

Scaling Ontology Engineering with LLMs: Semantic Alignment and Coverage Evaluation

Sebastian Neumaier^{1,2}, Tobias Dam¹ and Fabian Kovac¹

¹University of Applied Sciences St. Pölten, Austria

²Vienna University of Economics and Business, Austria

Abstract

We present NeOn-ACE (Alignment and Coverage Evaluation), a NeOn-GPT extension that integrates LLMs into ontology engineering for regulation-driven application profiles. NeOn-ACE extends the base NeOn-GPT pipeline with two novel steps: (i) LLM-assisted semantic alignment to reference ontologies and (ii) iterative LLM-based coverage evaluation against regulatory competency questions. We demonstrate the methodology using the EU AI Act Annex IV as one example regulatory context, with prototypical results and tools available at <https://w3id.org/aidoc-ap/>.

Keywords

Ontology Engineering, LLM, EU AI Act, Technical Documentation, Compliance, AI System

1. Introduction

Recent regulatory developments such as the EU AI Act [1] impose detailed technical documentation obligations on high-risk AI systems. Current documentation practices remain largely unstructured and text-based, hindering automated compliance and cross-system comparison. We present NeOn-ACE, an extended methodology for regulation-driven ontology engineering that adapts NeOn-GPT [2], an LLM-powered framework, with two key extensions: (i) LLM-assisted semantic alignment to reference ontologies and (ii) iterative LLM-based coverage evaluation against regulatory competency questions (CQ). In this paper, the EU AI Act is used as a motivating example; the methodology itself is designed for regulatory texts in general.

2. Background and Related Work

Large language models have been integrated into several stages of the ontology engineering lifecycle. NeOn-GPT [2] operationalises competency-question-driven development by automating terminology extraction, definition generation, taxonomy construction, and axiomatisation, with reasoning and human review in the loop. A separate strand targets ontology matching: LLMs4OM applies LLMs for zero-shot alignment across ontologies [3], prompt-based frameworks address more complex alignment tasks [4], and multi-agent dialogue has been explored as a means of negotiating mappings [5]. A third strand uses LLMs to generate or critique competency questions, e.g., to detect semantic modelling pitfalls [6].

These strands remain disjoint. NeOn-GPT and related pipelines build ontologies bottom-up from source texts but provide no systematic way to (i) reuse existing vocabularies through alignment or (ii) measure how well the evolving ontology covers an externally fixed set of requirements. Conversely, LLM-based matchers operate on finished ontologies and do not feed back into iterative refinement. As a result, when an ontology is driven by an external requirement source — a standard, a regulation, or a domain reference — engineers must manually orchestrate vocabulary reuse and completeness tracking across iterations. NeOn-ACE closes this loop: to our knowledge it is the first methodology to

Posters, Demos, Blue Sky, and Tutorials at SEMANTiCS 2026, Sep 2026, Ghent, Belgium

✉ sebastian.neumaier@ustp.at (S. Neumaier); tobias.dam@ustp.at (T. Dam); fabian.kovac@ustp.at (F. Kovac)

🆔 0000-0002-9804-4882 (S. Neumaier); 0000-0002-2463-5831 (T. Dam); 0000-0002-1611-2598 (F. Kovac)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

than final modelling decisions. The ontology authors review, revise, and consolidate the terminology to keep it faithful to the regulatory source text and consistent with domain usage.

(2) *Taxonomy Construction*. Using the refined vocabulary, we prompt the LLM to suggest subclass relations and broader category placements. For instance, it may propose that documentation about training datasets should be treated as a more specific type of dataset documentation. We manually validate these subsumptions and reject overly broad or ambiguous links. The taxonomy is iteratively adjusted until it supports intended competency-question patterns and produces a coherent class hierarchy.

(3) *Axiomatisation*. We then prompt the LLM to propose OWL axioms that operationalise the conceptual model, including domain and range restrictions. For example, a relation such as “uses training data” can be constrained so it only links training activities to dataset entities. Proposed axioms are examined by ontology engineers for logical soundness and modelling utility, and only accepted axioms are integrated into the OWL ontology.

(4) *Automated Reasoning* & (5) *Pitfall Detection*. After each modelling iteration, we run the HermiT reasoner to detect logical issues (unsatisfiable classes, conflicting constraints, unintended inferences) and periodically run OOPS! for common ontology pitfalls such as incomplete annotations or ambiguous naming. In the current implementation, reasoning is performed only over the generated ontology and does not include imported or aligned external ontologies. This reflects the workflow order: reasoning is used as quality control already before the dedicated alignment step establishes cross-ontology mappings. Detected issues are manually reviewed and resolved through axiom or hierarchy updates, providing a quality-control loop for each modelling cycle.

3.2. Step 6: LLM-assisted Alignment to Reference Ontologies

To ensure interoperability with other vocabularies, we systematically align the resulting ontology to reference ontologies such as those reviewed in section 2. As manual alignment is time-consuming and error-prone, we introduce an LLM-assisted alignment step. We applied this step to the ontology described in section 4, and the resulting alignments are available online.¹

We chose this custom alignment procedure over a standalone alignment system because our objective was an LLM-based method with lightweight integration into the iterative NeOn-GPT workflow, reusing the same prompt-driven engineering setup across modelling and evaluation steps. We therefore perform alignment after the initial ontology draft is constructed (Steps 1–3), to prioritise regulation-grounded concept elicitation, reduce early anchoring to external ontology design choices, and keep prompt context manageable when multiple reference ontologies are considered. Comparative evaluation against established ontology alignment systems is planned as future work.

We first generate candidate pairs using lexical similarity (normalised Levenshtein distance and token overlap) over labels and comments. We chose this lexical filter as it provides a practical, language-independent baseline before semantic matching with the LLM. For each pair, we present the LLM with the labels, definitions and usage contexts of both terms, and ask it to classify the relationship and assign a confidence score. Only correspondences with scores above 0.75 are retained. Alignments are manually reviewed by domain experts, who can accept, reject or refine the suggested relation.

Example alignment prompt (simplified):

Role: “You are an experienced knowledge engineer.”

Inputs: a pair of ontology terms with labels and definitions, e.g., `aidoc:AIDeployer` (“AI Deployer” – “...where the AI system is used in the course of a personal non-professional activity”) and `airo:AIDeployer` (same label and definition).

Task: classify the relationship as `owl:equivalentClass`, `skos:closeMatch`, `skos:broader`, `skos:narrower`, or `n/a`; assign a confidence score in $[0, 1]$; return a JSON object with fields `relation`, `confidence`, `explanation`.²

¹<https://certain-project.github.io/aidoc-ap/alignments.html>

²The full prompt is available in the project repository.

Example resulting LLM output:

```
1 { "relation": "owl:equivalentClass",
2   "confidence": 0.99,
3   "explanation": "The labels and descriptions are identical. The airo concept provides a URI,
   ↳ indicating it's formally defined; however, the descriptive semantics are the same. Therefore,
   ↳ they are equivalent.",
4   "source": "aidoc:AIDeployer",
5   "target": "https://w3id.org/airo#AIDeployer" }
```

3.3. Step 7: LLM-based Iterative Coverage Evaluation

A key challenge in building a regulation-driven ontology is ensuring that it adequately covers all legal requirements. We operationalise this notion of “coverage” through CQs: for each requirement in the target regulation, we define a set of CQs that a knowledge graph must be able to answer. We then use an LLM-based evaluator to estimate how well the current ontology supports these CQs (in our example, these are Annex IV requirements). We applied this step to the ontology described in section 4, and the resulting coverage results are available online.³

Coverage evaluation as an iterative process. The evaluator runs after major revisions; scores are compared across iterations to track progress and reveal gaps, guiding refinements such as adding missing classes, properties, or axioms.

Input and output schema. The evaluator receives plain-text requirements with CQs and candidate ontology entities, returning a JSON object with fields: "id" (requirement id), "score" (c_r), "matched_terms", and "explanation".

Metric interpretation. The coverage score c_r is an LLM-estimated indicator, not a validated metric. It is a subjective estimate of how well the ontology supports the requirement's CQs; we limit c_r to the unit interval and require justification with concrete classes or properties. Although unvalidated, the definition is stable across runs and versions, making it useful for comparative refinement. It should be treated as decision support, not a compliance guarantee.

Example coverage evaluator prompt (simplified):

Role: “You are an experienced knowledge engineer and expert on the EU AI Act.”

Inputs: a requirement (e.g., “A general description of the AI system including...name of the provider...”) with its CQs (e.g., “What is the intended purpose?”, “What is the provider name?”...); ontology terms as label-description pairs (aidoc:AISystem, aidoc:AIProvider, ...).

Task: pick at most 10 terms, assign an overall coverage score in $[0, 1]$, justify in 3–5 sentences; return a JSON object with fields id, score, matched_terms, explanation.⁴

Example resulting LLM output:

```
1 { "id": "aiact:req1",
2   "score": 0.85,
3   "matched_terms": ["aidoc:AIProvider", "aidoc:AISystem", "aidoc:isProvidedBy"],
4   "explanation": "The ontology directly addresses the requirement through 'AI System' as the core
   ↳ entity, 'AI Provider' which captures the provider's identity, and 'is provided by' establishing
   ↳ the link between the system and provider. While the requirement asks for a 'version' and
   ↳ 'relation to previous versions', this isn't explicitly modeled, though it could be derived with
   ↳ more complex provenance tracking of AI Artifacts or through a dedicated versioning property. The
   ↳ 'intended purpose' is implicitly captured within the definition of 'AI System' itself." }
```

³<https://certain-project.github.io/aidoc-ap/coverage.html>

⁴The full prompt is available in the project repository.

4. Implementation and Use Case Application

The NeOn-ACE methodology is currently being applied and validated within the framework of the Horizon Europe project CERTAIN.⁵ In this project, we apply the methodology to the development of a formal EU AI Act Annex IV ontology and a supporting knowledge graph populated with data from several pilot use cases. This implementation serves as a practical testbed for the iterative coverage evaluation (Step 7), allowing us to observe how the ontology evolves to meet the specific documentation requirements of high-risk AI systems. The prototypical implementation includes a first ontology model AIDOC-AP [13], the results of the LLM-based coverage analysis step, and the semantic alignment mappings to reference vocabularies; the public web interface is available at <https://w3id.org/aidoc-ap/>.

5. Conclusion and Future Work

We present NeOn-ACE, an extended NeOn-GPT methodology for regulation-driven ontology engineering. By combining LLM-assisted semantic alignment and iterative coverage evaluation, we systematize the integration of LLMs into complex ontology projects. The methodology balances scalability (LLM automation of candidate generation) with quality (domain expert curation). Future work includes empirical validation of coverage indicators, automation of knowledge graph construction, extension to additional regulatory contexts, evaluation of other alignment strategies that reuse established systems such as AgreementMakerLight [14] and OntoAligner [15], and post-alignment reasoning that includes imported/aligned ontologies to assess cross-ontology consistency. The current EU AI Act Annex IV implementation within the Horizon Europe project CERTAIN is one example application of this approach.

Declaration on Generative AI

During the preparation of this work, the author(s) used GitHub Copilot in order to: Draft content, Paraphrase and reword, Improve writing style, Abstract drafting, Grammar and spelling check. Further, the author(s) used ChatGPT (Overleaf AI Assistant) in order to: Formatting assistance. Claude Sonnet was used to support the development of source code. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

Acknowledgments

This research was funded by the European Union's Horizon Europe research and innovation programme HORIZON-CL4-2024-DATA-01-01 under grant agreement No. 101189650, and from the Swiss State Secretariat for Education, Research and Innovation (SERI).

References

- [1] European Parliament and Council, Regulation (eu) 2024/1689 of the european parliament and of the council of 13 june 2024 on artificial intelligence (ai act), <https://eur-lex.europa.eu/eli/reg/2024/1689/oj>, 2024. Official Journal of the European Union L 206.
- [2] N. Fathallah, A. Das, S. D. Giorgis, A. Poltronieri, P. Haase, L. Kovriguina, Neon-gpt: A large language model-powered pipeline for ontology learning, in: The Semantic Web: ESWC 2024 Satellite Events, volume 15344 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 36–50. doi:10.1007/978-3-031-78952-6_4.
- [3] H. Babaei Giglou, J. D'Souza, F. Engel, S. Auer, Llms4om: Matching ontologies with large language models, in: The Semantic Web: ESWC 2024 Satellite Events, Springer, 2025, pp. 25–35.

⁵<https://certain-project.eu/>, last accessed 2026-02-02

- [4] R. Amini, S. S. Norouzi, P. Hitzler, R. Amini, Towards complex ontology alignment using large language models, 2024. URL: <https://arxiv.org/abs/2404.10329>. arXiv: 2404.10329.
- [5] S. Zhang, Y. Dong, Y. Zhang, T. R. Payne, J. Zhang, Large language model assisted multi-agent dialogue for ontology alignment, in: Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS '24, Richland, SC, 2024, p. 2594–2596.
- [6] H. Choi, S. Hwang, K.-H. Lee, Vspo: Validating semantic pitfalls in ontology via llm-based cq generation, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 40, 2026, pp. 18977–18984. URL: <https://arxiv.org/abs/2511.07991>. arXiv: 2511.07991.
- [7] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, T. Gebru, Model cards for model reporting, in: Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29–31, 2019, ACM, 2019, pp. 220–229. doi:10.1145/3287560.3287596.
- [8] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. D. III, K. Crawford, Datasheets for datasets, Commun. ACM 64 (2021) 86–92. doi:10.1145/3458723.
- [9] D. Golpayegani, H. J. Pandit, D. Lewis, AIRO: an ontology for representing AI risks based on the proposed EU AI act and ISO risk management standards, in: Proceedings of the 18th International Conference on Semantic Systems (SEMANTiCS 2022), 2022, Vienna, Austria, volume 55 of *Studies on the Semantic Web*, IOS Press, 2022, pp. 51–65. doi:10.3233/SSW220008.
- [10] D. Golpayegani, H. J. Pandit, D. Lewis, To be high-risk, or not to be - semantic specifications and implications of the AI act's high-risk AI applications and harmonised standards, in: Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, FAccT 2023, ACM, 2023, pp. 905–915. doi:10.1145/3593013.3594050.
- [11] D. Esteves, A. Ławrynowicz, P. Panov, L. Soldatova, T. Soru, J. Vanschoren, ML Schema Core Specification, Technical Report, W3C Machine Learning Schema Community Group, 2016. URL: <https://ml-schema.github.io/documentation/ML%20Schema.html>.
- [12] J. Hernandez, D. Golpayegani, D. Lewis, An open knowledge graph-based approach for mapping concepts and requirements between the EU AI Act and international standards, AI and Ethics (2025). doi:10.1007/s43681-025-00708-6.
- [13] S. Neumaier, T. Dam, F. Kovac, AIDOC-AP: An application profile for technical documentation of AI systems, Semantic Web Journal (2026). Under review. Resource available at <https://w3id.org/aidoc-ap/>.
- [14] D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, F. M. Couto, The agreementmakerlight ontology matching system, in: R. Meersman, H. Panetto, T. Dillon, J. Eder, Z. Bellahsene, N. Ritter, P. De Leenheer, D. Dou (Eds.), On the Move to Meaningful Internet Systems: OTM 2013 Conferences, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 527–541.
- [15] H. Babaei Giglou, J. D'Souza, O. Karras, S. Auer, Ontoaligner: A comprehensive modular and robust python toolkit for ontology alignment, in: E. Curry, M. Acosta, M. Poveda-Villalón, M. van Erp, A. Ojo, K. Hose, C. Shimizu, P. Lisena (Eds.), The Semantic Web, Springer Nature Switzerland, Cham, 2025, pp. 174–191.