

Exploring Neuro-symbolic Pipelines for Structured Knowledge Extraction

Nikolaos Laoutaris

December 4, 2025

Contents

1	Introduction	1
2	Systematic Literature Review	2
2.1	Introduction	2
2.2	Methodology	2
2.2.1	Research Questions	2
2.2.2	Search Strategy	3
2.2.3	Inclusion/Exclusion Criteria	4
2.3	Results	5
2.3.1	PRISMA Flow Diagram	5
2.3.2	Data Extraction	5
2.4	Thematic Analysis	5
2.4.1	From Text to Structured Models	5
2.4.2	Automated Logic Generation (Text-to-SPARQL)	7
2.4.3	Validation and Hallucination Control	7
2.5	Discussion and Research Gap	7

Abstract

Abstract here.

Chapter 1

Introduction

Nothing here yet.

Chapter 2

Systematic Literature Review

2.1 Introduction

This chapter details the Systematic Literature Review (SLR) conducted to establish the theoretical foundations of Neuro-Symbolic AI. This chapter analyzes the current state of research in Neuro-Symbolic AI, specifically focusing on how Large Language Models (LLMs) and Knowledge Graphs (KGs) are combined to automate public services. We aim to identify existing approaches for extracting rules from text and generating formal logic (SPARQL/SHACL), highlighting the lack of rigorous regression testing frameworks in current literature.

2.2 Methodology

We used PRISMA guidelines. Overview of the process?

2.2.1 Research Questions

The primary objective of this review is to investigate the applications and reliability of Neuro-Symbolic AI. To achieve this, we defined three specific Research Questions (RQs) that guide the data extraction and synthesis process:

- **RQ1:** How are Large Language Models (LLMs) currently utilized to extract structured knowledge (RDF, Ontologies) from unstructured domain text?
- **RQ2:** What are the state-of-the-art approaches for translating natural language requirements into formal validation logic (such as SPARQL, SHACL, or SWRL)?
- **RQ3:** What methodologies exist for validating the semantic accuracy and syntactic correctness of LLM-generated logic?

RQ1 explores the initial phase of the pipeline (Text-to-Graph), while RQ2 focuses on the core challenge of logic generation. RQ3 allows us to critically analyze how existing studies ensure trust and correctness, identifying the gap that this dissertation aims to address.

2.2.2 Search Strategy

To identify relevant primary studies, we conducted an automated search on the **Scopus** database. Scopus was selected as the primary source due to its extensive coverage of computer science, information systems, and semantic web literature. The search was executed in **December 2025**. The search query was constructed to find the intersection of Generative AI and Semantic Web technologies. We employed Boolean logic to combine three key conceptual blocks:

1. **Generative AI Terms:** ("Large Language Model" OR "LLM")
2. **Target Logic/Language:** ("SHACL" OR "SPARQL")
3. **Symbolic Context:** ("Semantic Web" OR "Knowledge Graph")

These blocks were combined using the AND operator. The final search string applied to the Title, Abstract, and Keywords fields was:

("Large Language Model" OR "LLM") AND ("SHACL" OR "SPARQL") AND ("Semantic Web" OR "Knowledge Graph")

To ensure the review captured the most recent advancements, we applied strict metadata filters during the retrieval phase:

- **Date Range: 2023–2026.** This narrow window was selected because the application of Large Language Models to formal constraint languages (like SHACL) is a nascent field that emerged primarily after the widespread adoption of GPT-4 class models.
- **Language: English.** Only papers written in English were considered to ensure consistent analysis of terminology.
- **Document Type:** We focused on Articles and Conference Papers, excluding trade journals and errata.

This search strategy yielded an initial set of candidates which were then subjected to the screening process described in the following section.

2.2.3 Inclusion/Exclusion Criteria

To ensure the review focused specifically on the intersection of generative AI and structured compliance validation, we established strict inclusion and exclusion criteria. These were applied in two phases: initially to Titles and Abstracts (Practical Screening), and subsequently to Full Texts (Quality Screening). Table 2.1 summarizes the criteria used to select primary studies.

Table 2.1: Inclusion and Exclusion Criteria

Category	Inclusion Criteria	Exclusion Criteria
Domain	Normative Domains: Public Administration, Law, Regulatory Compliance, domain-agnostic pipelines.	Descriptive Domains: Natural sciences, where the goal is pattern discovery rather than rule validation.
Task Focus	Text-to-Graph extraction, Text-to-SPARQL/SHACL generation, GraphRAG architectures.	Pure NLP (summarization), Chatbots without symbolic grounding, low-level graph mechanics (Entity Alignment, Link Prediction), Subgraph Extraction.
Methodology	Neuro-Symbolic architectures (LLM + KG), Prompt Engineering for logic generation, Fine-tuning for query translation.	Traditional Machine Learning (non-generative), Reinforcement Learning without LLMs.
Data Flow	Forward: Transforming unstructured text into formal logic or structured data (Text \rightarrow Logic).	Reverse: Transforming structured data into natural language (Verbalization/Explanation) or pure retrieval without validation.
Mode	Textual inputs with or without pre-processing.	Multimodal studies (Speech/Image), Computer Vision.
Type	Peer-reviewed Articles and Conference Papers.	Conference Proceedings (Meta-entries), Posters, Editorials, non-English papers.

Of the papers sought, some could not be retrieved due to access restrictions. The remaining articles were assessed for eligibility. In this phase, we applied rigorous quality exclusion criteria (QE) to ensure the selected studies contributed specifically to the Neuro-Symbolic compliance pipeline:

- **QE1 (Domain Misalignment):** We excluded studies that relied too heavily on the domain specifics and could not be generalized well.
- **QE2 (Study Maturity & Focus):** We excluded studies that were primarily *benchmarking reports* or *preliminary experiments*. While relevant to the broader field, these papers focused on specific performance metrics rather than proposing novel architectural pipelines.
- **QE3 (Task Relevance):** We excluded studies lacking a specific validation or compliance component.

The next section summarizes the results following this quality assessment.

2.3 Results

From an initial set of 122 records, 14 studies were identified as meeting all eligibility criteria.

2.3.1 PRISMA Flow Diagram

The search and screening process is summarized in the PRISMA flow diagram (Figure 2.1).

Figure 2.1: PRISMA Flow Diagram of the selection process.

2.3.2 Data Extraction

Table 2.2 presents the data extraction summary for the 14 included studies. The studies are categorized by their primary contribution to the neuro-symbolic pipeline: (1) Domain-Specific Pipelines, (2) Automated Logic Generation, (3) Validation Frameworks, and (4) Retrieval (GraphRAG).

2.4 Thematic Analysis

2.4.1 From Text to Structured Models

Current research demonstrates that LLMs are highly effective at the initial 'extraction' phase, successfully mapping unstructured text into RDF or SHACL skeletons.

Table 2.2: Summary of Included Studies (Data Extraction)

Study	Domain / Input	Task	Target Logic	Validation Method
<i>Category 1: Domain-Specific Neuro-Symbolic Pipelines</i>				
Konstantinidis (2025) Konstantinidis2025	Public Service Regulations	Framework Proposal	RDF SHACL +	Conceptual Prototype (No regression testing)
Hanuragav (2025) Hanuragav2025	Clinical Study Reports	Compliance Check	SHACL + SPARQL	Deterministic Rule Execution
Oranekwu (2026) Oranekwu2026	IoT Security (NIST)	Compliance Check	Ontology + SWRL	Ontology-driven Reasoning
Spyropoulos (2025) Spyropoulos2025	Police Reports	Text-to-Graph	RDF Triples	Human-in-the-loop Verification
<i>Category 2: Automated Logic Generation (Text-to-Logic)</i>				
Walter (2026) Walter2026271	General (Wiki-data)	Text-to-SPARQL	SPARQL	Execution Accuracy (Zero-shot)
Soularidis (2024) Soularidis2024	NL Rules	Text-to-SWRL	SWRL	LLM-assisted Generation
Jiang (2025) Jiang202528	Scholarly QA	Text-to-SPARQL	SPARQL	Ontology-Guided Prompting
Mashhaditafreshi (2025) Mashhaditafreshi202536	JSON Data	Modeling	SHACL Shapes	Human Evaluation of Models
Avila (2025) Avila2025223	General QA	Text-to-SPARQL	SPARQL	Benchmark Execution (Auto-KGQA)
<i>Category 3: Validation & Hallucination Control</i>				
Perevalov (2025) Perevalov2025563	Multilingual QA	Query Filtering	SPARQL	LLM-based Probabilistic Filtering
Gashkov (2025) Gashkov2025177	QA Systems	Query Judging	SPARQL	LLM-as-a-Judge
Tufek (2025) Tufek202592	Industrial Standards	Requirement Translation	SPARQL	F1 Score on Logic Translation
<i>Category 4: Retrieval Frameworks (GraphRAG)</i>				
Ongriş (2025) Ongriş2025116	General (Wiki-data)	GraphRAG	SPARQL	Jaccard Similarity
Ahmed Khan (2026) AhmedKhan2026	Data Center Telemetry	Text-to-Query	SPARQL	Execution Accuracy vs NoSQL

2.4.2 Automated Logic Generation (Text-to-SPARQL)

Several studies focus on translating natural language directly into query languages. Walter et al. achieved state-of-the-art results in zero-shot SPARQL generation, while Soularidis et al. explored generating SWRL rules. However, these approaches often struggle with complex, nested logic without guidance.

2.4.3 Validation and Hallucination Control

A critical challenge is ensuring the generated logic is correct. Perevalov et al. propose using an LLM to 'judge' or filter the SPARQL queries. In contrast, Tufek et al. use F1 scores against a gold standard. Crucially, most existing validation methods are probabilistic (LLM-based) rather than deterministic.

2.5 Discussion and Research Gap

While the literature shows success in extraction (2.4.1) and generation (2.4.2), there is a gap in deterministic validation. Papers like Konstantinidis propose the theoretical framework for public services, and Hanuragav applies similar logic to clinical reports. However, no study has yet implemented a comprehensive Mutation Testing framework to rigorously test the structural stability of LLM-generated SHACL shapes for public service eligibility. This dissertation fills that gap.