Triples-Driven Ontology Construction with LLMs for Urban Planning Compliance

Rania Bennetayeb[1,2], Giuseppe Berio[1], Nicolas Bechet[1] and Albert Murienne[2]
[1]*Research Institute of Computer Science and Random Systems, Université Bretagne Sud, Vannes, France*
[2]*Algorithm Department, Institute of Research and Technology b-com, France*

Keywords: Text-to-Ontology, Ontology Learning, Knowledge Graphs, Large Language Models, Few-Shot Prompting, Chain-of-Thought, Triple Extraction, Semantic Web.

Abstract: Ensuring compliance with urban planning regulations requires both semantic precision and fully interpretable decision processes. In this paper, we present a semi-automated methodology that combines the flexibility of large language models with the rigour of Semantic Web technologies to develop an urban planning ontology from regulatory texts. First, the paper presents a systematic evaluation of eight state-of-the-art large language models on the WebNLG dataset for semantic triple extraction task, using few-shot and chain-of-thought prompting. It then discusses the engineering of a domain-adapted prompt. The resulting triples are partially validated through a two-step procedure that takes into account the topological properties of an underlying graph (corresponding to a raw version of a knowledge graph) and the assessment of Human domain experts.

1 INTRODUCTION

Recent advances in artificial intelligence and large language models (LLMs) have significantly improved AI-driven systems for automation. Such systems process large datasets and handle tasks such as summarization, translation, code generation, and question answering (Li et al., 2024). Their use spans from general content generation and chatbots to specialized fields, such as medical diagnosis and legal or technical document analysis (Chattoraj and Joshi, 2024). However, domain-specific tasks require high precision, structured data, and verifiable outputs. Regulatory compliance verification exemplifies this need and can benefit from semantic web (SW) technologies such as knowledge graphs (KGs) and ontologies (Vanapalli et al., 2025). Indeed, these technologies offer formal semantic representations enabling inference, consistency checks, and transparent decision paths, while constraining facts to schemas and supporting neuro-symbolic fact checking by combining neural flexibility with symbolic rigour. LLMs, excelling in language processing tasks and adapting to domains, may complement KGs and ontologies for effectively performing compliance verification, producing evolvable and complete systems.

According to this key idea, we develop a system to verify building permit (BP) applications against the Local Urban Planning (LUP) regulations of Rennes Métropole (RM), France. The system assists instructors in reviewing BPs efficiently while preserving statutory precision. The pipeline ingests an ontology representing both the LUP and the BPs, built semi-automatically using one LLM. The LLM provides suggestions for ontological relationships and concepts or instances in the form of triples (subjet, predicate, objet). Interested readers are invited to consult the figure illustrating the overall architecture via this link.

This paper presents the ontology generation process from the LUP. The main contributions are:

1. An evaluation of eight state-of-the-art (SOTA) LLMs on the triple extraction (TE) task with the WebNLG+2020 (Gardent et al., 2017) dataset, using few-shot prompting and Chain-of-Thought (CoT);
2. A domain-adapted prompt and CoT method with context augmentation, improving triples accuracy and graph connectivity.

The paper is organised as follows: Section 2 reviews the SOTA in TE. Section 3 presents our approach and Section 4 describes the datasets. Section 5 presents the LLM evaluation methodology and results. Section 6 details ontology generation, Section 6.2 covers graph analysis, and Section 7 concludes.

Graphs and prompts are available in the GitHub

Bennetayeb, R., Berio, G., Bechet, N. and Murienne, A.
Triples-Driven Ontology Construction with LLMs for Urban Planning Compliance.
DOI: 10.5220/0013838700004000

In *Proceedings of the 17th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2025) - Volume 2: KEOD and KMIS*, pages 184-191
ISBN: 978-989-758-769-6; ISSN: 2184-3228

repository : Triples-Driven-ontology-construction-with-LLMs-for-Urban-Planning-compliance.

2 RELATED WORKS

Traditional methods for constructing KGs and ontologies typically follow a structured pipeline involving data identification, ontology creation, knowledge extraction, refinement, and maintenance (Tamašauskaitė and Groth, 2023). In the SOTA, knowledge extraction is commonly framed as named entity recognition, classification, relation prediction, and entity disambiguation. However, with the rise of generative language models, this multistep approach has evolved into a more direct TE task, where information is captured as (subject, predicate, object) relationships.

Two main classical TE approaches include : Open Information Extraction (OpenIE) (Kolluru et al., 2020) and Claused Information Extraction (CIE). OpenIE offers a flexible framework capable of extracting information from diverse data sources, without relying on predefined schemas. By contrast, CIE operates within fixed constraints on pre-established schemas. Hybrid approaches that combine schema-free extraction with clause-based constraints have also been proposed to balance flexibility (Del Corro and Gemulla, 2013).

The emergence of LLMs has improved TE capabilities by demonstrating strong natural language understanding and generation abilities. (Petroni et al., 2019) showed that LLMs can act as implicit KBs, retrieving factual information from learned parameters without fine-tuning. However, as noted by (Razniewski et al., 2021), they lack explicit schemas, consistency, and update mechanisms, making them better suited to augment rather than replace KBs.

The use of LLMs for KG and ontology generation is nowadays quite common. Among the works addressing this direction, (Ghanem and Cruz, 2025a) study TE in order to structure extracted facts into KG, comparing fine-tuning and prompting strategies. Other studies, such as (Kommineni et al., 2024) propose a pipeline guided by competency questions with minimal human intervention.

3 GLOBAL APPROACH

The proposed ontology construction process relies on the identification and extraction of semantic triples from LUP. As explained in the Introduction, the designed process benefits from the extensive usage of LLMs. In this sense, to maximize automation, we must carefully select the best performing model. Because no LUP specific annotated dataset exists for evaluating extracted triples, we employ the public WebNLG+2020 dataset, which provides reference sentences annotated with ground truth triples. We then provide a comprehensive LLM evaluation strategy to continuously assess performance of current and future models(5).

The LLM-centred ontology construction process encompasses 4 interconnected components (6):

- **Text processing module**, segmenting documents into semantically coherent chunks, as defined in section 6.1 and performing preprocessing.
- **Knowledge extraction engine**, extracting triples with the selected LLM and ensuring a terminological coherence.
- **Validator**, assessing semantic quality of extracted triples against expert annotations.
- **Graph construction module**, assembling validated triples into one consistent knowledge structure.

Two design points can be highlighted. First, some triples are explicit in the given text. For instance, the sentence “The total area of building named le soleil is about 2330 m^2”, may suggest triple (“le_soleil", “has_total_area", “2330 m^2"). Other implicit relations must be inferred and named by the extraction engine, e.g. (“le_soleil", “is_a", “building"), (“2330 m^2", “has_unit", “m^2") and (“2330 m^2", “has_value", “2330").

Secondly, assembling a coherent (and consistent) ontology requires deciding whether triple elements are concepts or instances, clustering synonymous terms (e.g., “construction" vs. “building"), normalising relation variants (e.g., “in” vs. “includes”), and carefully identifying “is-a” links to build hierarchies.

4 DATA

In the next subsections, WebNLG dataset and LUP document are briefly presented.

WebNLG is an English corpus that pairs RDF triples from DBpedia with crowdsourced reference texts (sets up to seven triples) and, in its 2020 release, spans 16 DBpedia categories (e.g., Airport, Astronaut, Building, City). It can be accessed through Hugging Face’s GEM/WebNLG. Each complete WebNLG dataset entry, consisting of structured triples and their corresponding natural language text, constitutes a sample identified by a unique identifier

named *gem_id*. The WebNLG challenge targets two tasks: RDF → text generation and text → RDF semantic parsing. An example of what dataset entry looks like can be found below.

Sample WebNLG: text → triples.

```
Input:
{'gem_id': 'web_nlg_en−test−864',
 'input': 'Akeem Ayers, who started his career in 2011,
        debuted for the Tennessee Titans.'}

Output:
{'gem_id': 'web_nlg_en−test−864', 'target': ['
      Akeem_Ayers | debutTeam | Tennessee_Titans', '
      Akeem_Ayers | activeYearsStartYear | 2011']}
```

Analysis of the SOTA reveals the relevance of WebNLG's for KG and ontology generation. Text2KGBench assessed fact extraction, ontology conformance, and hallucination rates over a DBpedia–WebNLG subset of 4,860 sentences across 19 ontologies (Mihindukulasooriya et al., 2023). More recently, (Ghanem and Cruz, 2025b) systematically used WebNLG to compare Zero-Shot-Learning (ZSL), One-Shot-Learning (OSL), Few-Shot-Learning (FSL) and fine-tuning for TE, to generate a KG.

LUP is a regulatory document drafted by the Urban Planning Department in RM, available in both Word and PDF formats. It comprises 240 pages and 83,790 words. It is characterized by the specialized administrative language employed in the urban planning domain, which requires specific expertise for proper interpretation. This language manifests through formal terminology, detailed regulatory provisions and constraints. However, the application of regulations exhibits some flexibility through deontic modality, where "must" expresses obligation, "may" expresses possibility and "shall" expresses obligation or permission. This paper focuses on two LUP chapters with quite different content. The first, "Présentation du règlement" (Regulation Overview), contains the main taxonomy, presenting the classification of urban zones and sub-zones alongside with their characteristics and denominations. The second, "Parking" chapter, was selected for its complexity and its coverage of diverse cases and regulations. Additionally, parking compliance requirements are required for the majority of BPs, making this chapter central for compliance checks. A PDF version of the document is available online via link.

5 LLM EVALUATION

In this section, we provide the strategy used for evaluating the eigth relevant LLMs and the metrics used for summarizing the results.

5.1 Evaluation Strategy

An efficient sampling strategy has been adopted for working efficiently. A subset of data in WebNLG has been identified ($N = 150$ distinct identifiers *gem_id*) by randomly selecting from each categorical subset while ensuring all categories ((e.g., sports, geography, movies) being represented and maintaining their associated triple structures. To enhance model performance and output consistency, we have deliberately diversified our sample selection to include various relation types, and incorporated examples containing temporal information (dates) and other specific formats. This diversification strategy has been designed to expose the model to the expected output patterns, thereby facilitating improved normalization of the extracted triples.

We have designed the prompt to specify the system task and its role as an expert in information extraction. The task is decomposed in sequential steps to guide the model through the extraction process. The input format using dictionary structures containing *gem_id* input and target keys, along with the expected output format for RDF triples are also covered by the prompt. Finally, the prompt is enriched with diverse examples, including unit measurements, date formats, and other complex data structures.

The following LLMs: Claude 3.5 Sonnet, Copilot (version 14 February 2025), Gemini 2.0 Flash, GPT-4o, Grok2, Meta Llama3.3 70B Instruct, Mistral Nemo Instruct 2407 and Qwen2.5 72B Instruct have been evaluated in two distinct ways: strict or exact matching (i.e. extracted triples are compared as they are), and similarity-based matching using multiple metrics over extracted triples (Section 5.2). The detailed results are presented in Table 1.

5.2 Similarity Metrics

This section describes the similarity matching metrics used to evaluate extracted triples against expected triples. For each selected *gem_id* (i) the corresponding sentence and gold triple set R_i were paired. Each model then processed the 150 selected samples in batches of 20 and produced for each sample i a predicted triple set S_i.

To reduce surface mismatches between terms, every R_i and S_i are normalized by lower-casing, remov-

ing non-alphanumeric characters, standardising numeric and temporal formats, and trimming whitespaces.

We have implemented two lexical/string similarity metrics. First, the Levenshtein distance (Levenshtein, 1966) is computed and converted to a normalized Levenshtein ratio (Lev). Secondly, a suffix-tree similarity (Stree) is also computed (Marteau, 2018). In both cases, the scores across the sets R_i and S_i have been calculated as:

$$\text{Score}(R_i, S_i) = \frac{1}{|R_i|} \sum_{r \in R_i} \max_{s \in S_i} M(r,s), \;\; M \in \{Lev, Stree\}$$

To overcome more complex differences beyond lexical differences, we have used a pre-trained BERT model (Devlin et al., 2019) to generate semantic embeddings. For each triple $t = (s, p, o)$, we independently extract the embeddings of the "subject", "predicate" and "object" of the four final hidden states of BERT (producing vectors $\mathbf{e}_s, \mathbf{e}_p, \mathbf{e}_o \in \mathbb{R}^d$). Let a reference triple be $r = (s_r, p_r, o_r)$ and a predicted triple $s = (s_t, p_t, o_t)$. We define the semantic similarity between r and t as the average cosine similarity of the corresponding components:

$$\begin{aligned}\text{sim}_{\text{sem}}(r,s) \;=\; \frac{1}{3}\Big(&\cos(\mathbf{e}_{s_r}, \mathbf{e}_{s_s}) \\ &+ \cos(\mathbf{e}_{p_r}, \mathbf{e}_{p_s}) \\ &+ \cos(\mathbf{e}_{o_r}, \mathbf{e}_{o_s})\Big)\end{aligned}$$

For each predicted triple $s \in S_i$, the best match score m is defined as:

$$m(s) = \max_{r \in R_i} \text{sim}_{\text{sem}}(r,s),$$

and the set of accepted predicted triples at threshold τ is defined as:

$$A_i(\tau) = \big\{\, s \in S_i \;\big|\; m(s) \geq \tau \big\}$$

where a threshold. $\tau = 0.84$ has been fixed to guarantee an acceptable level of similarity.

Precision (*Pre*), recall (*Rec*) and F_{1score} (F_1) for sample i are then computed as:

$$\text{Pre}_i(\tau) = \frac{|A_i|}{|S_i|}, \; \text{Rec}_i(\tau) = \frac{|A_i|}{|R_i|},$$

$$F_{1\ i}(\tau) = \frac{2\,\text{Pre}_i(\tau)\,\text{Rec}_i(\tau)}{\text{Pre}_i(\tau) + \text{Rec}_i(\tau)}$$

$|A_i|$ number of predicted triples whose maximum similarity to any reference triple is $\geq \tau$.

$|S_i|$ total number of predicted triples for sample i.

$|R_i|$ total number of reference triples for sample i.

Finally, we compute macro and micro-averaged precision and recall over all the N extractions. Let's consider the all retrieved triples as the best matching triples (Ta), the predicted triples (S) and the reference (expected) triples (R) and the corresponding cardinalities:

$$T_A = \sum_{i=1}^{N} |A_i(\tau)|, \quad S = \sum_{i=1}^{N} |S_i|, \quad R = \sum_{i=1}^{N} |R_i|.$$

Then, the *global* (micro-averaged) *Pre*, *Rec* and F_1 are defined as:

$$\text{Pre}_{\text{global}}(\tau) = \frac{T_A}{S}, \qquad \text{Rec}_{\text{global}}(\tau) = \frac{T_A}{R},$$

$$F_{1\ global}(\tau) = \frac{2\,\text{Pre}_{\text{global}}(\tau)\,\text{Rec}_{\text{global}}(\tau)}{\text{Pre}_{\text{global}}(\tau) + \text{Rec}_{\text{global}}(\tau)}$$

Macro-averaged metrics are then computed as the arithmetic mean over samples:

$$\text{Pre}_{\text{macro}}(\tau) = \frac{1}{N} \sum_{i=1}^{N} \text{Pre}_i(\tau)$$

$$\text{Rec}_{\text{macro}}(\tau) = \frac{1}{N} \sum_{i=1}^{N} \text{Rec}_i(\tau)$$

$$F_{1,\text{macro}}(\tau) = \frac{1}{N} \sum_{i=1}^{N} F_{1,i}(\tau)$$

Table 1 summarises the comparative performance of the eight evaluated LLMs on TE task. Each row reports overall scores and individual results for strict matching, semantic similarity, suffix-tree and Levenshtein metrics. Notably, Claude 3.5 Sonnet shows the best results across all metrics.

6 APPLICATION TO LUP CORPUS

Following quite satisfactory preliminary results obtained with Claude Sonnet 3.5, we have upgraded to Claude Sonnet 4 for the TE task on the LUP. This decision has been motivated by several key improvements documented in the literature. Claude Sonnet 4 represents a significant upgrade over its predecessor, delivering superior reasoning capabilities while responding more precisely to complex instructions. These enhancements are quite relevant for sophisticated natural language processing tasks such as triple extraction, where understanding contextual relationships between entities is crucial for accurate knowledge representation.

Claude 4's context window has also been expanded to 200k tokens, making it ideally suited for

lengthy documents, generating triples without truncation or ever cutting off part of the output. Even if specific benchmarks for French triple extraction are not available in the current literature, Claude 4 demonstrates slight improvements in multilingual Q&A tasks, making it relevant for our French regulatory text processing task.

6.1 LUP Segmentation and Shots Preparation

The LUP is structured in chapters, sections, subsections, and sub-subsections. Content appears in paragraphs, lists, and cross-references. This interconnected structure makes sentence-level triple extraction ineffective because of the usage of several implicit or explicit references within or across distant sections. For instance, the subsection *"Areas to be Urbanized: AU zones"* states that *"Two types of AU zones are distinguished"* without naming them, while the next subsection *"Zone 1AU"* gives details but never mentions its implicit inclusion within AU zones. Processing isolated sentences or subsections thus breaks logical links (e.g., ("Zone_AU", "contains", "sub_Zone_AU1")), weakening coherence and connectivity. Conversely, processing the entire document at once leads to a quite limited number of triples. Thus, a balance is needed between the maximum text size an LLM can process and the minimum size required to preserve completeness.

To address this key point, we have implemented an iterative segmentation strategy to maintain semantic coherence while ensuring model efficiency. The document is divided in sections, with titles included; images are excluded, and tables are set aside. Using Claude's tokenizer, sections exceeding the token limit are divided in balanced chunks. If a chunk exceeded 500 tokens, it is further split starting from capital letters to the first occurrence of a colon (":"), as natural boundaries.

The first extraction has covered the seven initial chunks (chapter 1), introducing key terms, acronyms, and general guidelines. While some irrelevant triples have been generated, the extraction provided:

- Fundamental entities and relations forming the ontology's top layer;
- A high-level taxonomy of the urban planning domain.

In order to maximize the quality of the extracted triples, we selected sentences from the "Parking" chapter. Subsequently, these sentences were manually annotated by a domain expert. The annotations encompassed implicit-to-explicit relations, quantitative constraints (e.g., distance or height), and vague formulations (e.g., "immediate surroundings"), which are unsuitable for precise representation. To improve FSL, examples containing such vague formulations were deliberately included in the prompt set, with the objective of providing guidance to the model. The domain expert also normalised vocabulary and added implicit predicates where necessary to ensure consistency and accuracy. The resulting sentence–triple pairs served as shots for prompting. Figure 1 illustrates one such annotated example.

6.2 Prompt Engineering

We have defined two distinct methods for processing text chunks to extract triples. In the first method, each chunk has been treated independently: the model receives one chunk at a time and extracts triples based only on the content within that chunk. In the second method, the chunks are still processed individually, but the model is made aware of triples extracted from all previously processed chunks. This setup allows us to compare the impact of providing contextual information such as the previously extracted triples.

Both methods employ an almost identical prompt, with one key difference: the context-aware method comprises a dedicated section injecting the previously extracted triples. This contextual information is accompanied by specific instructions guiding the model to maintain terminological consistency and to ensure connections with previously identified or generated terms whenever possible.

We have adopted a FSL with five shots as described in subsection 6.1. However, rather than simply asking the model to extract triples, we have developed an enhanced CoT approach breaking down the task into well-defined sequential steps. This structured strategy has emerged from extensive experimentation where we iteratively refined the instructions to better guide the model.

The model has been configured with a temperature setting of 1, which is mandatory for activating Claude's reasoning capabilities. The reasoning budget has been set to 5000 tokens, providing the model with sufficient resources for the complex multiple step analysis. Finally, we have used XML tags (e.g., <triple> ... </triple>) to delimit portions and structure the prompt. This strategy, recommended in Anthropic's guidelines, creates clear boundaries between prompt sections, reduces ambiguity, and improves parsing of responses. The full prompt is provided in both English and French via this link.

Table 1: Performance metrics.

Model	Strict matching			Semantic similarity			STree	Levenshtein
	P	R	F1	P	R	F1		
claude_3.5 sonnet	58,91	58,84	58,88	88,70	90,98	89,36	92,92	90,12
Gemini 2.0 Flash	49,97	49,58	49,77	87,95	91,57	89,30	88,04	87,17
Grok 2	42,80	42,80	42,80	80,14	86,33	82,34	86,51	84,79
GPT 4o	42,33	41,67	42,00	78,30	84,81	80,57	85,68	84,04
meta-llamaLlama-3.3-70B-Instruct	40,85	40,71	40,78	77,53	85,42	80,11	85,33	83,02
copilot	39,84	39,54	39,69	73,71	81,39	76,65	84,03	82,18
Qwen2.5-72B-Instruct	35,52	35,79	35,66	59,63	64,30	61,04	84,78	82,47
Mistral-Nemo-Instruct-2407	30,57	30,29	30,43	71,18	78,99	73,87	81,00	80,10

Text :
"Les emplacements de stationnement exigés doivent être réalisés sur le terrain d'assiette de la construction ou dans son environnement immédiat. Dans ce cas, ils doivent être facilement accessibles à pied et situés à moins de 300 m du terrain de la construction pour la destination Habitation"

Triples:

Location constraints : `(emplacement_stationnement, situé_sur, terrain_assiette_construction)`

Accessibility requirements : `(moyen_accès, à_type, à_pied)`

Distance limitations : `(emplacement_stationnement, à_distance_de, terrain_construction)`

Figure 1: Example of annotated text in RDF triples.

6.3 Triple Validator

The process of constructing a coherent ontology depends on the quality of extracted triples. Since we lacked reference triples for the LUP corpus (unlike (Debattista et al., 2016) and (Ghanem and Cruz, 2025b)), the validator component operates in two distinct and complementary ways, presented below.

6.3.1 Graph Based Method Validator

We first compare the two extraction methods (context-less and context-aware) by constructing graphs from the extracted triples and analysing their topological properties using NetworkX (Hagberg et al., 2008). Indeed, graphs underlying the extracted triples represent the raw ontology and therefore should exhibit desirable topological properties, highlighted by:

Connectivity Analysis: identification of weak and strong connectivity and isolated knowledge clusters;

Structural Quality: detection of isolated terms, measurement of graph density and compactness;

Centrality Analysis: identification of important or highly connected nodes, revealing terms that correspond to potential key domain entities.

The results are presented in subsection 6.4. Triples produced by the method generating the graph with the best topological properties have then been submitted to the expert validator described below.

6.3.2 Expert Validator

A qualitative assessment has been performed by asking two domain experts to validate the extracted triples. Following precise guidelines and examples, they have been asked to classify each extracted triple in one or more of the following categories:

Category 0: incorrect triples that do not appear in the reference chunk or any previously extracted triples, or that are semantically meaningless (e.g., those relying on vague notions such as "immediate surroundings" or "in close proximity" without precise context);

Category 1: correctly formulated triples whose information is directly sourced from the input text;

Rule Category: triples expressing regulatory rules that contain numeric constraints;

Correction Category: triples violating normalization rules defined in the prompt's CoT steps, such as predicates not formulated affirmatively or those including deontic terms ("must", "may", "requires");

Pertinence: noisy triples that are not relevant for verifying the validity of a BP.

It should be noted that, even with a great insightful knowledge and experience, domain experts can still be biased and their understanding of triples may be partial. Consequently, additional validation methods should be developed. However, the graph method validator can be reapplied to assess the global impact of expert validator.

6.4 Graph Validator Results

Table 2 presents the evaluation of topological properties for the two graphs under consideration: Graph 1 corresponds to the context-less extraction method, and Graph 2 corresponds to the context-aware extraction method.

Table 2: Comparison.

Graph / Metric	Graph 1	Graph 2
Cleaned triples	278	331
Nodes	266	257
Edges	278	331
Disconnected triples	True	False
Density	0.0039	0.0050
Connected components	23	1
Main connectivity component	0.33	1

Graph 2 comprised 257 nodes and 331 edges, whereas Graph 1 comprised 266 nodes and 278 edges. Although Graph 1 exhibited a slightly higher node count (+3.5%), Graph 2 showed a greater number of edges (+19%), reflecting enhanced concept interconnectivity.

Notably, neither graph contained isolated nodes (i.e., nodes with degree zero): every node participated in at least one edge. The improvement in interlinking is further reflected by graph density: Graph 2 achieved a density of 0.00503 compared to 0.00394 for Graph 1, indicating a richer interconnection between potential concepts and instances. The graphs are available online (see Figure 2 and Figure 3).

Also, when edge direction was ignored (i.e., considering the graphs as undirected), Graph 2 formed a single cohesive component: it was fully weakly connected with a `largest_component_ratio` of 1, ensuring that all potential concepts/instances and predicates are reachable across the entire graph. Conversely, Graph 1 is split into 23 disconnected subgraphs, with the main component covering only 33% (88 out of 266) of nodes. This fragmentation degrades inference, SPARQL queries, and global reasoning, as many entities exist in isolated "semantic silos".

6.5 Graph Validation after Expert Validation

As noted above, the processed introductory LUP chapter contained a large amount of information that was not relevant for BP compliance verification. In particular, experts judged the first 111 extracted triples as irrelevant; some triples were also corrected. We therefore recomputed the topological metrics to assess the impact after expert validation. The most important results concern graph connectivity and are shown in Table 3.

It can be noted that, despite extensive triple removal and modification, the majority of the validated triples fall into two large, coherent subgraphs: Component 1 contains 91 triples (41.7%) spanning 85 nodes, and Component 2 contains 122 triples (56.0%) spanning 95 nodes. Together, these two components account for 97.7% of all generated triples in the graph. Components 3 and 4 represent residual fragments. The presence of these minor components suggests residual "semantic silos" — isolated facts or edge cases that are not connected to the core graph and which therefore require further analysis and more extractions of triples from the next chunks in this chapter.

Table 3: Distribution of triples on connected components.

Components	Triples	% of Total	Nodes
1	91	41,7%	85
2	122	56,0%	95
3	1	0,5%	2
4	4	1,8%	5

7 CONCLUSIONS

This paper describes a comprehensive method for semi-automatic domain ontology construction from regulatory documents using LLMs. A systematic evaluation of eight SOTA LLM platforms on the WebNLG dataset leads to a triple extraction performance-driven selection of the LLM.

The proposed domain-adapted prompt engineering strategy, combined with optimized document segmentation, preserves both semantic coherence and terminological consistency. Additionally, the experimented context-augmentation is promising even if facing scalability issues as the number of extracted triples increases, and specifically whenever document chunks contain diverse themes that introduce irrelevant information for subsequent chunks.

To partially address this limitation, future work will implement triple selection mechanisms using semantic similarity measures to determine which previously extracted triples are relevant to include as context for the chunk being processed (Papaluca et al., 2024). The next phase of the work will focus on extracting triples from all document segments and organizing them in a hierarchical ontology. Given that the LUP contains several normative rules and constraints, future development will integrate deontic logic modelling capabilities. We will employ OWL-

DL for expressing basic constraint definitions and taxonomic relationships, while leveraging the Semantic Web Rule Language (Lawan and Rakib, 2019) for encoding complex regulatory rule patterns exceeding OWL's expressivity. This will be complemented by Shapes Constraint Language rules for automated compliance validation. The integration of these formal logic frameworks will enable the ontology to systematically verify whether BPs satisfy regulatory requirements by encoding both structural and semantic constraints.

Finally, incorporating provenance metadata will ensure traceability of each ontology element back to its originating text segment in the source document. This provenance will facilitate precise updates when regulations evolve and ensure long-term reliability for automated compliance verification applications.

REFERENCES

Chattoraj, S. and Joshi, K. P. (2024). MedReg-KG: KnowledgeGraph for Streamlining Medical Device Regulatory Compliance. In *4th Workshop on Knowledge Graphs and Big Data in Conjunction with IEEE BigData 2024*. IEEE.

Debattista, J., Auer, S., and Lange, C. (2016). Luzzu—a methodology and framework for linked data quality assessment. *J. Data and Information Quality*, 8(1).

Del Corro, L. and Gemulla, R. (2013). Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 NAACL HLT*, pages 4171–4186. Association for Computational Linguistics.

Gardent, C., Shimorina, A., Narayan, S., and Perez-Beltrachini, L. (2017). Creating training corpora for NLG micro-planners. In Barzilay, R. and Kan, M.-Y., editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Ghanem, H. and Cruz, C. (2025a). Fine-tuning or prompting on llms: evaluating knowledge graph construction task. *Frontiers in Big Data*, 8:1505877.

Ghanem, H. and Cruz, C. (2025b). Fine-tuning or prompting on llms: evaluating knowledge graph construction task. *Frontiers in Big Data*, Volume 8 - 2025.

Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008). Exploring network structure, dynamics, and function using networkx. In Varoquaux, G., Vaught, T., and Millman, J., editors, *Proceedings of the 7th Python in Science Conference*, pages 11–15, Pasadena, CA USA.

Kolluru, K., Adlakha, V., Aggarwal, S., Chakrabarti, S., et al. (2020). Openie6: Iterative grid labeling and coordination analysis for open information extraction. *arXiv preprint arXiv:2010.03147*.

Kommineni, V. K., König-Ries, B., and Samuel, S. (2024). From human experts to machines: An llm supported approach to ontology and knowledge graph construction. *arXiv preprint arXiv:2403.08345*.

Lawan, A. and Rakib, A. (2019). The semantic web rule language expressiveness extensions-a survey. *arXiv preprint arXiv:1903.11723*.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707.

Li, Z., Fan, S., Gu, Y., Li, X., Duan, Z., Dong, B., Liu, N., and Wang, J. (2024). Flexkbqa: A flexible llm-powered framework for few-shot knowledge base question answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):18608–18616.

Marteau, P.-F. (2018). Sequence covering similarity for symbolic sequence comparison. *arXiv preprint arXiv:1801.07013*.

Mihindukulasooriya, N., Tiwari, S., Enguix, C. F., and Lata, K. (2023). Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text. In Payne, T. R., Presutti, V., Qi, G., Poveda-Villalón, M., Stoilos, G., Hollink, L., Kaoudi, Z., Cheng, G., and Li, J., editors, *The Semantic Web – ISWC 2023*, pages 247–265, Cham. Springer Nature Switzerland.

Papaluca, A., Krefl, D., Rodríguez Méndez, S., Lensky, A., and Suominen, H. (2024). Zero- and few-shots knowledge graph triplet extraction with large language models. In Biswas, R., Kaffee, L.-A., Agarwal, O., Minervini, P., Singh, S., and de Melo, G., editors, *Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024)*, pages 12–23, Bangkok, Thailand. Association for Computational Linguistics.

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., and Miller, A. (2019). Language models as knowledge bases? In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Razniewski, S., Yates, A., Kassner, N., and Weikum, G. (2021). Language models as or for knowledge bases. *CoRR*, abs/2110.04888.

Tamašauskaitė, G. and Groth, P. (2023). Defining a knowledge graph development process through a systematic review. *ACM Transactions on Software Engineering and Methodology*, 32(1):1–40.

Vanapalli, K., Kilaru, A., Shafiq, O., and Khan, S. (2025). Unifying large language models and knowledge graphs for efficient regulatory information retrieval and answer generation. *COLING 2025*, page 22.