Where, How, and What to Ground: Evaluating Ontology-Guided LLM-based Knowledge Graph Construction

Itisha Yadav[1,2], Roman Klinger[2] and Diana Peters[1]

[1]Institute of Data Science, German Aerospace Center (DLR), Jena, Germany
[2]Fundamentals of Natural Language Processing, University of Bamberg, Germany

Abstract

Ontology-guided information extraction transforms unstructured documents into structured knowledge graphs by grounding extracted entities and relations in predefined ontologies. Although recent large language model (LLM)-based approaches typically inject ontologies into prompts, their applicability to large and complex industrial ontologies remains underexplored. This work investigates three aspects of ontology-guided LLM-based knowledge graph construction: where grounding should occur, how alignment should be performed, and what type of ontology should be used. To understand these questions, we compare two ontology-grounding strategies. Early grounding performs extraction and ontology linking jointly through prompt-based ontology injection, whereas late grounding first performs unconstrained extraction and subsequently aligns the extracted knowledge to the ontology using embedding-based semantic matching. These strategies address the *where* and *how* aspects of ontology grounding. To investigate *what* ontology should be used, both approaches are evaluated on manually annotated semi-structured safety datasheets (SDSs) using two ontologies: the domain-specific SDS ontology developed in this work and the pre-existing generic upper-level ontology ORIONT. Results show that early grounding outperforms late grounding, indicating that prompt-based ontology injection is more effective than post-hoc alignment. Furthermore, domain-specific ontologies produce richer ABox representations compared to generic upper-level ontologies. These findings motivate a hierarchical grounding strategy, where information is first grounded in a domain ontology and then linked to generic ontological concepts for broader semantic interoperability. Although late grounding is affected by semantic alignment errors, its explicit mappings improve interpretability and provide valuable insights for refining the ontology-grounding pipeline. Across all settings, ontology hallucinations are negligible, with errors primarily arising from missed extractions and incorrect mappings. We conclude by discussing the limitations of this work and outlining directions for future research.

Keywords

Knowledge Graph Construction, Ontology-Based Information Extraction, Large Language Models, Prompt Engineering

1. Introduction

Information extraction converts unstructured or semi-structured documents into structured representations that can be consumed by downstream applications such as search, analytics, and decision-support systems [1, 2, 3]. When grounded in an ontology, extracted information gains a shared semantic structure, enabling reuse, querying, and interoperability across applications [4]. In a knowledge graph (KG), the ontology defines the TBox, while extracted instances populate the ABox [5, 6]. Recent advances in large language models (LLMs) have enabled ontology-guided knowledge graph construction through prompt-based approaches, where ontology concepts are injected directly into the model prompt [7, 8]. However, these approaches have primarily been evaluated using small demonstration ontologies. Extending these approaches to large-scale industrial ontologies with open-weights models remains challenging, especially for confidential document settings where proprietary LLM APIs such as Claude [9] and ChatGPT [10] cannot be used. As ontology size and complexity increase, the number of

Proceedings of the 5th International Workshop on Natural Language Processing for Knowledge Graph Construction (NLP4KGC 2026), co-located with the 25th International Semantic Web Conference (ISWC 2026), Bari, Italy, October 26, 2026
*Corresponding author.
itisha.yadav@dlr.de (I. Yadav); roman.klinger@uni-bamberg.de (R. Klinger); diana.peters@dlr.de (D. Peters)
0009-0002-7957-9600 (I. Yadav); 0000-0002-2014-6619 (R. Klinger); 0000-0002-5855-2989 (D. Peters)
© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-19

CEUR-WS.org/Vol-4335/paper4.pdf

classes, properties, and constraints that must be represented and considered during generation grows substantially, making reliable ontology-guided extraction more difficult. This raises fundamental design questions regarding ontology grounding in LLM-based extraction pipelines: *where* should ontology grounding occur, *how* should ontology information be incorporated, and *what* type of ontology is most suitable for extraction?

To address the *where* and *how* questions, we compare two ontology-grounding strategies. In *early grounding*, ontology concepts are verbalised and provided within the LLM prompt, enabling joint entity/relation extraction and ontology alignment, as adopted in prior prompt-based ontology-guided approaches [7, 8]. In *late grounding*, extraction is performed independently of the ontology, after which the extracted concepts are mapped to ontology entities and relations using embedding-based semantic alignment, inspired by prior work on post-hoc concept canonicalization [11]. To investigate the *what* question, we evaluate both strategies using two types of ontologies: SDS, a domain-specific ontology developed in this work, and ORIONT [12], a pre-existing generic upper-level ontology developed for life cycle assessment. We evaluate these approaches on safety datasheets (SDSs) [13], semi-structured industrial documents that combine tables and free-form text. Unlike plain textual documents, SDSs require reasoning over heterogeneous layouts, fragmented contexts, and domain-specific terminology, making them representative of real-world industrial document processing challenges.

Our experiments compare the resulting triples against manually annotated SDS data as ground truth. We analyse both extraction quality and ABox structure to understand how grounding strategy and ontology choice influence the resulting representations.

Our contributions are:

1. We provide a controlled comparison of early and late ontology grounding strategies for LLM-based knowledge graph construction, analysing the impact of ontology injection during generation versus post-hoc embedding-based alignment. We show that early grounding achieves higher overall extraction performance, while late grounding provides explicit alignment information that improves interpretability and supports ontology refinement.
2. We develop and release a domain-specific SDS ontology tailored to safety datasheet information extraction. The ontology is present in our code repository [14].
3. We investigate the impact of ontology selection and demonstrate that domain-specific ontologies provide richer ABox representations than generic upper-level ontologies. This highlights the importance of ontology-document vocabulary alignment for industrial knowledge graph construction.
4. We provide an end-to-end ontology-guided extraction pipeline for SDS documents, producing RDF/Turtle outputs from semi-structured PDF documents. Unlike prior SDS information extraction approaches that focus primarily on field extraction or named entity recognition (NER) [15], our approach targets full knowledge graph construction.

The code developed in this work is publicly available at [14]. The paper is organised as follows: Section 2 reviews related work. Section 3 describes the grounding approaches, SDS ontology, and experimental setup. Section 4 presents the results, which are discussed in Section 5. Section 6 concludes the paper and outlines future work.

2. Related Work

Classical knowledge graph construction pipelines decompose the task into supervised named entity recognition, coreference resolution, relation extraction, and entity linking [16, 17]. Joint models such as SpERT [18] and REBEL [19] improved end-to-end triple extraction but require large annotated corpora which is a bottleneck in domain-specific industrial settings. Instruction-following LLMs enabled prompt-based extraction in low-resource settings [20, 21], either as traditional three-part pipeline, including desiging an LLM-based NER model, an LLM-based relation extraction model, and an LLM-based entity linking model [22, 20, 23] or as a single multi-task step [8, 7].Text2KGBench [7], a single-step multi-task

approach, incorporates the ontology through prompt verbalization and reveals the limitations of this strategy when scaling to larger ontologies.

Our late grounding approach follows the post-hoc canonicalization paradigm, where information is extracted without ontology constraints and subsequently aligned to ontology concepts using semantic matching [24, 25, 11]. Recent approaches explore embedding-based alignment to bridge extracted vocabulary and predefined ontological schemas. Ontology embedding methods such as RDF2Vec [26] and OWL2Vec [27] further support this process by representing ontology concepts through their structural and semantic characteristics. However, late grounding remains limited by alignment errors, and sensitivity to embedding quality.

3. Methods

3.1. Task Definition and Design Space

Given a document D and an ontology $O = (C, P_{obj}, P_{data})$, the goal is to construct an ABox containing typed entities, object property assertions, and data property assertions grounded in the ontology. We organize ontology-guided KG construction along three dimensions: *where* ontology grounding occurs, *how* ontology knowledge is incorporated, and *what* type of ontology is used. The first dimension distinguishes *early grounding*, where ontology information is provided during generation, from *late grounding*, where extraction is performed independently and ontology alignment is conducted afterwards. The second dimension captures the mechanism of grounding: early grounding relies on prompt-based ontology verbalisation, allowing the LLM to jointly generate ontology-compliant triples, whereas late grounding performs explicit post-hoc semantic alignment between extracted concepts and ontology entities and relations. The third dimension concerns ontology specificity, where we compare a domain-specific ontology (SDS, constructed as part of this work; Section 3.3) with a generic upper-level ontology (ORIONT).

3.2. Grounding Approaches

Early grounding (ontology in the prompt). The prompt contains task instructions, a verbalisation of the ontology, and the document segment. The LLM performs entity extraction, entity-type resolution, relation extraction, and linking jointly, restricted to the supplied classes and relations, producing ontology-conformant triples directly. Because the ontology is consumed inside the model's forward pass, there is no intermediate artifact one can inspect to explain why a given concept was chosen. The prompt template is provided in the appendix (section A.2).

Late grounding (post-hoc alignment). Late grounding separates extraction from ontology alignment. First, information is extracted without ontology constraints, producing free-form entity mentions, entity types, and relation labels (prompt mentioned in appendix section A.1). Second, the extracted vocabulary is mapped to ontology concepts through embedding-based semantic alignment. This follows the post-hoc canonicalization paradigm explored in ontology matching and knowledge base construction, where extracted concepts are subsequently linked to predefined schemas [24, 25, 11]. Mappings are retained only when their cosine similarity exceeds a predefined threshold. Unlike early grounding, late grounding provides explicit alignment traces with candidate concepts and similarity scores, enabling error analysis and ontology refinement. To support post-hoc ontology alignment, we generate vector representations of ontology concepts following the OWL2Vec methodology [27]. We extend the original approach by replacing the Word2Vec encoder with the pretrained nomic-embed-text model [28]. Ontology concepts are represented through a textual description comprising their label, definition, and available example instances, following Song et al. [29]. Our initial experiments demonstrated that label-only representations were insufficient for reliable alignment. For extracted concepts, embeddings are generated from the complete extracted triple context, thereby capturing both entity semantics and relational context. We introduce two refinements to improve relation mapping.

Table 1
Structural statistics of the SDS and ORIONT ontologies. Cls., Obj., Data, Ind., Ax., Log., and Decl. denote classes, object properties, data properties, individuals, total axioms, logical axioms, and declaration axioms, respectively.

Ontology	Cls.	Obj.	Data	Ind.	Ax.	Log.	Decl.
SDS	95	75	24	–	1069	327	200
ORIONT	49	46	2	96	746	278	192

First, we apply *type-constrained* mapping, where entity types are mapped before relations. Our manual evaluation indicates that entity-type mapping achieves higher reliability compared to relation mapping. Consequently, we leverage the mapped subject and object types to restrict the candidate ontology properties to those compatible with their declared domains and ranges. This reduces semantically incorrect matches. Second, *ontology expansion via RDFS closure*: since domain and range constraints may be inherited from superclasses or superproperties, we materialise RDFS entailments using `owlrl`[1] before alignment. This makes implicit constraints explicit and ensures that relation candidates are filtered using the complete ontology semantics.

3.3. The SDS Ontology

Safety Data Sheets (SDSs) contain domain-specific concepts and relations that are not sufficiently represented by generic upper-level ontologies. Existing resources, such as ORIONT [12], lack the granularity required for SDS-specific knowledge representation. Our initial experiments with ORIONT resulted in lower triple recall and validity. Refer to results in (Section 4). This motivated the development of a dedicated SDS ontology. We developed the ontology through a semi-automatic three-step process. First, an LLM was used to propose an initial class and property structure based on the standardized 16-section SDS layout defined by OSHA [13]. This represents a low-risk application of LLMs, as the overall document structure is externally standardized. Second, human refinement was applied to class granularity, naming conventions, and domain/range constraints, while incorporating relevant concepts from ChEBI [30]. Third, the ontology was validated using the OOPS! pitfall scanner [31] to detect common OWL modelling issues. The validation revealed no critical or important pitfalls and identified only two minor issues, which are described in Appendix B. The resulting ontology statistics is summarized in Table 1, which presents the structural characteristics of the SDS ontology alongside those of ORIONT.

3.4. Experimental Setup

Data and model. We evaluate our methods on a safety datasheet selected from the dataset [32]. Since conventional PDF extraction tools such as pdfplumber[2] do not preserve table structure essential for SDS interpretation, we use PaddleOCR's PP-StructureV3 layout parser[3] to generate table-aware markdown for prompting. [4] For the information extraction part, both approaches use an instruction-tuned open-source model (quantised Qwen3.6-35B-A3B[5] via Ollama[6]) with default hyperparameters.

Ontologies and ground truth. We use two ontologies of different specificity: ORIONT [12], an upper-level ontology with generic concepts such as *Product system*, *Method*, *Flow classes*, *Measure*, and *Location*; and the SDS ontology (Section 3.3), whose concepts directly mirror SDS content. Structural statistics for both ontologies are reported in Table 1.

[1] https://owl-rl.readthedocs.io/
[2] https://github.com/jsvine/pdfplumber
[3] https://www.paddleocr.ai/main/en/version3.x/algorithm/PP-StructureV3/PP-StructureV3.html
[4] Among the evaluated document parsers, PaddleOCR achieved superior table structure preservation, producing more coherent representations than Docling [33], MinerU [34], and TableTransformer [35].
[5] https://huggingface.co/Qwen/Qwen3.6-35B-A3B
[6] https://ollama.com/

Table 2
Grounding results for SDS and ORIONT ontologies. Pr, Re, and F_1 denote precision, recall, and F_1 score; Cls and Rel denote ontology conformance for classes and relations. SH, OH, RH, and MH denote hallucinated subject types, object types, relations, and mentions, respectively.

Ont.	Approach	Thr.	Pr	Re	F_1	Cls	Rel	SH	OH	RH	MH
SDS	Early	–	0.486	0.279	0.355	1	1	0	0	0	0.054
	Late	0.73	0.295	0.042	0.074	1	1	0	0	0	0.033
	Late	0.60	0.116	0.071	0.089	1	1	0	0	0	0.026
ORIONT	Early	–	0.166	0.147	0.156	1	1	0	0	0	0.000
	Late	0.63	0.031	0.017	0.022	1	1	0	0	0	0.038
	Late	0.50	0.016	0.017	0.016	1	1	0	0	0	0.026

Ground-truth annotations were created for both ontologies by a single annotator with background in information extraction, in the format (subject_type, subject, predicate, object, object_type). It resulted in 308 annotated triples for the SDS ontology and 176 triples for ORIONT, on a single selected PDF.

Evaluation measures. Our evaluation measures are motivated by Text2KGBench [7]. We evaluate the generated knowledge graphs along three dimensions: (i) *fact accuracy*, measured using precision, recall, and F_1 over complete triples. A generated triple is considered correct when its predicate matches the reference triple and its arguments satisfy the corresponding matching criteria, using embedding similarity for literal values and type consistency for object properties; generated instance identifiers are disregarded. (ii) *ontology conformance*, which measures the validity of extracted classes and relations with respect to the target ontology; and (iii) *hallucination*, which quantifies unsupported vocabulary, including subject types (SH), relations (RH), and object types (OH) absent from the ontology, as well as extracted mentions that cannot be grounded to the source document through fuzzy matching (MH).

4. Results

4.1. Impact of grounding strategy

We evaluate early and late grounding across the SDS and ORIONT ontologies (Table 2). For the SDS ontology, early grounding achieves an F_1 score of 0.355, whereas late grounding achieves F_1 scores of 0.074 and 0.089 at similarity thresholds of 0.73 and 0.60, respectively. For ORIONT, early grounding achieves an F_1 score of 0.156, while late grounding achieves 0.022 and 0.016 at thresholds of 0.63 and 0.50, respectively. For each ontology, the first threshold was derived by averaging semantic-similarity scores between extracted entities and their nearest ontology class match, while the second was heuristically lowered.

4.2. Impact of ontology choice

The SDS ontology achieves higher extraction performance than ORIONT under both grounding strategies. Early grounding with SDS achieves an F_1 score of 0.355 compared with 0.156 for ORIONT. Similarly, late grounding achieves higher performance with SDS than ORIONT across the evaluated thresholds. Both ontologies achieve complete ontology conformance for extracted classes and relations (Cls = Rel = 1.0).

4.3. Hallucination and ABox analysis

Across all configurations, no hallucinated subject types, object types, or relations are observed (SH = OH = RH = 0). Mention hallucination remains low across all settings (MH $\leq$ 0.054). The structural analysis of generated ABoxes (Table 3) shows differences in the number of extracted classes, properties, individuals, and axioms across grounding strategies and ontologies.

Table 3
Structural metrics of generated ABoxes grouped by ontology and grounding approach. Cls., Obj., Data, Ind., Ax., Log., and Decl. denote classes, object properties, data properties, individuals, total axioms, logical axioms, and declaration axioms, respectively.

Approach	Cls.	Obj.	Data	Ind.	Ax.	Log.	Decl.
SDS							
Gold	47	32	21	135	78	63	15
Early	52	25	18	122	346	303	43
Late (.73)	10	4	11	19	78	63	15
ORIONT							
Gold	24	22	2	100	301	277	24
Early	18	12	7	122	301	250	19
Late (.63)	12	3	10	45	154	109	13

5. Discussion

5.1. Grounding strategy trade-offs

The results indicate that early and late grounding represent different trade-offs. Providing ontology information during generation enables the LLM to jointly perform extraction and ontology alignment, whereas late grounding separates these stages and relies on the quality of semantic matching. Although late grounding achieves lower extraction performance, its explicit alignment stage provides transparency into mapping decisions and can support error analysis and ontology refinement.

5.2. Importance of ontology-document alignment

The higher F_1 scores achieved by the SDS ontology over ORIONT across both grounding strategies indicate that ontology selection is a critical factor in ontology-guided extraction. Domain-specific ontologies provide finer-grained concepts closer to document terminology, while generic upper-level ontologies offer broader semantic coverage. This motivates a hierarchical grounding strategy, where extracted information is first aligned with domain-specific concepts and subsequently mapped to upper-level ontologies for interoperability and linked data integration [36]. Furthermore, the absence of ontology hallucinations indicates that the main challenges are not the generation of spurious ontology concepts, but rather incomplete knowledge extraction and inaccurate alignment of extracted facts with ontology concepts.

6. Conclusion and Future Work

This work investigated ontology grounding in LLM-based knowledge graph construction by comparing early grounding through prompt-based ontology injection with late grounding through embedding-based post-hoc alignment. Experiments on semi-structured safety datasheets show that grounding strategy, ontology specificity, and ontology-document alignment strongly influence extraction quality. Early grounding provides stronger overall performance, while late grounding offers explicit alignment decisions. Furthermore, domain-specific ontologies produce richer and more accurate representations than generic upper-level ontologies. Across all settings, ontology hallucination remains negligible, with errors primarily arising from incomplete extraction or incorrect alignment. As this work represents an initial investigation, it is limited to a single PDF and document type. Future work can focus on evaluating the approaches on larger dataset and additional document types. Additionally, exploring hybrid grounding pipelines that combine ontology-guided extraction with adaptive post-hoc alignment and human-in-the-loop validation. Systematic benchmarking of different LLMs for extraction and embedding generation for post-hoc ontology alignment can further provide insights into their impact on pipeline performance.

Declaration on Generative AI

During the preparation of this work, the author(s) used internally hosted large language models for the purposes of text paraphrasing. After using these LLMs, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] J. Jiang, Information Extraction from Text, Springer US, 2012, pp. 11–41. doi:10.1007/978-1-4614-3223-4_2.

[2] A. Hogan, C. Gutierrez, M. Cochez, G. d. Melo, S. Kirrane, A. Polleres, R. Navigli, A.-C. N. Ngomo, S. M. Rashid, L. Schmelzeisen, S. Staab, E. Blomqvist, C. d'Amato, J. E. L. Gayo, S. Neumaier, A. Rula, J. Sequeda, A. Zimmermann, Knowledge Graphs, volume 54, Springer International Publishing, 2022. doi:10.1007/978-3-031-01918-0.

[3] A. Ittoo, L. M. Nguyen, A. van den Bosch, Text analytics in industry: Challenges, desiderata and trends, Computers in Industry 78 (2016) 96–107. doi:10.1016/j.compind.2015.12.001.

[4] T. R. Gruber, A translation approach to portable ontology specifications, Knowledge Acquisition 5 (1993) 199–220. doi:10.1006/knac.1993.1008.

[5] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, S. Rudolph, et al., OWL 2 web ontology language primer, W3C recommendation 27 (2009) 123.

[6] G. De Giacomo, M. Lenzerini, et al., Tbox and abox reasoning in expressive description logics., KR 96 (1996) 10.

[7] N. Mihindukulasooriya, S. Tiwari, C. F. Enguix, K. Lata, Text2KGBench: A Benchmark for Ontology-Driven Knowledge Graph Generation from Text, Springer Nature Switzerland, 2023, pp. 247–265. doi:10.1007/978-3-031-47243-5_14.

[8] Z. Cauter, N. Yakovets, Ontology-guided knowledge graph construction from maintenance short texts, in: Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024), Association for Computational Linguistics, 2024, pp. 75–84. doi:10.18653/v1/2024.kallm-1.8.

[9] Anthropic, The Claude 3 Model Family: Opus, Sonnet, Haiku, Technical Report (2024). URL: https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.

[10] OpenAI, GPT-4, https://openai.com/research/gpt-4, 2023. Accessed: 2026-07-24.

[11] B. Zhang, H. Soh, Extract, define, canonicalize: An LLM-based framework for knowledge graph construction, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 9820–9836. URL: https://aclanthology.org/2024.emnlp-main.548/. doi:10.18653/v1/2024.emnlp-main.548.

[12] L. Zampori, T. M. Bachmann, Data Ontologies: Documentation of the ORIENTING LCSA Ontology (ORIONT), Technical Report, ORIENTING Project, 2024.

[13] J. W. Ronald, Understanding a safety data sheet (SDS) in regards to process safety, Procedia Engineering 45 (2012) 857–867. doi:10.1016/j.proeng.2012.08.250.

[14] I. Yadav, Early vs late ontology grounding, 2026. URL: https://doi.org/10.5281/zenodo.21511935. doi:10.5281/zenodo.21511935.

[15] M. Khan, J. Penfield, A. Suman, S. Crowell, A machine learning driven automated system to extract multiple information fields from safety data sheet documents, Heliyon 11 (2025) e42215. doi:10.1016/j.heliyon.2025.e42215.

[16] G. Weikum, X. L. Dong, S. Razniewski, F. Suchanek, Machine knowledge: Creation and curation of comprehensive knowledge bases, Foundations and Trends in Databases 10 (2021) 108–490. doi:10.1561/1900000064.

[17] S. Ji, S. Pan, E. Cambria, P. Marttinen, P. S. Yu, A survey on knowledge graphs: Representation,

acquisition, and applications, IEEE Transactions on Neural Networks and Learning Systems 33 (2022) 494–514. doi:10.1109/tnnls.2021.3070843.

[18] E. Markus, U. Adrian, Span-Based joint entity and relation extraction with transformer pre-training, in: ECAI 2020, IOS Press, 2020. doi:10.3233/faia200321.

[19] P.-L. Huguet Cabot, R. Navigli, REBEL: Relation extraction by end-to-end language generation, in: Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, 2021, pp. 2370–2381. doi:10.18653/v1/2021.findings-emnlp.204.

[20] S. Wadhwa, S. Amir, B. Wallace, Revisiting relation extraction in the era of large language models, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2023, pp. 15566–15589. doi:10.18653/v1/2023.acl-long.868.

[21] X. Wei, X. Cui, N. Cheng, X. Wang, X. Zhang, S. Huang, P. Xie, J. Xu, Y. Chen, M. Zhang, Y. Jiang, W. Han, ChatIE: Zero-shot information extraction via chatting with chatgpt, arXiv preprint arXiv:2302.10205 (2023). doi:10.48550/arXiv.2302.10205.

[22] S. Wang, X. Sun, X. Li, R. Ouyang, F. Wu, T. Zhang, J. Li, G. Wang, C. Guo, GPT-NER: Named entity recognition via large language models, in: Findings of the Association for Computational Linguistics: NAACL 2025, Association for Computational Linguistics, 2025, pp. 4257–4275. doi:10.18653/v1/2025.findings-naacl.239.

[23] A. Xin, Y. Qi, Z. Yao, F. Zhu, K. Zeng, B. Xu, L. Hou, J. Li, LLMAEL: Large language models are good context augmenters for entity linking, in: Proceedings of the 34th ACM International Conference on Information and Knowledge Management, CIKM '25, ACM, 2025, pp. 3550–3559. doi:10.1145/3746252.3761156.

[24] S. Vashishth, P. Jain, P. Talukdar, CESI: Canonicalizing open knowledge bases using embeddings and side information, in: Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18, WWW '18, ACM Press, 2018, pp. 1317–1327. doi:10.1145/3178876.3186030.

[25] S. Hertling, H. Paulheim, OLaLa: Ontology matching with large language models, in: Proceedings of the 12th Knowledge Capture Conference 2023, K-CAP '23, ACM, 2023, pp. 131–139. doi:10.1145/3587259.3627571.

[26] P. Ristoski, H. Paulheim, RDF2Vec: RDF Graph Embeddings for Data Mining, Springer International Publishing, 2016, pp. 498–514. doi:10.1007/978-3-319-46523-4_30.

[27] J. Chen, P. Hu, E. Jimenez-Ruiz, O. M. Holter, D. Antonyrajah, I. Horrocks, OWL2Vec*: embedding of owl ontologies, Machine Learning 110 (2021) 1813–1845. doi:10.1007/s10994-021-05997-6.

[28] Z. Nussbaum, J. X. Morris, B. Duderstadt, A. Mulyar, Nomic embed: Training a reproducible long context text embedder, arXiv preprint arXiv:2402.01613 (2024).

[29] Y. Song, J. Chen, R. A. Schmidt, GenOM: ontology matching with description generation and large language models, World Wide Web 29 (2026). doi:10.1007/s11280-026-01413-y, arXiv:2508.10703.

[30] K. Degtyarenko, P. de Matos, M. Ennis, J. Hastings, M. Zbinden, A. McNaught, R. Alcantara, M. Darsow, M. Guedj, M. Ashburner, ChEBI: a database and ontology for chemical entities of biological interest, Nucleic Acids Research 36 (2007) D344–D350. doi:10.1093/nar/gkm791.

[31] M. Poveda-Villalón, A. Gómez-Pérez, M. C. Suárez-Figueroa, OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation, International Journal on Semantic Web and Information Systems (IJSWIS) 10 (2014) 7–34.

[32] M. Khan, J. Penfield, Safety data sheet (SDS) dataset, 2024. URL: https://doi.org/10.5281/zenodo.14580098. doi:10.5281/zenodo.14580098.

[33] N. Livathinos, C. Auer, M. Lysak, A. Nassar, M. Dolfi, P. Vagenas, C. B. Ramis, M. Omenetti, K. Dinkla, Y. Kim, et al., Docling: An efficient open-source toolkit for ai-driven document conversion, arXiv preprint arXiv:2501.17887 (2025).

[34] B. Wang, C. Xu, X. Zhao, L. Ouyang, F. Wu, Z. Zhao, R. Xu, K. Liu, Y. Qu, F. Shang, et al., Mineru: An open-source solution for precise document content extraction, arXiv preprint arXiv:2409.18839 (2024).

[35] B. Smock, R. Pesala, R. Abraham, Pubtables-1m: Towards comprehensive table extraction from unstructured documents, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2022, pp. 4624–4632. doi:10.1109/cvpr52688.2022.00459.
[36] T. Berners-Lee, Linked Data, 2006. URL: https://www.w3.org/DesignIssues/LinkedData.html.

A. Prompt Templates

We reproduce the two prompt templates used for the extraction step. Both instruct the model to output a simple, parseable format (`type(...)` and `relation(...)` statements) in Prolog style rather than JSON, which we found more robust.

A.1. Unconstrained (no-ontology) extraction prompt

```
You are an information extraction system for Safety Data Sheets (SDS).
Given a text segment from an SDS document, extract entities and triples.

Output format -- two types of lines, one per line:
1. Entity typing: type(entity, ClassName)
2. Relation triple: relation(subject, object)

Rules:
- subject and object are short snake_case identifiers (e.g. product_1, xxx_dioxide_1, ghs_HXXX).
- For literal values, wrap the object in double quotes: relation(subject, "value").
- Use consistent identifiers across triples (e.g. always product_1 for the same product).
- Every entity identifier that appears as a subject or object (non-literal) must have
  at least one type(...) line.
- Output only type declarations and triples, no explanation or markdown.

Example output:
type(product_1, SomeProduct)
type(xxx_dioxide_1, HazardousComponent)
hasComponent(product_1, xxx_dioxide_1)
casNumber(xxx_dioxide_1, "12347-67")
```

A.2. Ontology-grounded extraction prompt

```
You are an ontology-based information extraction system for Safety Data Sheets (SDS).
Given a text segment from an SDS document, extract triples according to the relations
and concepts defined in the ontology below.

Ontology concepts (entity types): {concepts}
Ontology relations: {relations}

Output format -- two types of lines, one per line:
1. Entity typing: type(entity, ClassName)
2. Relation triple: relation(subject, object)

Rules:
- subject and object are short snake_case identifiers (e.g. product_1, xxx_dioxide_1, ghs_HXXX).
- For literal values, wrap the object in double quotes: relation(subject, "value").
- Use consistent identifiers across triples (e.g. always product_1 for the same product).
- Only extract relation triples for relations that appear in the ontology relations list above.
- Only assign entity types that appear in the ontology concepts list above.
- Every entity identifier that appears as a subject or object (non-literal) must have
  at least one type(...) line.
- If an entity could belong to a subclass (e.g. sds:HazardousComponent rather than
  just sds:ChemicalComponent), prefer the most specific class that is justified by the text.
- Output only type declarations and triples, no explanation or markdown.

Example output:
type(product_1, SomeProduct)
type(xxx_dioxide_1, HazardousComponent)
hasComponent(product_1, xxx_dioxide_1)
casNumber(xxx_dioxide_1, "12347-67")
```

In the ontology-grounded prompt, `{concepts}` and `{relations}` are populated at run time with the verbalised classes and object properties of the ontology.

B. SDS Ontology Validation Report (OOPS!)

We validated the SDS ontology with the OOPS! (OntOlogy Pitfall Scanner!), which checks for common syntax-based pitfalls in ontology, independent of any specific extraction task. Table 4 summarises the scan's findings: two pitfalls, both of *minor* importance.

Table 4
OOPS! validation report for the SDS ontology.

Code	Pitfall	Severity	Affected element(s)
P21	Using a miscellaneous class	Minor	Section16OtherInformation (1 element)
P13	Inverse relationships not explicitly declared	Minor	requiresPPE, hasRouteOfExposure, hasFirstAidMeasure (3 elements)

P21 flags classes created solely to hold instances that do not belong to any sibling class; in our case, Section16OtherInformation mirrors the SDS standard's own "Section 16: Other Information" catch-all, so the pitfall reflects the source document standard rather than an unintended modelling choice. P13 flags object properties without a declared `owl:inverseOf`; none of the three flagged properties (requiresPPE, hasRouteOfExposure, hasFirstAidMeasure) required a semantically meaningful inverse for the extraction task at hand. No pitfalls of Important or Critical severity were reported.