Ontology Alignment Validation using LLM and KG

Abdoulaye Diallo[1], Claudia d'Amato[2] and Mouhamadou Thiam[3]

[1]*Gaston Berger University of Saint Louis Senegal*

[2]*University of Bari Aldo Moro,Italy*

[3]*University Iba Der Thiam of Thiès*

Abstract

This work explore the applicability of Large Language Models (LLMs) and knowledge graph (KGs) for validation of ontology alignements (OA) without using reference alignments. We propose a systematic solution, grounded on the exploitation of the power of LLMs and KGs, for automatic the validation of ontology alignments. We add ontology concepts and information from a knowledge graph to the prompt to improve LLM reasoning and provide correct validation while minimizing errors. We experimentally proved that our approach reduces significantly the need for human intervention in the validation process. We demonstrate the effectiveness of our solution with experiments on OAEI (Ontology Alignment Evaluation Initiative) campaign and additional alignment benchmarks.

Keywords

ontology matching, Large Language Model, Knowledge Graph, Ontology Alignment

1. Introduction

In the Semantic Web and, more in general, in data integration, the use of ontologies has become essential, as they enable formal representation and sharing of domain knowledge. However, with their diversity and complexity, Ontology Matching (OM) emerged as a key interoperability enabler dealing with the semantic heterogeneity problem.

OM aims at determining ontology alignments (OAs), that is, a set of correspondences between the semantically related entities (classes/properties/instances respectively) of the input ontologies. These correspondences enable the knowledge and data expressed with the matched ontologies to interoperate. OAs are exploited for various tasks, such as ontology merging, data interlinking, query answering or navigation over Knowledge Graphs (KGs) [1].

Although many OM tools have been developed over the years [2] with the goal of improving their overall effectiveness (in terms of precision and recall-score), validating the alignments that are returned by these systems remains a fundamental step. For this reason, the Ontology Alignment Evaluation Initiative (OAEI) has been established, over the years, as the major OM benchmark. Indeed, the OAEI provides **reference alignments** to be used as benchmarks to measure alignment quality. They represent a set of correspondences between ontological entities (classes, properties, instances) that are deemed to be correct either by a group of experts or by high-performing OM systems. A reference alignment is defined as a set of quadruples, each of them is composed of two concepts drawn from the ontologies (Subject Area and Topic), the value of a similarity measure ranging from 0 to 1, and the corresponding type of relation ($=, \sqsubseteq, \sqsupseteq$). A repository of correct correspondences (reference alignments) is made available and used to compare the results of alignment systems. In several cases (as for the Conference track[1]), two types of reference alignment are made available: a) *Certain reference alignment*, , where the confidence value of the match is equal to 1.0; b) *Uncertain reference alignment*, where the confidence

[1]https://oaei.ontologymatching.org/2024/conference/index.html

value reflects the degree of agreement among a group of twenty experts on the validity of the match. Despite these benchmarks being a fundamental resource for OM, they show some limitations:

Incomplete coverage: the reference alignments may not cover all possible matches returned by the systems which may identify an alignment that is not part of the reference alignment, which may also impact the system's accuracy.

Subjectivity and Uncertainty: experts may have different interpretation of elements within the (input) ontologies thus bringing to disagreements that affect the assessment of the reference alignment.

Continuous updating: reference alignments need updates due to ontological changes.

Dependence on application domains: alignments usually refer to domain ontologies for which finding experts would be non trivial in case of highly specialized domains.

While most of the OM research focuses on performance improvement of OM systems [2], rather limited attention has been devoted to operationalizing, benchmarking and evaluation of OAs [3].Over the past decades, many approaches to ontology alignment have been proposed [4][5][6][7]. These have led to the development of various validation techniques, ranging from the use of reference alignments to (logical) consistency checks sometimes involving also domain experts [8] [9][10] [11]. This paper aims at filling this important gap while overcoming the limitation of the existing benchmarks as discussed above. Particularly, we present the first solution, to the best of our knowledge, for (semi-) automatizing the assessment of OAs and limiting the need for experts to be involved in the validation process. By exploiting Large Language Models (LLMs) [12] jointly with KGs, we formalize a semi-automated process that is also able to support OA assessment in case of changes within ontologies. LLMs, which have shown their ability to generate and understand text [12], have recently been adopted in ontology engineering [13, 14, 15] and OM [3, 16, 17, 18, 19, 20] with encouraging results. This proved ability motivated the idea of exploiting LLMs for the validation of OAs. However, the use of LLMs is not without limitations, as they may suffer from hallucination[21], that is, they may generate answers that appear correct at first sight when they are, in fact, incorrect. When using LLMs for OM, different types of hallucination may occur [22]. For instance, for ontologies focusing on the 'conferences' domain, with polysemous words and it would not be able to differentiate between 'chair', as related to 'conference chair' and 'chair', as related to the object that is used to seat.

To alleviate this problem, we integrate suitable KGs [23] into the OA validation process. Particularly, in order to validate OAs, we adopt suitable prompt engineering techniques using KGs as external resources. Indeed, KGs are well-structured data sources that represent domain knowledge adopting a graph-based data model and that can be enriched with schema level information in the form of ontologies [1]. The exploitation of a KG when prompting (namely querying) LLMs for validating OAs will allow to provide: additional context [24] to the query, for an improved understanding of the OA to be analyzed; up-to-date knowledge, as provided by the KG; and reduce hallucination, as introduced above. The main contributions of our work are as follows: a) automation of the OA validation process using LLM and KG; b) domain independent OA validation and reduction of the need for domain experts in the process; c) limitation the need of updating reference alignments after changes within ontologies. We demonstrate the reliability of our approach using ontologies from the OAEI and free LLMs and compare our results with existing (OAEI) benchmarks. The rest of the paper is organized as follows: Section 3 presents related works. Section 4 illustrates our proposed OA evaluation pipeline while experiments are provided and discussed in Sect. 3. Conclusions and future research directions are drawn in Sect. 4
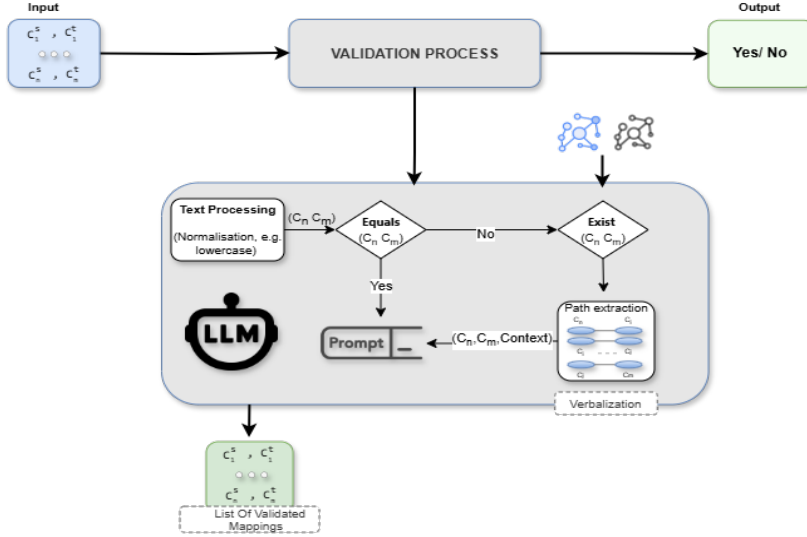
Figure 1: Pipeline Validation of Ontology Alignments using LLMs and KGs

2. Automatic Validation of Ontology Alignments via Large Language Models and Knowldge Graphs

We propose an automated ontology alignment validation pipeline that explores the power of LLMs coupled with KGs, by adopting suitable prompt engineering techniques. The overall pipeline is illustrated in Fig. 1. At the beginning of the process, candidate alignments (that is pairs of correspondences) are extracted and fed into the validation process, which consists of several steps. The first step is a text pre-processing (e.g. consisting in lowercase conversion and stemming) of the entities in the candidate alignment, in order to come up with normalized descriptions of these entities.

It is followed by the computation of the similarity value of the normalized entities. If the two entities result highly similar (with similarity value close to 1.0), a prompt is generated for inquiring the validation of the candidate alignment to the LLM. Alternatively, a check is performed in order to verify if the entities appear in the KG of reference and if there is a relational path between them. If so, the path is extracted and verbalized to be used as additional contextual information when prompting the LLM for inquiring the validation, jointly with the candidate entities. The LLM output is a binary Yes/No result to validate or reject the correspondence. Validated correspondences are added to a final list of alignments.

Please note that more articulated scenarios, such as the one in which only one entity is found in the KG, are not taken into account for the moment, since at this stage we are primarily interested in assessing the benefit of exploiting KGs. A more extensive discussion of the limitations and further improvements is provided in Sect. 4.

2.1. Text processing

This is the first phase in the validation process. It allows us to obtain the normalized representations of entities in the candidate alignment. A set of text processing techniques are used, which include:

- `Removal of special characters`: e.g., Program_Committee becomes ProgramCommittee).
- `Conversion to lowercase`: to ensure case consistency (e.g., ProgramCommittee becomes programcommittee).
- `Stemming`: to reduce words to their root form (e.g., Reviewes and isReviewing become review).
- `Removing stop words`: (non-informative words like "is", "the", "of") to focus on meaningful terms.

- `Expanding contractions or acronyms`: to their full form for clarity (e.g., PC becomes Program committee).

2.2. Similarity value

This step consists in measuring the similarity between pairs in a candidate alignment, after the text processing phase. We use the Jaro-Winkler [25] measure to assess the similarity between entities. The choise is motivated by the fact that it resulted well suited to short strings besides taking into account matching characters and common prefixes. The Jaro-Winkler meausure is defined as:

$$d_{\text{JW}} = d_{\text{Jaro}} + (l \cdot p \cdot (1 - d_{\text{Jaro}})) \tag{1}$$

where

- l: is the length of the common prefix (max 4 characters)
- p: is an adjustment factor (often 0.1)
- and d_{Jaro} is defined as

with

$$d_{\text{Jaro}} = \frac{1}{3} \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right) \tag{2}$$

- m: number of matching characters
- t: number of transpositions
- s1, s2: the two strings to be compared

This measure also assigns bonus scores when the initial characters of the strings are identical, thereby enhancing precision in cases of closely related labels. When the similarity value between the two entities is equal to 1, a specific prompt for both entities is formulated by filling in the information in the prompt template described in Sect. 2.4; otherwise, the validation process continues by exploiting the KG, as detailed in the next section.

2.3. Knowledge Graph

This step adds context for ontology alignment validation using a domain-specific Knowledge Graph (KG). The enrichment process via the KG requires linking entities from the source and target ontologies to entities in the KG. This operation, is performed as follows: 1) **Label Extraction:** The label corresponding to the ontology class (such as ProgramCommittee) is extracted. 2) **Keyword-based SPARQL Querying:** A SPARQL query is executed on the DBpedia endpoint to find resources whose label (rdfs:label) or name (foaf:name) matches the entity's label after text processing (lowercasing, stemming). We use a string similarity search built into DBpedia's SPARQL (bif:contains) to handle minor variations. The related triples are extracted if a relationship between them is discovered. The Graph2Text model [26] is then used to convert these triples into natural language, converting structured data into understandable sentences. Lastly, the generated prompt used for the final validation includes the verbalized context along with the two entities.

2.4. Prompt Generator

Specific prompt templates were crafted for entity pairs and their contextual representations involving triples from the knowledge graph. Regardless of the scenario, once the syntactic similarity is computed, the same prompt structure is reused. The prompting strategies employed include zero-shot, few-shot, and knowledge-enriched prompting to enhance reasoning capabilities.
In the zero-shot prompting setup, four alternative formulations were considered to evaluate concept similarity. Prompt 0 asked to *classify whether two given concepts are the same, given as "First concept: left, Second concept: right, Answer:".* Prompt 1 formulated the *task as an ontology matching problem in the*

conference domain, asking whether the two concepts are similar. Prompt 2 provided a shorter instruction:*Is left and right the same? requiring a yes or no answer.* Finally, Prompt 3 explicitly requested *to decide whether the two concepts match or not, with the answer constrained to yes or no.*

In the few-shot prompting setup, several validation pairs were used as illustrative examples. For Prompt 5, the pairs included 'Chairman' vs 'Chair' (answer: yes), 'Conference Chair' vs 'Session Chair' (answer: no), and 'Conference event' vs 'Conference activity '(answer: yes/no). In prompt 6, several correct and incorrect examples were also added before querying the LLM. In the KG-enriched prompting setup, additional contextual definitions from the knowledge graph were incorporated into the prompt. For instance, the concept "Chair" was defined as a *leadership role in an academic committee*, while "ProgramCommitteeChair" was *specified as a particular type of Chair responsible for organizing conference programs*. The model was then asked, based on this information, to determine whether the two concepts are equivalent.

2.5. LLM

The prompts generated in the previous steps are used to query LLM(s), e.g. GPT[27], LLama3[28] or DeepSeek[29] for the final validation of the candidate alignment. The LLM outputs is a Boolean response: if *yes*, the candidate alignment is validated; if *no* the candidate alignment is not validated.

3. Experiments

For our experiment, we used the 2021 OAEI Conference Track[30], which consists of 16 heterogeneous ontologies well-suited for ontology matching tasks. The ontology alignment tool used in this study was developed concurrently, which justifies the use of 2021 benchmarks. This methodology guarantees methodological coherence and facilitates a straightforward comparison with pre-existing baseline research. Among these, only 7 ontologies are included in the reference alignment: Cmt, ConfTool, Edas, Ekaw, Iasted, Sigkdd, and Sofsem. We selected version ra1[31] as the reference alignment, which is the original alignment where all match confidence values are 1.0. By focusing only on the ontologies covered by ra1, we ensured a direct comparison between the reference alignment and our proposed solution. For ontology matching, we used AML[32] (Alignment API) version 3.2 (2021 release) with a threshold set to 0.5 to generate a broader set of candidate pairs for validation. This configuration produced 375 candidate pairs across the seven ontologies, allowing us to better evaluate our approach. In the preprocessing phase, we applied NLP techniques using Python's NLTK and spaCy libraries to preprocess concept pairs, reducing them to their simplest forms. To enhance our prompts, we leveraged DBpedia[33] (queried via SPARQL Endpoint) and ConceptNet[2] for semantic enrichment. All triplets from the KG were processed with Graph2Text[26] before being added to the prompt. We used Ollama[3] to locally test multiple LLMs. We tested our approach with the following models: GPT4.0[4], Llama3[5], and Deepseek-r1[6].

The experiment is conducted as follows: A candidate pair (generated by AML 3.2) was considered correctly validated if : Our solution confirmed the match, and the pair existed in the reference alignment (ra1). Pairs failing either condition were deemed incorrect.

3.1. Results

With zero-shot the results are shown in the table 1. We observe that the LLama3 model for P0 and P2 obtains a very high result [97.5-98.3]%. The two prompts P1 and P3 obtain values equal to 92.5%. For the deepseek-r1 model, the results are also satisfactory, and the best prompts, such as P2, have a

---

percentage equal to 97.5%. With this model, the lowest validation rate is equal to 94.7. The chatGPT4.0 model remains very balanced with minimal variation between prompts. In this model, the minimum validation rate is around 95.5. An inter-model comparison shows the following statistics: LLama3 has the best score, while GPT4.0 is the most consistent and finally deepseek-r1 combines accuracy and stability.

Model / Prompt	P0(%)	P1(%)	P2(%)	P3(%)
LLama3	97.5	92.5	98.3	92.5
DeepSeek-r1	97.0	95.0	97.5	94.7
GPT-4.0	97.0	95.5	96.0	95.5

Table 1
Performance of our approach on the conference track with the zero-shot technique

Model / Prompt	P5 (%)	P6 (%)
LLama3	97.75	96.8
DeepSeek-r1	98.00	97.5
GPT-4.0	98.50	98.2

Table 2
Performance with Few-shot technique

Model/Prompt	P7 (%)
LLama3	97.45
DeepSeek-r1	98.60
GPT-4.0	98.40

Table 3
Performance with Knowledge enhanced prompting

In few-shot prompting with the addition of examples in the prompts, we obtained the results shown in the table 2.
For the few-shot technic, we tested our solution with prompt 5 and prompt 6. An overall analysis shows that the shot gives better results. We have stable and very high results, which is due to the addition of examples of correct and incorrect matches in the prompt.

After integrating data from a KG, we observed an improvement in the minimum coverage rate for all models: LLaMa3 from 96.80 to 97.45%, DeepSeek-r1 from 97.5 to 98.60% and finally GPT-4.0 from 98.2 to 98.40%. The results are shown in the table 3. Knowledge graphs provide semantically structured information. This result shows that coupling an LLM with a KG can improve understanding, accuracy, and scientific reasoning.

3.2. Discussion

The use of LLMs makes it possible to validate the alignments produced and reduces the need to update reference alignments. Our approach can also automate the validation of the alignments produced. It achieves results that are almost equal to the reference alignments. This automation paves the way for the scalability of the process of comparing ontology alignment approaches. By combining the reasoning capabilities of an LLM with explicit information from a KG, our system surpasses the capabilities of a domain expert in the validation process. The information from the KG compensates the lack of an expert. Our results demonstrate the reliability of our approach in the validation process. The use of a KG acts as a means of reducing hallucinations with factual data. The improvement in scores compared to zero-shot clearly shows that the KG-enriched prompt reduces errors.

4. Conclusion and future work

In this paper, we have presented a new approach that allows the validation of ontology alignments produced by any alignment system using an LLM combined with a knowledge graph. With an evaluation using a large number of alignments produced by our approach, our framework shows that LLMs, when combined with KGs and guided by prompt engineering techniques, can outperform benchmarks in

terms of reference. But Various problems need to be addressed including the cases where entities do not appear in the knowledge graph and where only one of the entities appears in the knowledge graph are not handled. There are also no specific knowledge graphs in certain domains, and compound concepts (e.g. ConferencePart) generally do not exist. In our experiments, we show that the proposed method achieves results that are almost similar to the proposed benchmarks. Furthermore, we intend to expand our methodology to manage the verification of intricate alignments, like situations in which *Reviewer ≡ Person AND (authorOfSomeReview).*

5. Acknowledgments

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly in order to grammar and spell check, and improve the text readability. After using the tool, the authors reviewed and edited the content as needed to take full responsibility for the publication's content.

References

[1] A. Hogan, C. Gutierrez, M. Cochez, G. de Melo, S. Kirrane, A. Polleres, R. Navigli, A.-C. Ngonga Ngomo, S. M. Rashid, L. Schmelzeisen, S. Staab, E. Blomqvist, C. d'Amato, J. E. Labra Gayo, S. Neumaier, A. Rula, J. Sequeda, A. Zimmerman, Knowledge Graphs, number 22 in Synthesis Lectures on Data, Semantics, and Knowledge, Springer, 2022. URL: https://kgbook.org/. doi:10.2200/S01125ED1V01Y202109DSK022.

[2] P. H. Portisch J, Hladik M, Background knowledge in ontology matching: A survey., Semantic Web (2022) 2639–2693. doi:10.3233/SW-223085.

[3] G. Macilenti, A. Stellato, M. Fiorelli, Prompting is not all you need evaluating gpt-4 performance on a real-world ontology alignment use case, Procedia Computer Science 246 (2024) 1289–1298.

[4] P. Shvaiko, J. Euzenat, Ontology matching: state of the art and future challenges, IEEE Transactions on knowledge and data engineering 25 (2011) 158–176.

[5] P. Ochieng, S. Kyanda, Large-scale ontology matching: State-of-the-art analysis, ACM Computing Surveys (CSUR) 51 (2018) 1–35.

[6] S. Anam, Y. S. Kim, B. H. Kang, Q. Liu, Review of ontology matching approaches and challenges, International Journal of Computer Science and Network Solutions 3 (2015) 1–27.

[7] E. Thiéblin, O. Haemmerlé, N. Hernandez, C. Trojahn, Survey on complex ontology matching, Semantic Web 11 (2020) 689–727.

[8] Z. Dragisic, V. Ivanova, P. Lambrix, D. Faria, E. Jiménez-Ruiz, C. Pesquita, User validation in ontology alignment, in: International Semantic Web Conference, Springer, 2016, pp. 200–217.

[9] S. Zhang, O. Bodenreider, Lessons learned from cross-validating alignments between large anatomical ontologies, Studies in Health Technology and Informatics 129 (2007) 822.

[10] L. Zhou, E. Thiéblin, M. Cheatham, D. Faria, C. Pesquita, C. Trojahn, O. Zamazal, Towards evaluating complex ontology alignments, The Knowledge Engineering Review 35 (2020) e21.

[11] E. Beisswanger, U. Hahn, Towards valid and reusable reference alignments—ten basic quality checks for ontology alignments and their application to three different reference data sets, Journal of biomedical semantics 3 (2012) S4.

[12] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, J. Gao, Large language models: A survey, arXiv preprint arXiv:2402.06196 (2024).

[13] J. Li, D. Garijo, M. Poveda-Villalón, Large language models for ontology engineering: A systematic literature review (2025).

[14] P. Mateiu, A. Groza, Ontology engineering with large language models, in: 2023 25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), IEEE, 2023, pp. 226–229.

[15] B. Zhang, V. A. Carriero, K. Schreiberhuber, S. Tsaneva, L. S. González, J. Kim, J. de Berardinis, Ontochat: a framework for conversational ontology engineering using language models, in: European Semantic Web Conference, Springer, 2024, pp. 102–121.

[16] H. Babaei Giglou, J. D'Souza, F. Engel, S. Auer, Llms4om: Matching ontologies with large language models, in: European Semantic Web Conference, Springer, 2024, pp. 25–35.

[17] Y. He, J. Chen, H. Dong, I. Horrocks, Exploring large language models for ontology alignment, arXiv preprint arXiv:2309.07172 (2023).

[18] Z. Qiang, W. Wang, K. Taylor, Agent-om: Leveraging llm agents for ontology matching, arXiv preprint arXiv:2312.00326 (2024).

[19] J. Sampels, S. Efeoglu, S. Schimmler, Exploring prompt generation utilizing graph search algorithms for ontology matching, in: Knowledge Graphs in the Age of Language Models and Neuro-Symbolic AI, IOS Press, 2024, pp. 2–19.

[20] O. Zamazal, Towards pattern-based complex ontology matching using sparql and llm, in: Proceedings of the 20th International Conference on Semantic Systems (SEMANTiCS 2024), SEMANTiCS, Amsterdam, Netherlands, 2024.

[21] V. Rawte, A. Sheth, A. Das, A survey of hallucination in large foundation models, arXiv preprint arXiv:2309.05922 (2023).

[22] Z. Qiang, K. Taylor, W. Wang, J. Jiang, Oaei-llm: A benchmark dataset for understanding large language model hallucinations in ontology matching, arXiv preprint arXiv:2409.14038 (2024).

[23] G. Agrawal, T. Kumarage, Z. Alghamdi, H. Liu, Can knowledge graphs reduce hallucinations in LLMs? : A survey, in: K. Duh, H. Gomez, S. Bethard (Eds.), Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 3947–3960. URL: https://aclanthology.org/2024.naacl-long.219/. doi:10.18653/v1/2024.naacl-long.219.

[24] Y. Li, R. Zhang, J. Liu, An enhanced prompt-based llm reasoning scheme via knowledge graph-integrated collaboration, in: International Conference on Artificial Neural Networks, Springer, 2024, pp. 251–265.

[25] W. E. Winkler, The state of record linkage and current research problems, Statistical Research Division, US Bureau of the Census, Wachington, DC (1999).

[26] G. Amaral, O. Rodrigues, E. Simperl, Prove: A pipeline for automated provenance verification of knowledge graphs against textual sources, Semantic Web 15 (2024) 2159–2192.

[27] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., Training language models to follow instructions with human feedback, Advances in neural information processing systems 35 (2022) 27730–27744.

[28] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al., The llama 3 herd of models, arXiv e-prints (2024) arXiv–2407.

[29] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, et al., Deepseek-v3 technical report, arXiv preprint arXiv:2412.19437 (2024).

[30] ConferenceTrack, Conferencetrack, 2024. URL: https://oaei.ontologymatching.org/2024/conference/index.html, accessed on July 24.

[31] O. Zamazal, V. Svátek, The ten-year ontofarm and its fertilization within the onto-sphere, Journal

of Web Semantics 43 (2017) 46–53.

[32] D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, F. M. Couto, The agreementmakerlight ontology matching system, in: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", Springer, 2013, pp. 527–541.

[33] DBpedia, Dbpediakg, 2024. URL: https://www.dbpedia.org/resources/knowledge-graphs/, accessed on July 16.