Beyond Predefined Schemas: TRACE-KG for Context-Enriched Knowledge Graph Generation

Mohammad Sadeq Abolhasani [1] **Yang Ba** [1] **Yixuan He** [2] **Rong Pan** [1]

Abstract

Knowledge graph generation typically relies either on predefined ontologies or on schema-free extraction. Ontology-driven pipelines enforce consistent typing but require costly schema design and maintenance, whereas schema-free methods often produce fragmented graphs with weak global organization, especially in long technical documents with dense, context-dependent information. We propose **TRACE-KG** (**T**ext-d**R**iven schem**A** for **C**ontext-**E**nriched **K**nowledge **G**raphs), a framework that jointly constructs a context-enriched knowledge graph and an induced schema without assuming a predefined ontology. TRACE-KG captures conditional relations through structured qualifiers and organizes entities and relations using a data-driven schema that serves as a reusable semantic scaffold while preserving full traceability to the source evidence. Experiments show that TRACE-KG produces structurally coherent, traceable knowledge graphs and offers a practical alternative to both ontology-driven and schema-free construction pipelines.

1. Introduction

Knowledge graphs (KGs) are increasingly used as long-lived substrates for organizing, integrating, and querying knowledge across documents, systems, and tasks (Hogan et al., 2021). Unlike raw text or traditional data models, graphs make structure explicit by representing entities and multi-relational links directly, supporting path-based queries and enabling *gradual schema commitment*: a graph can begin with a lightweight structure and evolve as requirements change rather than requiring a fixed schema from the outset. At the same time, large language models (LLMs) (Naveed et al., 2025) have enabled powerful text-centric workflows such as retrieval-augmented generation (RAG) (Lewis et al., 2020), which can answer many *local* questions directly from unstructured corpora.

Recent work suggests that explicit structured knowledge becomes particularly valuable when queries are *global* rather than local—when answers must integrate evidence across documents, enforce consistency, or remain inspectable and consistently organized across tasks and time (Khorashadizadeh et al., 2024; Bian, 2025). In such settings, a KG acts not merely as an index over text but as a persistent structured representation layer that complements LLMs. Yet constructing high-quality KGs directly from complex technical documents remains challenging and largely unsolved (Xue & Zou, 2022; Zhong et al., 2023). A key reason is that useful KGs require more than extracting plausible triples: they require *corpus-level organization*, typically provided by a *schema*—a shared semantic vocabulary of entity and relation types that promotes normalization and more consistent organization across documents (Hogan et al., 2021). However, most existing text-to-KG pipelines operate in one of two common extremes.

At one extreme, schema-free approaches extract entities and relations directly from text without enforcing a shared type vocabulary or corpus-level consolidation (Mo et al., 2025; Mihindukulasooriya et al., 2023). While this often yields high *local recall*, the resulting graphs fragment globally: the same entity may appear under multiple lexical variants, and semantically similar relations proliferate as near-duplicate predicates, in extreme cases producing relation vocabularies nearly as large as the edge set (Hofer et al., 2024; Bian, 2025; Mo et al., 2025).

At the opposite extreme, ontology-driven pipelines map text into a predefined ontology with fixed entity and relation categories (Mihindukulasooriya et al., 2023; Hofer et al., 2024). Although this enforces consistent typing and semantics (Hogan et al., 2021), it assumes a suitable ontology exists and remains aligned with evolving terminology and document granularity. In practice, ontology construction and maintenance require substantial expert effort and often

[1]School of Computing and Augmented Intelligence, Arizona State University, Tempe, USA [2]School of Mathematical and Natural Sciences, Arizona State University, Tempe, USA. Correspondence to: Mohammad Sadeq Abolhasani <mabolhas@asu.edu>.

The Workshop on Graph Foundation Models at the 43rd International Conference on Machine Learning (ICML 2026), Seoul, South Korea.

become a bottleneck (Hofer et al., 2024; Abolhasani & Pan, 2024). Moreover, restricting extraction to a fixed ontology can exclude information that does not map cleanly to predefined classes or relations, while relation vocabularies often remain comparatively coarse even when entity typing is enforced (Mo et al., 2025; Bai et al., 2025).

These limitations are particularly acute in complex domain-specific corpora, where terminology is dense, lexical variation is high, external reference knowledge may be unavailable, and critical information is distributed across narrative text as well as figures, tables, diagrams, and equations (Sun et al., 2025; Bian, 2025). We therefore use a corpus of engineering and maintenance technical documents as a motivating case study, highlighting key challenges for text-to-KG pipelines, including consolidation, schema induction, and context handling. A central difficulty in such settings is the interplay between *lexical* and *semantic* heterogeneity (Hofer et al., 2024; Chen et al., 2024): the same concept may appear under multiple surface forms, while similar expressions may denote different concepts depending on context. Unresolved lexical variation fragments evidence across duplicate nodes, whereas over-merging semantically distinct entities introduces incorrect relations (Wang et al., 2025a; Pons et al., 2024; Ding et al., 2024; Wang et al., 2025b). Moreover, relations in technical documents are often conditional—holding only under specific operating modes, temporal intervals, or constraints—so encoding them as unconditional triples can produce oversimplified or contradictory graphs (Hogan et al., 2021; Jiang et al., 2019; Xu et al., 2024). Capturing contextual qualifiers is therefore essential for faithful and inspectable knowledge representations.

These observations motivate a third approach between ontology-driven and schema-free settings. Ontology-driven pipelines assume a predefined ontology, $(D, O) \rightarrow G$, whereas schema-free pipelines extract a graph directly, $D \rightarrow G_{\text{sf}}$. To address the gap between these settings, we introduce **TRACE-KG** (**T**ext-d**R**iven schem**A** for **C**ontext-**E**nriched **K**nowledge **G**raphs), a data-driven framework for *schema-inducing KG generation*: given only a corpus D, TRACE-KG constructs a context-enriched graph $G = (E, R, S)$, where E contains resolved entities, R contains grounded relation instances, and S is an induced schema over both entities and relations. TRACE-KG supports multimodal document ingestion, iterative entity and relation resolution, relation canonicalization, structured qualifiers for conditional relations, and source-grounded traceability. Our main contributions are as follows:

- We introduce **TRACE-KG**, a multimodal, end-to-end framework for constructing **context-enriched** knowledge graphs from complex documents, with **explicit modeling of condition-aware relations** and provenance-preserving traceability.
- We propose **TRACE-Schema**, a data-driven schema induction mechanism that organizes **both** entities and relations into a **reusable semantic scaffold**, enabling corpus-level consolidation without relying on a predefined ontology.
- We develop an **evaluation framework** that compares the induced schema with human-created ontologies, enabling systematic evaluation of **schema coverage**, **granularity alignment**, and **structural consistency**.

2. Related Work

Schema commitment in text-to-KG construction. A central distinction in LLM-based text-to-KG pipelines is *when* the schema is fixed. Ontology-driven approaches populate a predefined conceptual model, improving normalization and semantic consistency when the ontology matches the corpus (Mihindukulasooriya et al., 2023; Ameri et al., 2026; Abolhasani & Pan, 2024; Tahsin et al., 2024), but they depend on costly ontology design and are brittle under evolving terminology and corpus-specific granularity (Hofer et al., 2024). At the other extreme, schema-free pipelines avoid predefined ontologies but often fragment entity and relation vocabularies. KGGen addresses this via post-hoc clustering and LLM-guided consolidation (Mo et al., 2025), and more recent work moves toward induced schema (Bai et al., 2025; Sun et al., 2025); however, such schemas remain partial, uneven across entities and relations, and weakly coupled to corpus-level consolidation.

Local resolution, context, and reliability. Another line of work focuses on improving local grounding and extraction quality. EntGPT and comparison-based prompting improve entity linking and matching by constraining LLM decisions (Ding et al., 2024; Wang et al., 2025b), KG-assisted disambiguation exploits existing graph structure (Pons et al., 2024), and schema-conditioned extractors such as GLiNER2 support structured extraction under user-provided label sets (Zaratiana et al., 2025). Separately, prior work shows that many scientific and technical relations are conditional rather than unconditional triples, motivating qualifier- or context-aware representations (Jiang et al., 2019; Xu et al., 2024; Qin et al., 2025), and robustness-oriented workflows such as KARMA (Lu & Wang, 2025). The core challenge is not extraction quality in isolation, but open-world *co-construction* of a graph and its schema: a reusable schema over both entities and relations must be induced from the corpus while entity identity, relation canonicalization, conditional context, and provenance remain jointly consistent across the entire graph. Extended discussion appears in Appendix A.

3. Preliminaries

3.1. Corpus and text units

Let $D = \{d_1, \ldots, d_N\}$ denote a corpus of documents. Each document is converted into a provenance-preserving text stream, where textualized descriptions of non-text elements such as figures, tables, diagrams, or equations may be integrated when available. The stream is partitioned into sentence-preserving chunks. Let C denote the set of sentence-preserving chunks. Each chunk $c \in C$ has an identifier $\mathrm{id}(c)$, textual content $\mathrm{text}(c)$, and provenance metadata $\mathrm{prov}(c)$ linking it to its source region.

3.2. Entities and relations

An *entity mention* is a text span m in a chunk c that refers to an object, event, process, or abstract concept. Let M denote the set of extracted mentions, where each mention stores its source chunk $\mathrm{chunk}(m) \in C$, span, name, description, type hint, confidence score, and justification excerpt. For a chunk c, let $M(c) = \{m \in M : \mathrm{chunk}(m) = c\}$.

A *resolved entity* is a canonical graph-level object obtained by consolidating mentions that refer to the same underlying object or concept. Let E denote the set of resolved entities, with $\mathrm{ResolvedEnt} : M \to E$ mapping each mention to its resolved entity. The entities supported by chunk c are $E(c) = \{\mathrm{ResolvedEnt}(m) : m \in M(c)\}$. Each entity $e \in E$ may also carry intrinsic properties $\mathrm{intr}(e)$, represented as typed key–value annotations.

Let R denote the set of relation instances. A relation instance $r \in R$ is a directed, source-grounded statement $r = (e_s, \ell_{\mathrm{raw}}, e_t, Q, \pi, s)$, where $e_s, e_t \in E$ are the source and target entities, ℓ_{raw} is the raw predicate label, Q is a set of contextual qualifiers, π is provenance metadata, and $s \in [0, 1]$ is a confidence score. Multiple relation instances may connect the same ordered entity pair, so the graph is represented as a directed multigraph. We denote the set of canonical relation labels by RelCan and use $\mathrm{CanonicalRel} : R \to \mathrm{RelCan}$ to assign each relation instance to a canonical label. Qualifiers Q encode contextual conditions such as temporal, spatial, operational, conditional, uncertainty, or causal constraints.

3.3. Induced schema and context-enriched graph

TRACE-KG organizes entities and relations through an induced schema. In this paper, a *schema* denotes a data-driven vocabulary and hierarchy of entity and relation types used to organize the generated graph.

At the entity level, EntCls denotes induced entity classes and $\mathrm{EntClsGroup}$ denotes broader entity class groups, with mappings $\tau_{\mathrm{ent}} : E \to \mathrm{EntCls}$ and $\gamma_{\mathrm{ent}} : \mathrm{EntCls} \to \mathrm{EntClsGroup}$.

At the relation level, RelCls denotes induced relation classes and $\mathrm{RelClsGroup}$ denotes broader relation class groups, with mappings $\tau_{\mathrm{rel}} : \mathrm{RelCan} \to \mathrm{RelCls}$ and $\gamma_{\mathrm{rel}} : \mathrm{RelCls} \to \mathrm{RelClsGroup}$.

Together, these define the induced schema $S = (\mathrm{EntCls}, \mathrm{EntClsGroup}, \mathrm{RelCls}, \mathrm{RelClsGroup}, \tau_{\mathrm{ent}}, \gamma_{\mathrm{ent}}, \tau_{\mathrm{rel}}, \gamma_{\mathrm{rel}})$.

A *context-enriched knowledge graph* is defined as $G = (E, R, S)$, where entities are associated with semantic descriptions, intrinsic properties, schema assignments, confidence estimates, justification excerpts, and provenance links, while relation instances additionally carry raw and canonical predicate labels together with contextual qualifiers. Intrinsic properties and qualifiers are represented as annotations attached to nodes and edges rather than as separate graph elements. Provenance links connect entities and relations to their supporting source evidence, enabling source-grounded traceability and auditability.

4. Methodology

Given a corpus D, TRACE-KG constructs a context-enriched knowledge graph $G = (E, R, S)$ grounded in document evidence and organized through an induced schema. The framework is built around a repeated three-stage pattern: **(i) high-recall recognition**, which extracts local candidates while deferring global normalization; **(ii) semantic neighborhood formation**, which embeds multi-field representations and clusters them into candidate neighborhoods; and **(iii) constrained resolution**, where an LLM selects structured actions over explicit identifiers and deterministic validators execute and log the graph updates. The LLM therefore proposes semantic decisions but does not directly mutate the graph state. Algorithm 1 abstracts this mechanism for EntRes, EntClsRes, and RelRes, while Figure 1 shows the full workflow. Implementation details are provided in Appendix B.

4.1. Document ingestion and chunking

Documents are converted into unified, provenance-preserving text streams. When non-text regions such as figures, tables, diagrams, or equations are available, TRACE-KG supports textualizing them with a vision-language document processor while preserving page- and region-level provenance. The resulting streams are segmented into sentence-preserving chunks $c \in C$. Later stages may use neighboring chunks for disambiguation, but extracted entities and relations must be grounded in the focus chunk.

4.2. Entity layer: recognition and iterative resolution

Entity recognition (EntRec). EntRec performs high-recall, chunk-local mention extraction. For each focus

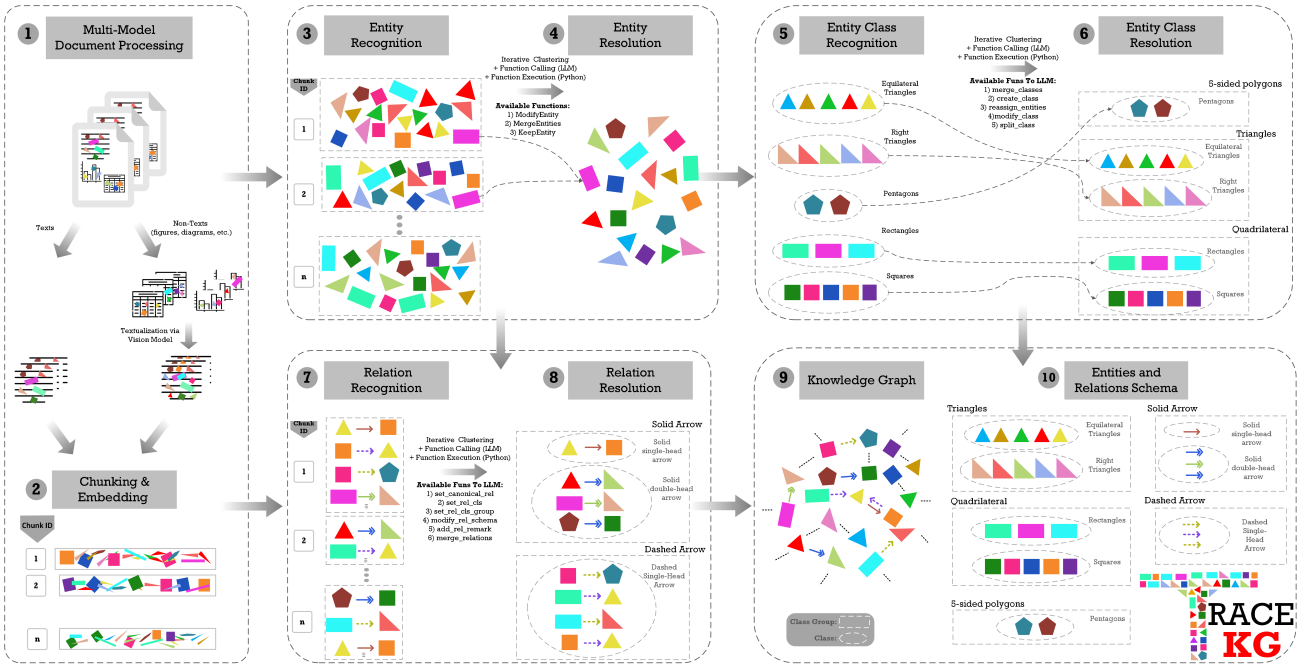

Figure 1. TRACE-KG pipeline. (1–2) **Processing/Chunking**: documents are textualized and chunked (scattered, unorganized shapes). (3) **EntRec**: local entity mentions are extracted but remain unresolved (e.g., duplicate pink rectangles). (4) **EntRes**: duplicate mentions are consolidated into resolved entities (e.g., one pink rectangle). (5) **EntClsRec**: resolved entities are grouped into entity classes (e.g., rectangles grouped together). (6) **EntClsRes**: entity classes are refined into broader class groups (e.g., rectangles and squares under a broader quadrilateral group). (7) **RelRec**: grounded relation instances are extracted with qualifiers (e.g., mixed solid single-headed, solid double-headed, and dashed single-headed arrows). (8) **RelRes**: relation labels are canonicalized and organized into relation classes and class groups (e.g., arrow types separated into solid single-headed, solid double-headed, and dashed single-headed arrows, then grouped into broader solid and dashed relation groups). (9) **KG**: resolved entities and relation instances form the context-enriched KG. (10) **Schema**: induced entity and relation hierarchies form the schema. Stages (4), (6), and (8) use semantic clustering, structured LLM actions, and deterministic execution.

chunk c, the extractor receives $\mathrm{text}(c)$ and a short window of preceding chunks for disambiguation, but extracts mentions only from the focus chunk. Each mention includes a mention-level name, description, broad type hint, confidence score, and justification excerpt grounded in $\mathrm{text}(c)$. Explicit intrinsic properties are extracted as typed key–value annotations when supported by the text. At this stage, TRACE-KG does not impose a fixed schema; mentions are collected before global identity and class decisions are made.

Entity resolution (EntRes). EntRes consolidates mention-level candidates into resolved entities. Mentions are represented using multiple fields, including name, description, type hint, justification evidence, and resolution context, then clustered with HDBSCAN into candidate identity neighborhoods. Within each neighborhood, the LLM selects structured actions over provided identifiers, such as MERGEENTITIES, MODIFYENTITY, or KEEPENTITY. The graph state is updated only after deterministic validators check identifier validity, action format, and operation constraints. EntRes follows Algorithm 1: after each round, mention-to-entity assignments and multi-field representations are refreshed, neighborhoods are recomputed, and the process stops when merge activity plateaus or the iteration budget is reached. This keeps identity edits auditable while allowing local resolution decisions to propagate across later neighborhoods.

4.3. Entity schema induction: classes and class groups

After entity resolution, TRACE-Schema organizes resolved entities into reusable schema-level types without requiring a predefined ontology.

Entity class recognition (EntClsRec). EntClsRec embeds each resolved entity using its name, description, intrinsic properties, broad type hint, and representative supporting evidence. These representations are clustered into semantic neighborhoods, and the LLM proposes candidate entity classes with labels, descriptions, and member entity identifiers. A neighborhood may yield multiple class proposals when it contains heterogeneous entities, avoiding forced assignment to an overly broad class. A fallback pass handles entities not confidently covered by earlier proposals.

Entity class resolution (EntClsRes). EntClsRes refines candidate classes into a validated schema hierarchy. Candidate class representations are embedded and clustered, then resolved through Algorithm 1 using structured actions such as MERGECLASSES, SPLITCLASS, CREATECLASS, REASSIGNENTITIES, and MODIFYCLASS. The output assigns each resolved entity to an entity class and broader class group, yielding $\mathrm{EntClsGroup} \rightarrow \mathrm{EntCls} \rightarrow E$.

4.4. Relation layer: chunk-centric recognition with qualifiers

Relation recognition (RelRec). RelRec extracts grounded relation instances after entities have been resolved and assigned to entity classes. For each chunk c, the extractor receives $\mathrm{text}(c)$ and the resolved entities supported by that chunk, $E(c)$. It may output only directed relations whose endpoints are in $E(c)$ and whose evidence appears in the focus chunk. Each relation instance records subject and object identifiers, a raw predicate label, an instance description, confidence, justification evidence, provenance, and a qualifier dictionary.

Qualifiers capture the local validity context of a relation, including temporal, spatial, operational, conditional, uncertainty, or causal constraints. They are attached as edge-level annotations rather than converted into additional graph nodes or edges. This preserves *when*, *where*, or *under what conditions* a relation holds while keeping the graph topology focused on entity–entity relations.

4.5. Relation canonicalization and relation schema induction

Relation processing follows the same recognition–neighborhood–resolution pattern as the entity layer, but its effect on the graph differs. Entity resolution collapses duplicate mentions into shared nodes. In contrast, relation instances are preserved as grounded evidence because each instance is tied to a specific source context and ordered entity pair. Therefore, relation resolution primarily normalizes predicate semantics and assigns schema annotations rather than deleting edges.

Relation resolution (RelRes). RelRes embeds each relation instance using a multi-field representation combining the raw predicate label, relation description, endpoint names, endpoint schema metadata, qualifiers, and supporting evidence. The resulting neighborhoods are resolved through Algorithm 1 using structured actions such as SETCANONICALREL, SETRELCLS, SETRELCLSGROUP, MODIFYRELSCHEMA, ADDRELREMARK, and MERGERELATIONS. The action MERGERELATIONS is permitted only when relation instances connect the same ordered entity pair and express equivalent semantics; otherwise, instances are retained and only their canonical labels or schema assignments are updated.

Algorithm 1 TRACE-KG constrained resolution pattern

Require: Candidate objects X; representation function ϕ; action set $\mathcal{A}$; budget T
Ensure: Refined objects X' with logged edits

```
for t = 1, ..., T do
    Embed X using φ and cluster into neighborhoods B
    for all B ∈ B do
        Select structured LLM actions from A over IDs in B
        Validate identifiers, action format, and operation constraints
        Execute approved actions deterministically and log edits
    end for
    Refresh representations after applied edits
    if edits plateau then break
    end if
end for
return refined candidates X'
```

After each round, canonical relation labels and schema assignments are refreshed, relation representations are recomputed, and refinement continues until edits plateau or the iteration budget is reached. The output is a set of grounded relation instances R, a canonical relation vocabulary RelCan, and the relation schema hierarchy $\mathrm{RelClsGroup} \rightarrow \mathrm{RelCls} \rightarrow \mathrm{RelCan}$.

4.6. Graph assembly and audit controls

The final stage assembles the context-enriched knowledge graph $G = (E, R, S)$. Resolved entities become nodes annotated with descriptions, intrinsic properties, schema assignments, confidence scores, justification excerpts, and provenance links. Relation instances become directed edges annotated with raw and canonical predicate labels, relation-schema assignments, qualifier dictionaries, confidence scores, justification excerpts, and provenance.

TRACE-KG applies audit controls throughout construction rather than treating LLM outputs as direct graph updates. Entity and relation extraction are chunk-local and must be grounded in focus-chunk evidence. Resolution is restricted to candidate neighborhoods, limiting unconstrained global edits. LLM decisions are expressed as structured actions over existing identifiers, while deterministic validators execute the approved updates and log the resulting edits. These controls do not guarantee correctness, but they make generated nodes, edges, schema assignments, and resolution decisions inspectable against their supporting evidence.

Table 1. Knowledge retention results on MINE-1, grouped by construction setting.

Method	Ret.Acc	EGU↑	RWA↑	SCI↑	Leak↓	TriCR→ 1	Conn.↑	AvgRank↓
Schema-free extraction / post-hoc consolidation								
OpenIE	56.0%	40.5%	41.5%	0.024	2.3%	3.201	74.0%	3.56
KGGen	63.6%	30.1%	30.1%	0.005	**0.0%**	0.494	46.1%	4.00
Retrieval-oriented graph baseline								
GraphRAG	48.4%	45.1%	45.1%	0.146	**0.0%**	0.394	**91.5%**	2.67
Dynamic schema induction baseline								
AutoSchemaKG	**95.1%**	37.3%	58.7%	0.047	36.5%	3.599	61.5%	3.33
Proposed schema-inducing context-enriched construction								
TRACE-KG	90.2%	**78.5%**	**79.6%**	**0.270**	1.3%	**0.956**	88.5%	**1.44**

5. Experiment & Analysis

TRACE-KG is evaluated along two complementary axes: (i) whether the constructed graph supports reliable graph-based retrieval of factual evidence, and (ii) whether the induced schema recovers reusable entity and relation types without access to a predefined ontology. Implementation details appear in the appendix, and an anonymized codebase with full experiment scripts is available at https://anonymous.4open.science/r/TRACE-KG.

5.1. Experiment 1: Knowledge Retention on MINE-1

To assess factual knowledge retention under graph-based retrieval, we adopt the MINE-1 benchmark (Mo et al., 2025). Given a source article and a set of derived factual statements, the task is to determine whether each statement can be supported by a retrieved subgraph of the constructed graph. We compare TRACE-KG against four baselines representing different construction settings: schema-free extraction or post-hoc consolidation (OpenIE (Angeli et al., 2015), KGGen (Mo et al., 2025)), a retrieval-oriented graph baseline (GraphRAG (Larson & Truitt, 2024)), and dynamic schema induction (AutoSchemaKG (Bai et al., 2025)). All methods are evaluated on identical benchmark instances.

Protocol (retrieval and strict judging). We follow the retrieval-and-judge procedure of MINE-1 (Mo et al., 2025). Each factual statement is embedded and matched to KG entities; the top-k entities are retrieved, expanded within a fixed hop budget to form an induced subgraph, and evaluated by a strict LLM judge. The judge determines whether the statement is supported using *only* the retrieved subgraph (binary decision; no external knowledge; no inferred edges). To ensure comparability, we use a fixed judge configuration across all methods, along with identical retrieval hyperparameters and embedding models.

Beyond raw retrieval accuracy. The standard MINE-1 score (Ret.Acc) measures whether facts can be supported from retrieved subgraphs. However, high Ret.Acc can arise from undesirable artifacts, such as copying long text spans into entity strings or retrieving fragments from globally disconnected graphs. To better capture graph quality, we complement Ret.Acc with structural and representational metrics reflecting connectivity, compression, and information leakage. We report three composite metrics: **RWA** (Reachability-Weighted Accuracy), **EGU** (Effective Graph Utilization), and **SCI** (Structural Coherence Index):

$$\mathbf{RWA} = \text{Ret.Acc} \times \text{Conn.} \tag{1}$$

$$\mathbf{EGU} = \text{Ret.Acc} \times \text{Conn.} \times (1 - \text{Leak\%}) \tag{2}$$

$$\mathbf{SCI} = \text{AvgDeg} \times \text{Clust.} \times \text{Conn.} \tag{3}$$

RWA adjusts retrieval accuracy by graph connectivity; EGU further penalizes lexical leakage; and SCI measures structural quality independently of retrieval accuracy (details in Appendix C.0.1).

Results. Table 1 shows that Ret.Acc alone yields an incomplete ranking. Although AutoSchemaKG achieves the highest Ret.Acc (95.1%), it also exhibits high leakage (36.5%) and a large TriCR (3.599), indicating that much of the retained evidence remains close to the source text rather than compactly encoded in the graph. TRACE-KG, by contrast, maintains high Ret.Acc (90.2%) while achieving low leakage (1.3%), a TriCR close to 1 (0.956), and strong connectivity (88.5%), resulting in the best EGU, RWA, SCI, and AvgRank. These results show that TRACE-KG balances retrieval accuracy with structural coherence, rather than relying on lexical overlap. See Appendix C.0.1 for additional diagnostics and baseline analysis. Figure 2 visualizes the effects of discounting leakage and structural fragmentation.

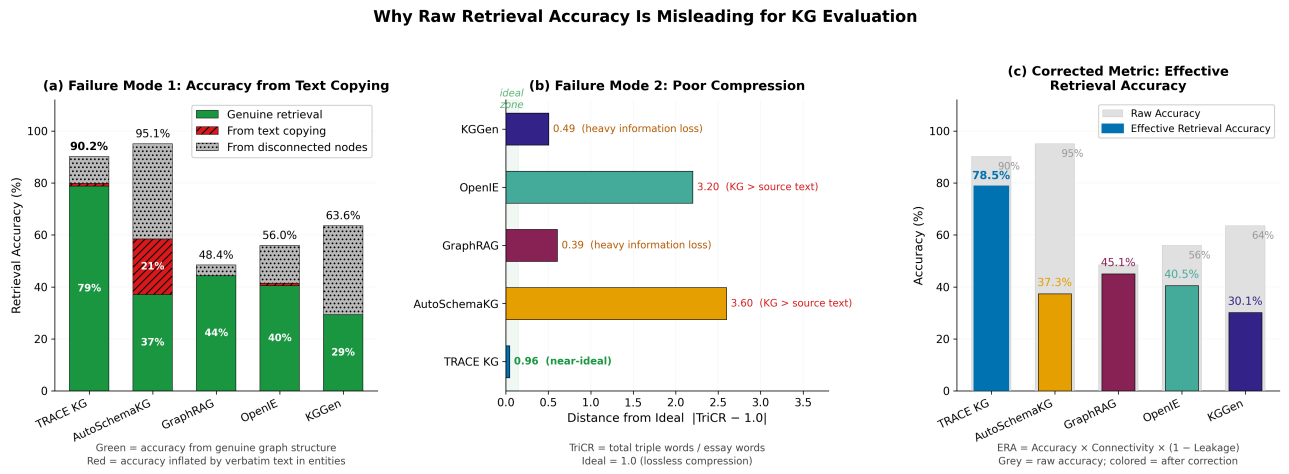

Figure 2. Failure modes behind raw retrieval accuracy on MINE-1. Left: Ret.Acc retained after discounting by connectivity and leakage. Middle: deviation of TriCR from the ideal value 1. Right: Ret.Acc versus EGU.

5.2. Experiment 2: Ontology-Held-Out Schema Evaluation

Goal and setting. We evaluate TRACE-KG's ability to induce a reusable schema directly from text using a novel *ontology-held-out* protocol. TRACE-KG operates solely on the input corpus and induces a hierarchical schema of entity and relation types as part of extraction and resolution, without access to any predefined ontology. A curated reference ontology is introduced only after schema induction for post hoc mapping and evaluation, ensuring that the induced schema is assessed against an independent, human-designed conceptual model (details in Appendix C.0.2).

Why this dataset. This setting requires (i) human-curated ontologies as held-out ground truth and (ii) domain-specific text from which schemas must be induced. We therefore use the DBpedia-WebNLG domain ontologies and sentence splits distributed by Text2KGBench and OSKGC (Mihindukulasooriya et al., 2023; Wang & Iwaihara, 2025). The dataset spans 19 domains with diverse entity and relation vocabularies, enabling cross-domain evaluation of schema induction. Inputs consist of *disjoint sentences* rather than full documents; thus, document-level structural cues (e.g., titles, sections) are absent, and schema must be inferred solely from local linguistic evidence.

Protocol. For each domain, we run TRACE-KG on the dataset-provided `train` sentences to induce a fixed TRACE schema. The induced schema contains hierarchical structures for both entities and relations: entity classes and broader entity class groups, and canonical relation labels, relation classes, and broader relation class groups. The reference ontology is then used only for post hoc alignment and evaluation. We report results under three *evaluation scopes* defined by gold triples: **Source** (`train` gold triples), **Held-out** (`test` gold triples), and **Combined** (all gold triples). Alignment judgements are produced by a structured LLM verifier, with targeted human auditing of low-confidence cases.

Schema mapping and verification. To compare the induced TRACE schema with the held-out reference ontology, we retrieve the top-K TRACE candidates for each active reference anchor and verify each pair using a structured verifier. Judgements are EQUIVALENT, NARROWER, BROADER, or UNRELATED; EQUIVALENT counts as exact recovery and NARROWER as compatible refinement. Relation alignment is direction-relaxed, low-confidence outputs are manually audited, and all candidates and judgements are retained for inspection.

Active anchors and metrics. For each evaluation scope (Source, Held-out, Combined), we evaluate only *active* reference anchors, defined as ontology relations appearing in the gold triples together with their domain and range concepts. Anchors are frequency-weighted according to their occurrence in the corresponding split. We report coverage (Exact and Compatible), frequency-weighted MRR@5, and domain/range consistency (D/R).

Results and discussion. Table 2 reports TRACE-KG's ontology-held-out schema evaluation, macro-averaged over 19 domains. Concept alignment is strong across all scopes, with near-saturated compatible coverage (97–99%). Lower exact recovery ($\approx 50\%$) reflects frequent NARROWER refinements rather than failure to recover the underlying concept. Relation alignment is more challenging, especially in the Held-out scope, but still reaches 84.8% compatible coverage. Retrieval quality remains high overall, with Combined MRR@5 of 0.883 for concepts and 0.795 for relations. Aligned relations also show strong domain/range consis-

Table 2. Ontology-held-out schema evaluation on DBpedia-WebNLG (19 domains).

Metric	Source	Held-out	Combined
Active reference anchors (avg. per ontology)			
Concepts	11.2	12.8	14.3
Relations	28.4	31.1	42.7
Concept alignment			
Coverage (Compat)	99.1%	97.2%	98.3%
Coverage (Exact)	50.2%	51.2%	50.2%
Coverage (Narrower)	48.9%	46.0%	48.1%
MRR@5	0.889	0.877	0.883
Relation alignment			
Coverage (Compat)	92.4%	84.8%	89.9%
Coverage (Exact)	61.8%	49.7%	56.8%
Coverage (Narrower)	30.6%	35.1%	33.1%
MRR@5	0.885	0.694	0.795
Domain/range consistency			
D/R (L1–L3)	91.0%	82.7%	88.1%

tency, reaching 82.7% under Held-out weighting. Overall, these results indicate that TRACE-KG induces schema elements directly from text that remain broadly compatible with a reference ontology.

6. Conclusion

We introduced TRACE-KG, a framework for constructing context-enriched and fully traceable knowledge graphs from multimodal technical documents while simultaneously inducing a reusable schema over both entities and relations without relying on a predefined ontology. By combining high-recall extraction, clustering-based semantic neighborhood formation, and constrained LLM-guided resolution with deterministic execution, TRACE-KG enables schema structure to emerge during graph construction rather than being fixed in advance or omitted entirely. Across complementary evaluations, TRACE-KG produces graphs that are structurally coherent, traceable, and robust to common failure modes such as lexical leakage and structural fragmentation, while consistently outperforming existing approaches on composite graph-quality metrics. These results suggest that integrating schema induction directly into KG construction provides a practical and scalable foundation for retrieval, and auditing over multimodal technical document collections.

7. Limitations and Future Work

The quality of schema induction and resolution depends on the stability of semantic neighborhoods formed through embedding and clustering. While iterative refinement mitigates early noise, errors in neighborhood formation can affect downstream canonicalization, particularly for low-frequency or semantically ambiguous entities and relations. Future work can explore more robust neighborhood formation, stability-aware refinement, and alternative clustering strategies.

Although TRACE-KG enforces structured decision-making through constrained action interfaces and deterministic execution, the selection of actions remains dependent on LLM outputs. As a result, performance is influenced by the capabilities and consistency of the underlying model, especially in complex or ambiguous cases. Future work includes developing more model-agnostic decision strategies, improving validation and repair mechanisms, and exploring efficient model cascades.

TRACE-KG is designed to be domain-agnostic, but schema quality can benefit from domain-specific linguistic patterns and terminology. Incorporating domain-adaptive or fine-tuned language models, as well as leveraging domain-specific corpora during schema induction, is a promising direction for improving performance in specialized settings.

References

Abolhasani, M. S. and Pan, R. Leveraging llm for automated ontology extraction and knowledge graph generation. *arXiv preprint arXiv:2412.00608*, 2024.

Ameri, F., Abolhasani, M. S., and Li, Y. From tabular maintenance work orders to ontology-guided knowledge graph via llm: Prioritizing content over structure. *Available at SSRN 6017381*, 2026.

Angeli, G., Premkumar, M. J. J., and Manning, C. D. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 344–354, 2015.

Bai, J., Fan, W., Hu, Q., Zong, Q., Li, C., Tsang, H. T., Luo, H., Yim, Y., Huang, H., Zhou, X., et al. Autoschemakg: Autonomous knowledge graph construction through dynamic schema induction from web-scale corpora. *arXiv preprint arXiv:2505.23628*, 2025.

Bian, H. Llm-empowered knowledge graph construction: A survey. *arXiv preprint arXiv:2510.20345*, 2025.

Chen, S., Zhang, Q., Dong, J., Hua, W., Li, Q., and Huang, X. Entity alignment with noisy annotations from large language models. *Advances in Neural Information Processing Systems*, 37:15097–15120, 2024.

Christmann, P., Saha Roy, R., and Weikum, G. Explainable conversational question answering over heterogeneous

sources via iterative graph neural networks. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pp. 643–653, 2023.

Ding, Y., Poudel, A., Zeng, Q., Weninger, T., Veeramani, B., and Bhattacharya, S. Entgpt: Entity linking with generative large language models. *arXiv preprint arXiv:2402.06738*, 2024.

Esmaeili, N. and Bamdad Soofi, J. Expounding the knowledge conversion processes within the occupational safety and health management system (osh-ms) using concept mapping. *International Journal of Occupational Safety and Ergonomics*, 28(2):1000–1015, 2022.

Euzenat, J., Ehrig, M., and Castro, R. Towards a methodology for evaluating alignment and matching algorithms. *Technical Report, Ontology Alignment Evaluation Initiative (OAEI)*, 2005.

Hofer, M., Obraczka, D., Saeedi, A., Köpcke, H., and Rahm, E. Construction of knowledge graphs: Current state and challenges. *Information*, 15(8):509, 2024.

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G. D., Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., et al. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37, 2021.

Jiang, T., Zhao, T., Qin, B., Liu, T., Chawla, N. V., and Jiang, M. The role of" condition" a novel scientific knowledge graph representation and construction model. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1634–1642, 2019.

Khorashadizadeh, H., Amara, F. Z., Ezzabady, M., Ieng, F., Tiwari, S., Mihindukulasooriya, N., Groppe, J., Sahri, S., Benamara, F., and Groppe, S. Research trends for the interplay between large language models and knowledge graphs. *arXiv preprint arXiv:2406.08223*, 2024.

Larson, J. and Truitt, S. Graphrag: Unlocking llm discovery on narrative private data. *Microsoft Research Blog*, 2024.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

Lin, C.-Y. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Lu, Y. and Wang, J. Karma: Leveraging multi-agent llms for automated knowledge graph enrichment. *arXiv preprint arXiv:2502.06472*, 2025.

Ma, Y., Zang, Y., Chen, L., Chen, M., Jiao, Y., Li, X., Lu, X., Liu, Z., Ma, Y., Dong, X., et al. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. *Advances in Neural Information Processing Systems*, 37:95963–96010, 2024.

Mihindukulasooriya, N., Tiwari, S., Enguix, C. F., and Lata, K. Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text. In *International semantic web conference*, pp. 247–265. Springer, 2023.

Mo, B., Yu, K., Kazdan, J., Cabezas, J., Mpala, P., Yu, L., Cundy, C., Kanatsoulis, C., and Koyejo, S. Kggen: Extracting knowledge graphs from plain text with language models. *arXiv preprint arXiv:2502.09956*, 2025.

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., and Mian, A. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 16 (5):1–72, 2025.

Newman, M. E. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

Newman, M. E. J. *Networks: An Introduction*. Oxford University Press, 2010. doi: 10.1093/ACPROF:OSO/9780199206650.001.0001.

Pons, G., Bilalli, B., and Queralt, A. Knowledge graphs for enhancing large language models in entity disambiguation. In *International Semantic Web Conference*, pp. 162–179. Springer, 2024.

Qin, Y., Yang, H., Cui, L., Zhang, Y., Feng, G., Qiao, Y., and Yao, Y. Semantic-aware fusion of mineral exploration knowledge streams towards dynamic geological knowledge graphs. *Minerals*, 15(12):1257, 2025.

Shannon, C. E. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Sun, J., Qian, S., Han, Z., Li, W., Qian, Z., Yang, D., Cao, J., and Xue, G. Lkd-kgc: Domain-specific kg construction via llm-driven knowledge dependency parsing. *arXiv preprint arXiv:2505.24163*, 2025.

Tahsin, R., Li, Y., Abolhasani, M. S., and Ameri, F. Generation of semantic knowledge graphs from maintenance work orders data. *J. Maintenance Engineering*, 11(2): 45–60, 2024.

Van Rijsbergen, C. J. Information retrieval. 2nd. newton, ma, 1979.

Wang, D. and Iwaihara, M. Oskgc: A benchmark for ontology schema-based knowledge graph construction from text. In *CEUR Workshop Proceedings*, volume 4041. CEUR-WS, 2025.

Wang, F., Tao, Z., Wang, M., Hu, M., and Bai, X. AELC: Adaptive entity linking with LLM-driven contextualization. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 4313–4327, Suzhou, China, November 2025a. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.231. URL https://aclanthology.org/2025.findings-emnlp.231/.

Wang, T., Chen, X., Lin, H., Chen, X., Han, X., Sun, L., Wang, H., and Zeng, Z. Match, compare, or select? an investigation of large language models for entity matching. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 96–109, 2025b.

Watts, D. J. and Strogatz, S. H. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

Xu, C., Li, M., Yang, C., Jiang, X., Tang, L., Qi, Y., and Guo, J. Context graph. *arXiv preprint arXiv:2406.11160*, 2024.

Xue, B. and Zou, L. Knowledge graph quality management: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4969–4988, 2022.

Zaratiana, U., Pasternak, G., Boyd, O., Hurn-Maloney, G., and Lewis, A. Gliner2: Schema-driven multi-task learning for structured information extraction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 130–140, 2025.

Zhong, A., Mo, D., Liu, G., Liu, J., Lu, Q., Zhou, Q., Wu, J., Li, Q., and Wen, Q. Logparser-llm: Advancing efficient log parsing with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4559–4570, 2024.

Zhong, L., Wu, J., Li, Q., Peng, H., and Wu, X. A comprehensive survey on automatic knowledge graph construction. *ACM Computing Surveys*, 56(4):1–62, 2023.

A. Extended Related Work

Ontology-driven construction and ontology population. A prominent direction treats the schema as an explicit *input*, framing text-to-KG construction as ontology population. TEXT2KGBENCH evaluates this setting by measuring whether generated triples remain faithful to text while conforming to ontology constraints, showing that errors persist even when the target schema is fixed (Mihindukulasooriya et al., 2023). In industrial and maintenance settings, Ameri et al. operationalize ontology-guided construction using a curated thesaurus and work-order ontology to map noisy descriptions into RDF representations aligned with domain concepts (Ameri et al., 2026). OntoKGen similarly employs LLMs in an interactive workflow to extract and refine an ontology from technical documents before generating a schema-consistent KG (Abolhasani & Pan, 2024). While these approaches improve normalization through schema alignment, they remain sensitive to ontology design, costly to maintain over time, and brittle under evolving terminology and corpus-specific granularity (Hofer et al., 2024).

Schema-free extraction and post-hoc consolidation. A complementary direction performs schema-free extraction, producing entities and relations directly from text. Such approaches often yield fragmented vocabularies at corpus scale: semantically similar relations appear under many surface predicates, and entities are duplicated across abbreviations and near-synonyms. KGGen highlights this issue, showing that naive extraction can produce relation vocabularies nearly as large as the edge set, and addresses it through embedding-based clustering with LLM-guided consolidation of entities and relations (Mo et al., 2025). While this improves representational reuse, it does not induce *schema connectivity*: in schema-free graphs, objects are related primarily through extracted edges, whereas type- or class-level structure could connect entities (and relations) via shared membership even in the absence of direct edges.

Inducing schema from text. To move beyond post-hoc label normalization, recent approaches treat schema as an *output* of construction. AutoSchemaKG induces abstract concepts for entities and relations and uses these assignments as a learned schema layer (Bai et al., 2025). This enables data-driven organization without a predefined ontology, but the resulting schema elements can remain verbose and heterogeneous, and their coupling with robust corpus-level identity resolution remains limited. LKD-KGC targets domain-specific repositories by modeling cross-document dependencies: it orders documents, incorporates retrieved context during summarization, and induces an entity schema (type names and definitions) to guide extraction (Sun et al., 2025). While this leverages repository structure, the resulting organization is often uneven across entities and relations, and extracted predicates and nodes remain close to natural-language expressions, limiting stable and reusable schema formation across long corpora.

Entity resolution, entity matching, and schema-aware IE. Mention-level ambiguity remains a central bottleneck in KG construction. EntGPT frames entity linking as a structured generative workflow based on explicit candidate sets, showing that grounding decisions can materially improve downstream reasoning (Ding et al., 2024). Wang et al. compare LLM prompting strategies for entity matching and show that structured comparison and selection outperform naive generation, highlighting the importance of controlling the decision process (Wang et al., 2025b). KG-assisted entity disambiguation similarly leverages existing graph structure to constrain candidate spaces and provide structured evidence (Pons et al., 2024). At the extraction layer, schema-driven IE models such as GLiNER2 enable flexible extraction under user-specified label sets, demonstrating the practicality of schema-conditioned structured extraction (Zaratiana et al., 2025). While these approaches improve grounding and extraction, they typically rely on external reference structures (KBs or predefined schemas) and are evaluated in isolation, rather than in end-to-end, corpus-scale settings where schema must be induced and entities and relations must be jointly canonicalized.

Context and conditions in knowledge representations. In scientific and technical domains, unconditional triples are often insufficient, as facts may hold only under specific conditions. Jiang et al. explicitly model *conditions* to mitigate contradictions when aggregating scientific statements across incompatible contexts (Jiang et al., 2019). Context Graph extends this perspective by attaching structured contextual attributes (e.g., provenance, time, location, quantitative qualifiers) to relations and shows that treating context as first-class improves downstream reasoning (Xu et al., 2024). In applied domains such as mineral exploration, dynamic KGs similarly emphasize fusing heterogeneous evidence streams where interpretation depends on local contextual cues (Qin et al., 2025). Despite these advances, many construction pipelines still default to flat triples or weak qualifier structures, limiting reasoning about *when* and *under what assumptions* a relation holds.

Reliability, verification, provenance, and multimodal evidence. LLM-based extraction is vulnerable to hallucinations and structural inconsistencies, motivating constrained workflows and explicit verification stages. KARMA exemplifies this direction by coordinating specialized LLM agents for extraction, schema alignment, and conflict resolution when extending an existing KG (Lu & Wang, 2025). Benchmarks such as TEXT2KGBENCH expose faithfulness and conformance failures even under schema constraints (Mi-

hindukulasooriya et al., 2023), while KGGen emphasizes retention and reuse under consolidation (Mo et al., 2025). However, many pipelines do not preserve fine-grained provenance as a first-class artifact—i.e., explicit links from each node or edge to supporting spans and structured elements such as tables, figures, and equations—and explicit confidence signals for prioritized, ad hoc review remain poorly standardized. These limitations are particularly pronounced in technical corpora, where critical information is distributed across narrative text and semi-structured or diagrammatic content.

Collectively, prior work advances key components of text-to-KG construction—ontology alignment, vocabulary consolidation, schema induction, entity matching, contextual modeling, and reliability-oriented orchestration. However, these components are largely developed in isolation and under incompatible assumptions about schema commitment, grounding, and evidence structure. The central gap is the open-world setting in which the schema itself must be induced from the corpus while entity identity, relation normalization, conditional context, and provenance remain jointly consistent across the entire construction process. TRACE-KG addresses this setting by integrating schema induction, resolution, contextualization, and traceability within a unified pipeline.

B. Extended Methodology Details

This appendix provides implementation-level details supporting reproducibility and auditing that are omitted from the main paper, including multi-field representation construction, clustering and subclustering strategies, constrained action interfaces ("function calling"), and iteration and stopping criteria. It also describes qualifier-aware relation merging and the traceability artifacts produced by the pipeline.

B.1. Intermediate artifacts and traceability

TRACE-KG persists intermediate JSONL artifacts at each stage to preserve traceability and enable inspection: chunk records (id, text, provenance); mention-level entity extractions; resolved entities with aggregated evidence and intrinsic properties; candidate and resolved entity classes with class groups; raw relation instances with qualifiers and evidence; and resolved relations with canonical labels and schema annotations. Each resolution stage additionally stores the exact prompt provided to the LLM, the raw LLM output for each processed cluster, and a structured action log.

B.2. Multi-field representations and embedding

Each resolution layer embeds *multi-field representations* rather than single strings. Let $\phi(\cdot)$ denote the embedding function introduced in the preliminaries. Given a record with fields $\mathbf{f} = (f_1, \ldots, f_k)$, we compute per-field embeddings $\phi(f_i)$ and combine them via a weighted sum:

$$\Phi(\mathbf{f}) = \mathrm{norm}\left(\sum_{i=1}^{k} w_i \, \phi(f_i)\right),$$

where w_i are layer-specific hyperparameters.

Entity representations (EntRes). Fields include mention name, description, type hint, intrinsic-property strings, and short evidence or context snippets.

Entity class representations (EntClsRec/EntClsRes). Fields include class label, class description, evidence summaries, and member-entity summaries.

Relation representations (RelRes). Fields include raw relation label, relation description, endpoint context (subject and object names together with entity-schema metadata), coarse relation-type hints, and qualifiers.

B.3. Clustering and subclustering

TRACE-KG uses density-based clustering (HDBSCAN in our implementation) over representation vectors to form semantic neighborhoods. To keep LLM calls tractable, we apply two controls: (i) **local subclustering** for oversized clusters, where only members of a large cluster are reclustered; and (ii) **bounded prompt batching**, which processes at most K items per LLM call by splitting a cluster into batches of size K.

Practical note. Clustering output is treated as *suggestive*, not authoritative: the LLM may decide to merge none of the candidates, and multi-run reclustering can correct imperfect neighborhoods.

B.4. Constrained action interfaces ("function calling")

In each resolution stage, the LLM outputs *only* a JSON array of action objects selected from a fixed vocabulary. Deterministic code validates identifiers, executes edits, and logs every applied or rejected action.

B.4.1. Entity resolution actions (EntRes)

- MergeEntities: merge multiple entity mentions into one canonical entity.
- ModifyEntity: revise the name, description, or type hint to prevent incorrect merges.

- KEEPENTITY (optional): explicitly keep a candidate unchanged.

Illustrative action schemas.

```
{ "action": "MergeEntities",
  "entity_ids": ["En_...","En_..."],
  "canonical_name": "...",
  "canonical_description": "...",
  "canonical_type": "...",
  "rationale": "..." }

{ "action": "ModifyEntity",
  "entity_id": "En_...",
  "new_name": "... or null",
  "new_description": "... or null",
  "new_type_hint": "... or null",
  "rationale": "..." }

{ "action": "KeepEntity",
  "entity_id": "En_...",
  "rationale": "..." }
```

B.4.2. ENTITY CLASS RESOLUTION ACTIONS (ENTCLSRES)

- MERGE_CLASSES: merge redundant or synonymous classes.
- SPLIT_CLASS: split overloaded classes into coherent subclasses.
- CREATE_CLASS: create a missing class for a coherent subset of entities.
- REASSIGN_ENTITIES: move entities between classes.
- MODIFY_CLASS: revise class metadata, including class-group assignment.

Each action must include a one-line justification; optional confidence scores and remarks may also be provided. Newly created or merged classes may be referenced within the same output via provisional identifiers, which are resolved deterministically by the pipeline.

B.4.3. RELATION RESOLUTION ACTIONS (RELRES)

- SET_CANONICAL_REL: assign a canonical predicate label and description.
- SET_REL_CLS: assign a relation class.
- SET_REL_CLS_GROUP: assign a relation class group.
- MODIFY_REL_SCHEMA: revise canonical and schema fields jointly.
- ADD_REL_REMARK: attach explanatory remarks without changing schema.
- MERGE_RELATIONS: merge duplicate relation instances between identical endpoints (including direction-normalized cases), only when semantics are equivalent.

Macro relation groups. During RelRec, each relation instance includes a required single-token coarse group hint (`rel_hint_type`) selected from: {`IDENTITY`, `COMPOSITION`, `CAUSALITY`, `TEMPORALITY`, `SPATIALITY`, `ROLE`, `PURPOSE`, `DEPENDENCY`, `COUPLING`, `TRANSFORMATION`, `COMPARISON`, `INFORMATION`, `ASSOCIATION`}. RelRes may revise this hint by assigning `rel_cls_group`.

B.5. Qualifier extraction and normalization

RelRec emits a qualifier dictionary with exactly eight fields:

```
{ "TemporalQualifier": ...,
  "SpatialQualifier": ...,
  "OperationalConstraint": ...,
  "ConditionExpression": ...,
  "UncertaintyQualifier": ...,
  "CausalHint": ...,
  "LogicalMarker": ...,
  "OtherQualifier": ... }
```

Missing qualifiers are normalized to JSON `null` for consistency across relation instances.

B.6. Safe duplicate relation merging and conflict handling

When two relation instances are candidates for merging—i.e., they connect the same endpoints (after direction normalization) and express equivalent canonical predicates—TRACE-KG merges provenance and evidence while reconciling qualifiers conservatively.

Non-conflicting qualifiers. If one qualifier dictionary is a subset of the other, the merged relation retains the superset.

Conflicting qualifiers. If overlapping qualifier keys have different values (e.g., distinct temporal windows), both relation instances are retained, with an explanatory remark indicating the conflict.

Direction normalization. If duplicates are expressed in opposite directions, RelRes may normalize them to a single direction by swapping endpoints, while preserving evidence spans.

B.7. Iteration and stopping criteria

TRACE-KG applies iterative refinement in three stages.

EntRes iteration. Repeat: cluster → resolve → collapse until merges fall below a threshold or a maximum number of rounds is reached.

EntClsRes multi-run refinement. Repeat: cluster candidate classes → apply structural actions until structural edits plateau over a patience window or a maximum number of runs is reached.

RelRes multi-run refinement. Repeat: cluster relation instances → apply schema actions until schema edits plateau over a patience window or a maximum number of runs is reached.

C. Detailed Experiment Descriptions

C.0.1. Extended Experiment 1 Details (MINE-1)

Metric definitions. Average degree (AvgDeg) is $|A|/|V|$, where A denotes the exported edge set. In addition to the standard MINE-1 retrieval accuracy (Ret.Acc), we evaluate structural and representational properties of the generated graphs.

Connectivity (Conn.) is the fraction of nodes in the largest weakly connected component (Newman, 2003). **Average degree (AvgDeg)** is $|E|/|V|$ (Newman, 2010). **Clustering coefficient (Clust.)** measures average local clustering (Watts & Strogatz, 1998; Newman, 2010). **Leakage (Leak%)** quantifies lexical overlap between source text and entity strings using 4-gram overlap (Lin, 2004). **Triple compression ratio (TriCR)** is the ratio of total triple word count to source word count; values near 1 indicate balanced compression, while deviations indicate information loss or redundancy (Shannon, 1948).

Composite metrics. RWA (Reachability-Weighted Accuracy) adjusts retrieval accuracy by connectivity, reflecting that disconnected components cannot support multi-hop retrieval. EGU (Effective Graph Utilization) further discounts retrieval by leakage, reducing credit for verbatim copying (Van Rijsbergen, 1979). SCI (Structural Coherence Index) measures structural quality independently of retrieval accuracy by combining relational density, clustering, and connectivity.

Average rank (AvgRank) measures the average rank position of the first retrieved supporting evidence for each query; lower values indicate more efficient retrieval.

Structural diagnostics. Table 3 complements Table 1 by exposing graph properties underlying retrieval performance: graph size ($|V|$, $|E|$), entity granularity (AvgEW), relational density (AvgDeg), global connectivity (Conn.), and local cohesion (Clust.). While these are descriptive rather than direct objectives, they explain why methods with similar Ret.Acc diverge after accounting for leakage and reachability.

TRACE-KG combines compact entities (AvgEW = 2.4) with the highest average degree and clustering, indicating dense and locally coherent neighborhoods. This aligns with its strong SCI and EGU scores in the main table. GraphRAG achieves the highest connectivity, but on much smaller graphs, suggesting connectivity driven by compression rather than broad relational coverage. AutoSchemaKG produces substantially longer entities (AvgEW = 6.6), consistent with its high leakage. OpenIE yields the largest graphs ($|V|$, $|E|$), but low clustering indicates weak local structure despite expansion. KGGen produces smaller, sparse graphs, explaining its low connectivity and reachability-adjusted performance.

Baseline behavior and trade-offs. AutoSchemaKG achieves the highest Ret.Acc but relies on lexical retention, reflected in high leakage and long entity strings. TRACE-KG instead maintains compact entities and strong structural properties, achieving the highest AvgDeg and clustering.

GraphRAG achieves high connectivity but on small graphs ($|V| = 11$, $|E| = 11$), limiting coverage and reducing Ret.Acc. OpenIE expands the source heavily (TriCR = 3.201) with weak structure, whereas KGGen compresses aggressively (TriCR = 0.494) but produces fragmented graphs (Conn. = 46.1%). These patterns highlight trade-offs between lexical retention, compression, and structural coherence.

Supplementary visualizations. Figure 3 provides a normalized multi-metric view. It is diagnostic rather than primary and should be interpreted alongside Table 1. The radar view shows that TRACE-KG remains consistently strong across retrieval accuracy, effective retrieval after discounting, connectivity, clustering, and compression quality, rather than excelling on a single axis.

Figure 4 visualizes EGU variability across benchmark instances. Its purpose is to show that the observed performance gap is stable and not driven by outliers; the ordering in Table 1 remains consistent across instances.

C.0.2. Extended Experiment 2 Details

Dataset and evaluation setup. We use the DBpedia-WebNLG collection distributed in Text2KGBench/OSKGC (Mihindukulasooriya et al., 2023; Wang & Iwaihara, 2025), which provides 19 domain ontologies paired with sentence-level gold triples (4,860 sentences in total). Table 4 summa-

Table 3. Additional structural diagnostics complementary to Table 1. $|V|$ and $|A|$ denote average node and edge counts; AvgEW is average words per entity. AvgDeg is $|E|/|V|$ (Newman, 2010); Conn. is largest weakly connected component fraction (Newman, 2003); Clust. is average clustering coefficient (Watts & Strogatz, 1998; Newman, 2010). Higher is better only for AvgDeg, Conn., and Clust.

Method	$\|V\|$	$\|E\|$	AvgEW	AvgDeg↑	Conn.↑	Clust.↑
TRACE-KG	63	90	2.4	**1.35**	88.5%	**0.199**
GraphRAG	11	11	1.6	0.98	**91.5%**	0.150
AutoSchemaKG	107	105	6.6	0.98	61.5%	0.066
OpenIE	170	253	3.2	1.26	74.0%	0.027
KGGen	84	65	1.2	0.75	46.1%	0.011

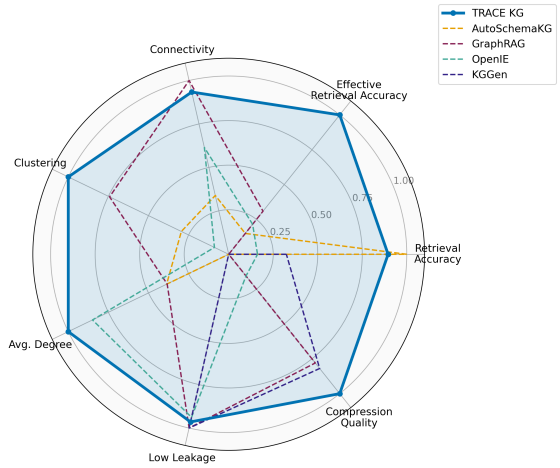

Figure 3. Normalized multi-metric profile on MINE-1. Leak is inverted, and compression is expressed as proximity to TriCR = 1 so that higher values are preferable on all axes.

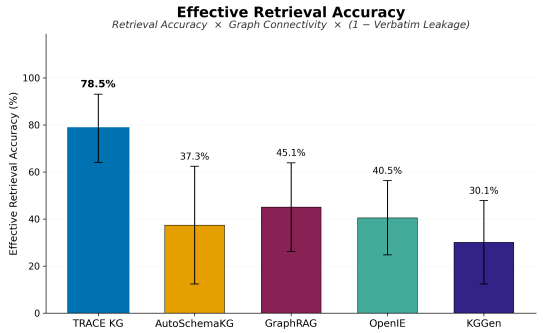

Figure 4. Effective Graph Utilization (EGU) on MINE-1. Error bars indicate variability across benchmark instances.

rizes per-domain statistics.

For each domain, TRACE-KG is run once on the `train` sentences to induce a fixed TRACE schema. Schema-to-ontology alignment judgements are then computed once per domain and reused across evaluation scopes. The three scopes in the main paper (**Source**, **Held-out**, **Combined**) differ only in which gold triples activate reference anchors and in their frequency weights; the induced TRACE schema and alignment mapping remain fixed.

Schema mapping. Schema mapping is performed at the schema level after induction. Each reference ontology anchor is aligned against the induced TRACE schema rather than sentence-level extractions. Because TRACE induces hierarchical schema structures, valid correspondences may occur at multiple abstraction levels. We therefore preserve the hierarchy during evaluation instead of flattening the schema into a single label space.

Candidate retrieval. For each reference anchor, we retrieve a small candidate set from the induced TRACE schema prior to semantic verification. Retrieval uses cosine similarity over weighted multi-evidence embeddings. Entity-side evidence includes schema labels across hierarchy levels and representative instance cues. Relation-side evidence includes canonical labels, higher-level relation classes, lexical variants, and representative subject–object examples with induced type signatures. Reference-side evidence includes ontology labels, domain/range constraints, and examples from `train` triples.

We retrieve $K{=}5$ candidates per anchor and apply a controlled assignment step to limit fan-out and reduce spurious many-to-many correspondences. For reproducibility, we persist the induced schema, candidate lists with similarity scores, LLM prompts and outputs, parsed judgements, and per-domain summaries.

LLM-based alignment and audit. Lexical similarity alone is insufficient for schema alignment under synonymy and granularity variation. We therefore use an LLM as a structured verifier for all retrieved (reference anchor, TRACE candidate) pairs. Each candidate is presented together with its hierarchical context, allowing alignment at the most appropriate abstraction level.

The judge assigns one of EQUIVALENT, NARROWER, BROADER, or UNRELATED, along with a confidence score. EQUIVALENT is treated as exact recovery and NARROWER as compatible refinement; BROADER and UNRELATED do not contribute to coverage. For relations, alignment is evaluated in a direction-relaxed manner by considering both orientations and retaining the better match.

Table 4. DBpedia-WebNLG statistics for Experiment 2.

#	Ontology	#Types	#Relations	#Sentences
1	University	15	46	156
2	Music	15	35	290
3	Airport	14	39	306
4	Building	14	38	275
5	Athlete	17	37	293
6	Politician	19	40	319
7	Company	10	28	153
8	Celestial	8	27	194
9	Astronaut	16	38	154
10	Comics	10	18	102
11	Transport	20	68	314
12	Monument	14	26	92
13	Food	12	24	398
14	Written Work	10	44	322
15	Sports Team	14	24	235
16	City	11	23	348
17	Artist	20	39	386
18	Scientist	15	47	259
19	Film	18	44	264
			Total Sentences	**4,860**

Table 5. Best-matching TRACE level among accepted alignments (macro-averaged, Combined).

Level	Relations	Concepts
Finest level	64.2%	50.8%
Middle level	28.7%	35.4%
Coarsest level	7.1%	13.8%

The verifier is run deterministically (temperature 0). Low-confidence cases (confidence < 0.88) undergo targeted human audit to ensure that aggregate trends are not artifacts of unstable judgements.

Active anchors and metrics. Metrics are computed under three scopes: **Source**, **Held-out**, and **Combined**. In each scope, only *active* reference anchors are evaluated: ontology relations appearing in the scope's gold triples together with their domain and range concepts. Anchors are frequency-weighted according to scope-specific gold triple counts (Euzenat et al., 2005).

We report **Coverage** (Exact and Compatible), frequency-weighted **MRR@5**, and **Domain/Range Consistency** (D/R). The D/R check is direction-relaxed and hierarchy-aware, allowing backoff to coarser TRACE types when fine-grained typing is sparse. Primitive datatypes (e.g., `xsd:date`, `xsd:string`) are excluded, as they cannot be induced as text-derived classes.

Granularity and alignment behavior. Because TRACE induces a hierarchy rather than a flat label space, successful mappings may occur at different abstraction levels. Table 5 shows that most relation alignments occur at the finest level, while concept alignments are more evenly distributed across levels. This supports the interpretation in the main paper that many successful matches arise through compatible refinement rather than exact equivalence.

Interpretation of alignment results. The Exact–Compatible gap in Table 2 primarily reflects granularity mismatch rather than alignment failure. TRACE often induces more specific schema elements than the reference ontology, leading to NARROWER matches. Thus, compatible coverage is a more appropriate indicator of successful schema recovery than exact equivalence alone.

A second pattern is that retrieval is not the primary bottleneck: MRR@5 remains high, indicating that compatible candidates are typically retrieved near the top. The larger drop in relation performance under the Held-out setting reflects the difficulty of stabilizing relation canonicalization under sparse evidence and lexical variability.

Failure modes and structural consistency. Residual failures concentrate in two regimes: (i) low-frequency relations with insufficient evidence to stabilize canonicalization, and (ii) semantically adjacent predicates whose distinction depends on context not fully captured at the sentence level. In both cases, NARROWER matches dominate, reflecting consistent but more fine-grained schema induction.

D/R consistency is computed only over relations with compatible mappings and therefore reflects structural coherence rather than coverage. The high D/R values indicate that, once aligned, TRACE relations connect semantically appropriate endpoint types, supporting the claim that the induced schema is structurally consistent and reusable.

D. Implementation Details

Setup. An anonymized implementation of TRACE-KG is available at https://anonymous.4open.science/r/TRACE-KG, with additional code included in the supplementary material. All experiments were conducted on a single compute node with one GPU (`gpu:1g.20gb:1`), 16 CPU cores, and 40 GiB RAM.

Pipeline configuration. TRACE-KG is implemented in Python and combines sentence-preserving chunking, transformer-based embeddings, density-based clustering, and step-specific LLM prompting within a unified pipeline. Unless otherwise specified, all stages use `gpt-5.1` with a maximum token budget of 16,000. Multimodal ingestion uses a vision-capable model (`gpt-5.1`) to convert non-text elements into structured text.

Chunking and context. Documents are segmented into sentence-preserving chunks of 100–200 tokens using spaCy (`en_core_web_sm`). Entity recognition operates at the chunk level with a context window of up to four preceding chunks.

Embeddings and clustering. Multi-field representations are embedded using `BAAI/bge-large-en-v1.5` (batch size 32, mean pooling, L2 normalization). Semantic neighborhoods are formed with HDBSCAN, with optional UMAP preprocessing. LLM calls are bounded by batching at most 10 items per prompt.

LLM interaction. LLM decisions are constrained through structured action interfaces, while execution is deterministic and validated by the pipeline. Entity recognition uses an 8,000-token limit, while resolution stages use 16,000 tokens. Temperature is omitted for GPT-5 models to ensure stable outputs. We use six stage-specific prompt templates—*Entity Recognition*, *Entity Resolution*, *Entity Class Recognition*, *Entity Class Resolution*, *Relation Recognition*, and *Relation Resolution*—defined in `TKG_Prompts.py`; all prompts are available at https://anonymous.4open.science/r/TRACE-KG.

Experiment-specific settings. For Experiment 1, retrieval uses top-$k = 8$ entities with a 2-hop expansion (up to 250 nodes and 300 edges), and evaluation uses a fixed LLM judge (`gpt-5.1`). For Experiment 2, candidate retrieval uses top-$K = 5$ schema elements per anchor; alignment uses an LLM verifier (1,400-token limit, top-$k = 6$), with anchored matching threshold 0.20 and at most 3 assignments per element. Low-confidence outputs (confidence < 0.88) are subject to targeted human audit.

E. Case Study

Document and motivation. As a running case study, we use *SEMI Document 6578*, a revision to *SEMI E10-0814E*, which specifies the definition and measurement of equipment reliability, availability, maintainability (RAM), and utilization. Maintenance and reliability records are a natural target for knowledge-graph construction because prior work has shown that such knowledge is difficult to reuse when it remains in noisy, semi-structured, or non-computable records rather than formalized semantic representations (Ameri et al., 2026). The document is a strong stress test for TRACE-KG because it is both technically dense and multimodal: even its opening pages combine formal purpose/scope statements, mutually exclusive equipment states, subsystem and cluster-tool terminology, and metric definitions such as MTBF, MCBF, MWBF, uptime, maintainability, and utilization. More broadly, this case study reflects long-context document understanding settings in which evidence is distributed across text, layout, tables, charts, and images rather than concentrated in a single local passage (Ma et al., 2024).

Why this document is challenging. The document mixes several levels of abstraction that are difficult to organize with flat triple extraction alone. It refers to equipment systems, subsystems, modules, states, substates, downtime categories, and performance measures, while also specifying how these concepts interact. Such hierarchical dependencies, dense technical terminology, and limited availability of external reference knowledge are characteristic challenges in domain-specific KG construction (Sun et al., 2025). In addition, essential information is distributed across prose, equations, and diagrams rather than appearing in a single textual form. A useful representation therefore requires more than local extraction: it must consolidate repeated technical concepts, distinguish closely related but non-identical terms, preserve provenance to the original evidence, and make granularity choices explicit. Analogous granularity issues have been observed in LLM-based log parsing, where different levels of specificity and applicability can yield different but plausible structured outputs (Zhong et al., 2024).

What TRACE-KG recovers. TRACE-KG converts the document into a consolidated, context-enriched graph while preserving direct links to source evidence. Figure 5 shows the resulting graph at document scale. Figure 6 shows that the induced schema is not flat: resolved entities are organized into interpretable groups such as manufacturing processes and operations, equipment time modeling, downtime and state semantics, performance and reliability analysis, and equipment condition/health. Organizing technical knowledge into structured and interpretable forms is important for effective representation, transmission, and operational use of domain knowledge, and knowledge graphs provide a natural mechanism for supporting such organization (Esmaeili & Bamdad Soofi, 2022). On the relation side, the induced schema is dominated by causal, trigger, and metric-impact patterns, reflecting the fact that RAM standards define performance through dependencies among states, events, and measurements.

Traceability and inspection. A key property of the case study is that the graph remains auditable against multimodal source material. This emphasis on evidence-grounded graph representations is consistent with prior work on explainable graph-based reasoning over heterogeneous sources, where entities and supporting evidence are linked explicitly to enable transparent and inspectable outputs (Christmann et al., 2023). Figure 7 shows a representative provenance example: the resolved ARAMS equipment time/state model is grounded back to its source diagram, and the resolved $MTBF_u$ calculation method (shown in Figure 8)

is grounded back to its originating equation region. Figures 9 and 10 show inspected graph instances at the node and relation levels. The resolved node *intended function of equipment* exposes induced class assignments, confidence, description, and supporting chunks, while the canonical relation `prevents_function` exposes evidence, provenance, and structured qualifier fields. Together, these examples illustrate that TRACE-KG preserves not only a consolidated graph structure, but also the evidence needed for technical interpretation and review.

Takeaway. This case study illustrates the type of document for which TRACE-KG is intended: standards-like technical material in which critical knowledge is distributed across text, equations, and diagrams, and in which analysts need a reusable semantic structure rather than isolated snippets. In this setting, the value of TRACE-KG lies not only in extracting entities and relations, but in organizing them into a coherent schema while maintaining end-to-end traceability.

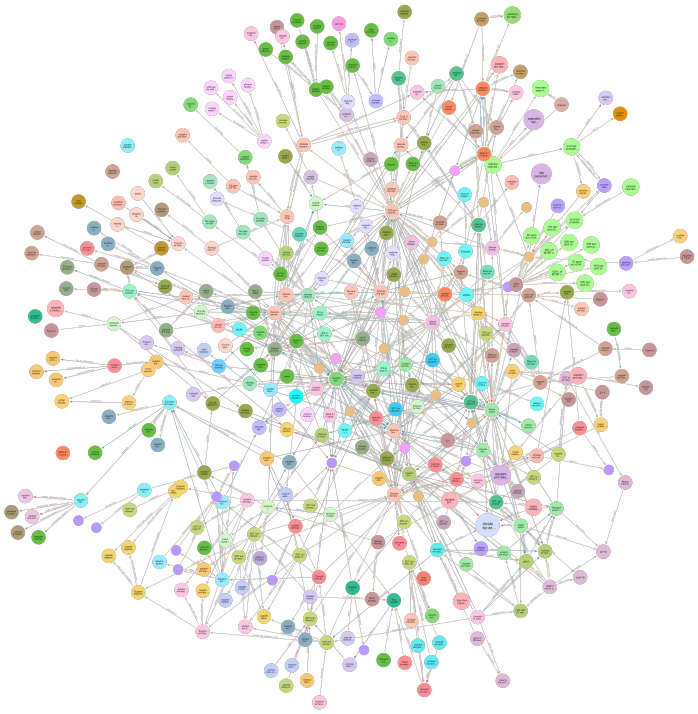

Figure 5. Overview of the constructed context-enriched knowledge graph for the case-study document.

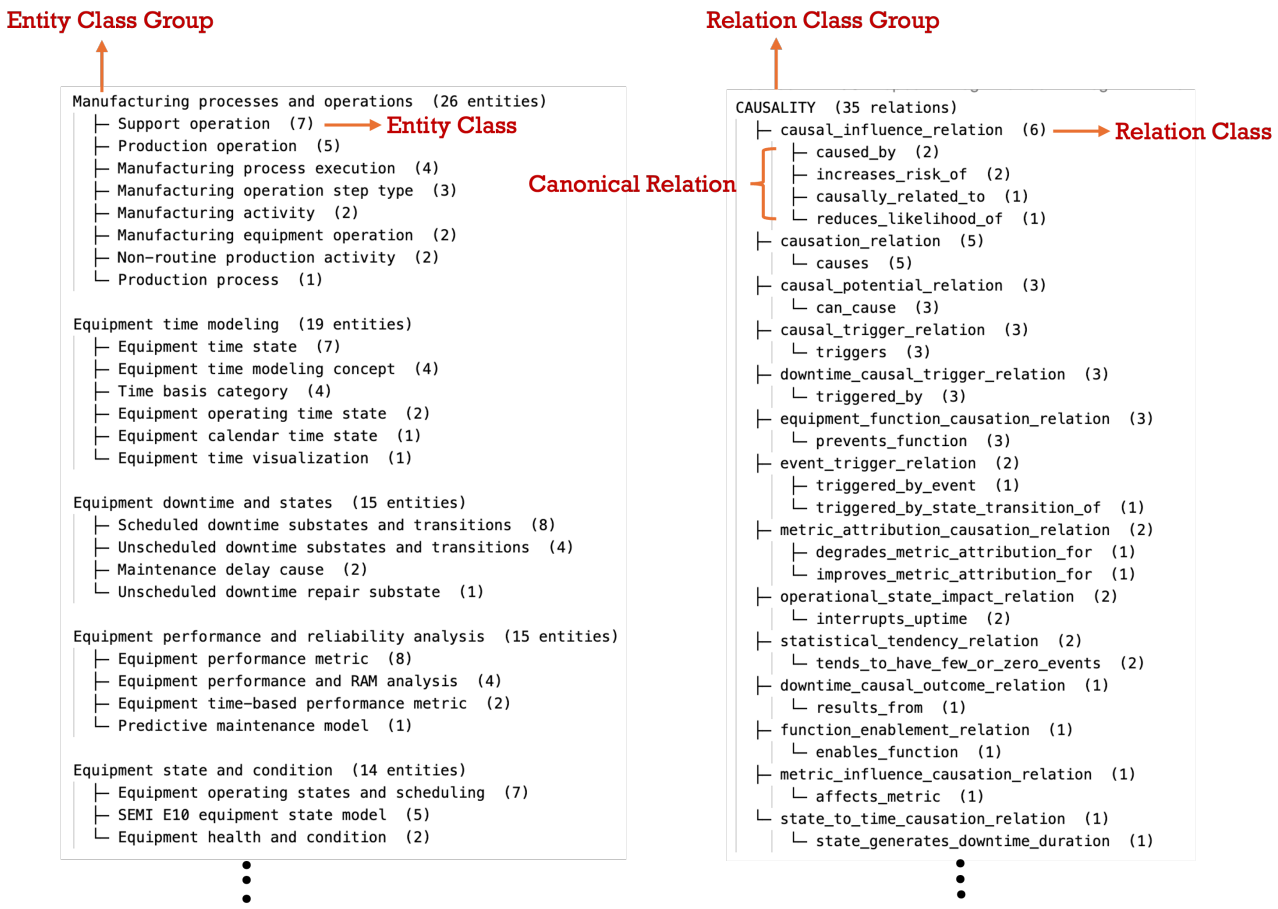

Figure 6. Induced entity and relation schema hierarchy for the case-study document.

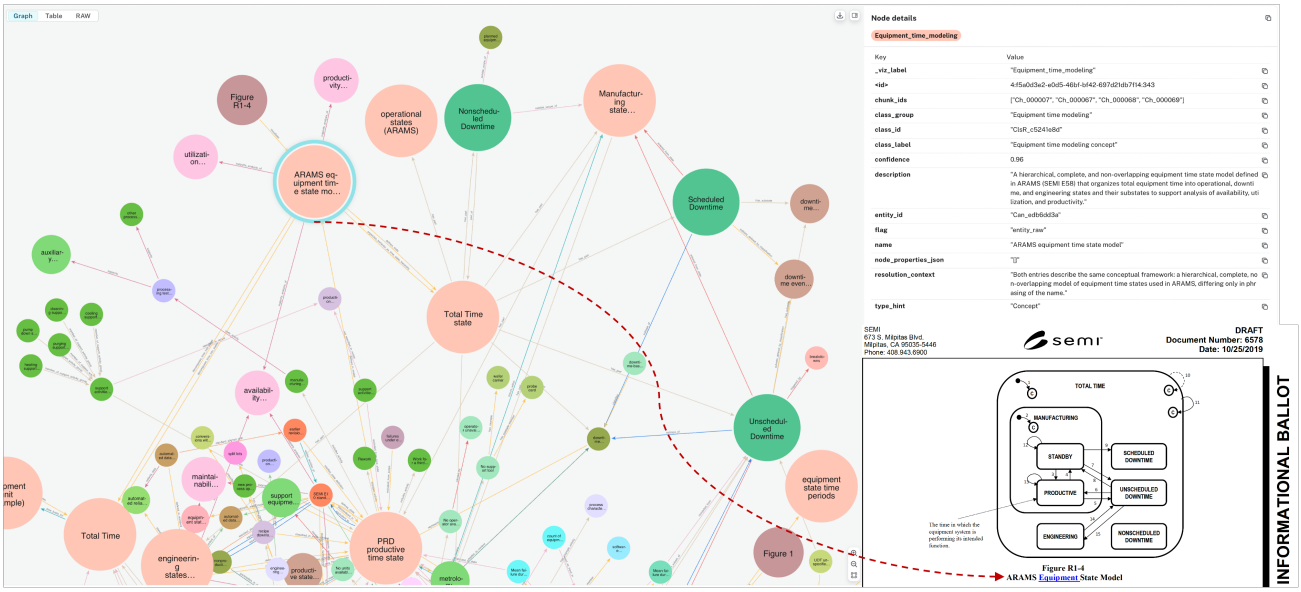

Figure 7. Traceability example (figure grounding): KG node linked to source diagram.

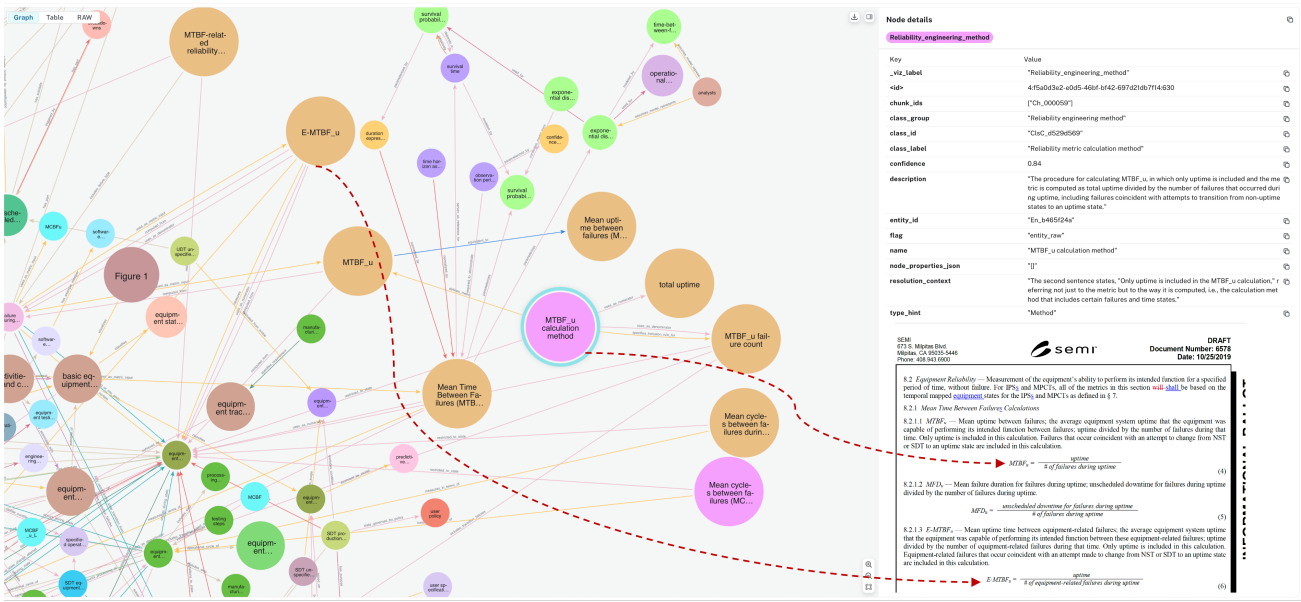

Figure 8. Traceability example (equation grounding): KG node linked to source equation.

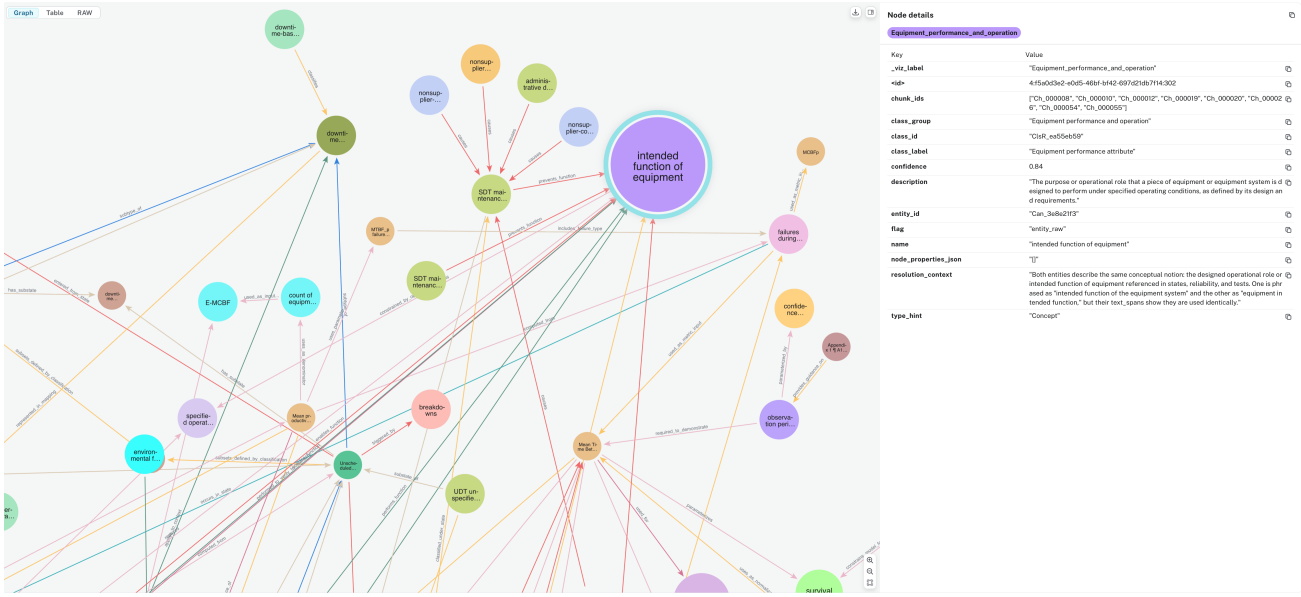

Figure 9. Example resolved entity with schema, confidence, and provenance.

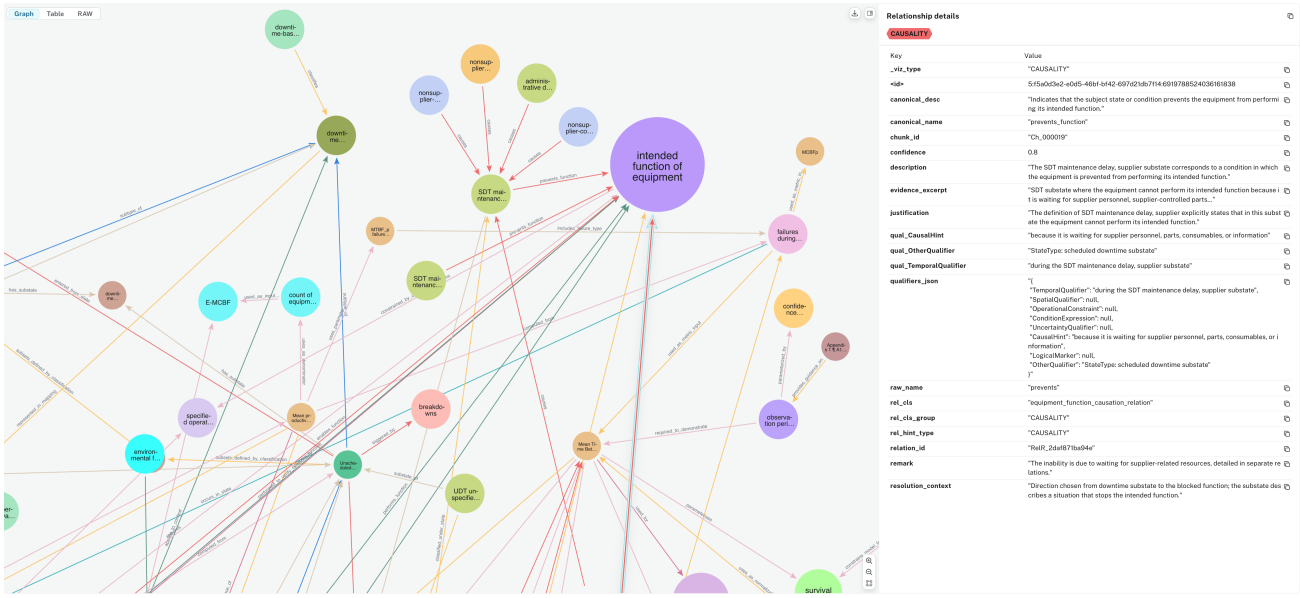

Figure 10. Example context-enriched relation with qualifiers and provenance.