

Ontology-Guided Procedural Knowledge Extraction and Validation Case Study on the Lockout-Tagout Documents

Cennet Oguz^{1,2} Noon Pokaratsiri Goldstein¹
Jörg Steffen¹ Reetam Biswas² Günter Neumann¹

¹German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany

²Saarland University, Germany

{cennet.oguz, noon.pokaratsiri, steffen, guenter.neumann}@dfki.de
rebi00001@stud.uni-saarland.de

Abstract

We present a case study developing an ontology-guided framework for extracting and validating procedural knowledge from safety-critical documents with limited annotations, specifically the Lockout-Tagout (LOTO) procedural documents. We build multimodal Procedural Knowledge Graphs (PKGs) grounded in the Procedural Knowledge Ontology (PKO) from text and diagrams using LLMs and VLMs. The extracted PKGs are evaluated on instruction- and graph-level metrics with validation against a domain expert curated ontology for action–entity relations, step order, and cross-modal alignment. Embedding-based metrics capture semantic alignment, while lexical metrics expose safety-critical mismatches. Our experiments on proprietary and public LOTO documents demonstrate the feasibility and limitations of multimodal procedural extraction and motivate ontology-guided validation.

1 Introduction

Large Language Models (LLMs) and Vision-Language Models (VLMs) have shown competencies at extracting structured knowledge from technical documents (Huguet Cabot and Navigli, 2021; Dagdelen et al., 2024; Bhattarai et al., 2024). Procedural knowledge extraction (PKE), however, requires representing not only entities and relations, but also ordered actions, dependencies, and their associated objects (Gupta et al., 2018; Dannenfelser et al., 2023). These requirements become particularly important in safety-critical industrial settings, where procedural representations must also preserve domain-specific operational constraints.

Lockout-Tagout (LOTO) provides a representative safety-critical use case. LOTO documents contain machine-specific procedures for controlling hazardous energy, with procedural information distributed across text, tables, document structure, and diagrams. These procedures specify ordered

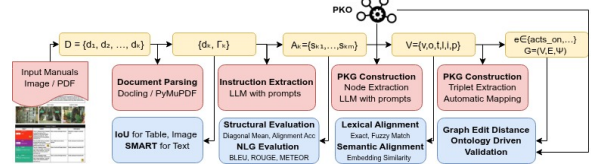


Figure 1: The end-to-end workflow from document parsing to PKG construction and evaluation.

actions together with equipment targets, tools, locations, verification steps, and supporting visual evidence. Gupta et al. (2018); Dannenfelser et al. (2023); Wu et al. (2022) have investigated procedural extraction and multimodal procedural understanding, exploiting the explicit structured representation of a Procedural Knowledge Graph (PKG) to associate executable actions with entities through semantic, temporal, and cross-modal relations.

Determining whether the extracted representation satisfies the safety requirements of the underlying procedure remains challenging. Although lexical, semantic, and graph-level similarities quantify the correspondence with a reference, they do not directly assess safety compliance. An extraction may remain highly similar to the reference while containing safety-critical errors, e.g. associating an action with the wrong energy-control component, omitting a required action, or altering the required execution order.

To address these challenges, we develop a multi-step ontology-guided framework, building on the Procedural Knowledge Ontology (PKO) (Carriero et al., 2025), to construct multimodal PKGs from heterogeneous LOTO documents. Ontologies and knowledge bases have been used to guide and verify LLM-based information extraction (Bhattarai et al., 2024; Chepurova et al., 2024), providing a basis for explicit domain-specific validation. By comparing this approach with end-to-end VLM-based extraction, we address the question: to what extent does the PKO-guided multi-step framework

improve PKE? Additionally, to address the question of the effectiveness of existing standardized metrics in detecting safety-related errors, our evaluation examines the framework step-by-step and extend the final PKG assessment beyond reference-based text and graph similarity metrics to include safety validation against expert-verified LOTO constraints.

Our main contributions are as follows:

- We present a baseline framework for real-world LOTO PKE by extending Ren et al. (2023) multi-dependency approach to an ontology-guided multi-modal LLM-based setting.
- We provide a multi-modal dataset and annotation features for evaluating the proposed pipeline.
- We demonstrate potential limitations of existing similarity-based text and graph metrics in validating safety compliance.

2 Background & Related Works

Lockout-Tagout (LOTO) Procedure. LOTO is a regulated industrial safety practice that prevents accidental machine activation during maintenance.¹ LOTO documents contain machine-specific instructions in text, diagrams, and tables, and are subject to both regulatory and confidentiality requirements.

Procedural Knowledge Extraction (PKE). LLMs have streamlined knowledge graph construction sub-tasks, e.g. named entity recognition (NER) and relations extraction (RE), into generative tasks (Mujtaba and Mahapatra, 2019; Huguet Cabot and Navigli, 2021; Chia et al., 2022; Trajanoska et al., 2023; Chepurova et al., 2024; Castle and Moreno Schneider, 2025), including extracting both (Dagdelen et al., 2024; Gong and Eldardiry, 2024) at the same time. Whereas these works examine *declarative knowledge*, we extend this paradigm towards constructing *procedural knowledge* graphs (PKG) as explored in Yang and Nyberg (2015); Mujtaba and Mahapatra (2019); Wu et al. (2022); Ren et al. (2023).

¹Regulated by the U.S. Occupational Safety and Health Administration (OSHA) (see <https://www.osha.gov/control-hazardous-energy>) and other agencies in other jurisdictions (see <https://www.safetylock.net/learn-lockout-tagout/lockout-tagout-legislation/>).

Ψ	Similarities in PKO Relations
$(v_i, \textit{precedes}, v_j)$	S_{k_m} step order, $(\textit{Step}, \textit{nextStep}, \textit{Step})$
$(v_i, \textit{acts_on}, o_i)$	$(\textit{Procedure}, \textit{hasProcedureTarget}, \textit{Thing})$
$(v_i, \textit{uses}, t_i)$	$(\textit{Action}, \textit{requiresTool}, \textit{Tool})$
$(v_i, \textit{occurs_at}, l_i)$	$(\textit{Procedure}, \textit{isAdoptedBy}, \textit{Organization})$
$(s_{k_j}, \textit{verify}, v_i)$	S_{k_m} step order, $(\textit{Step}, \textit{hasStepVerification}, \textit{Verification})$
$(p_i, \textit{image_of}, n_i)$	$(\textit{Procedure}, \textit{wasExtractedFrom}, \textit{Resource})$

Table 1: Non-exhaustive examples of Ψ formalization according to how the semantic roles correspond to the inter-class relations in the PKO (Carriero et al., 2025) as shown in Figure 2. Relations can be transitive, e.g. $(\textit{Procedure}, \textit{wasExtractedFrom}, \textit{Resource})$ implies $(\textit{Tool}, \textit{wasExtractedFrom}, \textit{Resource})$ because there are relations between *Procedure*, *Action*, and *Tool*.

PKG construction has been studied in specialized domains such as biomedical (Dannenfelser et al., 2023; Bhattarai et al., 2024), material science (Dagdelen et al., 2024; Castle and Moreno Schneider, 2025), legal (Zheng et al., 2024), and technical support (Gupta et al., 2018). Research into PKG for internal manufacturing processes (as opposed to consumer product operations) remains underexplored. Ren et al. (2023) investigate PKG with multiple dependency types; they examine PKG where nodes and edges represent information and relations either *between sentences* or *within each sentence*. We extend this paradigm to support **both inter-sentence and intra-sentence relations, including cross-modal (text-visual) relations** in the manufacturing domain.

Procedural Knowledge Ontology (PKO). The PKO provides an expert-curated schema for representing domain-specific procedures, including LOTO; Carriero et al. (2025) has also demonstrated its generalizability beyond original use cases. Bhattarai et al. (2024); Chepurova et al. (2024); Wang et al. (2024) demonstrate that ontologies and knowledge bases can improve LLM extraction and validate outputs via schema validation and in-context learning (ICL). This work investigates the application of the PKO (Carriero et al., 2025) in both PKG construction and evaluation. As Gretkowski et al. (2022) and Carriero et al. (2025) observe that procedural documents often omit implicit knowledge, we restrict our scope to document-explicit information.

3 Methodology

3.1 Multi-Step PKG Formalization

Figures 1 & 2 illustrate how a document is transformed into machine-interpretable PKG according

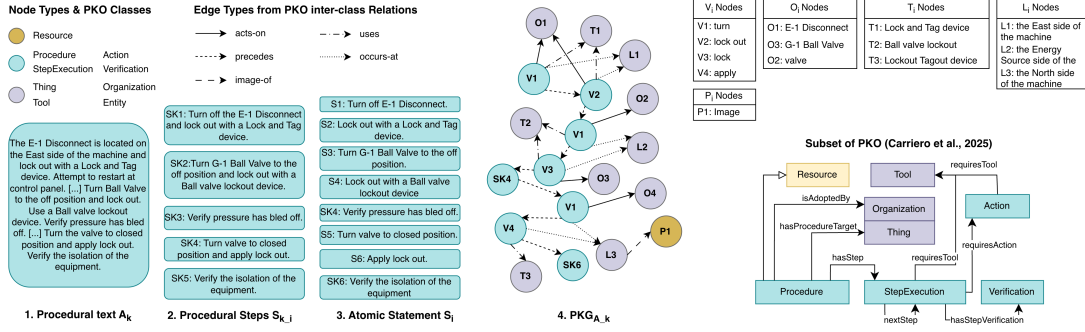


Figure 2: An example demonstrating how each extracted procedural text A_k is processed into executable procedural steps $S_{k,m}$. Atomic actions S_i along with semantic roles are extracted from each $S_{k,i}$ to form PKG nodes. The PKO classes and the inter-class relations corresponding to semantic information define edge types, facilitating the formation of triplets, which are sequentially composed into the PK G_{A_k} as explained in Section 3.1.

to our proposed framework: 1) **Document Parsing** extracts multimodal text–layout representation, 2) **Instruction Extraction** identifies executable steps and semantic roles via LLM prompting, and 3) **PKG Construction** is finalized from node inference and triplet formalization with PKO (Carriero et al., 2025) guidance by re-prompting the same LLM.²

Document Parsing. A document $D = \{d_{1 \rightarrow K}\}$ is decomposed into k -page text blocks $b_{k,i} = (t_{k,i}, \lambda_{k,i})$, where $t_{k,i}$ is text and $\lambda_{k,i} = (x_1, y_1, x_2, y_2)$ encodes the layout. Structural meta-data $\Gamma_k = \{\gamma_{k,1}, \gamma_{k,2}, \dots\}$ (e.g. headers, lists, tables) capture hierarchical organization. The resulting representation $D = \{(d_k, \Gamma_k) \mid k=1 \rightarrow K\}$ serves as input for instruction extraction.

Instruction Extraction. Executable steps $A_k = \{s_{k,1}, \dots, s_{k,m}\}$ are extracted using LLM few-shot prompting with each parsed page (d_k, Γ_k) as input. Formally, this is defined by a mapping $f : D \rightarrow A$, where $A = \bigcup_{k=1}^K A_k$ represents the complete set of procedures extracted from D .

Procedural Knowledge Graph. Extending from Ren et al. (2023), a PKG $G_{A_k} = (V, E, \Psi)$ is constructed from A_k , where V is the set of node types, E the directed relations, and $\Psi = \{(n_i, e_{ij}, n_j)\}$ the *procedural unit*. Each node $n_i \in V = \{v_i, o_i, t_i, l_i, S_{k,i}, p_i\}$ represents one of $\{action, target, tool, location, step, image\}$. A step description $S_{k,i}$ can denote an *executable action* or a *condition*, e.g. prerequisite, precaution,

informative, or verification. Each edge $e_{ij} \in E$ represents one of the procedural, semantic, and temporal (sequential order) relations inherently informed by the PKO (Carriero et al., 2025) (see Table 1) such as *acts_on*, *uses*, *occurs_at*, *image_of*, *precedes*, and *verify*. Some relations, which are not apparent, are transitive or analogous. For example, *occurs_at* is informed by *isAdeptedBy* in the PKO as *Procedure* has step *Action* (v_i) and *Organization* can be a *Location* (l_i).

Re-prompting the LLM decomposes each step into *atomic statements* $s_i = (v_i, \{o_i, t_i, l_i\})$, where v_i is the main verb and $\{o_i, t_i, l_i\}$ its argument set. A triplet representing a *procedural unit* (n_i, e_{ij}, n_j) is instantiated by connecting a main verb with its argument for intra-sentence dependencies or by connecting an *atomic statement* to another (inter-sentence dependencies). The final PKG $G_D = \{G_{A_1}, \dots, G_{A_K}\}$, where transitive *precedes* edges also capture inter-step (inter-sentence) dependencies, can represent nested procedures.

Cross-Modal Alignment. The cross-modal triplets $(p_i, image_of, n_i)$ or $(n_i, wasExtractedFrom, p_i)$ are formed by maximizing the cosine similarity between the vision-language embeddings of each extracted node and the candidate bounding box regions.

3.2 Baseline VLM-based PKG Construction

As a contrastive baseline to our multi-step framework, in an end-to-end PKE approach, a k -page document $D = \{d_{1 \rightarrow K}\}$ is represented as an ordered sequence of page images $X_d = \{x_{d,1}, x_{d,2}, \dots, x_{d,k}\}$ for input into a VLM via prompting to output the final PKG G_D directly. Formally, the extraction can be viewed as a func-

²We use few-shot prompting where templates include PKO class definitions and representative examples to enhance semantic coverage and output consistency (see Appendix C for task-specific details).

tion $f_m : X_d \rightarrow G_m$, where $G_m = (V_m, E_m)$ is the graph extracted by a model m . Each node $n_i \in V_m = \{v_i, o_i, t_i, l_i, S_{k_i}\}$ and each edge $e_{ij} \in E_m$ represent the node and relation types of the final PKG as formalized in Section 3.1.

Cross-Modal Triplets. To obtain cross-modal triplets $(n_i, \text{hasVisualEvidence}, p_i)$, the same model is re-prompted to extract visual grounding reference nodes p_i that can be associated with any of the procedural steps or entities in V_m .

Dataset	Documents	Procedures	Instructions	Visual
Proprietary	86	236	1840	213
Public	18	36	62	21
All	104	272	1,902	234

Table 2: Dataset composition: “Procedures” denotes procedural groups or sections in a document, “Instructions” denotes step descriptions, and “Visual” denotes visual evidence items.

3.3 Dataset & Annotation

We introduce an evaluation dataset consisting of proprietary and public LOTO documents annotated with layout and semantic information; Table 2 summarizes the dataset statistics. Public documents are obtained from generic LOTO templates in the public domain (web searches); proprietary LOTO documents are written with confidential machine- and facility-specific instructions.³

Annotation. To provide ground truths for the evaluation of intermediary steps in our pipeline, we use the Roboflow interface⁴ to annotate layout bounding boxes for page elements: *image*, *image-part*, *title*, *icon*, *paragraph text*, *itemized text*, *table*, and *table header*. The extracted text segments are manually aligned with their corresponding layout regions to provide visual-text grounding references.

Texts are further annotated as an ordered and nested list of *executable actions* and *conditional step* descriptions that capture hierarchy and sequence order. Each action is labeled with predefined PKG node and edge types based on the ontology mapping described in Section 3.1.

Executable actions are concrete steps performed by a practitioner and are associated, when stated, with entities (targets, tools, locations, verification

targets), or expected conditions. *Conditional steps* describe precautions, informative statements, prerequisite steps, additional requirements, and verification statements that affect safe execution. *Visual evidence* (bounding boxes) is annotated when a figure, image, table region, or page-level visual element supports an *executable action*, action-associated *entity*, *precaution*, or *verification*.

Inter-Annotator Agreement. A Computer Science Master’s student performs layout annotation. The Master’s student and a PhD researcher conduct procedural text annotations. A domain expert provides annotations for the proprietary documents and another PhD researcher annotates the public ones. We calculate the mean *Jaccard* and *Cohen’s κ* agreement scores (with exact lexical and normalized conceptual agreement matching⁵) for structured action tuples (*action*, *target*, *tool*, *location*), excluding auxiliary fields. The mean exact (and normalized) agreement scores for proprietary and public LOTO documents are: 0.64 (0.81) & 0.54 (0.72) and 0.61 (0.79) & 0.48 (0.67) respectively.

The exact matching agreement scores reflect lexical variations and differences in annotation granularity; normalized scores indicate substantial conceptual agreement. The slightly lower agreement on proprietary data can be attributed to the more fine-grained annotations introduced by the domain-expert annotator.

4 Experimental Setup

Safety-critical work requires careful evaluation because errors can have serious consequences. We propose three complementary evaluation angles to evaluate the proposed framework: **intermediary assessment** (*document parsing* and *instruction extraction*), final **PKG validation** (*reference alignment* and *safety compliance*), and **cross-modal alignment**. Our evaluation experiments follow the pipeline in Figure 1. We use Qwen3_14B⁶ for the evaluation in Sections 4.1 - 4.4.

Input. The initial input is a K -page procedural document $D = \{d_1, d_2, \dots, d_K\}$ in PDF format. The inputs and outputs for the subsequent stages in the PKG construction pipeline follow the formalization in Section 3.1.

³Therefore, LOTO documents are rarely available for research, and only public documents can be released to motivate future works and facilitate reproducibility; see Appendix A for annotation details.

⁴<https://roboflow.com/>

⁵Conceptual normalization considers lexical variations that do not hinder human reviewers’ understanding as a match, resulting in higher agreement scores.

⁶the Q4_K_M quantized distilled version from <https://ollama.com/library/qwen3:14b>

4.1 Intermediary Assessment–Document Parsing

We compare the extracted text elements t_k and the layout coordinates $\lambda_k = (x_1, y_1, x_2, y_2)$ for each page d_K from two parsing tools: *Docling*⁷ and *PyMuPDF*⁸. The outputs are compared with our annotated ground truths according to two criteria: 1) **layout accuracy** of extracted text, table, and image regions, and 2) **text quality** of extracted sentences. Both tools output text-layout representations (d_k, Γ_k) in reading order. However, we use *Docling* to parse inputs for the subsequent steps of the pipeline to construct the final PKG as it demonstrates higher layout accuracy.

4.1.1 Evaluation Metrics

Layout accuracy is assessed as an image segmentation task, using Intersection-over-Union (IoU) over annotated bounding boxes. Text extraction quality is assessed using the **SMART** (Amplayo et al., 2023) scorer,⁹ which measures sentence-level similarity between long texts. Together, these metrics quantify visual–text grounding fidelity.

Layout accuracy. We compute Intersection-over-Union (IoU) between predicted and annotated regions. Let L and L^* be the predicted and gold bounding regions: $\text{IoU} = \frac{|L \cap L^*|}{|L \cup L^*|}$.

Text quality. Because sentence boundaries and ordering can differ between parsers, we evaluate the extracted texts with **SMART** (Amplayo et al., 2023), which semantically aligns the predicted and reference sentences before computing *precision*, *recall*, and F_1 . We report **Smart-1** and **Smart-L**, which emphasize factual alignment and long-span coherence, respectively.

4.2 Intermediary Assessment–Instruction Extraction

We prompt Qwen3_14B to extract executable procedural actions from each parsed page (d_k, Γ_k) . The output from each page d_k , is a sequence of multi-step procedures $A_k = \{s_{k_1}, s_{k_2}, \dots, s_{k_m}\}$. We examine both the quality of the generated texts and the correctness of the procedural order.

⁷<https://github.com/docling-project/docling>

⁸<https://github.com/pymupdf/PyMuPDF>

⁹with their suggested BLEURT-20 (Pu et al., 2021) checkpoint to match unaligned sentences

4.2.1 Evaluation Metrics

We evaluate the extracted procedural steps with standard lexical metrics: **BLEU** (Papineni et al., 2002), **ROUGE** (Lin, 2004), and **METEOR** (Banerjee and Lavie, 2005)). To assess semantic correspondence and step-order alignment beyond surface overlap, we use a pairwise cosine similarity matrix between predicted and annotated reference steps:

$$S_{ij} = \frac{\mathbf{p}_i \cdot \mathbf{a}_j}{\|\mathbf{p}_i\| \|\mathbf{a}_j\|}, \quad (1)$$

where $\mathbf{p}_i$ and $\mathbf{a}_j$ are predicted and annotated step embeddings respectively. We calculate the **Diagonal Mean**, **Alignment Accuracy**, **Off-Diagonal Mean**, and **Margin Difference** from the pair-wise matrix. The **Diagonal_{mean}** captures the step-wise semantic similarity along the diagonal axis of the matrix. Mean scores use the lowest number of pairs as the sample size. **Accuracy_{alignment}** captures order preservation, quantifying how correct position-wise (i.e. according to the step order) the most similar instructions are. **Off-Diagonal_{mean}** and mean **Difference_{marginal}** measure the cross-step confusion and separability, i.e. how semantically different the unaligned instructions are. Formal definitions and an illustrative example are in Appendix B).

4.3 Final PKG–Reference Alignment

We prompt the same LLM as in Section 4.2 to extract the nodes and form *procedural unit* triplets from the procedural steps A_k according to the formalization in Section 3.1 to construct the final PKG. The resultant PKG is compared to a reference graph constructed from nodes and edges annotated according to Section 3.3. We assess *node type*, *node & edge features*, *triplet*, and *graph-level* alignments.

4.3.1 Evaluation Metrics

Node/Edge & Graph Alignments. We report node and edge features **Jaccard index**:

$$J = \frac{|X_{\text{pred}} \cap X_{\text{gold}}|}{|X_{\text{pred}} \cup X_{\text{gold}}|}.$$

Triplet matching. Given a predicted $\hat{\psi} = (\hat{v}, \hat{r}, \hat{o})$ and gold $\psi = (v, r, o)$, an *exact* lexical match requires identical triplet elements: $S_{\text{exact}}(\hat{\psi}, \psi) = \mathbb{I}[\hat{v} = v \wedge \hat{r} = r \wedge \hat{o} = o]$. *Fuzzy* uses normalized string similarity over triplet elements, and *Embedding-based* matching computes

cosine similarity between triplet text embeddings. We report **Precision**, **Recall**, and **F₁** for triplets and **average similarity** over matched pairs.

For graph-level structural similarity, we calculate the Graph Edit Distance (**GED**) and report the normalized GED (**nGED**); lower values indicate better structural agreement.

4.4 Final PKG–Procedural Safety Compliance

In addition to assessing the extracted PKG alignments with the annotated reference at varying granularity levels, we address the question: **is the extracted procedural knowledge admissible under LOTO safety constraints?** Therefore, we evaluate each extracted procedure block against safety-relevant criteria encoded in an expert-verified ontology in two settings: *Ontology-only* (deterministic checks) and *Ontology-LLM* (Qwen3-assisted, ontology-constrained). We verify the results of both settings by comparing them with a certified LOTO safety expert evaluation (see Appendix B.4).

Ontology-only. For each extracted procedural group, we infer (i) an energy type from the source text strings, (ii) procedural action targets according to the energy type, (iii) map step texts to canonical action categories, and (iv) apply safety constraints encoded in a JSON ontology to detect compliance violations, e.g. energy mismatches, action target incompatibilities, missing required categories, and step order violations. Violations are summarized per energy type and aggregated into a document-level score.

Ontology-LLM. We prompt the same LLM to evaluate the presence of safety relevant errors per energy type procedure block. The JSON safety-constraints ontology is included in the prompt. The model returns structured JSON outputs containing assessment along the same safety criteria as in the Ontology-only setting (*energy source*, *action order*, and *missing action*) to facilitate comparison.

4.4.1 Safety Compliance Metrics

Scores from procedural blocks by energy-type are aggregated into a document-level safety score. For the safety criteria encoded in the ontology (*energy source*, *action order*, and *missing procedural steps*), we report the **accuracy** and **agreement**, expressed as Pearson (r) and Spearman (ρ) scores from the two evaluation settings compared to the assessment performed by a certified LOTO expert.

4.5 Cross-Modal Alignment

To evaluate cross-modal alignment, whether procedural entities are grounded across text-visual modalities, we measure whether the predicted $(p_i, image_of, n_i)$ triplets match the annotated text-visual ground truth associations. For the proposed multi-step pipeline formalized in Section 3.1, we compare performance between *CLIP* (Radford et al., 2021) and *PaliGemma* (Steiner et al., 2024) VLM embeddings.

4.5.1 Evaluation Metrics

For image–text alignment consistency, we evaluate the $(p_i, image_of, n_i)$ relations using **Top- k Accuracy** (Acc@1, Acc@2). We further report **Mean Cosine Similarity** between aligned image and text embeddings to measure semantic closeness. These metrics quantify the extent to which each model is able to associate procedural text elements to their visual references.

4.6 Comparison with VLM-based PKG

To determine the benefits of our proposed multi-step pipeline, we compare the final PKG according to Section 3.1 and the one formalized according to Section 3.2 against the reference PKG constructed from annotations in Section 3.3. We use Qwen3.5¹⁰ and Gemma-4¹¹ as representative “best-available” vision-language models that meet the data protection requirements of domain stakeholders without increasing the hardware resource requirements.

Input. For VLM-based PKG construction, a K -page document is represented as an ordered sequence of page images $X_d = \{x_{d_1}, x_{d_2}, \dots, x_{d_K}\}$ (see Section 3.2).

4.6.1 Evaluation Metrics

We measure the cosine similarity¹² of the extracted *executable actions*, *conditional steps*, *procedural unit* aggregates consisting of action-associated entities (*action*, *target*, *tool*, *location*), and *cross-modal* edges in a pair-wise fashion. We report **F₁** scores. These criteria provide both sentence-level and intra-sentence extraction granularity measures.

¹⁰<https://huggingface.co/Qwen/Qwen3.5-4B>

¹¹<https://huggingface.co/google/gemma-4-E4B-it>

¹²<https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2> is used for embeddings; similarity threshold of 0.9 to allow for minor lexical variations while being strict for same-domain comparisons

Matching Strategy	Precision (%)	Recall (%)	F ₁ (%)	Avg. Sim. (%)	Node Jac. (%)	Edge Jac. (%)	nGED (1 - nGED) (%)
Exact	31.41	44.28	36.09	—	52.11	63.42	35.00 (65.00)
Fuzzy (0.85)	72.22	67.78	67.74	87.22	54.03	67.15	19.80 (80.20)
Embedding	81.70	77.20	77.10	95.60	56.27	69.38	13.80 (86.20)

Table 3: The evaluation results of node/edge overlaps (Node/Edge Jac.), triplet matching (Precision, Recall, F₁, and Avg. Sim), and graph-level similarity (nGED (1 - nGED)) comparing *Exact*, *Fuzzy* (threshold of 0.85), and *Embedding-based* matching strategies. Scores are macro-averaged across all procedures.

5 Results & Discussion

Document Parsing and Instruction Extraction.

Table 4 shows a trade-off between layout and text fidelity. *Docling* achieves slightly higher layout accuracy than *PyMuPDF* (78.98% vs. 76.80% IoU), whereas *PyMuPDF* obtains higher SMART scores. Since the downstream pipeline relies on structured document regions to preserve step boundaries and reading order, we prioritize layout fidelity and use *Docling* for subsequent multi-step PKG construction.

The extracted steps show moderate semantic and ordering correspondence with the reference (Table 5), with $\text{Diag}_{\text{mean}} = 64.27\%$ and $\text{Acc}_{\text{align}} = 64.29\%$. Their positive correlation ($r = 0.77$; Figure 3) indicates that higher semantic correspondence is associated with better procedural order alignment, although variation remains across examples. ROUGE-1 and METEOR are substantially higher than BLEU, consistent with substantial surface-form variation. However, small lexical changes can still alter LOTO meaning, e.g. *apply LOTO to G-1 first* versus *apply LOTO to E-1 first*. These results motivate combining semantic and lexical correspondence with domain-specific validation.

Framework	IoU	Smart-1 _{F1}	Smart-L _{F1}
PyMuPDF	76.80	58.07	51.73
Docling	78.98	50.47	47.41

Table 4: Parsing comparison showing the trade-off between layout accuracy (IoU, %) and text alignment (SMART F₁, %).

PKG Reference Alignment. Reference-based PKG scores depend strongly on the matching criterion (Table 3). Triplet F₁ increases from 36.09% with Exact matching to 67.74% with Fuzzy and 77.10% with Embedding matching, while nGED decreases from 35.00% to 13.80%. The relatively small change in Node Jaccard compared with the much larger improvement in triplet F₁ suggests

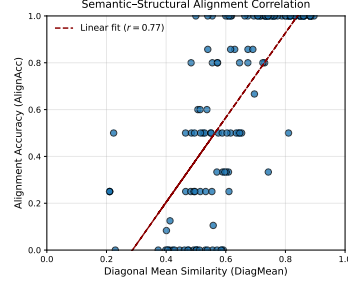


Figure 3: Correlation between semantic similarity ($\text{Diag}_{\text{mean}}$) and structural accuracy ($\text{Acc}_{\text{align}}$) across procedural steps ($r = 0.77$).

Diag	Acc _{align}	Off-Diag	Diff _{margin}
64.27	64.29	40.87	11.80
BLEU	METEOR	ROUGE-1	Human
37.06	61.92	72.25	86.02

Table 5: Mean embedding- and lexical-based scores (%). Diag and Acc_{align} capture semantic and order alignment; Off-Diag and Diff_{margin} quantify cross-step confusion.

that semantic matching primarily resolves alternative lexical realizations of procedural relations rather than substantially changing the set of extracted node types. The high average similarity under Embedding matching further indicates that many predicted triplets are semantically close to the reference despite surface-form differences.

At the same time, this result also exposes a limitation of reference-based evaluation in a safety-critical setting. Semantically similar triplets may still differ in a safety-relevant argument, equipment identifier, or procedural dependency. High embedding-based correspondence should therefore be interpreted as semantic graph similarity rather than evidence that the extracted procedure satisfies LOTO safety constraints. This motivates the separate ontology-guided compliance evaluation reported below.

Safety Compliance. Table 6 shows that ontology-guided evaluation provides an additional assessment of explicit LOTO constraints. Energy Source

verification is reliable in both settings (96.2–96.6% accuracy), suggesting that explicitly stated energy cues can be checked effectively using deterministic ontology rules. The largest benefit from Onto-LLM occurs for Action Order, where accuracy increases from 50.0% to 86.2% and expert correlation from 3.7/3.7 to 68.0/68.0. Qualitative analysis indicates that deterministic rules struggle with indirectly expressed actions, e.g. *try to start the machine* as a verification step, whereas the ontology-constrained LLM more reliably maps such expressions to predefined procedural categories. This contrast suggests that deterministic checks are most effective when safety-relevant evidence is explicit, while LLM assistance is particularly useful when linguistic variation must first be resolved before applying the same ontology-defined constraints.

Missing Action detection remains more challenging, improving from 61.5% to 69.0% accuracy, with expert correlation increasing from 29.6/29.6 to 46.2/46.2. Detecting omissions requires determining whether an expected procedural element is absent, making it less directly observable than classifying an explicitly stated action. Overall, the ontology defines the validation criteria, while the LLM acts as a constrained semantic mapper from free-form procedural language to those criteria. This preserves expert-defined safety requirements while improving their applicability to linguistically variable procedural descriptions.

Criteria	Setting	Accuracy	$r/\rho (\times 100)$
Energy Source	Ontology	96.2	46.9/46.9
	Onto-LLM	96.6	69.4/69.4
Action Order	Ontology	50.0	3.7/3.7
	Onto-LLM	86.2	68.0/68.0
Missing Action	Ontology	61.5	29.6/29.6
	Onto-LLM	69.0	46.2/46.2

Table 6: Agreement with expert ontology-driven validation. Accuracy is reported in %; Pearson r and Spearman ρ are scaled by 100.

Cross-Modal Grounding. PaliGemma outperforms CLIP on action–image grounding (Table 7), increasing Acc@1 from 11.0% to 28.6% and Acc@2 from 24.4% to 55.6%. Nevertheless, the absolute accuracy remains modest, confirming that fine-grained association of procedural actions with visual evidence is a remaining challenge.

VLM-based PKG Comparison. The multi-step framework substantially outperforms direct end-to-end VLM extraction for the main structured extraction criteria (Table 8). Action F_1 improves by 34.56

Model	Acc@1	Acc@2
CLIP	11.0	24.4
PaliGemma	28.6	55.6

Table 7: Action–image grounding accuracy (%) for CLIP and PaliGemma 2 Mix (3B-448).

Model	Criteria	End-to-End	Multi-Step	ΔF_1
Qwen3.5	Action	53.14	87.70	+34.56
	Procedural Unit	12.01	51.44	+39.43
	Conditional	24.82	23.43	−1.39
	Cross-Modal	11.76	17.27	+5.51
Gemma4	Action	39.16	88.16	+49.00
	Procedural Unit	5.46	41.58	+36.12
	Conditional	6.94	20.95	+14.01
	Cross-Modal	5.00	11.67	+6.67

Table 8: Comparison of end-to-end VLM PKG extraction and the multi-step PKG framework. Scores are F_1 ; ΔF_1 denotes Multi-Step minus End-to-End.

points for Qwen3.5 and 49.00 points for Gemma4, while Procedural Unit F_1 improves by 39.43 and 36.12 points, respectively. Conditional extraction is less consistent, with a slight decrease for Qwen3.5 but a 14.01-point gain for Gemma4. Cross-Modal relations improve for both models, although their low absolute scores show that visual grounding remains challenging.

Overall Discussion. Taken together, the results highlight complementary roles for structured decomposition, semantic evaluation, and ontology-guided safety validation. The multi-step formulation is particularly effective for extracting actions and procedural units, whereas conditional and cross-modal relations remain more challenging. Reference-based semantic metrics accommodate substantial surface-form variation, but do not directly determine whether safety-relevant procedural constraints are preserved. Ontology-guided evaluation addresses this gap by making the validation criteria explicit, while LLM assistance improves the mapping of linguistically variable instructions to those predefined categories. These findings suggest that reliable procedural knowledge extraction in safety-critical settings benefits from combining model-based extraction with explicit, expert-defined validation rather than relying on either component alone.

6 Conclusion and Future Work

This case study demonstrates complementary benefits from multi-step PKG construction and

ontology-guided validation for safety-critical procedural documents. First, the multi-step framework substantially improves *Action* and *Procedural Unit* extraction over direct end-to-end VLM generation in both Qwen3.5 and Gemma4 models, supporting the decomposition of heterogeneous LOTO documents into structured intermediate representations. Second, cross-modal results show that visual evidence can be incorporated into the PKG, although fine-grained action-image grounding remains challenging. Third, ontology-guided evaluation complements reference-based text and graph similarity by explicitly checking energy-source consistency, procedural ordering, and required-action coverage. The Onto-LLM setting further improves agreement with a certified LOTO safety expert by mapping linguistically variable instructions to predefined ontology categories.

Together, these findings support using PKO as an interface for both structured procedural representation and constraint-specific validation, while also showing that semantic similarity alone is insufficient for evaluating safety-critical procedural knowledge. Conditional extraction, missing-action detection, and cross-modal grounding remain open challenges. Future work should improve these components, extend the encoded safety criteria, and evaluate transferability to other procedural and safety-critical domains.

Limitations

Single Procedure Type. Our experiments are conducted on a small real-world proprietary dataset supplemented by additional public LOTO procedural documents. Since our results are based on a single procedure type, future works are necessary to verify the transferability of the proposed pipeline and the cross-domain adaptability claimed by the PKO (Carriero et al., 2025). As one of the objectives for sharing our work is to establish a baseline on a real-world use case that is underexplored, we plan to investigate other procedure types in other safety-critical domains for future works. To encourage future research on procedural knowledge extraction from the LOTO procedure, we will also share an annotated LOTO dataset gathered from public sources.

PKO Inter-class Relations. The inter-class relations in the PKO we exploit for PKG triplet formation only provide coarse-grained cross-modal dependencies between procedural steps and visual

references. We rely on transitive relations (Table 1) to map between procedure-level dependencies and semantic roles within each atomic action sentence. Similarly, the PKO does not support direct dependencies for image segments and step-action classes. Future research to investigate dynamic ontology updates with training data may be exploited to facilitate fine-grained action-image-segment and entity-image-segment relation identification.

Procedural Sequence Ordering. We rely on the document layout structure, i.e. left-to-right top-to-bottom reading order, to infer the temporal order of a procedural step sequence. No claims can be made from our results regarding the evaluated models’ ability to infer procedural step ordering without layout information. We leave the procedural sequence ordering task for future works.

Multi-Component System Evaluation. Our proposed pipeline relies on multiple automatic metrics that are prevalent for the respective experiments at different stages in the workflow. As the results illustrate, curating a unified approach to evaluating multi-component systems holistically requires both technical and domain-specific insights. Ensuring that metrics commonly used in natural language processing (NLP) tasks provide reliable compliance with target-domain requirements and safety constraints remains a challenging task.

Ontology-guided Evaluation Our ontology-driven evaluation inherits the strengths and limitations of the underlying ontology. In particular, constraint checks are only as accurate as the ontology’s coverage of device types, action categories, and safety rules. Missing or overly coarse concepts can lead to false positives (flagging valid steps as violations) or false negatives (failing to detect unsafe omissions or mismatched energy controls). Specialized-domain procedures may change over time; terminology and regulations vary in different regions and organizations. Consequently, an ontology that is not regularly maintained may become outdated. Some critical information is implicit or context-dependent (e.g. non-standardized shorthands or steps distributed across tables and figures), which can be difficult to encode explicitly. These factors suggest that ontology-guided evaluation is best viewed as a structured reliability layer that facilitates expert collaboration and oversight in an LLM-assisted workflow.

Ethical and Legal Considerations

The publicly sourced LOTO documents are obtained exclusively from open and unrestricted online sources. We use them solely for academic research purposes, without altering their content or redistributing them outside the scope permitted by fair use. The proprietary documents are handled under the confidentiality agreement with the industry partner organization. The use of LLMs/VLMs is only via locally hosted models in compliance with the confidentiality agreement; no sensitive information, machine identifiers, or company identification details are shared externally. All processing and analysis steps are carried out in accordance with ethical guidelines to ensure responsible use of industrial safety documentation.

Acknowledgments

The authors thank anonymous reviewers for their constructive feedback, Stefania Baldassarre and Diego Beltrame for their domain expertise in the evaluation activities. The work was partially funded by the European Union (EU) through the project PERKS–Eliciting and Exploiting Procedural Knowledge in Industry 5.0 (ID: 101120323)¹³ under the “Digital, Industry, and Space” funding program and the German Federal Ministry of Research, Technology and Space (BMFTR) through the project TRAILS–Trustworthy and Inclusive Machines (ID: 16IW24005).¹⁴

References

- Reinald Kim Amplayo, Peter J Liu, Yao Zhao, and Shashi Narayan. 2023. [SMART: Sentences as basic units for text evaluation](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda*. OpenReview.net.
- Christoph Auer, Maksym Lysak, Ahmed Nassar, Michele Dolfi, Nikolaos Livathinos, Panos Vagenas, Cesar Berrospi Ramis, Matteo Omenetti, Fabian Lindlbauer, Kasper Dinkla, Lokesh Mishra, Yusik Kim, Shubham Gupta, Rafael Teixeira de Lima, Valery Weber, Lucas Morin, Ingmar Meijer, Viktor Kuropiatnyk, and Peter W. J. Staar. 2024. [Docling technical report](#). arXiv Preprint 2408.09869, arXiv.
- Satanjeev Banerjee and Alon Lavie. 2005. [METEOR: An automatic metric for MT evaluation with improved correlation with human judgments](#). In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.
- Kriti Bhattarai, Inez Y. Oh, Zachary B. Abrams, and Albert M. Lai. 2024. [Document-level Clinical Entity and Relation extraction via Knowledge Base-Guided Generation](#). In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, pages 318–327, Bangkok, Thailand. Association for Computational Linguistics.
- Valentina Anita Carriero, Mario Scrocca, Ilaria Baroni, Antonia Azzini, and Irene Celino. 2025. [Procedural Knowledge Ontology \(PKO\)](#). In *The Semantic Web: 22nd European Semantic Web Conference, ESWC 2025, Portoroz, Slovenia, June 1–5, 2025, Proceedings, Part II*, page 334–350, Berlin, Heidelberg. Springer-Verlag.
- Steffen Castle and Julian Moreno Schneider. 2025. [Entity Linking using LLMs for Automated Product Carbon Footprint Estimation](#). In *Proceedings of the 1st Workshop on Ecology, Environment, and Natural Language Processing (NLP4Ecology2025)*, pages 56–60, Tallinn, Estonia. University of Tartu Library.
- Alla Chepurova, Yuri Kuratov, Aydar Bulatov, and Mikhail Burtsev. 2024. [Prompt Me One More Time: A Two-Step Knowledge Extraction Pipeline with Ontology-Based Verification](#). In *Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing*, pages 61–77, Bangkok, Thailand. Association for Computational Linguistics.
- Yew Ken Chia, Lidong Bing, Soujanya Poria, and Luo Si. 2022. [RelationPrompt: Leveraging Prompts to Generate Synthetic Data for Zero-Shot Relation Triplet Extraction](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 45–57, Dublin, Ireland. Association for Computational Linguistics.
- John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S. Rosen, Gerbrand Ceder, Kristin A. Persson, and Anubhav Jain. 2024. [Structured information extraction from scientific text with large language models](#). *Nature Communications*, 15(1):1418. Publisher: Nature Publishing Group.
- Ruth Dannenfelser, Jeffrey Zhong, Ran Zhang, and Vicky Yao. 2023. [Into the Single Cell Multiverse: an End-to-End Dataset for Procedural Knowledge Extraction in Biomedical Texts](#). In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023) Datasets and Benchmarks Track*, volume 36, pages 10922–10934. Curran Associates, Inc.
- Jiaying Gong and Hoda Eldardiry. 2024. [Prompt-based Zero-shot Relation Extraction with Semantic Knowledge Augmentation](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13143–13156, Torino, Italia. ELRA and ICCL.

¹³<https://perks-project.eu/>

¹⁴<https://trails-dfki.github.io/>

- Andrzej Gretkowski, Dawid Wiśniewski, and Agnieszka Ławrynowicz. 2022. [Should We Afford Affordances? Injecting ConceptNet Knowledge into BERT-Based Models to Improve Commonsense Reasoning Ability](#). In *Knowledge Engineering and Knowledge Management*, pages 97–104, Cham. Springer International Publishing.
- Abhirut Gupta, Abhay Khosla, Gautam Singh, and Gargi Dasgupta. 2018. [Mining Procedures from Technical Support Documents](#). *CoRR*, abs/1805.09780.
- Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. [REBEL: Relation Extraction By End-to-end Language generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Jorj X. McKie and R Liu. 2025. [PyMuPDF: A Python Binding for MuPDF](#). <https://github.com/pymupdf/pymupdf>. Original-date: 2012-10-06T18:54:25Z.
- Dena Mujtaba and Nihar Mahapatra. 2019. [Recent Trends in Natural Language Understanding for Procedural Knowledge](#). In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 420–424.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Amy Pu, Hyung Won Chung, Ankur Parikh, Sebastian Gehrmann, and Thibault Sellam. 2021. [Learning compact metrics for MT](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 751–762, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.
- Haopeng Ren, Yushi Zeng, Yi Cai, Bihan Zhou, and Zetao Lian. 2023. [Constructing Procedural Graphs with Multiple Dependency Relations: A New Dataset and Baseline](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8474–8486, Toronto, Canada. Association for Computational Linguistics.
- Andreas Steiner, André Susano Pinto, Michael Tschanen, Daniel Keysers, Xiao Wang, Yonatan Bitton, Alexey Gritsenko, Matthias Minderer, Anthony Sherbondy, Shangbang Long, Siyang Qin, Reeve Ingle, Emanuele Bugliarello, Sahar Kazemzadeh, Thomas Mesnard, Ibrahim Alabdulmohsin, Lucas Beyer, and Xiaohua Zhai. 2024. [PaliGemma 2: A family of versatile vlms for transfer](#). *arXiv preprint. ArXiv:2407.07726* [cs].
- Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, Mayank Chaturvedi, Aditya Chawla, Victor Cotruta, Alice Coucke, Phil Culliton, Robert Dadashi, Lucas Dixon, Mohamed Elhawaty, Utku Evci, and 304 others. 2026. [Gemma 4 technical report](#). *Preprint*, arXiv:2607.02770.
- Milena Trajanoska, Riste Stojanov, and Dimitar Trajanov. 2023. [Enhancing Knowledge Graph Construction Using Large Language Models](#). *ArXiv:2305.04676* [cs].
- Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftexhar Naim, Joe Zou, Feiyang Chen, Daniel Cer, Alice Lisak, Min Choi, Lucas Gonzalez, Omar Sanseviero, Glenn Cameron, Ian Ballantyne, Kat Black, Kaifeng Chen, and 70 others. 2025. [Embeddinggemma: Powerful and lightweight text representations](#).
- Jianing Wang, Qiushi Sun, Xiang Li, and Ming Gao. 2024. [Boosting language models reasoning with chain-of-knowledge prompting](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4958–4981, Bangkok, Thailand. Association for Computational Linguistics.
- Te-Lin Wu, Alex Spangher, Pegah Alipoormolabashi, Marjorie Freedman, Ralph Weischedel, and Nanyun Peng. 2022. [Understanding Multimodal Procedural Knowledge by Sequencing Multimodal Instructional Manuals](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4525–4542, Dublin, Ireland. Association for Computational Linguistics.
- Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Lim Wang, Yunfei Chu, and Junyang Lin. 2026. [Qwen3.5-omni technical report](#). Technical report.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Ly, and 1 others. 2025. [Qwen3 technical report](#). Technical report.

Zi Yang and Eric Nyberg. 2015. [Leveraging Procedural Knowledge for Task-oriented Search](#). In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 513–522, New York, NY, USA. Association for Computing Machinery.

Chunmo Zheng, Xing Su, Yinqiu Tang, Junjing Li, and Mohamad Kassem. 2024. [Retrieve-Enhance-Verify: A Novel Approach for Procedural Knowledge Extraction from Construction Contracts Via Large Language Models](#). SSRN:4883720.

A Annotation Guidelines

A.1 LOTO Annotation

This section describes the annotation procedure used to convert LOTO documents into structured procedural knowledge. A Computer Science Master’s student performs layout annotation. The Master’s student and a PhD researcher conduct procedural text annotations. A domain expert provides annotations for the proprietary documents and the PhD researcher annotates the public ones.

Inter-Annotator Agreement. We calculate the mean *Jaccards* and *Cohen’s κ* agreement scores (with exact lexical and normalized conceptual agreement matching) for structured action tuples (*action*, *target*, *tool*, *location*), excluding auxiliary fields. The mean exact (and normalized) scores for proprietary and public LOTO documents are 0.64 (0.81) & 0.54 (0.72) and 0.61 (0.79) & 0.48 (0.67) respectively.

1. Document Parsing. Annotation begins by reviewing the full LOTO document page. Each document typically contains a set of lockout steps consisting of step numbers, action descriptions, required tools, target components, location information, verification steps, and corresponding images. The annotator identifies and isolates the lockout step table along with any associated images.

2. Action Extraction. For each lockout step, annotator extracts the following two fields:

- **step_order:** The sequential number of the step as written in the LOTO procedure.
- **action:** The textual action describing what must be performed (for example, “Turn Disconnect to the off position” or “lock out”).

The goal at this stage is to obtain a clean list of ordered natural language instructions exactly as they appear in the document.

3. Procedural Unit Extraction. For each extracted instruction, annotators expand the action into a structured node containing the following fields:

- **tool:** The tool specified in the document (for example, Lock and hasp device, a ball valve lockout device).
- **target:** The object or machine component being acted upon (for example, E-1 Disconnect, G-1 Ball Valve).
- **location:** The location text describing where the action takes place (for example, East side of the machine).
- **conditional units:** The verification or warning instruction required for safety (for example, “Attempt to restart at control panel” or “Verify pressure has bled off”).
- **cross-modal:** The bounding box coordinates for the figure associated with that step.

Each step is annotated independently, using terminology directly copied from the document to ensure accuracy.

4. PKG Construction. The annotated instruction sentences and action-associated entities are formalized into a PKG according to Section 3.1.

Annotation Costs. Annotation was completed during a period of 7 months, averaging 8 hours per week. Graduate student assistants are compensated at the rate of 18 EUR per hour in compliance with local regulations. Other annotators are compensated through project funding according to their respective consortium agreements.

B Evaluation Details

This appendix summarizes the implementation details of the evaluation strategies in Section 4.

B.1 Instruction Extraction

We assess semantic and order-level correspondence using a pairwise cosine similarity matrix between predicted and reference steps:

$$S_{ij} = \frac{\mathbf{p}_i \cdot \mathbf{a}_j}{\|\mathbf{p}_i\| \|\mathbf{a}_j\|}, \quad (2)$$

Predicted (row) & Reference (col) Steps	$Step_{Ref_1}$	$Step_{Ref_2}$	$Step_{Ref_3}$	$Step_{Ref_4}$	$Step_{Ref_5}$	$Step_{Ref_6}$	$Step_{Ref_7}$
notify affected personnel	71.07	28.25	21.12	13.65	15.66	18.68	23.48
review lockout procedure	27.92	88.57	34.32	26.24	57.64	22.73	31.11
perform machine stop	21.54	34.57	91.82	27.47	32.80	29.90	31.82
isolate energy	14.73	26.48	28.42	88.36	54.95	53.37	20.12
lockout energy	17.39	57.65	28.83	53.52	87.49	46.95	22.79
dissipate energy	20.54	24.40	30.03	55.57	49.65	86.75	14.67
attempt restart	21.52	31.67	29.34	20.38	25.26	15.85	88.43

Table 9: Scaled cosine similarity matrix S_{ij} between predicted (row) and annotated reference (column) steps (%). Bold diagonal values indicate correct semantic correspondences between predicted and reference instructions.

Table 10: $Step_{Ref_j}$ to Annotation reference step description.

$Step_{Ref_j}$	Annotation Step Description
$Step_{Ref_1}$	notify employees
$Step_{Ref_2}$	review lockout procedure
$Step_{Ref_3}$	perform machine stop
$Step_{Ref_4}$	isolate energy
$Step_{Ref_5}$	lockout energy
$Step_{Ref_6}$	dissipate energy
$Step_{Ref_7}$	attempt restart

where $\mathbf{p}_i$ and $\mathbf{a}_j$ are step embeddings. Using $L = \min(n, m)$, we report:

$$\mathbf{Diag}_{mean} = \frac{1}{L} \sum_{i=1}^L S_{ii}, \quad (3)$$

$$\mathbf{AlignAcc} = \frac{1}{L} \sum_{i=1}^L I \left[\arg \max_j S_{ij} = i \right], \quad (4)$$

$$\mathbf{OffDiag}_{mean} = \frac{1}{L(L-1)} \sum_{i \neq j} S_{ij}, \quad (5)$$

$$\mathbf{MarginDiff}_{mean} = \frac{1}{L} \sum_{i=1}^L \left(S_{ii} - \max_{j \neq i} S_{ij} \right). \quad (6)$$

Table 9 provides an example to demonstrate the pair-wise cosine similarity matrix. Table 10 provides the annotation reference for the step descriptions in Table 9.

B.2 Human Review

We collect human judgments for (i) instruction extraction and (ii) node (slot) extraction using a structured Google Form that mirrors document structure (procedure block $\rightarrow$ ordered action steps) to gain qualitative assessment insights.

Labels. For each *Source Identification & Details* item and each *Action Step*, annotators select *CORRECT*, *PARTIAL*, *WRONG*, or *N-SURE*. A free-text field supports error notes. *PARTIAL* label is when some details are correct, but there is some incorrect information, e.g. correct but incomplete energy source identification.

Node (slot) inspection. Each step is shown with extracted slots (*Action*, *Target*, *Tool*, *Location*). Annotators apply the same labels while inspecting both the full instruction and its slots.

B.3 Extended Node Extraction Results

Table 11 reports node extraction **Precision**, **Recall**, and **F₁** by node type according to *Exact*, *Fuzzy*, and *Embedding* matching strategies.

B.4 Expert Safety Compliance Evaluation

We verify ontology-driven PKG safety compliance evaluation with a certified LOTO safety expert. Each predicted procedure is split into energy-specific blocks and shown with a short description and the ordered steps. The expert answers:

- *Correct energy source?* (Yes/Partially/No)
- *Instruction order correct?* (Yes/No)
- *Missing required steps?* (Yes/No)
- *Structural error severity* (None/Minor/Major)
- *Approve this block?* (Yes/Needs Revision/No)
- *Optional Comments*

We report accuracy for binary criteria (with *Partially* treated as “No” per experiment) and Pearson r / Spearman ρ .

Node Type	Exact Match			Fuzzy Match			Embedding Similarity			
	P	R	F1	P	R	F1	P	R	F1	AvgSim
Action	84.75	84.93	84.84	87.00	87.19	87.10	88.75	88.95	88.85	0.97
Target	20.00	20.00	20.00	22.00	22.00	22.00	89.50	89.50	89.50	0.93
Tool	58.29	58.00	58.15	58.29	58.00	58.15	99.50	99.00	99.25	0.97
Location	42.64	42.00	42.32	48.22	47.50	47.86	88.32	87.00	87.66	0.93

Table 11: Precision (P), Recall (R), and F_1 scores (in %) for node type extraction under exact, fuzzy, and embedding-based matching. The **Action** row reports the mean over *Action* and *Verification*. Embedding similarities use google/embeddinggemma-300m (Appendix C)

B.5 Multimodal Grounding

In this study, multimodal grounding links each extracted step, slot, or PKG entity to visual evidence in the same manual, such as diagram labels, table cells containing device identifiers, and visual cues that disambiguate components or tools. Grounding accuracy is evaluated with strict Acc@1, but industrial manuals exhibit substantial domain shift: component terminology, abbreviations, and device IDs are rare in general-purpose VLM pretraining data, and many referents appear as schematic symbols or short alphanumeric labels rather than natural images with descriptive captions. These properties make exact top-1 matching difficult, even when the retrieved evidence is still useful as a shortlist for expert verification.

To reflect the operational objective, we report **Acc@k** and evidence recall at small k and treat multimodal grounding as evidence support for downstream validation, where the main benefit is conservative flagging of unsupported target IDs or component types. We expect grounding to improve with domain-specific, machine-specific, and brand-specific visual inputs, e.g. by adapting the retriever on manufacturer manuals and diagrams using symbol-to-component inventories, device-ID lexicons, and weak supervision from document structure.

C Implementation

Technical implementation details are provided here for reproducibility. Our code will be made available after publication at the repository: github.com/pokarats/Onto-LOTO-PKEval.

C.1 Frameworks and Models

We list the main tools and models used in our pipeline; links point to reference implementations.

1. **Document parsing:** *Docling* (Auer et al., 2024) (github.com/docling-project/docling),

PyMuPDF (McKie and Liu, 2025) (github.com/pymupdf/PyMuPDF).

2. **LLM (instruction extraction and PKG construction):** *Qwen3* distilled qwen3:14b (Yang et al., 2025) via Ollama (ollama.com/library/qwen3:14b), using default decoding settings.
3. **Similarity and embedding models:** BLEURT-20 (Pu et al., 2021) for SMART (Amplayo et al., 2023) (github.com/google-research/bleurt), and google/embeddinggemma-300m (Vera et al., 2025) (huggingface.co/google/embeddinggemma-300m) for embedding-based PKG matching.
4. **Vision-Language Models (Cross-Modal Grounding):** openai/clip-vit-large-patch14-336 (CLIP) (Radford et al., 2021) and google/paligemma2-3b-mix-448 (Steiner et al., 2024) (both from Hugging Face).
5. **Vision-Language Models (PKG end-to-end Baselines):** Qwen/Qwen3.5-4B (Xu et al., 2026) and google/gemma-4-E4B-it (Team et al., 2026) (both from Hugging Face).

C.2 Hardware

All experiments run locally on a server with 6 NVIDIA GeForce RTX 2080 Ti (11GB VRAM each) GPUs. The software environment uses openSUSE Leap 15.5 and locally managed Python environments. All experiments involving LLMs/VLMs were performed locally without external API calls.

C.3 Runtime and Cost

All components are executed locally. Document parsing and ontology-only validation are CPU-bound and scale linearly with document length

and number of extracted steps. Embedding-based matching and cross-modal grounding add GPU compute but remain batchable and deterministic.

LLM-based stages (instruction extraction, slot extraction, and ontology-LLM auditing) dominate runtime because they require autoregressive decoding and scale with the number of processed pages/blocks and prompt length. Since we run qwen3:14b locally via Ollama, there is no per-call monetary cost; the primary cost is GPU time. For API-based deployments, cost would be token-driven, and ontology-LLM auditing would be the most expensive component due to embedding the ontology text in every prompt.

C.4 Prompting

We adopt a two-stage prompting design. In the first stage, the LLM extracts only executable procedural instructions from the parsed pages in a few-shot setting and returns them in a structured JSON format. In the second stage, conditioned on the extracted steps, the LLM performs slot-level semantic extraction (action, target/object, tool, location, verification) aligned with a subset of PKO classes, using concise label definitions and few-shot exemplars to standardize outputs. All prompts will be released with the accompanying code repository.