LLMs for Ontology Engineering: A landscape of Tasks and Benchmarking challenges

Daniel Garijo[1,*], María Poveda-Villalón[1], Elvira Amador-Domínguez[1], ZiYuan Wang[1], Raúl García-Castro[1] and Oscar Corcho[1]

[1]*Universidad Politécnica de Madrid, Madrid, Spain*

Abstract

Large Language Models (LLMs) have emerged as a powerful technology for text generation tasks, showing promise in supporting the Ontology Engineering (OE) process. In this paper, we review current research on applying LLMs to OE tasks, aiming to identify commonalities and gaps in the state of the art. We categorize these efforts using the Linked Open Terms (LOT) methodology, characterizing them by their input and expected output. From this analysis, we highlight key challenges when creating benchmarks to evaluate LLM performance in OE tasks.

Keywords

Large Language Models, Ontology Engineering, Benchmark, Challenges

1. Introduction

Ontologies are a key component of Knowledge Engineering for integrating, validating and reasoning with data in Knowledge Graphs [1]. However, developing ontologies is a challenging and time consuming task. According to existing methodologies for ontology development [2], Knowledge Engineers should follow an iterative process to 1) distill the knowledge of the target domain by interviewing experts and understand their data-driven requirements, 2) implement a shared conceptualization by assessing existing standard ontologies described in the domain and validating it against the requirements, 3) make the ontology available on the web in both human and machine-readable manner, and 4) assess and maintain the ontology by addressing any new requirements that may arise from its use. While different tools have been developed by the scientific community to assist in the Ontology Engineerning process (e.g., for formalizing tests to assess requirements [3], creating human-readable documentation [4], ontology assessment [5, 6], etc.) a significant manual effort is still required from knowledge engineers for conceptualizing, reusing and validating existing ontologies.

In recent years, Large Language Models (LLMs) [7, 8, 9] have emerged as a disruptive AI technology for text generation tasks. On the one hand, LLMs have revolutionized the state of the art by providing impressive results in challenging AI tasks such as code generation [10], question answering [11] or text summarization [7], and becoming easy to adapt as chat bots such as ChatGPT.[1] On the other hand, LLMs have limited reasoning skills [12], hallucination problems (i.e., producing inaccurate answers and information) [13], lack transparency when providing results [14] and present bias problems [15].

A number of works have started using LLMs for aiding developers in ontology engineering tasks (e.g., proposing competency questions [16], learning ontologies from text [17], aligning concepts to existing taxonomies [18], etc.). However, the tasks addressed in existing works are usually defined in an heterogeneous manner, with different scope, inputs and expected outputs. In this paper we provide an overview of existing Ontology Engineering tasks addressed in the state of the art and map them
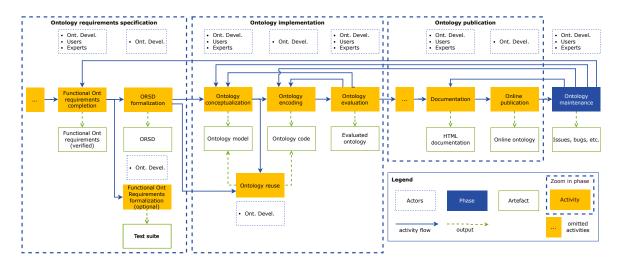
[1]https://chat.openai.com/

Figure 1: An overview of the LOT methodology, showing most relevant phases and activities.

against the different phases described in the Linked Open Terms (LOT) methodology [2]. In addition, we characterize each different task by their expected input and output. Our work helps characterizing existing tasks, describes existing gaps and discusses the main challenges when creating reference benchmarks for evaluation.

2. Mapping LLM for OE Tasks to the LOT methodology

Ontology development projects may involve a substantial number of activities. For example, the NeOn methodology identifies 10 processes and 49 activities involved in ontology engineering [19]. Some activities are carried out almost in every ontology development, for example ontology evaluation, while others are needed only in some specific cases, for example ontology customization. Ontology development methodologies, like Linked Open Terms (LOT), NeOn , METHONTOLOGY [20], DILIGENT [21] and SAMOD [22] among others, orchestrate the execution of ontology engineering activities in a guided way that usually involves requirements specification (written as Competency Question or affirmative statements), implementation and evaluation as core steps. We take the LOT workflow as basis for our analysis as it considers the core steps from traditional methodologies and extends them with additional steps for ontology publication and maintenance. Figure 1 depicts the core phases included in LOT (requirements specification, implementation, publication and maintenance) and main activities included in each phase. Boxes entitled "…" in Figure 1 group activities from the methodology that are not considered for this analysis due to the nature of the activity, e.g., proposing a candidate release to be published, or describing the purpose of the ontology.

In order to categorize existing works using LLMs for OE tasks, we reviewed and mapped each approach to the activities from the LOT methodology. During this process, we considered which input is given to the LLM (or system described in each work) and which output is expected by each approach, so as to select the corresponding LOT activity and characterize each task. For example, if the input is a set of competency questions and the output is OWL code, the task would be mapped under "Ontology encoding", while if the output is a diagram or other conceptualization formalism, it would be mapped to "Ontology conceptualization". Table 1 shows the results of our mapping, outlining the input and output of each approach.

As described in Table 1, most of the existing efforts on applying LLMs to OE are focused on the initial stages: requirement specification and implementation. Regarding requirements specification, existing approaches focus mostly on generating Competency Questions from text, like Kommineni et al. [26] and Antia and Keet [28]. Other works, e.g., Alharbi et al. [23], Rebboud et al. [24] and Ciroku et al. [25],

LOT Phase	LOT Task	Resource	Input	Output
Requirements Specification	Purpose, scope, and non-functional requirement writing	-		
	Functional requirements writing	Alharbi et al. [23]	Ontology file	CQs
	CQ writing	Rebboud et al. [24]	Ontology file	
		Ciroku et al. [25]		
		Kommineni et al. [26]	Text	
		Zhang et al. [27]	User story	
		Antia and Keet [28]	Text and CLaRO templates	
	Requirement improvement	-		
	Requirement formalization	Tufek et al. [29]	CQs	SPARQL Queries
Implementation	Reuse	Lopes et al. [18]	Term, informal definition and domain entity	Corresponding class in top-level ontology
		Hertling and Paulheim [30]	Ontologies and Text	Ontology Mappings
	Conceptualization	Babaei Giglou et al. [31]	Source text and terminologies	Taxonomy, relationships, and axioms
		Toro et al. [32]	Partially completed ontology term	JSON/YAML object with logical definitions and relationships
	Encoding	Mateiu and Groza [33]	Text	Ontology file
		Köhler and Neuhaus [34]	Text	
		Doumanas et al. [35]	Text (ORSD)	
		Saeedizade and Blomqvist [36]	CQs	
		Kommineni et al. [26]		
		Caufield et al. [37]	Schema and Text	
		Amini et al. [38]	Ontology file and additional information	
	Evaluation	-		
Publication	Documentation	Toro et al. [32]	Ontology term	Term definition
	Online publication	-		
Maintenance		-		

Table 1

Categorization of existing resources regarding the use of LLMs in Ontology Engineering per LOT phase and task. Phases with no existing resources are denoted with '-'.

analyzed the opposite activity, that is, extracting CQs from existing ontologies, which may be applied during a reverse ontological engineering process. It can also be observed that Saeedizade and Blomqvist [36] use LLMs to formalize ontological requirements as SPARQL queries.

Regarding the activities involved in the ontology implementation phase, we observe that most works aim to facilitate the ontology encoding activity using LLMs to generate ontology OWL files from text or CQs (e.g., Mateiu and Groza [33], Doumanas et al. [35], Saeedizade and Blomqvist [36], Kommineni et al. [26], Caufield et al. [37], Amini et al. [38] and Köhler and Neuhaus [34]). Other works, such as Babaei Giglou et al. [31], focus on a previous step by generating conceptualizations from texts and terminologies. In addition, some approaches, like Lopes et al. [18] and Hertling and Paulheim [30], have explored the use of LLMs for assisting the ontology matching activity, which may also be useful for ontology reuse (i.e., helping identify candidate terms in existing ontologies). Finally, one work was identified to aid ontology documentation: Toro et al. [32] present an LLM-powered ontology completion approach that contributes to the *ontology conceptualization* task, since it extracts relations and logical definitions for a given term, but it can also be categorized within *ontology documentation*, since it provides a definition for the target input term.

3. LLMs for Ontology Engineering tasks: Gaps and challenges

Table 1 illustrates some of the main **gaps** in the state of the art. Within the *requirement specification phase*, no approaches deal with non-functional requirement specification writing, an often neglected

task when building ontologies. Another important gap is on requirement improvement, where a new task may be derived from CQ writing in order to use LLMs to enhance a current set of requirements, given an initial set of CQs.

Next, the *implementation phase* has received most of the attention so far, especially on conceptualization and encoding of ontologies. However, there are two notable gaps. First, in the reuse phase no approaches have been presented to support ontology search, selection or adaptation by using LLMs. Second, no approaches cover ontology evaluation so far. While [29] generates SPARQL queries from CQs in order to validate an ontology against its instances, no approaches address the assessment of the ontology itself. New tasks may be proposed in order to feed the reports from ontology evaluation tools like OOPS! [5] or FOOPS! [6] to an LLM in order to translate their suggestions into changes in the ontology.

Finally, the *publication* and *maintenance* phases are barely addressed by the reviewed works, despite the potential of LLMs to contribute to tasks like generating ontology examples, proposing definitions and labels in multiple languages, or suggesting changes in an existing OWL file based on a new requirement.

Given the interest in the community to automatically assist OE activities with LLMs, it is becoming increasingly important to promote a common evaluation framework and benchmarks for OE in different OE tasks. Works such as Hertling and Paulheim [30] and Alharbi et al. [23] have started moving in this direction. We have identified three main challenges from the conducted review:

Challenge 1: Homogenizing OE Task Definitions. As shown in Table 1, different approaches tackle the same LOT tasks with different aims. For example, in CQ writing some approaches take plain text [23], while others take text and external templates [24]. Others reverse engineer the questions with an ontology file [23]. Therefore, the first challenge is to identify precisely each OE task, specifying the expected input and output for the LLM.

Challenge 2: Establishing common evaluation methods and metrics for OE tasks. Many tasks from Table 1 have a clear input and output, but may be challenging to evaluate. For example, for the ontology conceptualization and encoding tasks, the ontology models generated from LLMs may be compared to multiple valid representations instead of a single reference ontology. In addition, different similarity metrics may be used to consider similar entities in the ontology graph (e.g., if the LLM proposes object properties that are synonyms of a reference property) or the completeness of the proposed model (correct number of classes, properties and data properties). Similarly to the BLEU score [39] established in other areas like Machine Translation, new metrics may have to be developed to establish a fair result evaluation for OE tasks.

Challenge 3: Establishing OE task-specific curated benchmarks. In order to ensure a fair evaluation of LLM results in different tasks, different benchmarks must be defined and be adapted to the different inputs and outputs of the respective OE tasks. Some approaches have started working in this direction [23], e.g., by reusing existing CQ benchmarks like Coral [40], which collect hundreds on CQs across 14 ontologies. However, the quality among these CQs is heterogeneous (i.e., some CQs may be ambiguous, or not provide enough context) as they belong to different initiatives with no common set of practices or guidelines for their creation. In addition, existing resources may be subject to *contamination*, i.e., resources are ingested as pre-training or fine-tuned data by LLMs. New task-specific benchmarks should define common guidelines for curators, and ensure that a portion of the benchmark remains concealed from web crawlers (but available for full evaluations on demand).

4. Conclusions and Future work

In this paper we provide an overview of existing efforts using (L)LMs for Ontology Engineering tasks, grouping them according to the different phases described in the Linked Open Terms methodology and characterizing them in terms of the input and output each work proposes. Our efforts highlight unexplored areas where LLMs may be used to aid OE tasks (e.g., evaluation, documentation, maintenance) describing the main research challenges to be addressed in order to create reference benchmarks for each of these tasks. We believe that creating high-quality benchmarks will provide a common framework for

automated evaluation, promoting the use of LLMs for OE tasks and reducing human effort required in their evaluation. Extending the analysis started in this paper by including methodologies and activities for building Knowledge Graphs (or other types of ontology exploitation scenarios) will likely be highly valuable for assessing automated Knowledge Graphs construction processes.

5. Acknowledgments

References

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

[2] M. Poveda-Villalón, A. Fernández-Izquierdo, M. Fernández-López, R. García-Castro, LOT: An industrial oriented ontology engineering framework, Engineering Applications of Artificial Intelligence 111 (2022) 104755. doi:10.1016/j.engappai.2022.104755.

[3] A. Fernández-Izquierdo, R. García-Castro, Themis: a tool for validating ontologies through requirements, in: International Conference on Software Engineering and Knowledge Engineering, 2019. URL: https://api.semanticscholar.org/CorpusID:199571789.

[4] D. Garijo, WIDOCO: a wizard for documenting ontologies, in: International Semantic Web Conference, Springer, Cham, 2017, pp. 94–102. URL: http://dgarijo.com/papers/widoco-iswc2017.pdf. doi:10.1007/978-3-319-68204-4_9.

[5] 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.

[6] D. Garijo, O. Corcho, M. Poveda-Villalón, FOOPS!: An Ontology Pitfall Scanner for the FAIR Principles 2980 (2021). URL: http://ceur-ws.org/Vol-2980/paper321.pdf.

[7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of machine learning research 21 (2020) 1–67.

[8] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al., Palm: Scaling language modeling with pathways, Journal of Machine Learning Research 24 (2023) 1–113.

[9] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971 (2023).

[10] B. Roziere, J. Gehring, F. Gloeckle, S. Sootla, I. Gat, X. E. Tan, Y. Adi, J. Liu, R. Sauvestre, T. Remez, et al., Code llama: Open foundation models for code, arXiv preprint arXiv:2308.12950 (2023).

[11] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Curran Associates Inc., Red Hook, NY, USA, 2020.

[12] K. Valmeekam, A. Olmo, S. Sreedharan, S. Kambhampati, Large language models still can't plan (a benchmark for llms on planning and reasoning about change), in: NeurIPS 2022 Foundation Models for Decision Making Workshop, 2022.

[13] V. Rawte, A. Sheth, A. Das, A Survey of Hallucination in Large Foundation Models, 2023. URL: https://arxiv.org/abs/2309.05922. arXiv:2309.05922.

[14] G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, A. Cappelli, H. Alobeidli, B. Pannier, E. Almazrouei,

J. Launay, The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only, arXiv preprint arXiv:2306.01116 (2023).

[15] I. O. Gallegos, R. A. Rossi, J. Barrow, M. M. Tanjim, S. Kim, F. Dernoncourt, T. Yu, R. Zhang, N. K. Ahmed, Bias and fairness in large language models: A survey, Computational Linguistics (2024) 1–79.

[16] R. Alharbi, V. Tamma, F. Grasso, T. Payne, An experiment in retrofitting competency questions for existing ontologies, in: Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, SAC '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 1650–1658. URL: https://doi.org/10.1145/3605098.3636053. doi:10.1145/3605098.3636053.

[17] 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.

[18] A. Lopes, J. Carbonera, D. Schmidt, L. Garcia, F. Rodrigues, M. Abel, Using terms and informal definitions to classify domain entities into top-level ontology concepts: An approach based on language models, Knowledge-Based Systems 265 (2023) 110385. URL: https://www.sciencedirect.com/science/article/pii/S0950705123001351. doi:https://doi.org/10.1016/j.knosys.2023.110385.

[19] M. C. Suárez-Figueroa, A. Gómez-Pérez, M. Fernández-López, The NeOn Methodology framework: A scenario-based methodology for ontology development, Applied Ontology 10 (2015) 107–145.

[20] M. Fernández-López, A. Gómez-Pérez, N. Juristo, METHONTOLOGY: from ontological art towards ontological engineering, in: Proceedings of the Ontological Engineering AAAI97 Spring Symposium Series, American Asociation for Artificial Intelligence, 1997.

[21] H. Pinto, C. Tempich, S. Staab, Ontology engineering and evolution in a distributed world using diligent, in: S. Staab, R. Studer (Eds.), Handbook on Ontologies, International Handbooks on Information Systems, Springer Berlin Heidelberg, 2009, pp. 153–176.

[22] S. Peroni, A simplified agile methodology for ontology development, in: OWL: Experiences and Directions–Reasoner Evaluation: 13th International Workshop, OWLED 2016, and 5th International Workshop, ORE 2016, Bologna, Italy, November 20, 2016, Revised Selected Papers 13, Springer, 2017, pp. 55–69.

[23] R. Alharbi, V. Tamma, F. Grasso, T. Payne, An Experiment in Retrofitting Competency Questions for Existing Ontologies, in: Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, SAC '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 1650–1658. URL: https://doi.org/10.1145/3605098.3636053. doi:10.1145/3605098.3636053.

[24] Y. Rebboud, L. Tailhardat, P. Lisena, R. Troncy, Can LLMs generate competency questions?, in: Springer (Ed.), ESWC 2024, Extended Semantic Web Conference, Special Track on Large Language Models for Knowledge Engineering, 26-30 May 2024, Hersonissos, Greece, Hersonissos, 2024.

[25] F. Ciroku, J. de Berardinis, J. Kim, A. Meroño-Peñuela, V. Presutti, E. Simperl, RevOnt: Reverse engineering of competency questions from knowledge graphs via language models, Journal of Web Semantics 82 (2024) 100822. URL: https://www.sciencedirect.com/science/article/pii/S1570826824000088. doi:https://doi.org/10.1016/j.websem.2024.100822.

[26] V. K. Kommineni, B. König-Ries, S. Samuel, From human experts to machines: An llm supported approach to ontology and knowledge graph construction, 2024. URL: https://arxiv.org/abs/2403.08345. arXiv:2403.08345.

[27] 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, 2024. URL: https://arxiv.org/abs/2403.05921. arXiv:2403.05921.

[28] M.-J. Antia, C. M. Keet, Automating the generation of competency questions for ontologies with agocqs, in: F. Ortiz-Rodríguez, B. Villazón-Terrazas, S. Tiwari, C. Bobed (Eds.), Knowledge Graphs and Semantic Web, Springer Nature Switzerland, Cham, 2023, pp. 213–227.

[29] N. Tufek, A. Saissre, A. Hanbury, Validating Semantic Artifacts With Large Language Models, in: Proceedings of the 21th European Semantic Web Conference (ESWC), Krete, Greece, 2024, pp. 24–30.

[30] S. Hertling, H. Paulheim, Olala: Ontology matching with large language models, in: Proceedings of

the 12th Knowledge Capture Conference 2023, K-CAP '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 131–139. URL: https://doi.org/10.1145/3587259.3627571. doi:10.1145/3587259.3627571.

[31] H. Babaei Giglou, J. D'Souza, S. Auer, Llms4ol: Large language models for ontology learning, in: T. R. Payne, V. Presutti, G. Qi, M. Poveda-Villalón, G. Stoilos, L. Hollink, Z. Kaoudi, G. Cheng, J. Li (Eds.), The Semantic Web – ISWC 2023, Springer Nature Switzerland, Cham, 2023, pp. 408–427.

[32] S. Toro, A. V. Anagnostopoulos, S. M. Bello, K. Blumberg, R. Cameron, L. Carmody, A. D. Diehl, D. M. Dooley, W. D. Duncan, P. Fey, P. Gaudet, N. L. Harris, M. P. Joachimiak, L. Kiani, T. Lubiana, M. C. Munoz-Torres, S. O'Neil, D. Osumi-Sutherland, A. Puig-Barbe, J. T. Reese, L. Reiser, S. M. Robb, T. Ruemping, J. Seager, E. Sid, R. Stefancsik, M. Weber, V. Wood, M. A. Haendel, C. J. Mungall, Dynamic retrieval augmented generation of ontologies using artificial intelligence (dragon-ai), Journal of Biomedical Semantics 15 (2024) 19. URL: https://doi.org/10.1186/s13326-024-00320-3. doi:10.1186/s13326-024-00320-3.

[33] 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 Computer Society, Los Alamitos, CA, USA, 2023, pp. 226–229. URL: https://doi.ieeecomputersociety.org/10.1109/SYNASC61333.2023.00038. doi:10.1109/SYNASC61333.2023.00038.

[34] N. Köhler, F. Neuhaus, The Mercurial Top-Level Ontology of Large Language Models, 2024. URL: https://arxiv.org/abs/2405.01581. arXiv:2405.01581.

[35] D. Doumanas, A. Soularidis, K. Kotis, G. Vouros, Integrating llms in the engineering of a sar ontology, in: I. Maglogiannis, L. Iliadis, J. Macintyre, M. Avlonitis, A. Papaleonidas (Eds.), Artificial Intelligence Applications and Innovations, Springer Nature Switzerland, Cham, 2024, pp. 360–374.

[36] M. J. Saeedizade, E. Blomqvist, Navigating ontology development with large language models, in: A. Meroño Peñuela, A. Dimou, R. Troncy, O. Hartig, M. Acosta, M. Alam, H. Paulheim, P. Lisena (Eds.), The Semantic Web, Springer Nature Switzerland, Cham, 2024, pp. 143–161.

[37] J. H. Caufield, H. Hegde, V. Emonet, N. L. Harris, M. P. Joachimiak, N. Matentzoglu, H. Kim, S. A. T. Moxon, J. T. Reese, M. A. Haendel, P. N. Robinson, C. J. Mungall, Structured prompt interrogation and recursive extraction of semantics (spires): A method for populating knowledge bases using zero-shot learning, 2023. URL: https://arxiv.org/abs/2304.02711. arXiv:2304.02711.

[38] 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.

[39] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: a method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, Association for Computational Linguistics, USA, 2002, p. 311–318. URL: https://doi.org/10.3115/1073083.1073135. doi:10.3115/1073083.1073135.

[40] A. Fernández-Izquierdo, M. Poveda-Villalón, R. García-Castro, Coral: a corpus of ontological requirements annotated with lexico-syntactic patterns, in: The Semantic Web: 16th International Conference, ESWC 2019, Portorož, Slovenia, June 2–6, 2019, Proceedings 16, Springer, 2019, pp. 443–458.