

Seminar Report

Using Large Language Models to Generate Knowledge Base: LLM-Based Knowledge Graph Construction for Whole Slide Image Representation Workflow

Jihoo Chung

MatrNr: 20771035

Supervisor: Aasish-Kumar Sharma

Georg-August-Universität Göttingen
Institute of Computer Science

September 30, 2026

Abstract

Recent advancements in Large Language Model (LLM) have enabled automated Knowledge Graph (KG) construction, transforming complex literature into structured representations for specialized domains. However, while KGs have succeeded in biomedical domains, constructing reliable domain-specific KGs from literature on Whole Slide Image (WSI) representation workflows remains unexplored due to the highly specialized nature of the domain and the complex dependencies inherent in multi-step pathology pipelines.

To address these limitations, we propose an LLM-driven KG construction framework optimized for WSI representation workflows. By combining targeted prompt engineering and custom ontology refinement, the framework systematically mitigates hallucinations while preserving broad semantic coverage.

Our evaluations demonstrate that prompt and ontology refinement with input context expansion progressively enhance extraction fidelity, raising the overall F_1 -score. However, error analysis reveals that expanded context induces aggressive extractions, increasing false positives such as hallucinated triples and schema violations in complex workflows.

In summary, our key contributions are as follows: (1) a foundational LLM-based KG construction pipeline for WSI representation workflows, (2) a domain-targeted custom ontology and prompting strategy, and (3) a manual error analysis identifying key extraction failure patterns for the target domain.

Code Availability: <https://github.com/JihooChung/wsiKG>

I hereby declare that all uses of tools and AI as well as assistance from third parties in this work have been fully and accurately disclosed in the Transparency Report in Appendix Section B.

Missing or incorrect information may be considered an attempt to cheat.

Contents

List of Tables	iv
List of Figures	iv
List of Abbreviations	v
1 Introduction	1
2 Related Work	2
2.1 Knowledge Graph Using Large Language Models	2
2.1.1 Knowledge Graph	2
2.1.2 Using Large Language Models for Knowledge Graph Construction .	2
2.1.3 Domain Specific Knowledge Graph Construction using Large Lan- guage Models	2
2.2 Whole Slide Image Representation Workflow	2
3 Methods	3
3.1 Methods Overview	3
3.2 Dataset	3
3.3 Candidate Large Language Models	5
3.4 Ontology Generation	5
3.5 Prompting guided by Retrieval-Augmented Generation	6
3.6 Evaluation	6
3.6.1 Phase 0: Validation and Model Selection	7
3.6.2 Phase 1: Prompt Refinement	7
3.6.3 Phase 2: Ontology Refinement	7
3.6.4 Phase 3: Final Evaluation	7
4 Results	8
4.1 Phase 0	8
4.2 Phase 1	9
4.3 Phase 2	9
4.4 Phase 3	9
5 Discussion	10
5.1 Interpretation of Findings	10
5.1.1 Phase 0	10
5.1.2 Phase 1	11
5.1.3 Phase 2	11
5.1.4 Phase 3	12
5.2 Limitations	12
5.3 Future work	13
6 Conclusion	13
References	14

A	Ontology Design and Refinement	B1
A.1	Entity and Relation Definition	B1
A.2	Previous Ontologies	B1
A.2.1	Initial Ontology	B1
A.2.2	Revised Ontology	B1
B	Transparency Report: Tools and Third-Party Contributions	B2
B.1	Overview	B2
B.2	Per Section	B3

List of Tables

1	Results of Phase 0: Validation and Model selection	8
2	Results of Phase 1: Prompt Refinement	9
3	Results of Phase 2: Ontology Refinement	9
4	Results of Phase 3: Final Evaluation	10
5	Entity Definitions	B1
6	Relation Definitions	B2

List of Figures

1	Overview of developing the proposed pipeline for LLM-based KG construction for WSI Representation Workflow.	4
2	Overview of the proposed ontology schema for WSI representation workflows.	6
3	Distribution of extraction error types across baseline configurations in Final Evaluation (Phase 3)	10
4	Overview of the initial ontology schema for WSI representation workflows.	B3
5	Overview of the revised ontology schema for WSI representation workflows.	B3

List of Abbreviations

CLAM Clustering-constrained Attention Multiple Instance Learning

FNs False Negatives

FPs False Positives

Gemma Gemma 4 31B Instruct

GWDG Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

HPC High-performance Computing

KB Knowledge Base

KG Knowledge Graph

LLM Large Language Model

MoE Mixture-of-Experts

PANTHER Prototype AggregatioN-based framework for compacT HEterogenous slide
set Representation

Patch-GCN Patch-based Graph Convolutional Network

Qwen Qwen3 30B A3B Instruct 2507

RAG Retrieval-Augmented Generation

SAIA Scalable AI Accelerator

TITAN Transformer-based pathology Image and Text Alignment Network

TPs True Positives

WSI Whole Slide Image

1 Introduction

Recent advancements in LLMs have accelerated KG construction, leveraging their powerful natural language understanding and reasoning capabilities [Ibr+24]. Beyond capturing general knowledge, the research focusing on using LLMs to construct target domain-specific KGs has been also growing in different domains [Jel+26]. These domain-specific KGs structure complex literature into machine-readable representations, facilitating automated reasoning and knowledge discovery across specialized fields [Ver+22].

Despite this progress, several specialized target domains remain largely unexplored, including WSI representation workflows. WSIs are characterized by vast giga-pixel sizes, complex spatial semantics, and a promising source to be used in diagnostic pathology [KGG20]. However, researchers in digital pathology often adhere to a few well-known methodologies without a unified, structured understanding of the broader literature landscape [Hos+24]. This is driven by the exponential growth of scientific literature, which makes manually tracking and comparing biomedical experimental procedures increasingly labor-intensive [Luo+25]. Structuring WSI representation workflows into a comprehensive KG offers a promising avenue to standardize, query, and compare these complex analytical pipelines across scientific publications.

To address this challenge, we introduce a foundational pipeline for automatically constructing a KG focused on WSI representation workflows through iterative prompt and ontology refinement, where the ontology defines domain-specific concepts and relations (detailed in Section 3.4). Specifically, our framework guides LLMs via domain-specific custom ontology to extract structured entities and relationships directly from scientific literature. To systematically investigate the feasibility, optimization, and error types of this pipeline, we formulate three main Research Questions (RQs):

- **RQ1:** *Can an LLM extract structured WSI representation workflow knowledge from scientific literature according to a predefined ontology?*
- **RQ2:** *How do prompt and ontology refinement affect extraction quality?*
- **RQ3:** *What types of errors remain in ontology-guided LLM-based KG construction for WSI representation workflow?*

To answer these RQs, we design four sequential phases; (1) Phase 0 to validate feasibility of LLM-based KG construction for our target domain and backbone model selection, (2) Phase 1 to evaluate iterative prompt refinement strategies, (3) Phase 2 to optimize the underlying custom ontology, and (4) Phase 3 to perform a final evaluation including error type analysis. Based on manual evaluations with a curated literature dataset, we quantify performance, specify error types of current pipeline, and suggest directions for future enhancements.

Our work has three main contributions. First, we propose a foundational LLM-based framework for automatic KG construction targeting WSI representation workflows, addressing an unexplored domain in scientific knowledge extraction. Secondly, we design a specialized ontology and a corresponding prompt that effectively structures pathology workflows into RDF-based graph representations. Lastly, we conduct a manual evaluation for each phase using a manually curated dataset and identify key error types, such as hallucination entities and/or relations, showing insights for future developments.

2 Related Work

2.1 Knowledge Graph Using Large Language Models

2.1.1 Knowledge Graph

A KG can be defined as a graph structured Knowledge Base (KB) [Tia+22], where the KB means a structured repository that store facts in a relational format, providing an organized way to query and retrieve predefined facts and relationships [Yan+25b].

A KG is able to represent domain knowledge with nodes and edges [Tia+22]. It can be generally expressed as $G = \{E, R, F\}$, where E is the set of entities, R is the set of relationships defining connections between entities, and F is the set of facts [Tia+22]. Each fact is often defined as a triple formatted as (Entity, Relation, Entity). [Tia+22].

2.1.2 Using Large Language Models for Knowledge Graph Construction

LLMs are deep learning models trained with large text corpora, which are able to understand the context and generate responses like humans [Ibr+24]. By leveraging the strong generalizability and natural language processing capabilities of LLMs for knowledge extraction, KGs can overcome their inherent incompleteness and limited language understanding [Pan+24]. Specifically, LLMs can enhance KG construction in various ways, such as extracting entities and relationships from texts, completing missing parts of existing KGs through fact generation, and producing natural language descriptions of facts in existing KGs [Ibr+24].

For such KG construction, LLMs enables an end-to-end generative process via prompt engineering [Zhu+24]. Furthermore, providing directly an ontology guides the model to mitigate the risk of structural hallucination [Mih+23].

2.1.3 Domain Specific Knowledge Graph Construction using Large Language Models

Recently, the integration of LLMs into domain-specific KGs construction has developed across specialized fields such as biomedicine, cybersecurity, education, and law [Jel+26]. While LLMs show impressive general knowledge and reasoning abilities, there are limitations of the precise terminology, formalisms, and relationship patterns specific to scientific domains [ZH26]. To overcome the limited abilities, ontologies can guide LLMs to capture domain concepts and relations, serving as blueprints for graph construction [CTB25].

Although KGs are well suited for mapping step-by-step processes and sequential features in complex scientific workflows, KG studies on biomedical workflows remain surprisingly sparse. Instead, recent approaches have primarily been used to represent abstract concepts (e.g. Disease and Gene) and semantic relationships of them [Abu+23], leaving fine-grained algorithmic steps, and dynamic pipeline architectures largely unmodeled. To bridge this gap, our work propose an ontology-guided framework designed to construct a WSI representation workflow KG directly from computational pathology literature.

2.2 Whole Slide Image Representation Workflow

WSIs are giga-pixel digital pathology slides that have become an useful tool in digital pathology [KGG20], enabling slide-level visual assessment of tumor microenviron-

ments [Mun+21]. Due to the huge size of them, WSIs cannot be processed directly causing extreme memory constraints [Son+23].

To overcome this, WSIs are typically split into smaller patches at a fixed magnification level, non-tissue regions are filtered out using masks, and patch-level feature vectors are extracted using pretrained backbones [Son+23]. These patch embeddings are then often aggregated via attention-based or pooling mechanisms to obtain slide-level representations for downstream tasks [Son+23].

In this work, we focus on this end-to-end WSI representation workflow, including the entire pipeline to have slide-level embedding of each WSI.

3 Methods

3.1 Methods Overview

Our work proposes an ontology-guided KG construction pipeline tailored to the domain of WSI representation workflows with targeted prompting strategies, guided by Retrieval-Augmented Generation (RAG) based on a custom domain ontology.

As illustrated in Figure 1, our work consists of four sequential phases to establish, refine, and evaluate the LLM-based KG construction pipeline for our target domain. Due to the absence of existing benchmark datasets in this domain, we curated a literature-based dataset to incrementally refine and validate our approach. Across Phases 0 through 2, we optimize the pipeline configuration including backbone model selection, prompt design, and ontology construction, with a single representative research paper. In Phase 3, we evaluate the performance of the finalized setup across three unseen research papers selected based on their varying levels of workflow complexity (detailed in Section 3.2).

Specifically, each phase is designed to address specific RQs:

- **Phase 0 (Validation & Model Selection):** To address RQ1, we evaluate the feasibility of extracting structured WSI representation workflow KG. We compare 'Gemma 4 31B Instruct (Gemma)' and 'Qwen3 30B A3B Instruct 2507 (Qwen)' to identify the optimal primary backbone model for subsequent phases (with the rationale for selecting these two candidate models detailed in Section 3.3).
- **Phase 1 (Prompt Refinement):** In order to address RQ2, we iteratively refine prompt designs and identify limitations of the results for developing the ontology.
- **Phase 2 (Ontology Refinement):** Continuing to address RQ2, we optimize the domain-specific ontology structure for the corresponding prompt and RAG.
- **Phase 3 (Final Evaluation):** To address RQ3, we evaluate the finalized pipeline configuration. By progressively incorporating each component of the pipeline, we conduct ablation experiments to quantify individual contributions and analyze remaining error types in KG construction for our target domain.

3.2 Dataset

To evaluate our ontology-guided KG construction pipeline, we selected four representative research papers on WSI representation workflows. The Clustering-constrained Attention Multiple Instance Learning (CLAM) [Lu+21] research paper was leveraged as the data to validate framework feasibility, refine prompt templates, and optimize the custom ontology

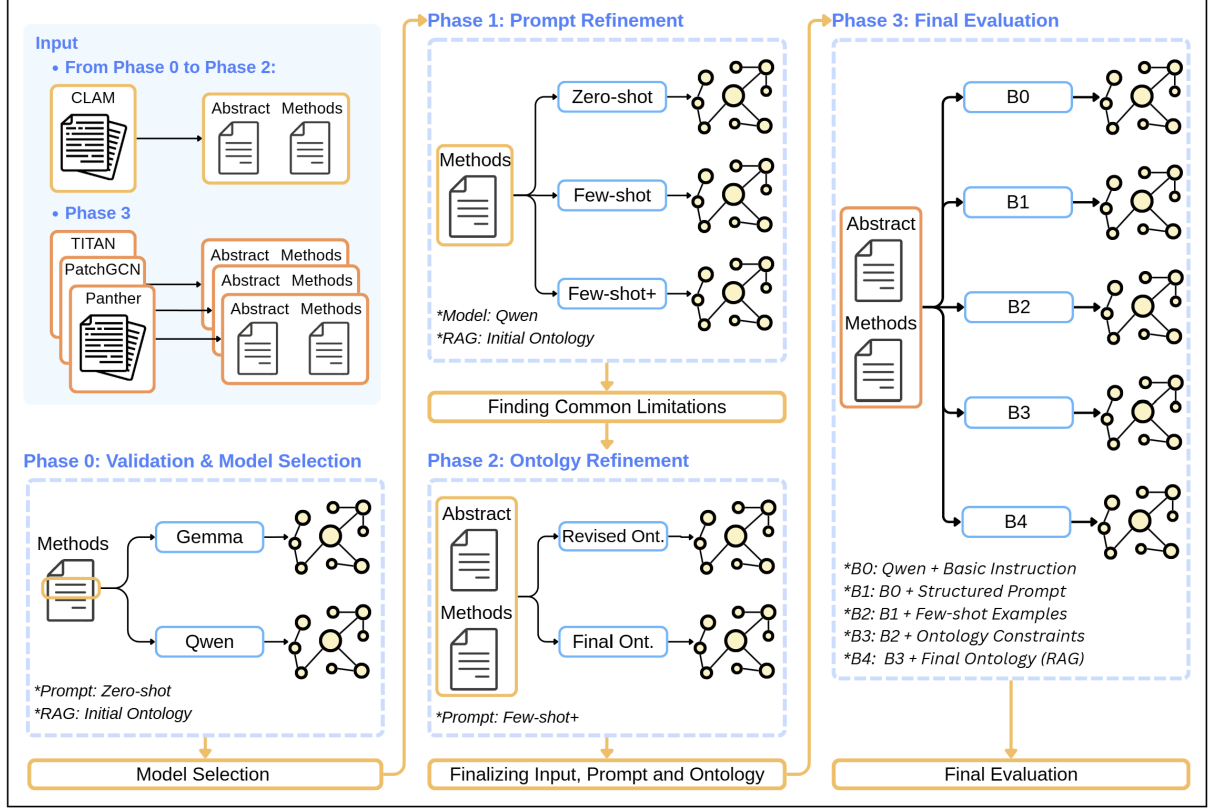


Figure 1: Overview of developing the proposed pipeline for LLM-based KG construction for WSI Representation Workflow.

(Phases 0 to 2). In Phase 3, considering the absence of existing benchmark datasets in this domain, we strategically selected three representative papers ranging from simple to highly complex workflows to optimize evaluation coverage within manageable resource limits. Specifically, Prototype AggregationN-based framework for compact HETerogenous slide set Representation (PANTHER) [Son+24], Patch-based Graph Convolutional Network (Patch-GCN) [Che+21], and Transformer-based pathology Image and Text Alignment Network (TITAN) [Din+25] were used as unseen data to evaluate comprehensive performance and ablation studies.

The detailed characteristics of each selected paper and the specific text section used as an input for each phase is explained below:

- *CLAM* [Lu+21]: Selected for pipeline development (Phases 0–2) due to its representative characteristic as a widely adopted workflow ($\geq 2,200$ citations). Different text inputs were provided across phases: Phase 0 utilized an excerpt of the Methods section covering tissue segmentation, patching and feature extraction; Phase 1 expanded the input to the complete Methods section; and Phase 2 incorporated both the Abstract and the Methods sections to provide broader context.
- *PANTHER* [Son+24]: Serves as a representative *low-complex* workflow example, requiring no extensive pretraining and relying on per-slide Gaussian Mixture Model fitting. Phase 3 used both the Abstract and the Methods sections as input text.
- *Patch-GCN* [Che+21]: Represents a *medium-complex* workflow, including a pre-

trained feature extractor backbone combined with a task-specific Graph Convolutional Network. The Abstract and the Methods sections were input to Phase 3.

- *TITAN* [Din+25]: Represents a *high-complex* workflow, including an end-to-end multimodal foundation model pretraining scheme. For Phase 3, both the Abstract and the Methods sections were used as input text.

The different algorithmic complexity of the unseen data in Phase 3 enables a structured evaluation of the proposed KG construction pipeline with increasingly sophisticated workflows. Notably, to preserve mathematical expressions, equations, and algorithmic notation without loss, the input text was in LaTeX format.

To establish a gold standard for evaluation, we manually constructed ground truth KG, representing the ideal, correct graph structures extracted from the inputs, aligned with phase-specific ontologies and serialized as RDF Turtle (`.ttl`) files. For the development phases using the CLAM [Lu+21] paper, Phase 0 utilized a subset of the *Initial Ontology* corresponding to the input excerpt, resulting in 43 triples, Phase 1 adopted the full *Initial Ontology* (56 triples), and Phase 2 used two refined ontologies progressively updated through testing (63 and 93 triples, respectively). Finally, for Phase 3, ground truth KGs were constructed based on the finalized ontology, resulting 235 triples in total. Detailed design principles and developments of the ontology are elaborated in Section 3.4.

3.3 Candidate Large Language Models

Following Dasgupta et al.’s evaluation of Qwen- and Gemma-based models for LLM triple extraction [DSS26], we selected Qwen [Yan+25a] and Gemma [Abd+26] as representative candidate models within the $\sim 30\text{B}$ parameter class and comparable native long-context support ($\sim 256\text{k}$ tokens). This pairing enable a comparison between Mixture-of-Experts (MoE) and dense architectures. While Qwen uses 3B active parameters with MoE architecture [Yan+25a], Gemma activates all 31B parameters as a dense architecture [Abd+26]. Evaluating results from both models allows us to examine whether and how prompting results varies across different architectures and methods.

Notably, all of our prompt-based approaches were executed within the Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG) environment. Inference was conducted using the GWDG Scalable AI Accelerator (SAIA) framework, which provides API access to local High-performance Computing (HPC) clusters [Doo+26]

3.4 Ontology Generation

As illustrated in Figure 2, our proposed ontology is divided into three main components: (1) *Metadata and Entry Points*, which capture fundamental scientific paper information and define the entry point of the computational workflow; (2) *Workflow Step Specification*, which explains operational details at each step and connects the steps sequentially; and (3) *Other Entity Specification*, which provides the additional information of the entities referenced within each workflow step.

To construct this ontology, we conducted two revisions across the phases. The ontology refinement process proceeded in three phases to establish a robust schema. In Phases 0 and 1, the *Initial Ontology* was established to define basic domain boundaries. In Phase 2, this was updated to a *Revised Ontology* before being polished into the *Final Ontology*, which was subsequently deployed in Phase 3. Section A provides a step-by-step overview of this process alongside full definitions for all entities and relations in the Final Ontology.

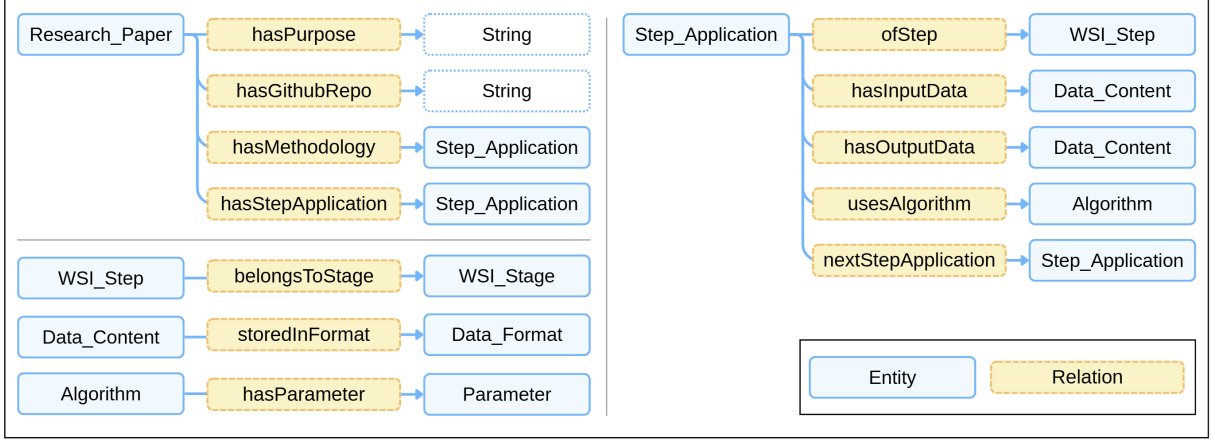


Figure 2: Overview of the proposed ontology schema for WSI representation workflows.

3.5 Prompting guided by Retrieval-Augmented Generation

We applied zero-shot and few-shot prompting [Bro+20] to establish a structural baseline and enforce explicit RDF syntax and entity type declarations. We also integrated RAG [Lew+20] via GWDG Arcana [GWD25] to retrieve relevant syntax specifications and the ontology to provide structural constraints on the generated outputs.

To investigate how prompt structures influence the quality of ontology-guided extraction, we evaluated three progressively refined prompt configurations: *Zero-shot*, *Few-shot (Structure)*, and *Few-shot (+Declarations)*. *Zero-shot* establishes the basic prompt, consisting only of the target ontology rules and output RDF format constraints. *Few-shot (Structure)* augments the zero-shot baseline by incorporating explicit structural syntax rules alongside basic triple connection examples to guide valid RDF graph generation. *Few-shot (+Declarations)* further enhances the prompt by integrating strict constraint specifications, expanded syntax guidelines, and explicit entity type declarations (e.g., `wsi:SVS rdf:type wsi:Data_Format`) to ensure complete output.

3.6 Evaluation

To systematically assess the performance of our KG construction pipeline across several phases, we define phase-specific evaluation metrics matching the objectives of each phase.

Due to the exploratory nature of this work and the absence of a predefined schema and datasets, automated evaluation was impractical. Therefore, we conducted a manual evaluation to compare the generated outputs against the ground truth graphs.

To perform evaluation, predictions are first categorized into True Positives (TPs), False Positives (FPs), and False Negatives (FNs). They are defined on a triple-level basis, in which TPs are exact matching TTL-formatted RDF triples, FPs are hallucinatory or invalid triples or extra natural language sentences, and FNs represent missed triples [GC25]. Notably, through all the phases, variations in entity annotations were ignored, focusing on semantic correctness to calculate triple-level Precision (P), Recall (R), and F_1 -score.

We selected these metrics because KG evaluation requires assessing both accuracy and coverage. Precision measures the proportion of correctly extracted triples among all predicted outputs, while recall measures the proportion of correctly extracted triples relative to the total ground-truth annotations [CJ25]. The F_1 -score represents the harmonic mean

of precision and recall, providing a balanced overall accuracy score [CJ25]. Specifically, these metrics are computed as:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F_1\text{-score} = 2 \times \frac{P \times R}{P + R} \quad (1)$$

3.6.1 Phase 0: Validation and Model Selection

In Phase 0, we assess the baseline feasibility of LLM-based approach for our target domain and select the primary backbone model (RQ1). We evaluate the extraction performance of candidate models (Gemma and Qwen) by measuring triple-level precision, recall, and F_1 -score against the Phase 0 ground truth KG. The model showing superior performance is selected as the primary backbone architecture for subsequent phases.

3.6.2 Phase 1: Prompt Refinement

In Phase 1, we evaluate prompt strategies across three configurations (*Zero-shot*, *Few-shot (Structure)*, and *Few-shot (+Declarations)*) for the prompt optimization (RQ2) by measuring triple-level precision, recall, and F_1 -score against the ground truth. Each iteration is assessed to track improvements of the output quality. Furthermore, by manually analyzing the extraction outputs across all three configurations, we identify common limitations of the *Initial Ontology*. These identified limitations are leveraged to update and refine the ontology in Phase 2.

3.6.3 Phase 2: Ontology Refinement

In Phase 2, we evaluate the impact of ontology schema optimization (RQ2). Building upon the limitations identified in Phase 1, we evaluate extraction quality using triple-level precision, recall, and F_1 -score across the updated ontology variants. Through this refinement process, we finalize the optimal pipeline components to be utilized in Phase 3.

3.6.4 Phase 3: Final Evaluation

In Phase 3, we evaluate the generalizability of the finalized pipeline across the three unseen research papers and perform an component-wise ablation study for error anylsis (RQ3). To quantify the individual contribution of each pipeline component, we evaluate five cumulative configurations:

- *B0 (Baseline)*: Direct extraction using the Qwen backbone model paired with basic plain-text extraction instructions.
- *B1 (B0 + Structured Prompt)*: Includes structured formatting rules and RDF syntax constraints to baseline instructions.
- *B2 (B1 + Few-shot Examples)*: Incorporates representative few-shot triples to demonstrate valid relational connections.
- *B3 (B2 + Ontology Constraints)*: Applies schema constraints and explicit entity type declarations.
- *B4 (B3 + RAG-guided Ontology Integration)*: Integrates RAG to provide relevant ontology guidelines into the generation process.

We focus on triple-level F_1 -score across all benchmark datasets for each configuration to validate the progressive performance gains delivered by each component.

To categorize the failure modes of LLM-based extraction, we adapt the error type analysis proposed by Ilves et al. [IBÜ26], which quantifies extraction errors across hallucinated and missed entities, hallucinated and missed triples, schema violations, and relation mismatches. Since our evaluation spans from baseline non-instructed configurations (B0) to fully guided pipelines (B4), the spectrum of output variations is broader. Therefore, we define nine fine-grained error categories to analyze structural, semantic, and syntactical extraction failures:

1. *Hallucinated Entities*: Entities generated by the LLM that do not exist in the Ground Truth graph.
2. *Missed Entities*: Valid ground truth entities that the model failed to extract.
3. *Hallucinated Relations*: Relational predicates generated by the model that do not exist in the ground truth graph.
4. *Missed Relations*: Relations present in the ground truth omitted in the output.
5. *Hallucinated Triples*: Extracted triples not existing in the ground truth graph.
6. *Missed Triples*: Valid ground truth triples omitted by the model.
7. *Schema Violations*: A specific subset of *Hallucinated Triples* where valid entities and relations are incorrectly linked, violating the ontology’s domain-range constraints.
8. *Extra Text*: Natural language texts generated alongside the RDF Turtle output.
9. *Invalid RDF*: Syntactic or structural formatting errors that violate the RDF Turtle grammar specification.

Under this taxonomy, total FPs are equal to the sum of *Hallucinated Triples*, *Extra Text*, and *Invalid RDF* errors, while FNs correspond to *Missed Triples*.

4 Results

4.1 Phase 0

As summarized in Table 1, the comparison between Gemma and Qwen demonstrates distinct trade-offs in extraction performance. Gemma achieved higher precision (0.552 vs. 0.431) with fewer FPs (13), but suffered from low recall (0.372). In contrast, Qwen showed superior coverage, capturing more correct triples ($TP = 25$) to achieve a higher recall (0.581) and a superior overall F_1 -score (0.495 vs. 0.444). Since capturing comprehensive domain knowledge is essential for iterative ontology refinement, Qwen was selected as the primary backbone model for subsequent pipeline phases.

Table 1: Results of Phase 0: Validation and Model selection

Models	TP	FP	FN	Precision	Recall	F_1 Score
Gemma	16	13	27	0.552	0.372	0.444
Qwen	25	33	18	0.431	0.581	0.495

4.2 Phase 1

As shown in Table 2, evaluating the three prompt configurations reveals the impact of structural and explicit guidance with examples during extraction. The baseline *Zero-shot* setup achieved a lower F_1 -score (0.283) with the longer input compared to Phase 0 results. Introducing structural rules in *Few-shot (Structure)* resulted in performance degradation across all metrics ($F_1 = 0.116$, $Precision = 0.091$, $Recall = 0.161$) due to a sharp increase in FPs (90). However, the prompt with explicit entity type declarations in *Few-shot (+Declarations)* substantially recovered extraction capability, yielding the highest precision (0.267), recall (0.554), and overall F_1 -score (0.360).

Table 2: Results of Phase 1: Prompt Refinement

Prompts	TP	FP	FN	Precision	Recall	F_1 Score
Zero-shot	18	53	38	0.254	0.321	0.283
Few-shot (Structure)	9	90	47	0.091	0.161	0.116
Few-shot (+Declarations)	31	85	25	0.267	0.554	0.360

4.3 Phase 2

Table 3 shows that optimizing the ontology schema alongside expanding the input context yielded immediate and substantial performance improvements. Even the *Revised Ontology* setup significantly outperformed the best Phase 1 result ($F_1 = 0.360$), achieving an F_1 -score of 0.486. Transitioning from *Revised Ontology* to *Final Ontology* further refined extraction precision by reducing FPs (decreased from 71 to 49) and increasing TPs to 53. Consequently, precision improved from 0.377 to 0.520, raising the overall F_1 -score to 0.544, while recall is decreased from 0.683 to 0.570.

Table 3: Results of Phase 2: Ontology Refinement

Ontology	TP	FP	FN	Precision	Recall	F_1 Score
Revised Ontology	43	71	20	0.377	0.683	0.486
Final Ontology	53	49	40	0.520	0.570	0.544

4.4 Phase 3

As presented in Table 4, the component-wise ablation study across the three unseen papers shows overall performance gains as structural guidance and RAG-based ontology constraints are progressively added. The unguided baseline (*B0*) completely failed across all datasets ($F_1 = 0.000$) due to non-standardized outputs and severe hallucination. Introducing structural formatting rules (*B1*) established a functional baseline ($F_1 = 0.164$), which further improved with few-shot examples (*B2*, $F_1 = 0.387$) and schema constraints (*B3*, $F_1 = 0.429$). Incorporating RAG-guided ontology retrieval (*B4*) achieved the highest overall extraction performance ($F_1 = 0.568$), driven by a major performance gain on the complex TITAN [Din+25] input (F_1 increased from 0.106 to 0.240).

To understand the underlying drivers of performance improvements, we analyzed error types distributions, as illustrated in Figure 3. In *B0*, invalid syntax and non-RDF formatting were observed with substantial *Extra Text* (150) and *Invalid RDF* (31) errors alongside massive *Hallucinated Triples* (598). Applying structured prompt in *B1*

Table 4: Results of Phase 3: Final Evaluation

	PANTHER	Patch-GCN	TITAN	Total
B0	0.000	0.000	0.000	0.000
B1	0.211	0.224	0.074	0.164
B2	0.631	0.428	0.126	0.387
B3	0.691	0.692	0.106	0.429
B4	0.686	0.685	0.240	0.568

instantly eliminated formatting violations (*Extra Text* and *Invalid RDF* dropped to 0) and drastically reduced *Hallucinated Relations* (from 147 to 22). From *B2* to *B4*, the integration of explicit schema rules completely mitigated *Missed Relations* (0). Furthermore, RAG integration in *B4* contributed to further decrease of *Missed Entities* to 24 and *Missed Triples* to 75.

Notably, a trade-off is observed from *B2* to *B4*. While *Missed Triples* and *Missed Entities* decrease, *Hallucinated Triples* increase from 212 in *B2* to 291 in *B4*.

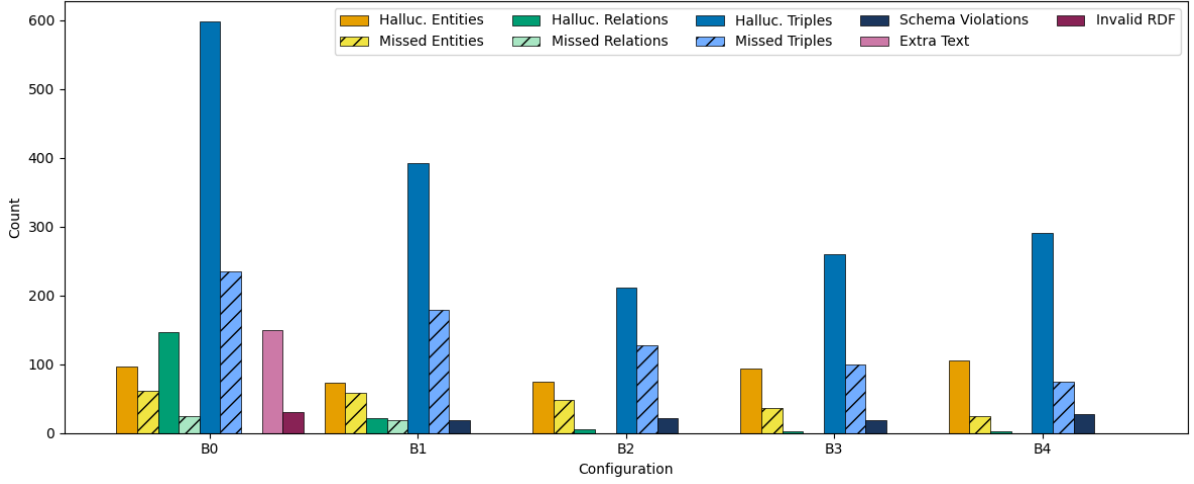


Figure 3: Distribution of extraction error types across baseline configurations in Final Evaluation (Phase 3)

5 Discussion

5.1 Interpretation of Findings

5.1.1 Phase 0

The results in Phase 0 address RQ1, confirming that LLMs are possible to extract structured WSI representation workflow knowledge from unstructured scientific literature according to a predefined ontology. As summarized in Table 1, both candidate models showed basic competency in parsing text into triples, though with distinct trade-offs.

Gemma exhibited a conservative extraction posture, achieving higher precision, while resulting lower recall. In contrast, Qwen achieved a higher recall with lower precision,

and a higher overall F_1 -score (0.495 vs. 0.444). This behavioral difference may be related to their architectural characteristics: Gemma is a dense model in which all parameters are activated for each input token, whereas Qwen is a sparse MoE model that activates only a subset of experts per token, thereby reducing the computational cost per token. However, the observed difference in performance should be interpreted as an empirical finding rather than attributed solely to the model architecture.

From a KG construction perspective, high recall is critical during the initial exploratory phase, as omitting core domain entities and relations poses a greater bottleneck for iterative refinement than filtering out noisy extractions. Consequently, Qwen was selected as the backbone LLM for subsequent optimization phases due to its broader coverage of domain semantics.

5.1.2 Phase 1

The findings in Phase 1 address RQ2 by showing that prompt engineering enhances extraction quality when structural rules are explicitly explained with ontology declarations. While the baseline *Zero-shot* prompt provided a moderate result ($F_1 = 0.283$), *Few-shot (Structure)* caused a performance collapse ($F_1 = 0.116$). This decline was driven by a dramatic increase in FPs (90), indicating that forcing only syntax constraints leads the model to generate structurally compliant yet semantically hallucinatory triples. In contrast, incorporating explicit entity type declarations in *Few-shot (+Declarations)* effectively anchored the model’s output, achieving the highest F_1 -score (0.360).

These observations demonstrate that precise entity type declarations prevent structural hallucinations. However, despite these prompt refinements, the overall performance scores achieved in Phase 1 remain constrained. This fundamental bottleneck stems from two main factors: (1) schema-level ambiguities within the *Initial Ontology*, and (2) an over-concentration on training the proposed workflow rather than on the complete workflow. Consequently, prompt adjustments alone proved insufficient to bridge these structural gaps. To address these limitations, Phase 2 expands the input scope by incorporating paper abstracts alongside full methods sections to enrich contextual grounding, while simultaneously refining the ontology schema into a more non-redundant hierarchy.

5.1.3 Phase 2

The results in Phase 2 address RQ2 by demonstrating that explicit ontology refinement, combined with input context expansion, plays a decisive role in performance improvement. Optimizing the schema alongside incorporating both the abstract and full methods sections yielded immediate structural gains. Specifically, *Revised Ontology* outperformed the best Phase 1 result ($F_1 = 0.360$), jumping to an F_1 -score of 0.486.

However, its schema remained restricted to representing single, isolated papers. Consequently, it exhibited structural limitations in distinguishing shared processing concepts from paper-specific implementations when scaling up to multi-paper integration, leading to collisions when step names overlapped across different publications. To overcome these limitations and enable cross-study graph merging, we extended the schema into *Final Ontology* by introducing explicit conceptual layers, specifically decoupling generic processing steps from study-specific executions.

Transitioning to *Final Ontology* further enhanced extraction precision by clarifying ambiguous class-relation boundaries, which reduced FPs. This reduction drove precision

up to 0.520 and the overall F_1 -score to 0.544. Notably, this schema also caused recall to decline from 0.683 to 0.570. This drop primarily stems from the model’s over-concentration on extracting entities and relations strictly bound to the proposed workflow of the paper, rather than capturing every component of the complete end-to-end workflow context. Nevertheless, this result still outperforms the outcomes from Phase 1, showing an overall performance improvement.

Overall, these findings indicate that resolving schema ambiguities and expanding context eliminate structural hallucinations, thereby improving extraction accuracy.

5.1.4 Phase 3

The findings in Phase 3 directly address RQ3 by quantifying the individual contributions of pipeline components and remaining error types. Specifically, the result confirms that structural constraints ($B1$) are foundational for syntax enforcement, eliminating formatting errors such as *Extra Text* and *Invalid RDF*. Adding few-shot examples ($B2$) and schema rules ($B3$) proves critical for reducing *Hallucinated Relations* to negligible levels (3) and entirely eliminating *Missed Relations*, along with reduced *Hallucinated Triples*. Finally, RAG integration ($B4$) further reduces *Missed Entities* and *Missed Triples* with the overall extraction performance gain to an F_1 -score of 0.568.

Despite these gains, residual errors reveal two key extraction dynamics. First, expanding schema coverage introduces a trade-off with hallucinations: from $B2$ to $B4$, *Hallucinated Triples* increased from 212 to 291, alongside 105 *Hallucinated Entities*. This occurs because explicit schema constraints and RAG encourage aggressive candidate extraction, particularly in complex workflows like *TITAN* [Din+25]. Second, this aggressive extraction increases domain-range mismatches, driving schema-level violations up to 27 in $B4$ as valid entities and predicates are incorrectly linked across workflow steps.

Together, these dynamics show that while ontology guidance and RAG improve extraction coverage, they also increase edge-case hallucinations and binding errors, highlighting the need for further research for filtering them.

5.2 Limitations

The primary limitation of this work lies in its restricted scope, particularly regarding dataset size and model selection. Due to the high manual annotation cost for ground truth KG construction and evaluation, our evaluation was conducted on a small targeted data using Qwen and Gemma. Consequently, the findings may not fully capture model behavior across larger-scale datasets or diverse LLM architectures.

Furthermore, analyzing the residual error types in Phase 3 highlights structural limitations of our approach, particularly when handling highly complex literature workflows. Although RAG integration and explicit schema constraints enhance semantic coverage, the increased hallucinations implies that an inherent limitation in our pipeline, where expanding contextual information causes the model to incorrectly bind or hallucinate outputs.

In addition, a fundamental challenge stems from the current lack of unified data across WSI representation workflows. Since computational pathology tools and models use different standards, outputs from different workflows cannot be easily combined or searched together. Consequently, our work is limited to constructing an individual graph for each research paper rather than merging them into a single unified knowledge graph.

Lastly, the evaluation framework is limited by manual verification rather than expert-driven benchmarks, introducing potential subjectivity. Specifically, the qualitative assessments depended on non-expert evaluation, which may not fully capture clinical or pathological correctness.

5.3 Future work

Regarding the scope limitations, future work should focus on expanding both the dataset scale and model diversity. Incorporating automated ground truth generation techniques to evaluate broader biomedical corpora and test additional state-of-the-art LLMs will help validate the generalizability and robustness of our work.

To mitigate hallucinations in complex workflows, future work can incorporate post-extraction validation and domain-specific fine-tuning. Automated rule-based validation will be able to refine candidate triples, while targeted fine-tuning can improve predicate alignment and entity extraction precision.

Moreover, future work should aim to establish a unified framework for digital pathology. Mapping different workflow entities to universal medical schema will enable cross-study KG integration, laying the foundation for a scalable, dynamic KB.

Finally, to address evaluation constraints, developing expert-validated methods is essential. Transitioning from manual non-expert assessments to semantic metrics verified by experts will allow objective evaluation for trustworthy clinical deployment.

6 Conclusion

Our work shows that LLMs effectively construct KGs for WSI representation workflows, when guided by the domain ontology and structured prompting. Progressing from Phase 0 to 2, explicit entity declarations and RAG-driven schema constraints improved the total F_1 -score from 0.000 to 0.544. Furthermore, Phase 3 reveals that integrating RAG with strict domain constraints contributes to reduce additional errors and missing triples. However, expanding context increases hallucinations, alongside minor schema violations in complex workflows. These limitations highlight that while prompt and ontology refinement effectively address main bottlenecks, additional improvement of performance remains essential to filter hallucinated outputs.

The main contributions of this work are summarized as follows. First, we propose an LLM-based foundational framework that automates KG construction for WSI representation workflows, bridging unstructured digital pathology literature with structured, RDF-based representations. Furthermore, we design a specialized digital pathology ontology paired with a prompting strategy. By combining explicit entity assertions with RAG-driven schema retrieval through iterative refinement, our approach successfully gains improvements related to the elimination of structural hallucinations and enforces strict domain constraints across complex pipelines. Lastly, evaluating this architecture on unseen research papers reveals critical error types. These empirical insights provide diagnostic clarity on how LLMs process complex texts of our target domain, establishing actionable guidelines for future research direction.

References

- [Abd+26] Gemma Team Sherif El Abd et al. “Gemma 4 Technical Report”. In: *ArXiv* abs/2607.02770 (2026). URL: <https://api.semanticscholar.org/CorpusID:289923375>.
- [Abu+23] Bilal Abu-Salih et al. “Healthcare knowledge graph construction: A systematic review of the state-of-the-art, open issues, and opportunities”. In: *Journal of Big Data* 10.1 (2023), p. 81.
- [Bro+20] Tom Brown et al. “Language models are few-shot learners”. In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.
- [Che+21] Richard J Chen et al. “Whole slide images are 2d point clouds: Context-aware survival prediction using patch-based graph convolutional networks”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2021, pp. 339–349.
- [CJ25] Seungmin Choi and Yuchul Jung. “Knowledge graph construction: Extraction, learning, and evaluation”. In: *Applied Sciences* 15.7 (2025), p. 3727.
- [CTB25] Tiago da Cruz, Bernardo Tavares, and Francisco Belo. “Ontology learning and knowledge graph construction: A comparison of approaches and their impact on RAG performance”. In: *arXiv preprint arXiv:2511.05991* (2025).
- [Din+25] Tong Ding et al. “A multimodal whole-slide foundation model for pathology”. In: *Nature medicine* 31.11 (2025), pp. 3749–3761.
- [Doo+26] Ali Doosthosseini et al. “SAIA: a seamless Slurm-native solution for HPC-based services”. In: *The Journal of Supercomputing* 82.7 (May 2026), p. 403. ISSN: 1573-0484. DOI: 10.1007/s11227-026-08508-3. URL: <https://doi.org/10.1007/s11227-026-08508-3>.
- [DSS26] Abhirup Dasgupta, Fernando Spadea, and Oshani Wasana Seneviratne. “From “Strings” to “Things” for Personal Knowledge Graphs: Evaluating LLM Triple Extraction for Recommendation Systems”. In: *TEXT2KG/BiKE@ESWC*. 2026. URL: <https://api.semanticscholar.org/CorpusID:289706423>.
- [GC25] Hussam Ghanem and Christophe Cruz. “Fine-tuning or prompting on LLMs: evaluating knowledge graph construction task”. In: *Frontiers in Big Data* 8 (2025). URL: <https://api.semanticscholar.org/CorpusID:279630265>.
- [GWD25] GWDG. *Arcana/RAG*. Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG). Accessed: 2026-08-26. 2025. URL: <https://docs.hpc.gwdg.de/services/ai-services/arcana/index.html>.
- [Hos+24] Mahdi S Hosseini et al. “Computational pathology: a survey review and the way forward”. In: *Journal of Pathology Informatics* 15 (2024), p. 100357.
- [Ibr+24] Nourhan Ibrahim et al. “A survey on augmenting knowledge graphs (KGs) with large language models (LLMs): models, evaluation metrics, benchmarks, and challenges”. In: *Discover Artificial Intelligence* 4 (2024). URL: <https://api.semanticscholar.org/CorpusID:273826701>.
- [IBÜ26] Markus Ilves, Eduard Barbu, and Jaan Übi. “Large Language Models for Knowledge Graph Extraction: A Schema-Constrained Evaluation Framework”. In: *Proceedings of the Knowledge Graphs and Large Language Models Workshop (KG-LLM)@ LREC26*. 2026, pp. 221–228.

- [Jel+26] Hamed Jelodar et al. “Integrating graphs, large language models, and agents: Reasoning and retrieval”. In: *Information Fusion* (2026), p. 104586.
- [KGG20] Neeta Kumar, Ruchika Gupta, and Sanjay Gupta. “Whole slide imaging (WSI) in pathology: current perspectives and future directions”. In: *Journal of digital imaging* 33.4 (2020), pp. 1034–1040.
- [Lew+20] Patrick Lewis et al. “Retrieval-augmented generation for knowledge-intensive nlp tasks”. In: *Advances in neural information processing systems* 33 (2020), pp. 9459–9474.
- [Lu+21] Ming Y Lu et al. “Data-efficient and weakly supervised computational pathology on whole-slide images”. In: *Nature biomedical engineering* 5.6 (2021), pp. 555–570.
- [Luo+25] Yi Luo et al. “From intention to implementation: automating biomedical research via LLMs”. In: *Science China Information Sciences* 68.7 (2025), p. 170105.
- [Mih+23] Nandana Mihindukulasooriya et al. “Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text”. In: *International semantic web conference*. Springer. 2023, pp. 247–265.
- [Mun+21] Felicitas Mungenast et al. “Next-generation digital histopathology of the tumor microenvironment”. In: *Genes* 12.4 (2021), p. 538.
- [Pan+24] Shirui Pan et al. “Unifying large language models and knowledge graphs: A roadmap”. In: *IEEE Transactions on Knowledge and Data Engineering* 36.7 (2024), pp. 3580–3599.
- [Son+23] Andrew H Song et al. “Artificial intelligence for digital and computational pathology”. In: *Nature Reviews Bioengineering* 1.12 (2023), pp. 930–949.
- [Son+24] Andrew H Song et al. “Morphological prototyping for unsupervised slide representation learning in computational pathology”. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2024, pp. 11566–11578.
- [Tia+22] Ling Tian et al. “Knowledge graph and knowledge reasoning: A systematic review”. In: *Journal of Electronic Science and Technology* 20.2 (2022), p. 100159.
- [Ver+22] Shilpa Verma et al. “Scholarly knowledge graphs through structuring scholarly communication: a review”. In: *Complex & Intelligent Systems* 9 (2022), pp. 1059–1095. URL: <https://api.semanticscholar.org/CorpusID:251467419>.
- [Yan+25a] An Yang et al. “Qwen3 technical report”. In: *arXiv preprint arXiv:2505.09388* (2025).
- [Yan+25b] Wenli Yang et al. “A comprehensive survey on integrating large language models with knowledge-based methods”. In: *Knowledge-Based Systems* 318 (2025), p. 113503.
- [ZH26] Erxi Zhu and Yuan Hu. “A multi-source fusion and dynamic verification framework for scientific knowledge graph construction: Methods, evaluation, and applications”. In: *AI Magazine* 47.2 (2026), e70062.
- [Zhu+24] Yuqi Zhu et al. “Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities”. In: *World Wide Web* 27.5 (2024), p. 58.

A Ontology Design and Refinement

This section shows the design and refinement of the ontology for describing WSI representation workflow.

A.1 Entity and Relation Definition

To systematically explain the target domain, we define eight core entity classes as described in Table 5

Table 5: Entity Definitions

Entity	Definition
Research_Paper	Represents the source scientific publication itself.
WSI_Step	Defines a canonical processing step (e.g., <i>Patch_Segmentation</i>) that is shared across papers to enable cross-study graph merging.
Step_Application	Captures a paper-specific execution of a <i>WSI_Step</i> , connected to execution-specific entities such as algorithms, parameters, inputs, outputs, and next executions.
WSI_Stage	Represents a standardized stage for high-level processing representation, restricted to three predefined entities: <i>Stage_Patching</i> , <i>Stage_Feature_Extraction</i> , and <i>Stage_Aggregation</i> .
Algorithm	Describes specific algorithms, or computational methods (e.g., <i>OTSU_Thresholding</i>).
Parameter	Explains key hyperparameters and configuration values of an <i>Algorithm</i> (e.g., <i>patch_size=256</i>).
Data_Content	Describes the semantic input or output data representation transferred between steps (e.g., <i>Whole_Slide_Image</i> , <i>Tissue_Patches</i>).
Data_Format	Specifies the file extension used for the data (e.g., <i>SVS</i>).

The interactions between entities are formally defined by the relations defined in Table 6.

A.2 Previous Ontologies

A.2.1 Initial Ontology

Figure 4 presents the initial ontology schema, which served as the starting point for our work.

A.2.2 Revised Ontology

Figure 5 shows the revised ontology schema after Phase 1, specifying input or output data of the workflow with 'Data_Content' and 'Data_Format' entities

Table 6: Relation Definitions
Definition

Relation	Definition
<code>hasPurpose</code>	Connects a <i>Research_Paper</i> to a literal text string describing the primary objective or focus of the publication.
<code>hasGithubRepo</code>	Connects a <i>Research_Paper</i> to a literal text string specifying its official open-source repository URL.
<code>hasMethodology</code>	Links a <i>Research_Paper</i> to the entry point <i>Step_Application</i> which is the beginning of its experimental pipeline.
<code>hasStepApplication</code>	Links a <i>Research_Paper</i> to all constituent <i>Step_Application</i> instances within its pipeline to preserve full paper provenance.
<code>ofStep</code>	Maps a paper-specific <i>Step_Application</i> to its corresponding shared <i>WSI_Step</i> entity.
<code>nextStepApplication</code>	Links <i>Step_Application</i> to another subsequent <i>Step_Application</i> within the same paper’s workflow.
<code>belongsToStage</code>	Maps a generic <i>WSI_Step</i> concept to exactly one predefined <i>WSI_Stage</i> individual.
<code>usesAlgorithm</code>	Links a <i>Step_Application</i> to the specific <i>Algorithm</i> implemented in that step.
<code>hasParameter</code>	Connects an <i>Algorithm</i> to its associated <i>Parameter</i> specification or hyperparameter value.
<code>hasInputData</code>	Specifies the <i>Data_Content</i> which is an input of a <i>Step_Application</i> .
<code>hasOutputData</code>	Specifies the <i>Data_Content</i> generated as an output of a <i>Step_Application</i> .
<code>storedInFormat</code>	Links a <i>Data_Content</i> entity to its corresponding physical file format or extension (<i>Data_Format</i>).

B Transparency Report: Tools and Third-Party Contributions

B.1 Overview

- Writing
 - LaTeX (Overleaf)
- Diagrams for Ontologies
 - Canva
- Plot for Error Analysis
 - Python Matplotlib
- Code
 - Cursor

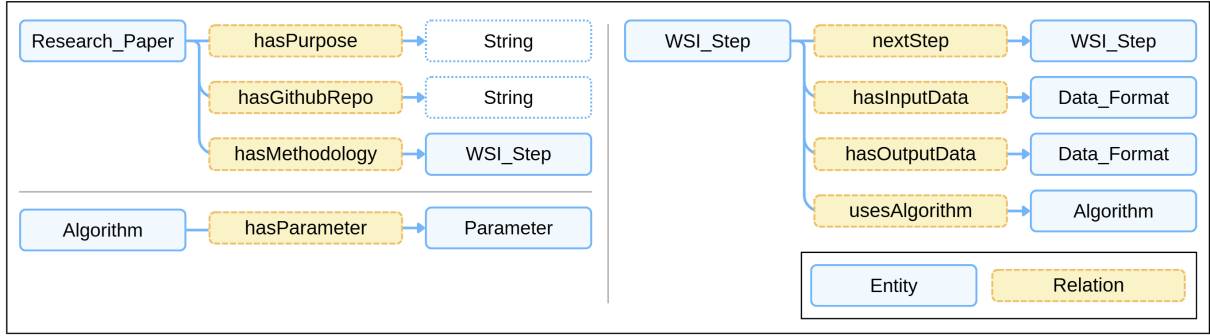


Figure 4: Overview of the initial ontology schema for WSI representation workflows.

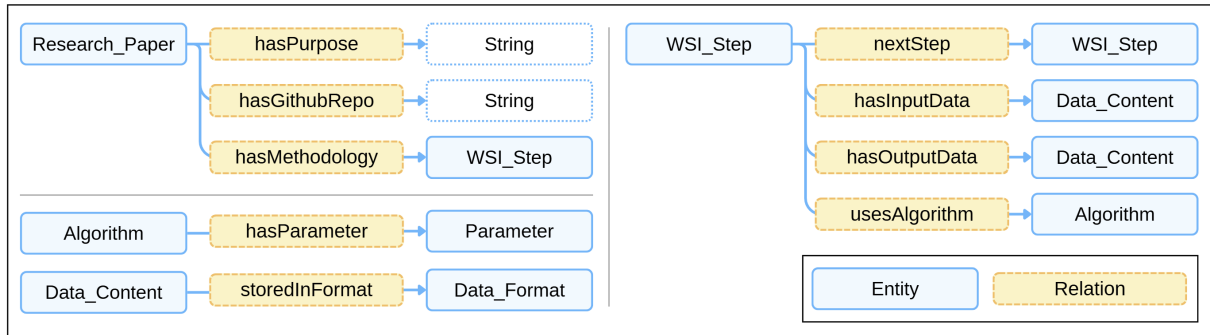


Figure 5: Overview of the revised ontology schema for WSI representation workflows.

- AI
 - Chat AI Model 'Gemma 4 31B Instruct'
 - Chat AI Model 'Qwen3 30B A3B Instruct 2507'
 - Gemini 3.6 Flash
 - Perplexity
 - Cursor
- Third-Party
 - Aasish-Kumar Sharma reviewed the structure.

B.2 Per Section

- Section 1 Introduction
 - Gemini 3.6 Flash was used for grammatical refinement, vocabulary enhancement, and proofreading.
 - Perplexity was used for literature searches.
- Section 2 Related Work
 - Gemini 3.6 Flash was used for grammatical refinement, vocabulary enhancement, and proofreading.

- Perplexity was used for literature searches.
- Section 3 Methods
 - Gemini 3.6 Flash was used for correcting the RDF syntax of the ontologies., grammatical refinement, vocabulary enhancement, and proofreading.
 - Cursor was used for entity and relation extraction for evaluation.
 - Chat AI Model 'Gemma 4 31B Instruct' and 'Qwen3 30B A3B Instruct 2507' are used as part of the pipeline.
- Section 4 Results
 - Gemini 3.6 Flash was used for grammatical refinement, vocabulary enhancement, and proofreading.
- Section 5 Discussion
 - Gemini 3.6 Flash was used for grammatical refinement, vocabulary enhancement, and proofreading.
- Section 6 Conclusion
 - Gemini 3.6 Flash was used for grammatical refinement, vocabulary enhancement, and proofreading.