are consistent with the target ontology. *Provenance shapes* constrain the annotation layer itself: every new statement is required to carry a minimum set of provenance annotations, so that provenance is enforced rather than optional.

3.3. Explore and Generate–Repair Loops

In the *explore* phase, the LLM inspects the T-Box and the current A-Box through `node_info` and `execute_sparql_query` in order to identify the ontology fragment relevant to the current task and to detect entities that already exist in the graph, thereby reducing hallucination and duplication. In the *generate–repair* phase, the LLM produces a set of candidate RDF 1.2 annotated statements, each combining a domain triple with its provenance annotations. These candidates are submitted to `validate_shacl` against both structure and provenance shapes. If the conformance report is positive,

the statements are integrated into the extended A-Box; otherwise, the report is returned to the LLM as structured feedback, which uses it to repair the offending statements. The phase is repeated until conformance is achieved or a predefined stopping criterion (e.g., a maximum number of repair iterations) is reached.

3.4. Non-Asserted Triples for Ontology Evolution

A recurring difficulty in LLM-assisted KG augmentation is that extracted knowledge often refers to concepts, relations, or attributes that are not yet part of the ontology. Discarding such information would waste useful signal, whereas asserting it directly would compromise semantic consistency. The framework addresses this trade-off by exploiting the RDF 1.2 distinction between asserted and non-asserted triples. When the LLM identifies knowledge that appears well supported by the source documents but cannot be aligned with the current T-Box, the corresponding statements are emitted as non-asserted triples: they are materialised in the graph and remain fully addressable, but the graph is not committed to their truth. Crucially, these non-asserted triples still carry the full RDF 1.2 provenance annotation. In this way, the graph accumulates a backlog of provenance-annotated candidate knowledge that an ontology engineer can later inspect and, if deemed valuable, use to drive the extension of the ontology itself.

4. Pilot Implementation and Preliminary Evaluation

A pilot implementation of the proposed design was developed during a two-day hackathon on RDF 1.2 and is available in an anonymised repository.[2] The prototype realises the core mechanism of the framework: an LLM orchestrator coupled to rudof through the MCP and driven by a SHACL-guided generate–repair loop. Several components of the full design are intentionally out of scope at this stage. Retrieval-augmented access to source documents and the non-asserted-triple mechanism for ontology evolution (Section 3.4) are not implemented. The explore phase is not scaffolded as a distinct stage: the LLM is nonetheless exposed to rudof's exploration tools (`node_info`, `execute_sparql_query`) through the MCP and may invoke them at any turn, so exploration is available on demand but not enforced. Finally, the MCP execution log envisaged in the design is not yet built; the prototype emits only an independent SHACL conformance report as an auxiliary output alongside the augmented graph.

4.1. Implementation

The system is a single-binary Rust pipeline organised into three modules. `mcp.rs` is a thin MCP client that spawns `rudof mcp` as a child process and forwards tool calls. `llm.rs` is an OpenAI-compatible tool-calling driver that re-shapes the MCP tool catalogue into function-tool schemas and mediates each LLM turn. `pipeline.rs` is the orchestrator: it pre-loads the ontology into the MCP datastore, builds the prompts, runs the bounded LLM loop (by default, up to 30 tool calls), writes the emitted Turtle graph, and re-validates it on a fresh `rudof mcp` instance to produce an independent conformance report. The LLM used is DeepSeek's `deepseek-chat`.

The ontology supplied to the prototype is a minimal fragment of the FoCOSA food-and-health vocabulary (Listing 1): it declares `focosa-f:FoodItem`, the abstract class `focosa-h:RestrictedDiet` with four concrete subclasses, the object property `schema:suitableForDiet`, and a single seed individual.

Listing 1: Ontology and seed A-Box supplied to the prototype (`inputs/gold_standard.owl`).

```
focosa-f:FoodItem        a owl:Class .

focosa-h:RestrictedDiet  a owl:Class .
focosa-h:VeganDiet       a owl:Class ; rdfs:subClassOf focosa-h:RestrictedDiet .
focosa-h:PescatarianDiet a owl:Class ; rdfs:subClassOf focosa-h:RestrictedDiet .
```

[2] https://anonymous.4open.science/r/provenance-aware_agentic_framework_for_data_integration_into_knowledge_graph

```
focosa-h:GlutenFreeDiet  a owl:Class ; rdfs:subClassOf focosa-h:RestrictedDiet .
focosa-h:LactosaFreeDiet a owl:Class ; rdfs:subClassOf focosa-h:RestrictedDiet .

schema:suitableForDiet
    a owl:ObjectProperty ;
    rdfs:domain focosa-f:FoodItem ;
    rdfs:range focosa-h:RestrictedDiet .

focosa-f:GrilledSalmon
    a focosa-f:FoodItem ;
    rdfs:label "Grilled salmon" ;
    schema:suitableForDiet focosa-h:PescatarianDiet .
```

For simplicity, the prototype adopts an ad-hoc provenance vocabulary limited to three predicates: `llm:LLM_id` (the identifier of the model that produced the assertion), `llm:validationMethod` (the validation method, fixed to `"shacl"`), and `ctx:extractedFrom` (the IRI of the source document). This minimal vocabulary is an implementation expedient; its alignment with the standard provenance vocabularies discussed in Section 3.2, such as PROV-O and the nanopublication model, is left to future work. Two SHACL shapes are defined on top of this vocabulary (Listing 2): a *structure* shape targeting `focosa-f:FoodItem` and requiring reification on every `schema:suitableForDiet` arc (`sh:reificationRequired true`), and a *provenance* shape imposing exactly one occurrence of each of the three predicates on every reifier.

Listing 2: SHACL shapes constraining the RDF 1.2 provenance annotations (`inputs/shapes/provenance_shapes.ttl`).

```
shapes:FoodItemShape
    a sh:NodeShape ;
    sh:targetClass focosa-f:FoodItem ;
    sh:property [
        sh:path schema:suitableForDiet ;
        sh:nodeKind sh:IRI ;
        sh:reifierShape shapes:ProvenanceShape ;
        sh:reificationRequired true
    ] .

shapes:ProvenanceShape
    a sh:NodeShape ;
    sh:property [
        sh:path llm:LLM_id ;
        sh:minCount 1 ;
        sh:maxCount 1 ;
        sh:datatype xsd:string
    ] ;
    sh:property [
        sh:path llm:validationMethod ;
        sh:minCount 1 ;
        sh:maxCount 1 ;
        sh:datatype xsd:string
    ] ;
    sh:property [
        sh:path ctx:extractedFrom ;
        sh:minCount 1 ;
        sh:maxCount 1 ;
        sh:nodeKind sh:IRI
    ] .
```

4.2. Preliminary Results

The prototype was evaluated on a short corpus describing six common foods across ten dietary categories. A representative sentence from the corpus is shown in Listing 3, and the corresponding fragment of the emitted graph in Listing 4: the individual is typed as `focosa-f:FoodItem`, labelled with its surface form, and connected to the four diet resources declared in the ontology through `schema:suitableForDiet` arcs, each carrying the three mandated provenance annotations as an RDF 1.2 reifier.

Listing 3: Sentence from the source corpus (`inputs/food_diet_expansion.txt`).

```
Brown rice, boiled, sits in the grains and cereals category. It is
naturally gluten free and lactose free, and it is widely accepted in
vegan, vegetarian, ovolactovegetarian, pescatarian and omnivore diets,
as well as in halal, kosher and hindu diets. [...]
```

Listing 4: Fragment of the emitted graph corresponding to the sentence in Listing 3 (`outputs/extracted_kg.ttl`).

```
focosa-f:BrownRiceBoiled
    a focosa-f:FoodItem ;
    rdfs:label "Brown rice, boiled" ;
    schema:suitableForDiet focosa-h:GlutenFreeDiet
    {|
        llm:LLM_id           "deepseek-chat" ;
        llm:validationMethod "shacl" ;
        ctx:extractedFrom    <file://.../food_diet_expansion.txt>
    |} ;
    schema:suitableForDiet focosa-h:LactosaFreeDiet {| ... |} ;
    schema:suitableForDiet focosa-h:VeganDiet {| ... |} ;
    schema:suitableForDiet focosa-h:PescatarianDiet {| ... |} .
```

The same pattern applies to the remaining five foods mentioned in the text (whole-grain wheat bread, boiled quinoa, boiled red lentils, firm tofu, and raw almonds), all of which appear in the emitted graph as typed and labelled individuals of `focosa-f:FoodItem`. Diet resources mentioned in the corpus but not declared in the ontology (vegetarian, ovo-lacto-vegetarian, omnivore, halal, kosher, and hindu) were correctly omitted rather than introduced as new vocabulary, as visible in Listing 4, where only the four ontology-declared diets are attached to `BrownRiceBoiled`. The independent verification pass on the emitted artefact returned `sh:conforms true`.

These preliminary results indicate that the LLM–MCP–SHACL workflow can produce ontology-consistent, provenance-annotated assertions that pass independent SHACL validation. We note, however, that in the reported run the emitted graph was already conformant at the first validation, so the repair branch of the generate–repair loop was not triggered; a systematic assessment of repair effectiveness, requiring inputs that induce non-conformance, is therefore left to future work (Section 5).

5. Conclusion and Future Work

This paper presented a provenance-aware agentic framework for Knowledge Graph augmentation from unstructured documents. The framework coordinates ontology exploration, RDF 1.2 annotated statement generation, provenance tracking, and iterative SHACL validation through an LLM and the Model Context Protocol. By integrating provenance and validation into the generation loop, it aims to improve the transparency, traceability, and semantic consistency of LLM-generated knowledge. A pilot implementation on a food-and-diet corpus showed that the LLM–MCP–SHACL workflow can produce ontology-consistent, provenance-annotated assertions that pass independent SHACL validation.

Several directions remain open. First, we will add retrieval over source documents in text, JSON, and RDF formats—parsing, chunking, embedding, and indexing them in a vector store—so that the LLM can ground each generated statement in a specific supporting passage. Second, we will align the ad-hoc provenance vocabulary with established models such as PROV-O and the nanopublication format, and implement the non-asserted-triple mechanism for ontology evolution. Third, we will conduct a systematic evaluation covering assertion correctness, SHACL conformance, provenance completeness, repair effectiveness, runtime, and computational cost, across different datasets and ontologies. This evaluation will include inputs that deliberately induce non-conformance, so as to exercise and quantify the repair branch of the generate–repair loop, which the present pilot did not activate. Finally, we will compare the framework against simpler extraction pipelines and study how different LLM families and model sizes affect assertion quality, repair behaviour, and resource requirements.

Declaration on Generative AI

During the preparation of this work, Claude and ChatGPT were used only for language editing and to improve the readability of the manuscript. These tools were not used to generate the core scientific ideas, experimental data, results, or technical contributions. The authors reviewed, verified, and edited all AI-assisted outputs, approved the final manuscript, and take full responsibility for the integrity of its content.

References

[1] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, ACM Computing Surveys 54 (2021) 1–37. URL: http://dx.doi.org/10.1145/3447772. doi:10.1145/3447772.

[2] L. Zhong, J. Wu, Q. Li, H. Peng, X. Wu, A comprehensive survey on automatic knowledge graph construction, 2023. URL: https://arxiv.org/abs/2302.05019. arXiv:2302.05019.

[3] M. Hofer, E. Rahm, Evaluation of pipelines for data integration into knowledge graphs, 2026. URL: https://arxiv.org/abs/2605.22304. arXiv:2605.22304.

[4] M. Hofer, D. Obraczka, A. Saeedi, H. Köpcke, E. Rahm, Construction of knowledge graphs: Current state and challenges, Information 15 (2024). URL: https://www.mdpi.com/2078-2489/15/8/509. doi:10.3390/info15080509.

[5] S. Pavel, Automatic ontology construction using llms as an external layer of memory (2026). URL: https://zenodo.org/doi/10.5281/zenodo.19696042. doi:10.5281/ZENODO.19696042.

[6] L. Peng, P. Yang, Y. Juexiang, L. Yuangan, The construction and refined extraction techniques of knowledge graph based on large language models, Scientific Reports 16 (2026) 8104. URL: https://doi.org/10.1038/s41598-026-38066-w. doi:10.1038/s41598-026-38066-w.

[7] 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.

[8] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen, L. Wang, A. T. Luu, W. Bi, F. Shi, S. Shi, Siren's song in the AI ocean: A survey on hallucination in large language models, CoRR abs/2309.01219 (2023). URL: https://arxiv.org/abs/2309.01219.

[9] H. Knublauch, D. Kontokostas, Shapes Constraint Language (SHACL), W3C Recommendation, 2017. URL: https://www.w3.org/TR/shacl/, accessed: 2026-07-24.

[10] Anthropic, Model Context Protocol, Specification, 2024. URL: https://modelcontextprotocol.io/specification, accessed: 2026-07-24.

[11] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, Y. Cao, ReAct: Synergizing reasoning and acting in language models, in: International Conference on Learning Representations (ICLR), 2023. URL: https://arxiv.org/abs/2210.03629.

[12] T. Lebo, S. Sahoo, D. McGuinness, PROV-O: The PROV Ontology, W3C Recommendation, 2013. URL: https://www.w3.org/TR/prov-o/, accessed: 2026-07-24.

[13] C. Kosyfaki, P. Groth, An (updated) overview of data provenance: Concepts, challenges and opportunities, in: Companion of the International Conference on Management of Data, SIGMOD Companion '26, Association for Computing Machinery, New York, NY, USA, 2026, p. 563–570. URL: https://doi.org/10.1145/3788853.3801877. doi:10.1145/3788853.3801877.

[14] O. Hartig, B. Thompson, Foundations of an alternative approach to reification in RDF, CoRR abs/1406.3399 (2014). URL: https://arxiv.org/abs/1406.3399.

[15] V. Nguyen, O. Bodenreider, A. Sheth, Don't like RDF reification? making statements about statements using singleton property, in: Proceedings of the 23rd International Conference on World Wide Web (WWW), 2014, pp. 759–770. doi:10.1145/2566486.2567973.

[16] P. Groth, A. Gibson, J. Velterop, The anatomy of a nanopublication, Information Services & Use 30 (2010) 51–56. doi:10.3233/ISU-2010-0613.
[17] W3C RDF-star Working Group, RDF 1.2 Concepts and Abstract Syntax, W3C Working Draft, 2025. URL: https://www.w3.org/TR/rdf12-concepts/, accessed: 2026-07-24.