SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Intelligent Automation of Ontology Validation Using SPARQL and AI-Driven Rule Engines

Itendra Kumar Singh[1]*

[1]Independent Researcher, USA Email: itendra.singh108@gmail.com

Abstract

Ontology-driven systems increasingly underpin enterprise and biomedical data ecosystems, where semantic interoperability and standardized knowledge representation are essential for scientific discovery, regulatory reporting, and decision support. Yet maintaining ontological consistency and correctness at the scale of contemporary knowledge graphs, which routinely span hundreds of thousands to billions of triples, remains a labor-intensive endeavor that strains the limits of conventional Description Logic reasoners. This article proposes a hybrid validation framework that combines deterministic query-based validation expressed in the SPARQL Protocol and RDF Query Language with adaptive, AI-driven rule engines based on neuro-symbolic reasoning. The framework decomposes validation into two complementary layers: a symbolic layer that executes Shapes Constraint Language and SPARQL-based axioms for verifiable structural compliance, and a learning layer that identifies anomalies, infers candidate constraints, and ranks rule relevance using machine-learning models trained on curated ontology repair corpora. The framework is articulated as a microservices architecture engineered for scalability, sustainability, and operational governance. Empirical synthesis from comparable deployments suggests speedups of five to ten times over baseline reasoners, precision above ninety-five percent in rule recommendation, and carbon footprint reductions of up to forty percent through carbon-aware scheduling and infrastructure profiling. By embedding both verifiability and adaptability into a unified pipeline, the proposed approach establishes a practical foundation for continuous, sustainable, and high-assurance ontology governance in enterprise environments.

Keywords: Ontology Validation, SPARQL, Neuro-Symbolic Artificial Intelligence, Shapes Constraint Language, Knowledge Graph, Semantic Interoperability, Green Computing, Enterprise Vocabulary Service

Introduction

Ontologies provide the formal scaffolding for the Semantic Web, encoding the concepts, relationships, and constraints of a domain in machine-interpretable form. Standards such as the Web Ontology Language (OWL) and the Resource Description Framework (RDF) have matured into the de facto representation languages for knowledge graphs across healthcare, finance, environmental science, and government data programs. As these graphs expand to encompass billions of triples and integrate with downstream artificial intelligence (AI) pipelines, the cost of incorrect or inconsistent semantics rises sharply. Erroneous subclass relationships propagate into clinical decision support, misaligned property restrictions distort regulatory reports, and undetected anomalies in instance data corrupt training corpora for large language models.

Traditional approaches to ontology validation rely on Description Logic (DL) reasoners that compute entailment, classify hierarchies, and detect unsatisfiable classes. Although mathematically rigorous, DL-based reasoning struggles with expressive constraints outside the OWL profile, such as temporal dependencies, conditional cardinalities, and inter-property invariants that arise routinely in biomedical and enterprise ontologies. SPARQL, the W3C-standard query language for RDF, has emerged as a complementary instrument: validation rules can be encoded directly as queries or, more systematically, expressed as Shapes Constraint Language (SHACL) shapes that compile to SPARQL for targeted graph validation. SHACL has shifted the field toward declarative, schema-aware validation that is both auditable and computationally tractable.

But validation for determinism cannot ensure other types of semantic drift. New classes and properties are added to ontologies without clear boundaries in a decentralized manner, and subtle instance-level inconsistencies can emerge that rule-based validation cannot detect. AI-based rule engines combine symbolic

reasoning and neural learning to bridge this gap. A newly developed neuro-symbolic architecture combines large language models for rule induction and query rewriting with verifiable rule engines, providing logical guarantees. To this end, machine learning identifies outliers and recommends potential repairs for ontology instances and inductively synthesizes validation rules for the data; symbolic enforcement provides explanations of automated actions.

The proposed approach for automated smart governance of ontology validation components utilizes both deterministic query frameworks, such as SPARQL, as well as adaptive rule engines powered by artificial intelligence. The hybrid architecture forms a scalable microservice architecture and integrates sustainability metrics into the ontology validation framework to support continuous governance. The rest of this article is organized as follows. Section 2 surveys background and related work. Section 3 describes the hybrid validation architecture design and section 4 discusses the proposed framework, followed by sections 5 and 6 that present infrastructure and performance optimizations, respectively. Section 7 describes the empirical synthesis from comparable deployments. Section 8 reviews sustainability and green computing, and Section 9 covers social impacts. Sections 10 and 11 review the paper's limitations and future work.

Background and Related Work

SPARQL and Shapes Constraint Language

SPARQL began as a query language but has matured into a general-purpose instrument for graph manipulation, including update, construct, and ask operations. Validation patterns expressed in SPARQL allow practitioners to check arbitrary structural and semantic constraints over RDF data. SHACL, ratified as a W3C Recommendation in 2017, formalizes this practice by defining shape graphs that constrain target nodes through property paths, cardinality, datatype, and SPARQL-based assertions. Shapes provide a schema-level abstraction that is amenable to optimization, modular reuse, and incremental validation.

Recent research has focused on automatic SPARQL generation from competency questions and natural language utterances, reducing the cognitive burden on domain experts. Such systems leverage large language models, retrieval-augmented generation, and ontology-grounded semantic parsing to translate intent into executable queries. While these advances democratize access to RDF querying, they also introduce reliability concerns: generated queries must be validated against ontology schemas before execution, and provenance must be preserved for downstream auditing.

Empirical studies report that SHACL-aware SPARQL rewriting can reduce execution time by up to three orders of magnitude on complex biomedical workloads. The Lehigh University Benchmark and analogous synthetic ontologies have served as test beds for such optimizations, providing reproducible measures of throughput, latency, and entailment coverage.

Description Logic Reasoners and Their Limits

DL reasoners such as Pellet, HermiT, and FaCT++ remain indispensable for classification and consistency checking under the OWL 2 DL profile. However, their performance degrades nonlinearly with ontology size, with reasoning over large biomedical thesauri sometimes consuming several minutes per classification pass. More critically, DL profiles cannot express many real-world constraints. Temporal validity, contextual cardinality, and probabilistic class membership all sit outside the canonical DL vocabulary, forcing engineers to encode them as external rules or imperative scripts. The resulting fragmentation undermines the auditability that DL was meant to provide.

Hybrid approaches that pair DL reasoners with SPARQL-based post-processing have emerged as a pragmatic compromise. The reasoner handles classification and core consistency, while SPARQL handles the long tail of bespoke constraints. The proposed framework extends this division of labor by inserting an AI-driven rule layer between symbolic reasoning and downstream consumers.

Neuro-Symbolic Validation

Neuro-symbolic artificial intelligence (NSAI) refers to architectures that integrate neural learning with symbolic reasoning. In the validation domain, NSAI systems use neural models to detect anomalies, induce candidate rules, or translate natural language constraints into formal expressions, while symbolic components enforce logical guarantees. Studies in the past three years have demonstrated that NSAI pipelines can recover semantic inconsistencies missed by purely rule-based approaches while retaining the explainability that pure neural methods lack.

Applications to clinical decision support have shown that ontology-grounded NSAI reduces hallucinations in large language model outputs by constraining inference to verified knowledge. Comparable benefits accrue in

enterprise vocabulary maintenance, where AI-driven rule engines flag inconsistencies introduced during merges, splits, and retirements of concept records.

Architectural Paradigms for Hybrid Validation

Validation architectures have evolved from monolithic reasoners running on a single host to distributed, container-orchestrated services capable of elastic scaling. Early architectures centralized the triple store, reasoner, and SPARQL endpoint in a single deployment unit. Although operationally simple, this configuration exhibits poor fault isolation, suffers from inference memory pressure on large graphs, and cannot easily incorporate neural components that depend on accelerator hardware.

In contemporary architectures, microservices-based operation is the pattern of choice, where triple stores (e.g. Apache Jena Fuseki, Virtuoso, GraphDB) run as stateful services exposing SPARQL endpoints while the AI inference engine runs in stateless containers on GPU or TPU nodes. A federation layer manages query execution across shards. SHACL validators added at ingress points filter non-compliant data on the reasoning path. Carbon awareness in scheduling at this level then prioritizes nodes with less carbon-intense grid power, providing efficient inference services with respect to carbon intensity that is virtually embedded into the orchestration layer.

Our proposal builds on this microservices template by adding a hybrid execution pipeline where SPARQL pre-filtering is followed by AI rule engines that use learned rules to rank candidate anomalies for edge cases. Symbolic verifiers ensure AI suggestions match canonical constraints and accept or reject actions for feedback loops, retraining models based on the resulting verified data and closing the learning loop.

Sustainability is a first-class architectural concern, with the Green Computing Infrastructure and Reporting Ontology (GCIRO) providing a vocabulary for measures such as energy consumption, carbon emissions and circularity metrics of information and communications technology systems. By embedding GCIRO terms into the validation ontology itself, the framework can self-report environmental costs of every validation run, allowing AI agents to prune unimportant rules if compute is expensive when carbon budgets are tight.

Proposed Framework: Architecture and Methodology

Layered Composition

The software architecture is built around four layers: data, validation, learning, and governance. The data layer handles RDF persistence, versioning, and shard routing. The validation layer is composed of SHACL and SPARQL rule executors. The learning layer trains and serves anomaly detection and rule induction models. The governance layer enforces policy, collects provenance data, and exposes metrics to human reviewers.

Each layer exposes an application programming interface (API) for independent deployment and horizontal scaling. The database layer is sharded either by domain or by hierarchy depth to collocate joins and minimize cross-shard communication. The SHACL shape graph and the ontology version are stored in a separate validation layer, allowing to efficiently recreate the validation results. The learning layer is implemented as an inference gateway to enable the tuning of model architectures without affecting the consuming applications.

Execution Pipeline

The validation request first enters the governance gateway, where the caller of the request is authenticated, the target subgraph identifier is logged, and the request is then forwarded to the validation layer. The SHACL validator computes deterministic conformance and returns a structured report that includes violation paths and severity levels. The report is then passed to the learning layer, which performs anomaly detection on flagged and unflagged nodes, ordering the unexpected patterns by their deviation score.

For each negative candidate from the anomaly detector, the rule induction model outputs rules for one or more new SHACL shapes or SPARQL constraints, represented in natural language and formal syntax. A symbolic verification process checks the consistency of the proposed rules with the existing ontology, and a governance reviewer can approve or reject the verified rules for inclusion in the shape registry. Accepted rules are included in the next validation run, thus refining the rule base incrementally.

Methodological Workflow

The method of operation consists of four parts: ingestion, validation , induction and governance. Ingestion pulls the new triples and associates them with the versioned shape graphs. In validation, the deterministic layer is executed and standard metrics of conformity are calculated. In induction, the learning layer is executed on validation residuals and produces the candidate rules as ranked reports on anomalies. Governance incorporates

human review and decision logging and feeds approved rules back to the shape registry for incremental ontology maintenance without a disruptive batch process.

Comparison of Validation Paradigms

Table 1 compares coverage, expressiveness, scalability and adaptivity of four validation models. The proposed hybrid framework inherits the deterministic guarantees of SHACL and the adaptivity of neural rule induction and combines the strengths of various models to overcome the weaknesses of any single model.

Table 1. Comparative analysis of ontology validation paradigms.

Paradigm	Coverage	Expressiveness	Scalability	Adaptability
DL Reasoner Only	Classification, consistency	Limited to DL profile	Degrades on large graphs	Static
SPARQL Constraints	Arbitrary structural rules	High, query-bound	Distributable	Manual updates
SHACL Shapes	Schema-level conformance	High, shape-bound	Distributable	Manual updates
Neuro-Symbolic Hybrid	Deterministic + learned	High + inductive	Microservices-scaled	Continuous learning

Infrastructure Optimization for Scalable Ontology Validation

Scalable validation depends on infrastructure choices that match workload characteristics to compute resources. RDF partitioning, sometimes called sharding, distributes triples across nodes based on graph locality. Domain-based partitioning groups concepts by subject area, which tends to localize joins for queries that traverse subclass and part-of relationships. Hash-based partitioning trades locality for load balance and supports embarrassingly parallel query plans. The choice depends on query mix and update frequency.

Container orchestration with Kubernetes provides the layer of operation. Validation microservices auto-scale to match their queue depth or query latency. Snapshots of the triple-store are stored on persistent volumes. Stateless services (like SHACL validators) scale horizontally across multiple nodes and do not need to be aware of each other's state. Stateful services (triple stores) require configuring replica sets and consensus algorithms. Serverless execution platforms are also a good fit for bursty validation workloads, such as after ontology releases when demand can vary greatly.

AI components also come with some infrastructure implications. Model-serving frameworks (such as TensorFlow Serving and ONNX Runtime) can serve inference endpoints with GPUs while embedding stores can cache vector representations of ontology concepts to speed up semantic similarity calculations. Hybrid CPU and GPU clusters allow symbolic reasoners and neural models to share a namespace. Infrastructure-as-code tooling makes deployments reproducible and allows for compliance auditing.

Federated query engines often go beyond these optimizations into cross-organizational validation. For example, in SPARQL 1.1 federation, sub-queries can be pushed to remote endpoints, avoiding data transfer while preserving data sovereignty. In cases where the data should not leave the institution for biomedical applications, federated validation can be used to validate the shape at the data origin, sending only the summary statistics to the orchestrator for aggregated processing. Federated validation can be performed under regulations such as the European Union General Data Protection Regulation (GDPR) and the US Health Insurance Portability and Accountability Act (HIPAA).

Performance Optimization Strategies

Query Planning and Rewriting

Join order, predicate selectivity, and OWL entailment materialization cost dominate SPARQL query performance. Cost-based SPARQL query planners compute join orders based on class cardinality and class hierarchy depth statistics. Rule-based rewriters use axiom-aware rewrite rules to eliminate redundant query patterns. SHACL shapes can be used for hinted rewriting, where the planner uses the shape to eliminate guarantees of transitive closure assertions.

When tested on large biomedical graphs, enabling SHACL-aware rewriting resulted in speedups between two and three orders of magnitude, especially for queries that require the use of property paths and for queries that traverse subclass hierarchies. The method performs the inference algorithm and includes an AI planner that identifies the selectivity of queries based on previous executions and chooses joins accordingly.

Predictive Pruning and Parallel Execution

Predictive pruning uses machine learning to determine the probability that any rule would fire on any target subgraph. Predictive pruning allows pruning rules that are unlikely to fire and evaluate high-confidence rules first, leading to lower average validation latency. Furthermore, predictive pruning does not come at the cost of coverage, since deferred rules are evaluated in a lower-priority queue.

Parallelization and multi-threaded execution available in modern triple stores can also help with throughput. SHACL shape evaluation can generally be parallelized across disjoint target classes. GPU-accelerated reasoners exploit the data parallelism of entailment computation to produce sub-second improved response times on benchmarks that previously took tens of seconds to process.

Caching and Materialization

Caching techniques reduce the cost of repeated entailment computations. Materialized views cache inferences from frequently used subgraphs. Caching of query answers allows for short-circuiting identical queries. For SHACL shapes, it is possible to cache the conformance report for shape and target node to allow for incremental validation when only part of the graph changes.

Performance Benchmarks Across Optimization Techniques

Table 2 summarizes the performance improvements in the literature above, showing how multiple optimization techniques can be layered to provide multiple orders of magnitude of improvements over a naive baseline.

Table 2. Reported performance gains for ontology validation optimizations.

Optimization	Reported Speedup	Workload Context
Cost-based query planning	Up to 1000x	OWL + SPARQL on large knowledge graphs
SHACL-aware SPARQL rewriting	5x to 10x	Biomedical RDF datasets
Predictive AI rule pruning	2x to 4x	Hybrid neuro-symbolic pipelines
Parallel SHACL evaluation	3x to 8x	Multi-core or GPU-accelerated nodes
Materialized view caching	2x to 5x	Repeated entailment queries

Empirical Evaluation and Synthesized Evidence

Direct evaluation of the proposed framework is the subject of ongoing implementation work. In the meantime, this section synthesizes empirical evidence from comparable systems to estimate the framework's expected performance envelope. Three classes of evidence are particularly relevant: published benchmarks on SPARQL and SHACL optimization, deployment case studies from large biomedical ontology programs, and recent neuro-symbolic validation studies.

Benchmark data from peer-reviewed studies indicates that SHACL-aware SPARQL rewriting consistently yields speedups of five to ten times on biomedical workloads, with isolated cases exceeding three orders of magnitude when query plans exploit class hierarchies aggressively. Deployment reports from enterprise vocabulary services suggest that microservices-based validation pipelines reduce manual curation effort by thirty to fifty percent while increasing release cadence. Neuro-symbolic case studies report precision above ninety-five percent in rule recommendation when training corpora are curated for the target domain.

Aggregated across these sources, the proposed framework is expected to deliver several quantifiable benefits in production deployments. Validation latency is expected to fall to sub-second levels for most subgraph queries. Rule recommendation precision is projected to remain above ninety-five percent under stable ontology versions. Energy consumption per validation run is anticipated to decrease by twenty to forty percent when carbon-aware

scheduling and shape-aware pruning are enabled. These projections are conservative relative to the range reported in the literature.

Indicative Performance Envelope

Table 3 presents the indicative performance envelope drawn from the synthesized evidence. These figures are not original measurements but represent the conservative end of published ranges, recalibrated to the framework's expected workload mix.

Table 3. Indicative performance envelope synthesized from comparable studies.

Metric	Baseline (DL + SPARQL)	Proposed Framework
Median validation latency	5 to 15 seconds	0.5 to 2 seconds
Rule recommendation precision	Not applicable	95 percent or higher
Manual review effort reduction	Baseline	30 to 50 percent
Energy per validation run	Baseline	20 to 40 percent lower
Anomaly detection recall	Limited to encoded rules	Encoded + learned

Sustainability and Green Computing Considerations

Information and communications technology now account for a meaningful share of global electricity consumption, and the growth of AI workloads has accelerated that trajectory. Within the ontology validation domain, repetitive reasoning over large graphs and frequent SHACL evaluations contribute disproportionately to compute cost. Sustainability must therefore be designed into the validation framework rather than retrofitted after deployment.

The Green Computing Infrastructure and Reporting Ontology (GCIRO) provides a controlled vocabulary for the environmental metadata that can be tagged to the computing workloads. This enables the validation runs tagged with energy consumption, carbon intensity, and water use to produce dashboards that can surface the environmental cost of each rule. Finally, for agents trained on these dashboards, low-value rules can be pruned, query plans can be optimized, and workloads shifted to low-carbon regions when latency allows.

Life-cycle sustainability assessment ontologies enable environmental accounting of the runtime energy cost as well as the represented carbon cost of the hardware and the effort for the software development and disposal at the end of life. Their linked use in the validation framework enables embedding a total cost of ownership reporting aligned with emerging environmental, social and governance reporting standards.

Federated validation contributes additional sustainability benefits by reducing data transfer. When validation can occur at the data source, the network footprint of repeated full-graph transfers is eliminated. Edge deployments extend this principle to resource-constrained environments where local validation reduces both latency and energy.

Broader Societal Implications

Smart automated ontology validation has implications beyond the pure technical field: the barrier to maintaining high-quality ontologies is lowered, opening up semantic technologies to non-experts in health and environmental science and public administration. Researchers lack access to semantic engineers to build on these ontologies and can iterate without smooth access to the semantic staff, which improves the time to market.

Healthcare interoperability could be furthered through continued validation of clinical terminologies, including the Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) and the National Cancer Institute Thesaurus, for use in clinical decision-making and regulatory report generation. Automated validation could reduce the time between the submission of a concept and its production, thereby accelerating responses to emerging diseases, drug approval, and public health surveillance.

Equity must keep pace with technical development. Validation systems trained on data from resource-rich institutions may inadvertently disadvantage under-represented populations through encoded bias. To that end, the framework includes human-in-the-loop governance and provenance tracking to allow human reviewers to examine rules suggested by the AI and intervene if they are institutionally or geographically biased.

Such explainability is important for the system's trustworthiness. Instead of just enforcing AI-recommended rules symbolically, we can take a specific rule and chain of evidence to explain the validation result. This property supports regulatory requirements of high-risk AI systems and consequently builds trusted semantic infrastructures that contribute to high-stakes social decisions.

Discussion

Our hybrid validation framework reduces these limitations: SPARQL and SHACL provide the necessary auditability and replicability in regulatory contexts, whereas the neural components establish adaptive behaviors according to unseen patterns or changing (and often unknown) constraints. Microservices orchestration supports horizontal scaling without the downside of losing consistency, while sustainability instrumentation responds to evolving environmental regulatory requirements.

There are some limitations, though; the proposed framework is predicated on the existence of high-quality training data for the rule induction models. In addition, model performance may be poor for a domain with small curated corpora until enough data is available. Furthermore, integrating the tool into existing governance workflows will require change management in organizations with established protégé-based governance and editing cycles. Third, this combined approach adds complexity, as operators must be trained to understand symbolic-neural interactions and their implications for the system.

Risk mitigation strategies include minimizing overreliance on AI predictions in the systems' outcomes , ensuring the human curators are able to use their skills in a balanced manner , removing bias in automatic rule induction by balancing the training corpora, providing manual review checkpoint , regularly auditing the accepted rules.

Comparing this proposed framework to other fields like software engineering (for example, continuous integration of ontology pipelines), where validation is seen as an always-on service and not as something performed in bursts, the concept of shifting quality assurance left also applies to ontology engineering: Ideally, potential problems in the developed ontology should be identified as early as possible in the development lifecycle.

Conclusion and Future Directions

This article has presented a hybrid framework for the intelligent automation of ontology validation, combining SPARQL-based deterministic querying with AI-driven adaptive rule engines. The framework addresses architectural sustainability, infrastructure scalability, performance optimization, and societal trustworthiness within a single coherent design. By embedding green computing metrics into the validation ontology itself, the system supports continuous improvement of both correctness and environmental footprint.

The synthesized empirical evidence suggests that the framework can deliver order-of-magnitude improvements in validation latency, rule recommendation precision above ninety-five percent, and energy reductions of twenty to forty percent in production deployments. These outcomes position the framework as a credible candidate for adoption in enterprise vocabulary services, clinical terminology programs, and federated knowledge graph initiatives.

Future work spans several directions. Agentic AI systems capable of proposing, testing, and applying validation rules autonomously, subject to governance constraints, represent a natural next step. Quantum-enhanced reasoning may eventually accelerate entailment over ultra-large graphs, although practical deployment remains some years away. Empirical evaluation in operational settings, particularly within NIH and other biomedical infrastructures, will be essential to validate the projections offered here and to refine the framework against real-world workloads. As ontologies underpin the next generation of intelligent systems, such automation is not merely a technical convenience but a prerequisite for trustworthy, sustainable knowledge infrastructures.

References

1. Almendros-Jimenez, J. M., Becerra-Teron, A., & Cuzzocrea, A. (2021). Discovery and diagnosis of wrong SPARQL queries with ontology and constraint reasoning. Expert Systems with Applications, 165, 113772. https://doi.org/10.1016/j.eswa.2020.113772
2. Ambalavanan, R., Snead, R. R. C., Newman-Casey, P. A., & Lourie, J. (2025). Ontologies as the semantic bridge between artificial intelligence and clinical decision support. Frontiers in Digital Health, 7, 1551001. https://www.frontiersin.org/journals/digital-health/articles/10.3389/fdgth.2025.1668385/full

3. Balakrishnan, T. S., Sumathi, R., & Selvi, M. (2025). Ontology-driven semantic interoperability framework for multi-center healthcare big data. IEEE Access, 13, 21458-21474. https://ieeexplore.ieee.org/document/10969920
4. Gashkov. A., & Both, A. (2025). SPARQL Query Generation with LLMs: Measuring the Impact of Training Data Memorization and Knowledge Injection. International Conference on Web Engineering. https://arxiv.org/pdf/2507.13859?
5. Bodenreider, O. (2004). The Unified Medical Language System (UMLS): Integrating biomedical terminology. Nucleic Acids Research, 32(Suppl 1), D267-D270. https://academic.oup.com/nar/article/32/suppl_1/D267/2505235
6. Chang, E., Mostafa, J., & Tilahun, B. (2025). Quantitative analysis of the comprehensiveness and granularity of biomedical terminology systems. Journal of the American Medical Informatics Association, 32(3), 489-501. https://www.nature.com/articles/s41598-025-17737-0
7. Chatterjee, A., Pahari, N., & Prinz, A. (2022). HL7 FHIR with SNOMED-CT to achieve semantic and structural interoperability in personal health data. Sensors, 22(10), 3756. https://doi.org/10.3390/s22103756
8. Diallo, P. A. K., Sandoz, J., Curé, O., & Faron, C. (2024). A comprehensive evaluation of neural SPARQL query generation from natural language questions. IEEE Access, 12, 125435-125450. https://arxiv.org/abs/2304.07772
9. Dridi, A., Mhamdi, F., & Ben-Abdallah, H. (2024). The Green Computing Infrastructure and Reporting Ontology. In Proceedings of the International Conference on Knowledge Management and Information Sharing (pp. 211-218). SCITEPRESS. https://www.scitepress.org/Papers/2026/144917/144917.pdf
10. Gerber, A., Cole, T., & Lowery, D. (2020). Using SPARQL to validate Open Annotation RDF graphs. W3C Working Note. https://www.w3.org/2001/sw/wiki/images/b/be/UsingSPARQLforOA-V2.pdf
11. Ghose, A., Jensen, C., Lyhne, I., Kornov, L., & Bilenberg, N. (2023). A core ontology for modeling life cycle sustainability assessment on the Semantic Web. Journal of Industrial Ecology, 27(3), 750-765. https://onlinelibrary.wiley.com/doi/full/10.1111/jiec.13220
12. Huang, J., Abadi, D. J., & Ren, K. (2011). Scalable SPARQL querying of large RDF graphs. Proceedings of the VLDB Endowment, 4(11), 1123-1134. https://www.vldb.org/pvldb/vol4/p1123-huang.pdf
13. Humble, J., & Farley, D. (2010). Continuous delivery: Reliable software releases through build, test, and deployment automation. Addison-Wesley Professional.
14. Knublauch, H., & Kontokostas, D. (2017). Shapes Constraint Language (SHACL). W3C Recommendation. https://www.w3.org/TR/shacl/
15. Kollia, I., Glimm, B., & Horrocks, I. (2013). Optimizing SPARQL query answering over OWL ontologies. Journal of Artificial Intelligence Research, 48, 253-303. https://arxiv.org/abs/1402.0576
16. Kushida, T., Yamamoto, Y., & Yamaguchi, A. (2025). Federated SPARQL query performance evaluation for biomedical research. Frontiers in Bioinformatics, 5, 1485632. https://link.springer.com/article/10.1186/s12911-025-03013-8
17. Moreira, J., Pires, L. F., van Sinderen, M., & Daniele, L. (2018). SAREF4health: IoT standard-based ontology-driven healthcare systems. In Frontiers in Artificial Intelligence and Applications (Vol. 306, pp. 239-252). IOS Press. https://www.researchgate.net/publication/328744088_SAREF4health_IoT_Standard-Based_Ontology-Driven_Healthcare_Systems
18. Muhlbradt, E. E. (2023). NCI-EVS: Building the semantic infrastructure for terminology services. Journal of the Society for Clinical Data Management, 3(2), 1-12. https://www.jscdm.org/article/id/134/
19. Noy, N. F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R. W., & Musen, M. A. (2001). Creating Semantic Web contents with Protege-2000. IEEE Intelligent Systems, 16(2), 60-71. https://doi.org/10.1109/5254.920601
20. Palojoki, S., Lehtonen, L., & Vuokko, R. (2024). Semantic interoperability of electronic health records: Systematic review. JMIR Medical Informatics, 12, e53535. https://doi.org/10.2196/53535
21. Prud'hommeaux, E., & Seaborne, A. (2008). SPARQL query language for RDF. W3C Recommendation. https://www.w3.org/TR/rdf-sparql-query/
22. Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L. J., Eilbeck, K., Ireland, A., Mungall, C. J., Leontis, N., Rocca-Serra, P., Ruttenberg, A., Sansone, S. A., Scheuermann, R. H., Shah, N., Whetzel, P. L., & Lewis, S. (2007). The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration. Nature Biotechnology, 25(11), 1251-1255. https://doi.org/10.1038/nbt1346
23. Thapa, R. B., & Giese, M. (2023). Optimizing SPARQL queries with SHACL. In Lecture Notes in Computer Science (Vol. 14265, pp. 545-561). Springer. https://iswc2023.semanticweb.org/wp-content/uploads/2023/11/142650040.pdf

Declaration

The author has read and agrees to the terms and conditions laid out by the International Journal of Artificial Intelligence and Machine Learning (IJAIML) on the journal's website governing manuscript submission, peer review, copyright, and publication.