Complex Ontology Alignment using LLMs: A Case Study

Adrita Barua[1,*], Reza Amini[2,*], Sanaz Saki Norouzi[1,*], Reihaneh Amini[1,*] and Pascal Hitzler[1,*]

[1]*Kansas State University, Manhattan, KS, USA*
[2]*University of Arkansas, Fayetteville, AR, USA*

Abstract
Ontology alignment is a key task in the semantic web with the goal of finding the semantic correspondences between two ontologies. While most existing approaches focus on simple (1-to-1) matching, complex matching consisting of m-to-n relationships between ontologies remains more challenging. Previous results [1] have indicated that complex ontology alignment through LLM prompting assistance may be possible if the input ontologies are appropriately modular – they also indicated that LLM prompting fails to provide reasonable outputs in the absence of such module information. As this previous study has only looked at one dataset, we herein provide a replication study on a completely different dataset, and the results support the usefulness of modules information in ontology matching.

Keywords
Complex ontology alignment, LLMs, Modular ontologies

1. Introduction

Ontology alignment/Ontology matching (OM) refers to the task of aligning semantically similar concepts among diverse knowledge systems. Ontologies serve as a key building block for applications, and in particular often serve as schema for knowledge graphs (KGs). However, each system often adopts its own schema and vocabulary. As a result, identifying correspondences between entities in different ontologies is essential across multiple domains for data integration.

Ontology alignment has been studied for a long time, and many alignment approaches and systems have been developed. However, the majority of these systems are designed to detect only "simple" 1-to-1 mappings between ontologies, typically by establishing equivalence relationships between classes (unary predicates) or between properties (binary predicates). While simple mappings provide value, they often fall short for data integration tasks, which demand mappings expressed as complex rules.[1] Unfortunately, automatically discovering such complex alignments remains challenging, as it requires significant domain expertise and manual labor. Any effort to automate or partially automate this process would therefore yield significant benefits.

Recent investigations have revealed a growing interest in ontology alignment using large language models (LLMs) [2, 3]. A core challenge in ontology alignment is that many ontologies are underspecified and lack the internal structure that could support self-explanation. It has been argued that adding internal structure, such as conceptual 'ontology modules', could facilitate ontology engineering tasks that are difficult to automate [4]. Building on this conceptual framework, [1] adopts a modular architecture to generate complex alignments, demonstrating notably improved outcomes when richer, module-based content is available. In this paper, we explore whether the use of modular information, as proposed in [1], remains effective in another use case when the dataset is different. Our goal is to assess the

OM 2025: The 20th International Workshop on Ontology Matching collocated with the 24th International Semantic Web Conference (ISWC 2025), November 2nd, 2025, Nara, Japan
*Corresponding author.
adrita@ksu.edu (A. Barua); ramini@walton.uark.edu (R. Amini); sanazsn@ksu.edu (S. Saki Norouzi); reihanea@ksu.edu (R. Amini); hitzler@ksu.edu (P. Hitzler)
https://github.com/AdritaBarua (A. Barua); https://people.cs.ksu.edu/~hitzler/ (P. Hitzler)
0000-0002-3287-7443 (A. Barua); 0009-0008-1584-7657 (R. Amini); 0009-0003-4441-108X (S. Saki Norouzi); 0000-0001-6192-3472 (P. Hitzler)

[1]See Section 4 for an example mentioned in 1.

robustness of this approach and identify opportunities for further improvement. In summary, as we will see, the results from [1] do carry over to the new dataset.

2. Related Work

Ontology alignment is an important part of the semantic web field, where finding matching entities between two different ontologies is important, which would help with knowledge discovery and knowledge sharing tasks [5]. Although extensive work has been done on ontology matching using methods such as similarity checking and fuzzy lexical matching [6], it is now essential to explore how LLMs can facilitate this task. In this section, we briefly review recent papers that utilize LLMs for ontology matching.

As initial steps toward applying LLMs to OM tasks through prompt engineering, [3] and [7] demonstrated the potential and challenges of zero-shot prompting on Ontology Alignment Evaluation Initiative (OAEI) datasets. [2] proposed a method that generates candidate ontology alignments using embeddings and then employs an LLM to perform the matching and make binary decisions on the OAEI datasets. Similarly, [8] introduced an agent-based approach for the retrieval and matching processes on the same datasets. [9] proposed a Retrieval Augmented Generation (RAG)-based approach using different LLMs, evaluated on 20 OAEI datasets. They also developed OntoAligner, a toolkit that combines traditional OM techniques with LLMs [10].

These studies primarily focus on simple matching, while only a few have addressed complex ontology alignment. For example, [1] proposed a Chain-of-Thought (CoT) based method that uses modular information, evaluated on the GeoLink dataset [11], which we use as the baseline of our work. [12] introduced an approach that integrates SPARQL query patterns with LLM-based validation for 1-to-n matchings. In another study, [13] proposed a novel approach that combines SPARQL-based subset extraction for both ontologies with prompt engineering to generate complex matchings in the Expressive and Declarative Ontology Alignment Language (EDOAL) format; this was tested on the GeoLink and Conference datasets. The same authors [14] further extended the CANARD system [15], which originally used SPARQL queries and subgraph matching, by incorporating LLM-based embeddings in multiple steps on the Conference dataset.

3. Generating Alignment Rules

In this study, we used a similar approach to that described in [1] to evaluate whether their use of modular information for ontology matching remains effective in other test cases. In particular, the modular information we used is based on the Modular Ontology Modeling (MOMo) methodology [4]. Here, a module is defined as a part of the ontology (i.e., a subset of the ontology axioms) that captures a key notion along with its key attributes, as a human expert would conceptualize it. In this paradigm, modules are defined by the ontology creators during the modeling process.

For this work, we aimed to generate alignment rules using LLMs for the Enslaved OAEI Complex Alignment benchmark [16].[2] This benchmark is non-synthetic—that is, it is based on a real-world data deployment scenario in which the Enslaved ontology [17] was used as the basis for developing a data deployment on a Wikibase installation, namely the Enslaved Hub.[3] Wikibase is the software underlying Wikidata, and it can also be used independently for knowledge graph creation and management. The Enslaved ontology was modeled using the MOMo methodology and serves as the schema for this knowledge graph. The Enslaved Hub is a centralized platform for engaging with historical slave-trade data from various sources. Its deployment on the Wikibase platform makes the data available in RDF through standard Wikibase interfaces; however, Wikibase imposes limitations on the use of RDF (see, e.g., [18]), and the Enslaved ontology could not be used as-is with Wikibase. As a result, the RDF export from the Enslaved.org hub differs significantly in structure from the Enslaved ontology ABox (i.e., the

[2]See also https://oaei.ontologymatching.org/2021/complex/index.html#popenslaved
[3]https://enslaved.org

RDF that uses the Enslaved ontology as a schema). This discrepancy gives rise to a *natural* complex alignment, which was captured in the benchmark reported in [16]. We used this benchmark as our baseline for evaluation.

First, we presented the module information of the Enslaved ontology by listing all axioms for each module (see [17] for detailed axiom information). For example, all the axioms for a particular module, such as the Age Record Module, are included in the module file (see Figure 2). For the Enslaved ontology, a total of 13 modules were given in the module file. Next, we extracted the triples from the Wikibase ontology's `.ttl` file and prompted the LLM to determine alignment rules for each Wikibase triple based on the Enslaved ontology's module information. Each `.ttl` file contains the triples relevant to the right-hand side of the alignment rule. For example, for the alignment rule shown in Equation 1, the corresponding `.ttl` file contains triples related to the three Wikibase entities mentioned in the rule (see Figure 3). We had a total of 124 such rules, and each `.ttl` file contained all the triples related to its respective alignment rule, which were then passed to the LLM iteratively. We invoked GPT-4o [19] via the OpenAI API with `temperature=0` and `top_p=1`. This prompt structure follows the prompting workflow from earlier work [1], ensuring that the structure remains consistent and supporting reproducibility. We use a zero-shot prompt, so no example format of the alignment rule was provided to the LLM. As a result, the LLM produces a discussion of the alignment rule between the two ontologies in the output instead of just giving the final rule (see Figure 4). This helps us determine where the LLM makes mistakes and gain insights from them. The prompt is as follows:

```
We have two ontologies: Enslaved and Wikibase.
We need to find the complex ontology alignment rules between these two ontologies.
All the information regarding the modules and patterns of the Enslaved ontology is provided here:
{module_file_read}
Now, consider the following triples from the Wikibase ontology:
Wikibase triple:
{WikiBase_ttl_read}
Now, find the alignment rules for the given Wikibase triples with respect to the Enslaved ontology,
     based on the module information provided. Provide the output in the format of Alignment Rules: the
      generated alignment rules.
```

Figure 1: Prompt Example

AgeRecord
Axioms:
(1) $AgeRecord \sqsubseteq AgentRecord$
(2) $AgeRecord \sqsubseteq \leq 1\ hasValue.AgeCategory$
(3) $AgeRecord \sqsubseteq \leq 1\ hasAgeValue.xsd{:}double$
(4) $AgeRecord \sqsubseteq \exists hasValue.AgeCategory \sqcup \exists hasAgeValue.xsd{:}double$
(5) $hasAgeRecord \sqsubseteq hasPersonRecord$

Figure 2: Example format of the `AgeRecord` module.

The LLM will try to match the Wikibase entities mentioned in the `.ttl` file with the modules given in the Enslaved module file and come up with the result file as an output. For example, for the given `.ttl` file mentioned in Figure 3, the output of the prompt is shown in Figure 4.

4. Evaluation and Results

After generating the alignment rules for each `.ttl` file in the Wikibase ontology we evaluated them using the reference alignment ([16]).

```
### https://lod.enslaved.org/entity/Q410
ed:Q410   rdf:type   owl:Class ;
schema:description   "A person. Subclass of Agent" ;
rdfs:label   "Person" .

### https://lod.enslaved.org/prop/P42
ep:P42   rdf:type   owl:AnnotationProperty ;
schema:description   "property to obtain the age record of a person" ;
rdfs:label   "hasAge".

### http://wikiba.se/ontology#Statement
wikibase:Statement   rdf:type   owl:Class .
```

Figure 3: Example format of the Wikibase `.ttl` file.

To evaluate our approach, we applied precision and recall on key entities from the complex alignment rules. Recall measures the share of correctly detected Enslaved instances among all expected, while precision captures the accuracy of those detected. Together, these standard metrics provide a balanced view of the method's effectiveness in identifying Enslaved-related cases.

$$\text{Recall} = \frac{\text{Number of Correctly Detected Pieces}}{\text{Total Number of Enslaved Pieces in Complex Alignment}}$$

$$\text{Precision} = \frac{\text{Number of Correctly Detected Pieces}}{\text{Total Number of Detected Pieces}}$$

We manually analyzed 100 complex alignment rules, focusing not on the full rule outputs but on whether the key components (predicates) were detected. While detecting these pieces (without actual composition of the pieces into a rule) is simpler than generating full rules, it reflects the core challenge of complex ontology alignment. Once the correct pieces are identified, assembling the complete rule is straightforward for a human or even a symbolic algorithm using the ontology and example data. More details on the evaluation of each rule are recorded in the *evaluation.csv* file here. [4]

In evaluating recall, we found that LLM accurately identified all three expected Enslaved entities in our example, yielding a recall of 1.0:

$$\begin{aligned} &\texttt{enslaved:Person}(x) \wedge \texttt{enslaved:hasAgeRecord}(x, y) \wedge \texttt{enslaved:AgeRecord}(y) \\ &\leftrightarrow \texttt{ed:Q410(Person)}(x) \wedge \texttt{ep:P42(hasAge)}(x, y) \wedge \texttt{wikibase:Statement}(y) \end{aligned} \tag{1}$$

For precision, LLM returned only the correct entities, with no irrelevant additions, resulting in a precision of 1.0. Notably, if extra entities such as "AgeValue" or "Place" had been included, precision would have dropped, as these would count as incorrect detections under the alignment rule. In practice, returning many unnecessary entities would also make human assembly of the final rule more difficult.

Table 1
Recall and Precision for detected Enslaved entities and GMO entities listed in [1] using module information

	Recall			Precision		
	≥ 0.5	≥ 0.75	$= 1$	≥ 0.5	≥ 0.75	$= 1$
Enslaved Entities	72.0%	51.0%	37.0%	69.0%	43.0%	33.0%
GMO Entities	73.3%	62.3%	45.0%	69.7%	59.6%	45.8%

As shown in Table 1, 72% of the complex alignment rules achieved a recall above 0.5, meaning more than half of the Enslaved entities per rule were successfully detected. Notably, recall exceeded 0.75 for about 50% of the rules, and a perfect recall of 1.0 was reached in 37% of cases, thanks to the integration

[4]See https://github.com/AdritaBarua/Complex-Ontology-Alignment-using-LLMs for supplementary materials for replication.

To align the given Wikibase triple with the Enslaved ontology, we need to map the concepts and properties from the Wikibase ontology to the corresponding concepts and properties in the Enslaved ontology. Let's break down the given Wikibase triple and find the corresponding elements in the Enslaved ontology.

Wikibase Triple Breakdown:

- **Class: Person**
  - URI: `http://example.org/entity/Q410`
  - Label: `"Person"`
- **Object Property: hasAgeRecord**
  - URI: `http://example.org/prop/P42`
  - Domain: `http://example.org/entity/Q410` (Person)
  - Range: `http://wikiba.se/ontology#Statement`
- **Class: Statement**
  - URI: `http://wikiba.se/ontology#Statement`

Corresponding Elements in the Enslaved Ontology:

- **Class: Person**
  In the Enslaved ontology, `Person` is a subclass of `Agent`:
  Axiom: `Person` ⊑ `Agent`
- **Object Property: hasAgeRecord**
  In the Enslaved ontology, `hasAgeRecord` is a subproperty of `hasPersonRecord`:
  Axiom: `hasAgeRecord` ⊑ `hasPersonRecord`
- **Class: Statement**
  In the Enslaved ontology, the concept of a record (like `AgeRecord`) is a subclass of `AgentRecord`:
  Axiom: `AgeRecord` ⊑ `AgentRecord`

Alignment Rules:
Based on the above analysis, we can generate the following alignment rules:

- **Class Alignment:**
  Align the Wikibase `Person` class with the Enslaved `Person` class:
  `http://example.org/entity/Q410` ≡ `Person`
- **Property Alignment:**
  Align the Wikibase `hasAgeRecord` property with the Enslaved `hasAgeRecord` property:
  `http://example.org/prop/P42` ≡ `hasAgeRecord`
- **Class Alignment:**
  Align the Wikibase `Statement` class with the Enslaved `AgentRecord` class, as `AgeRecord` is a subclass of `AgentRecord`:
  `http://wikiba.se/ontology#Statement` ≡ `AgentRecord`

Final Output:

```
Alignment Rules:
1. http://example.org/entity/Q410 ≡ Person
2. http://example.org/prop/P42 ≡ hasAgeRecord
3. http://wikiba.se/ontology#Statement ≡ AgentRecord
```

These alignment rules map the concepts and properties from the Wikibase ontology to the corresponding elements in the Enslaved ontology, based on the provided module information.

Figure 4: The LLM output to align the Wikibase triples with the Enslaved ontology.

of module information. These results closely match those reported in the original study [1], reinforcing the validity of the approach, which is mentioned in the second row of the Table 1.

For precision, which reflects how accurately the responses pointed to expected Enslaved entities, we found that 69% of the evaluated records achieved precision above 0.5. And 43% of the entities obtain a precision of 0.75 or more.

These numbers provide strong support for the approach proposed in the reference paper [1], as we successfully replicated it on new data.

The slight variations in our evaluation metrics compared to the original paper (GMO entities in Table 1) can be attributed to several factors. Notably, the naming conventions in Wikibase data tend to be more abstract and code-based, which introduces additional complexity in alignment. Furthermore, the alignments in our dataset are generally longer, potentially increasing the difficulty of precise matching. Despite these challenges, the overall results remain consistent and demonstrate the robustness of the original approach across varied data contexts.

5. Conclusion and Future Work

This paper presents a case study that shows that the use of module information in a complex ontology alignment task yields results comparable to those reported by [1] when applied in a different scenario. Since complex ontology alignment is a critical task, and it is important to assess whether our approach remains effective when applied to other use cases, a case study was necessary before attempting to generalize the system in an automated way, especially given that the results currently require manual evaluation. Our findings indicate the robustness of the method across different datasets and show the potential of leveraging LLMs for a challenging task like complex ontology matching at scale. It shows that having modular information about the underlying ontology significantly helps automate the matching process, as the LLM output shows how they used the module information as an anchor to map the related entities. Further improvements are needed to refine the system and develop an end-to-end architecture with improved accuracy.

Acknowledgments

The authors acknowledge partial funding under the National Science Foundation grant 2333782 "Proto-OKN Theme 1: Safe Agricultural Products and Water Graph (SAWGraph): An OKN to Monitor and Trace PFAS and Other Contaminants in the Nation's Food and Water Systems", and Kansas State University through the Game-changing Research Initiation Program (GRIP), as part of the "Towards a Global Food Systems Data Hub: Seeding the Center for Sustainable Wheat Production" project.

Declaration on Generative AI

During the preparation of this work, the author(s) used X-GPT-4 and Gramby in order to: Grammar and spelling check. 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.

References

[1] R. Amini, S. S. Norouzi, P. Hitzler, R. Amini, Towards complex ontology alignment using large language models, in: S. Tiwari, B. Villazón-Terrazas, F. Ortiz-Rodríguez, S. Sahri (Eds.), Knowledge Graphs and Semantic Web - 6th International Conference, KGSWC 2024, Paris, France, December 11-13, 2024, Proceedings, volume 15459 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 17–31. URL: https://doi.org/10.1007/978-3-031-81221-7_2.

[2] S. Hertling, H. Paulheim, Olala: Ontology matching with large language models, in: K. B. Venable, D. Garijo, B. Jalaian (Eds.), Proceedings of the 12th Knowledge Capture Conference 2023, K-CAP 2023, Pensacola, FL, USA, December 5-7, 2023, ACM, 2023, pp. 131–139. URL: https://doi.org/10.1145/3587259.3627571.

[3] Y. He, J. Chen, H. Dong, I. Horrocks, Exploring large language models for ontology alignment, in: I. Fundulaki, K. Kozaki, D. Garijo, J. M. Gómez-Pérez (Eds.), Proceedings of the ISWC 2023 Posters, Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with 22nd International Semantic Web Conference (ISWC 2023), Athens, Greece, November 6-10,

2023, volume 3632 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3632/ISWC2023_paper_427.pdf.

[4] C. Shimizu, K. Hammar, P. Hitzler, Modular ontology modeling, Semantic Web 14 (2023) 459–489. URL: https://doi.org/10.3233/SW-222886.

[5] P. Shvaiko, J. Euzenat, Ontology matching: State of the art and future challenges, IEEE Trans. Knowl. Data Eng. 25 (2013) 158–176. URL: https://doi.org/10.1109/TKDE.2011.253.

[6] M. Abd Nikooie Pour, A. Algergawy, P. Buche, L. J. Castro, J. Chen, A. Coulet, J. Cufi, H. Dong, O. Fallatah, D. Faria, et al., Results of the ontology alignment evaluation initiative 2023, in: P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, O. Hassanzadeh, C. Trojahn (Eds.), Proceedings of the 18th International Workshop on Ontology Matching co-located with the 22nd International Semantic Web Conference (ISWC 2023), Athens, Greece, November 7, 2023, volume 3591 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 97–139. URL: https://ceur-ws.org/Vol-3591/oaei23_paper0.pdf.

[7] S. S. Norouzi, M. S. Mahdavinejad, P. Hitzler, Conversational ontology alignment with chatgpt, in: P. Shvaiko, J. Euzenat, E. Jiménez-Ruiz, O. Hassanzadeh, C. Trojahn (Eds.), Proceedings of the 18th International Workshop on Ontology Matching co-located with the 22nd International Semantic Web Conference (ISWC 2023), Athens, Greece, November 7, 2023, volume 3591 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 61–66. URL: https://ceur-ws.org/Vol-3591/om2023_STpaper1.pdf.

[8] Z. Qiang, W. Wang, K. Taylor, Agent-om: Leveraging LLM agents for ontology matching, Proc. VLDB Endow. 18 (2024) 516–529. URL: https://www.vldb.org/pvldb/vol18/p516-qiang.pdf.

[9] H. B. Giglou, J. D'Souza, F. Engel, S. Auer, Llms4om: Matching ontologies with large language models, in: A. Meroño-Peñuela, Ó. Corcho, P. Groth, E. Simperl, V. Tamma, A. G. Nuzzolese, M. Poveda-Villalón, M. Sabou, V. Presutti, I. Celino, A. Revenko, J. Raad, B. Sartini, P. Lisena (Eds.), The Semantic Web: ESWC 2024 Satellite Events - Hersonissos, Crete, Greece, May 26-30, 2024, Proceedings, Part I, volume 15344 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 25–35. URL: https://doi.org/10.1007/978-3-031-78952-6_3.

[10] H. B. Giglou, J. D'Souza, O. Karras, S. Auer, Ontoaligner: A comprehensive modular and robust python toolkit for ontology alignment, CoRR abs/2503.21902 (2025). URL: https://doi.org/10.48550/arXiv.2503.21902. arXiv:2503.21902.

[11] L. Zhou, M. Cheatham, A. Krisnadhi, P. Hitzler, Geolink data set: A complex alignment benchmark from real-world ontology, Data Intell. 2 (2020) 353–378. URL: https://doi.org/10.1162/dint_a_00054.

[12] O. Zamazal, Towards pattern-based complex ontology matching using SPARQL and LLM, in: D. Garijo, A. L. Gentile, A. Kurteva, A. Mannocci, F. Osborne, S. Vahdati (Eds.), Joint Proceedings of Posters, Demos, Workshops, and Tutorials of the 20th International Conference on Semantic Systems co-located with 20th International Conference on Semantic Systems (SEMANTiCS 2024), Amsterdam, The Netherlands, September 17-19, 2024, volume 3759 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024. URL: https://ceur-ws.org/Vol-3759/paper8.pdf.

[13] G. H. S. Sousa, R. Lima, C. Trojahn, Towards generating complex alignments with large language models via prompt engineering, in: E. Jiménez-Ruiz, O. Hassanzadeh, C. Trojahn, S. Hertling, H. Li, P. Shvaiko, J. Euzenat (Eds.), Proceedings of the 19th International Workshop on Ontology Matching co-located with the 23rd International Semantic Web Conference (ISWC 2024), Baltimore, USA, November 11, 2024, volume 3897 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 43–56. URL: https://ceur-ws.org/Vol-3897/om2024_LTpaper4.pdf.

[14] G. Sousa, R. Lima, C. Trojahn, Complex ontology matching with large language model embeddings, CoRR abs/2502.13619 (2025). URL: https://doi.org/10.48550/arXiv.2502.13619. arXiv:2502.13619.

[15] É. Thiéblin, G. Sousa, O. Haemmerlé, C. Trojahn, CANARD: an approach for generating expressive correspondences based on competency questions for alignment, Semantic Web 15 (2024) 897–929. URL: https://doi.org/10.3233/SW-233521. doi:10.3233/SW-233521.

[16] L. Zhou, C. Shimizu, P. Hitzler, A. M. Sheill, S. G. Estrecha, C. Foley, D. Tarr, D. Rehberger, The enslaved dataset: A real-world complex ontology alignment benchmark using wikibase, in: M. d'Aquin, S. Dietze, C. Hauff, E. Curry, P. Cudré-Mauroux (Eds.), CIKM '20: The 29th ACM

International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020, ACM, 2020, pp. 3197–3204. URL: https://doi.org/10.1145/3340531.3412768.
[17] C. Shimizu, P. Hitzler, Q. Hirt, D. Rehberger, S. G. Estrecha, C. Foley, A. M. Sheill, W. Hawthorne, J. Mixter, E. Watrall, R. Carty, D. Tarr, The enslaved ontology: Peoples of the historic slave trade, J. Web Semant. 63 (2020) 100567. URL: https://doi.org/10.1016/j.websem.2020.100567.
[18] C. Shimizu, A. Eells, S. G. Estrecha, L. Zhou, P. Hitzler, A. M. Sheill, C. Foley, D. Rehberger, Ontology design facilitating wikibase integration - and a worked example for historical data, J. Web Semant. 82 (2024) 100823. URL: https://doi.org/10.1016/j.websem.2024.100823.
[19] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al., Gpt-4o system card, CoRR abs/2410.21276 (2024). URL: https://doi.org/10.48550/arXiv.2410.21276. arXiv:2410.21276.