CORAL: A Corpus of Ontological Requirements Annotated with Lexico-Syntactic Patterns

Alba Fernández-Izquierdo(✉), María Poveda-Villalón(✉), and Raúl García-Castro

Ontology Engineering Group, Universidad Politécnica de Madrid, Madrid, Spain
{albafernandez,mpoveda,rgarcia}@fi.upm.es

Abstract. Ontological requirements play a key role in ontology development as they determine the knowledge that needs to be modelled. In addition, the analysis of such requirements can be used (a) to improve ontology testing by easing the automation of requirements into tests; (b) to improve the requirements specification activity; or (c) to ease ontology reuse by facilitating the identification of patterns. However, there is a lack of openly available ontological requirements published together with their associated ontologies, which hinders such analysis. Therefore, in this work we present CORAL (Corpus of Ontological Requirements Annotated with Lexico-syntactic patterns), an openly available corpus of 834 ontological requirements annotated and 29 lexico-syntactic patterns, from which 12 are proposed in this work. CORAL is openly available in three different open formats, namely, HTML, CSV and RDF under "Creative Commons Attribution 4.0 International" license.

Keywords: Corpus · Linked data · Ontological requirements · Lexico-syntactic patterns

Resource type: Dataset
DOI: https://doi.org/10.5281/zenodo.1967306
URL: http://coralcorpus.linkeddata.es

1 Introduction

In recent years, the definition of functional ontological requirements [3,17,18], which represent the needs that the ontology to be built should cover, and their automatic formalization into axioms or tests (e.g., [5,13,19]) have been studied.

This work is partially supported by the H2020 project VICINITY: Open virtual neighbourhood network to connect intelligent buildings and smart objects (H2020-688467), by ETSI Specialist Task Force 534, and by a Predoctoral grant from the I+D+i program of the Universidad Politécnica de Madrid. The authors want to thank Agnieszka Ławrynowicz and her team for helping in the collection of ontological requirements.

P. Hitzler et al. (Eds.): ESWC 2019, LNCS 11503, pp. 443–458, 2019.
https://doi.org/10.1007/978-3-030-21348-0_29

The aim of these studies is to reduce the time consumed by ontology developers during the ontology verification activity, in which the ontology is compared against the ontological requirements, thus ensuring that the ontology is built correctly [16]. Functional ontological requirements can be written in the form of competency questions, which are natural language questions that the ontology to be modelled should be able to answer, or as statements, which are sentences that determine what should be built in the ontology.

However, to accurately define ontological requirements is not a trivial task and, therefore, neither is their automatic translation into a formal language. Due to the fact that some requirements are ambiguous [9] or vague, their transformation into axioms or tests is not usually direct and, consequently, it is very difficult to automate such translation. The analysis of these functional ontological requirements can be used in several domains, e.g., to improve ontology testing by easing the automation of requirements into tests, to improve the requirements specification activity by identifying problems in the definition of the requirements that should be avoided, or to ease ontology reuse by facilitating the identification of patterns to implement such requirements. However, there is a lack of openly available ontological requirements published together with their associated ontologies, which hinders such analysis.

In order to solve this limitation of data, the work presented here presents CORAL (Corpus of Ontological Requirements Annotated with Lexico-syntactic patterns), a corpus of 834 functional ontological requirements collected from different projects, websites and papers with the aim of providing a resource that could help the formalization of ontological requirements in an ontology. Additionally, this work provides a dictionary of lexico-syntactic patterns (LSPs) which includes LSPs collected from the state of the art and also defines new ones. This dictionary of LSPs have been used to annotate the set of ontological requirements. The aim of these annotations is to provide a mechanism for analyzing how these requirements are defined and, consequently, identifying whether there is ambiguity in their specification, with the ultimate goal of generating tests automatically.

The paper is organized as follows. Section 2 presents the process we followed to generate and annotate the corpus of ontological requirements. Section 3 describes the structure of the corpus, and how it can be maintained. Section 4 presents a set of statistics related to the corpus and Sect. 5 describes possible applications and usages. Section 6 describes the related work. Finally, Sect. 7 presents the conclusions we obtained and gives an overview on future work.

2 Building the Corpus

The corpus of annotated ontological requirements was generated through a set of five steps. This section summarizes them and also describes how this corpus is going to be extended and maintained.

2.1 Steps for Generating the Corpus

The steps carried out to generate the corpus, which are summarized in Fig. 1, were the following:

1. *To search for ontological requirements.* A set of 834 functional requirements, which have their corresponding ontology implementation available, was collected. These requirements were written as competency questions and as statements, and collected from several projects, as well as from papers and from resources available on the Web. These 834 functional requirements were associated to 14 different ontologies, whose names and requirements provenance are summarized in Table 1. These ontologies are built by different authors, and cover different topics and sizes.
2. *To search for existing LSPs.* In order to annotate the corpus of ontological requirements, a dictionary of LSPs was created. LSPs are understood as "formalized linguistic schemas or constructions derived from regular expressions

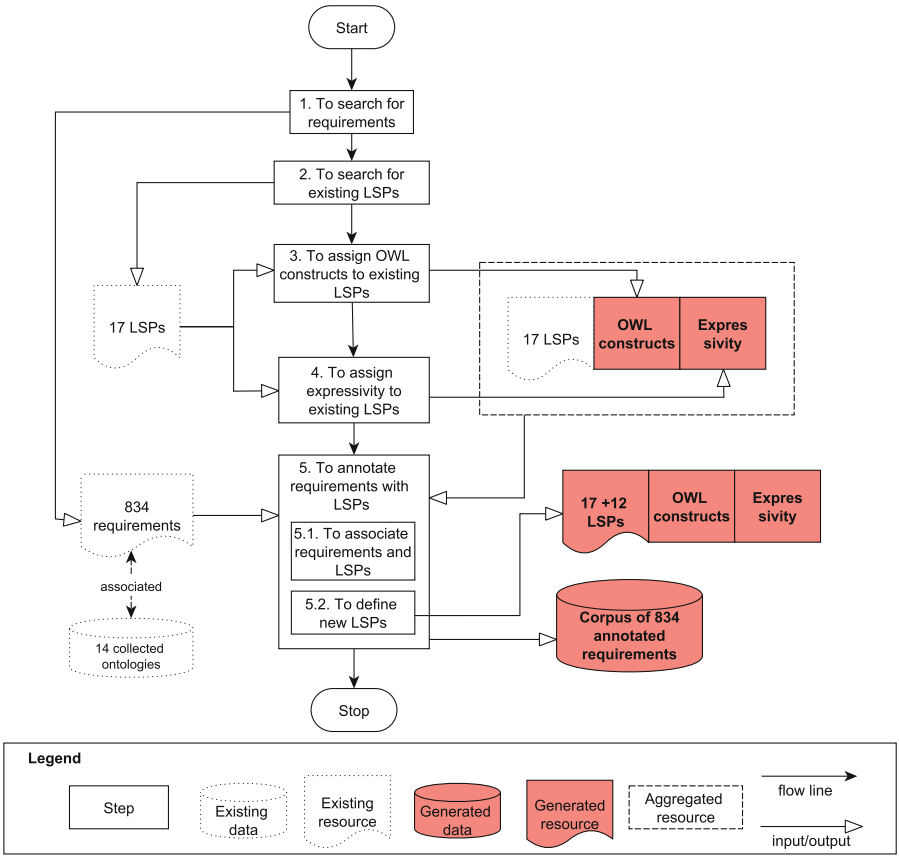

Fig. 1. Steps, with their inputs and outputs, carried out to conduct the analysis.

Table 1. Collected ontologies with their requirements

Ontology name	Provenance	Number of requirements
SAREF4ENVI	ETSI Technical Report [14]	58
OneM2M	ETSI Technical Report [14]	58
SAREF	ETSI Technical Report [14]	71
SAREF4BLDG	ETSI Technical Report [14]	98
BTN100	Github repository[a]	77
OntoDT	Paper [10]	14
Video Game	Paper [11]	66
Software Ontology	Paper [8]	90
Dem@care Ontology	Project deliverable[b]	107
ODRL	Website[c]	15
WoT mappings	Website[d]	15
Building Topology Ontology	Website[e]	18
WoT VICINITY	Website (See Footnote d)	24
VICINITY Core	Website (See Footnote d)	127

[a] https://github.com/oeg-upm/ontology-BTN100/tree/master/requirements.
[b] http://www.demcare.eu/downloads/D5.1SemanticKnowledgeStructures_and Representation.pdf.
[c] http://w3c.github.io/poe/ucr/.
[d] http://vicinity.iot.linkeddata.es.
[e] https://w3c-lbd-cg.github.io/bot/#Requirements.

in natural language that consist of certain linguistic and paralinguistic elements, following a specific syntactic order, and that permit to extract some conclusions about the meaning they express" [1]. Daga et al. [4] proposed a corpus of 17 LSPs[1], each of which has associated one or more ontology design patterns (ODPs) [6] that indicates how the LSP could be implemented in an ontology. It is possible that one LSP is associated with several disjoint ODPs, resulting in several possible ontology implementations. These LSPs are going to be called polysemous LSPs. Besides these ODPs, each LSP is also associated to: (a) an identifier, (b) formalization according to a BNF extension[2] and (c) examples of sentences in natural language which match with such LSP. It is worth noting that there can be more than one possible formalization for each LSP.

3. *To assign OWL constructs to existing LSPs.* Based on the ODPs associated to each LSP, we identify the set of OWL constructs needed to implement each LSP. The information related to how each ODP should be implemented was extracted from Suarez et al. [15], which defines a repository for ODPs along with their formal representation in an ontology. It was decided to consider

[1] http://ontologydesignpatterns.org/wiki/Submissions:LexicoSyntacticODPs.
[2] https://www.w3.org/Notation.html.

every OWL construct except for the annotation or versioning constructs, due to the fact that they do not add expressivity to the ontology and as they are used as metadata. Considering that in some cases the LSPs can be associated with several disjoint ODPs and, therefore, implemented in alternative models, each LSP has OWL constructs for each alternative model.

4. *To assign expressivity to existing LSPs.* According to the previously identified OWL constructs, the DL expressivity [2] of each LSP was determined. Because of the fact that some LSPs can be implemented into several models involving different OWL constructs, different expressivity has been associated to each model.
5. *To annotate ontological requirements with LSPs.* The annotation of ontological requirements is divided into two sub-steps:
   5.1 *To associate requirements and LSPs.* The association between ontological requirements and LSP was manually made based on the syntax of each requirement and on its mapping with the formalization of each LSP.
   5.2 *To define new LSPs.* For those ontological requirements which do not match with any of the state of the art LSPs, new patters were provided to support them. These new patterns include the same information as the state of the art LSPs, such as the formalization, ODPs associated (if exists) and examples of use. Altogether, a set of 12 new LSPs were added to the dictionary of patterns. If there are no ODPs associated to the LSP, then the OWL constructs were determined based on how each type of requirement is usually implemented in ontologies, e.g., a requirement related to a union between two concepts will be related with the OWL construct related to union, i.e., *owl:unionOf*.

2.2 Availability, Extensibility and Maintenance of the Annotated Corpus

The dictionary of patterns and the annotated corpus are openly available in HTML,[3] CSV and RDF format as Zenodo resources. They have a canonical citation using the DOI https://doi.org/10.5281/zenodo.1967306 and are published under Creative Commons Attribution 4.0 International license (CC BY 4.0).[4] Additionally, the corpus is also available in DataHub[5] and from Google Dataset search.

Maintenance of the corpus will be facilitated through the continuous process of gathering requirements from projects where ontologies are involved, as well as from periodic searches for gathering new openly available requirements on the Web and on papers. If new LSPs need to be defined to support these new ontological requirements, they will also be added to the dictionary of LSPs. For the time being the addition of new requirements and LSPs is manual. However, if CORAL is adopted by the community, it will be considered to provide an automatic service to allow users to add their requirements to the corpus.

[3] http://coralcorpus.linkeddata.es/.
[4] https://creativecommons.org/licenses/by/4.0.
[5] https://datahub.io/albaizq13/coralcorpus.

3 Corpus Description

The corpus presented here is divided into two resources, i.e., the dictionary of LSPs and the corpus of annotated requirements, which is annotated using the dictionary of LSPs. Each of these resources has a different structure, which is described below. Additionally, this section includes the description of the vocabulary used to publish the corpus as Linked Data and an example of use.

3.1 Dictionary of Lexico-Syntactic Patterns

For each LSP we have stored the following items:

- **Identifier of the LSP**. It contains an acronym composed of LSP (Lexico-Syntactic Pattern), plus the acronym of the relation captured by the ODP, plus the ISO-639[6] code for representing the name of the language for which the LSP is valid.
- **Description of the LSP**. It contains a brief description of the LSPs and the associated ODPs (if exist).
- **NeOn ODP identifier**. It determines the identifier used in the NeOn ODP repository [15] of the ODPs that can be used to implement the LSP. If the ODP was not contained in that repository, an acronym is created following the same rules. Identifiers are composed by the component type (e.g., LP standing for Logical Pattern), component (e.g., SC standing for SubClassOf), and number of the pattern (e.g., 01). If the LSP does not match with any ODP, then this information
- **Formalization**. The LSPs have been formalized according to the BNF notation.
- **Examples**. Sentences in natural language that exemplify the corresponding LSPs.
- **OWL constructs** associated to the LSP. These OWL constructs have been extracted from the ODPs, which indicates how it should be implemented in the ontology. As mentioned before, if there is no associated ODP then the OWL constructs are determined based on how that type of LSP is usually implemented in ontologies.
- **DL Expressivity**. It is related to the OWL constructs that are associated to each LSP.

3.2 Annotated Corpus of Ontological Requirements

Each ontological requirement includes the following information:

- **Identifier of the requirement**. This identifier can be used to differentiate each requirement.
- **Competency question or statement**. It represents the need the ontology is expected to cover.

[6] https://iso639-3.sil.org/code_tables/639/data.

- **Answer to the competency question**. If the requirement is written as a competency question, it needs an answer. As an example, the sentence "*There are two types of devices, sensor and actuator*" could be the answer to the previous competency question.
- **Provenance**. This information indicates the URI of the ontology for which the requirement was defined.
- **Corresponding LSP**. It indicates the LSP extracted from the proposed dictionary of LSPs (see Sect. 3.1) which matches the ontological requirement.

Due to the fact that for each LSP we have stored the OWL constructs and expressivity, it is possible to obtain the OWL constructs and expressivity of each requirement. However, because there are several polysemous LSPs, some of the requirements can be implemented in alternative models and, consequently, they have OWL constructs for each model.

3.3 Example of Use

Along this subsection an example of annotated requirement and LSP is presented in order to ease the understanding of how the corpus was built as well as how the corpus can be used to annotate ontological requirements.

In this example, a user needs to annotate a requirement with an appropriate LSP. The requirement is a competency question that states "There are different types of devices: sensor and actuator". To achieve this goal, the user has to manually look for the LSP in the corpus that best matches with such requirement. After this analysis, the user obtains that the requirement can be represented with the BNF formalization "*There are QUAN CN-CATV NP<superclass> PARA [(NP<subclass>,)* and] NP<subclass>*", which is the formalization associate to the LSP with identifier "LSP-SC-EN". Therefore, such requirement is annotated with the LSP "LSP-SC-EN", which has the following characteristics:

- **Identifier**: "LSP-SC-EN".
- **BNF formalization**: "There are QUAN CN-CATV NP<superclass> PARA [(NP<subclass>,)* and] NP<subclass>".
- **Description**: "The definition of a subsumption relation in an ontology".
- **Identifier of the associated ODP**: "LP-SC-01", which represent the ODP related to subclassOf relations.
- **Example**: "There are several kinds of memory: fast, expensive, short term memory and long-term memory".
- **OWL constructs**: "subclassOf, Class (Thing, Nothing)".
- **DL Expressivity**: "AL".

From this annotation it can also be deduced that the annotated requirement should be implemented following the ODP "LP-SC-01", which represent the ODP related to subclassOf relations. This ODP indicates that the two classes involved in the requirement need to be defined in the ontology together with the relation subclassOf between them.

3.4 Publishing the Corpus as Linked Data

The vocabulary used for publishing the corpus as linked data, depicted in Fig. 2, models the ontological requirements and their relations with the associated ontologies and LSPs. The proposed vocabulary can be extended with more information related to ontological requirements. Figure 3 shows an example of an ontological requirement following this vocabulary.

There are four main concepts in this proposed vocabulary[7], i.e., *ontological requirement*, *ontology*, *lexico-syntactic pattern* and *lexico-syntactic pattern implementation*.

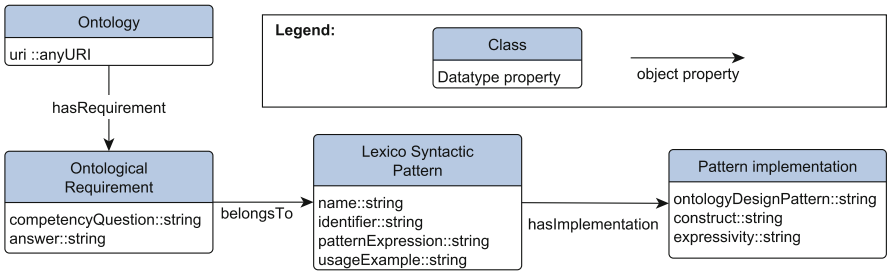

Fig. 2. Overview of the vocabulary for ontological requirements.

An *ontological requirement* is associated with two main types of elements: (1) *ontology*, which determines the provenance of the requirement, and (2) the correspondent *lexico-syntactic pattern*. The *ontological requirement* concept also includes as properties the competency question and its answer (if exists). Each *lexico-syntactic pattern*, in turn, is associated with the *lexico-syntactic pattern implementation*. It also includes as properties the name of the LSP, the identifier, the LSPs formalization and the usage example. The *lexico-syntactic pattern implementation* concept has as properties the ODPs, the constructs associated to the ODPs, and the expressivity of the implementation. Finally, the *ontology* concept, which represents the ontology from which the requirement is extracted, has as property its associated URI.

4 Corpus Statistics

This section introduces a set of statistics on the LSPs and ontological requirements present in the corpus.

Table 2 shows the total number of the collected ontological requirements and identified LSPs. Additionally, it shows the average and median ontological requirements and LSPs per ontology. According to Table 2, the analysed ontologies have an average of 56.15 ontological requirements in their specification, but these specifications are only related to an average of 7.07 LSPs.

[7] http://w3id.org/def/ontoreq.

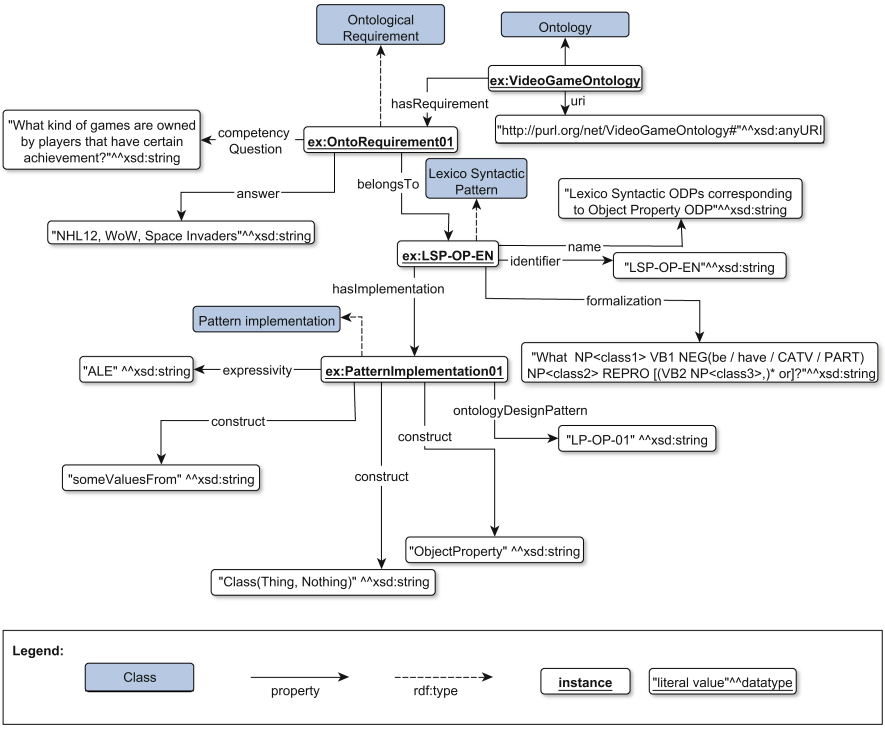

Fig. 3. Example of ontological requirement using the vocabulary proposed in this work.

The distribution of the LSPs according to their possible implementations, i.e., 1 LSP to 1 implementation in case that there is only one implementation for the LSP, and 1 LSP to N implementations in case that a LSP is related to several implementations, was also analysed. We observed that the majority of the LSPs, more precisely 88%, have a direct translation into a set of OWL constructs in an ontology. However, 12% of the ontological requirements are polysemous, that means, they can be implemented in several ontology models.

Table 2. Average and median of requirements per ontology and average of lexico-syntactic patterns per ontology

	Ontological requirements	Lexico-syntactic patterns
Average per ontology	56.15	7.07
Median per ontology	58	6
Total	834	29

Figures 4 and 5 show the number of ontologies in whose requirements specification each LSPs and each OWL constructs are used, respectively. According

to the results depicted in Fig. 4, the LSPs related to object properties are the most used in the ontological requirements, being present in the requirements specification of almost all the analyzed ontologies. However, there is also a set of LSPs which are never used in the requirements specification, such as the LSPs related to equivalent or disjoint classes.

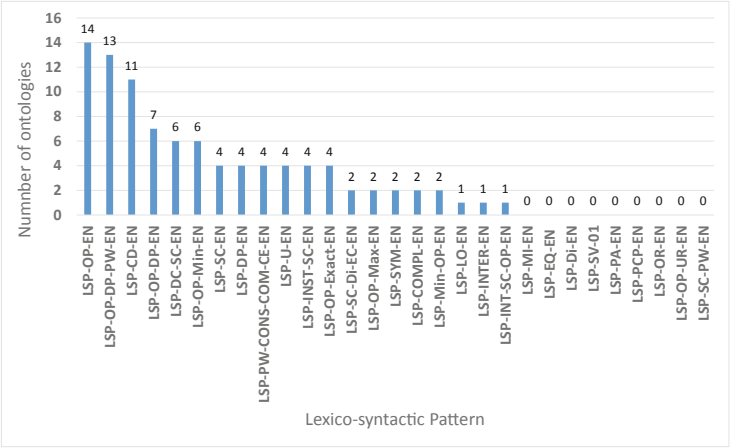

Fig. 4. Number of ontologies that use each lexico-syntactic pattern.

We can observe from Fig. 4 that polysemous LSPs, i.e., those LSPs that have more than one possible implementation in the ontology, are very common in the specification of the ontologies. Examples of this type of LSPs are LSP-OP-DP-PW-EN or LSP-OP-DP-EN, associated to "Object Property or Datatype Property or Simple Part-Whole relation ODPs" and "Object Property or Datatype Property ODPs", respectively. Therefore, even though there are few polysemous LSPs (only 12% of the set of LSPs), they have an important impact in the specification of ontological requirements.

The results shown in Fig. 5 indicate that all the requirement specifications includes the definition of classes, properties, domain and range. Additionally, the figure also depicts that there is a set of OWL constructs that are never associated to the requirements, such as those related to functional or transitive properties. Specifically, 19,23% of OWL constructs are present in all the ontologies requirements, 38,46% of OWL constructs are not present in any of the ontologies, and 61,54% of OWL constructs are present in at least one of the ontologies requirement specification.

Figure 6 shows the number of ontological requirements that are associated to each LSP. This figure only includes those LSPs with at least one requirement associated. We observed that the most common LSP in the requirement specifications is LSP-OP-EN, which is related to object properties. Other popular LSPs are LSP-OP-DP-EN and LSP-OP-DP-PW-EN, are polysemous LSPs and,

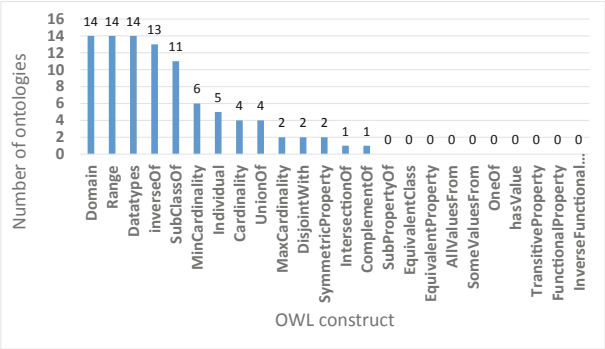

Fig. 5. Number of ontologies that uses each OWL construct.

therefore, they do not have direct translation to axioms or tests. It should be taken into account that there are several requirements that, due to they are long and include several sentences, they are associated to several LSPs. Therefore, the sum of the values in columns in Fig. 6 is higher than 834.

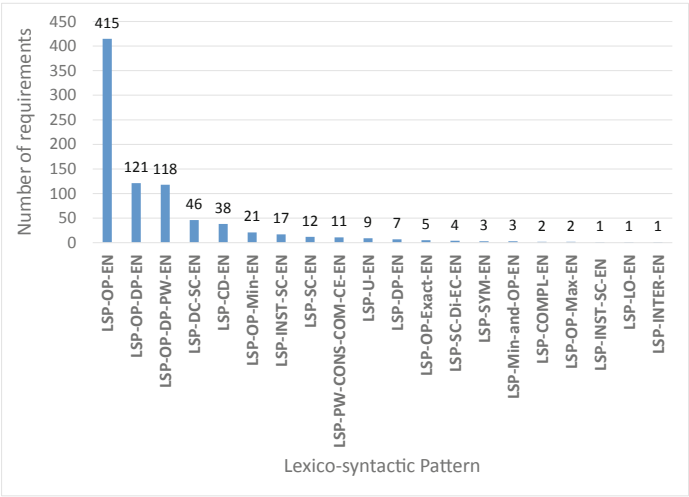

Fig. 6. Number of requirements per Lexico-Syntactic Pattern.

Finally, Fig. 7 illustrates the number of requirements associated to each DL expressivity. From this table it can be deduced that the majority of the requirements are related only to attributive language, which allows allows atomic negation, concept intersection, universal restrictions, and limited existential quantification, and to existential restriction, i.e., ALE. Only few requirements include

more complex logic related, for example, with cardinality or with inverse relationships, i.e., ALU or ALI. It should be taken into account that if an ontological requirement is associated with a polysemous LSP it will have several possible implementations, each one with a particular expressivity. Therefore, the sum of the values in columns in Fig. 7 is higher than 834.

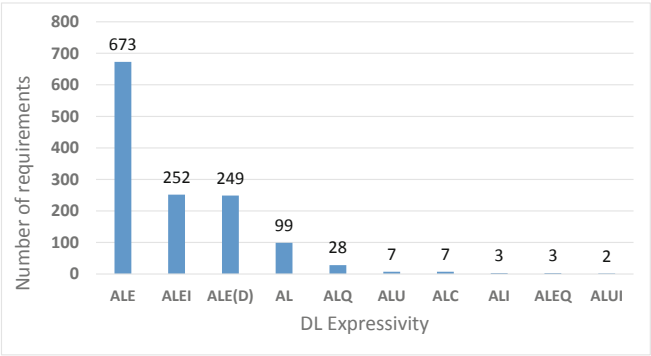

Fig. 7. Number of ontological requirements per DL expressivity.

5 Applications of the Corpus

The CORAL corpus can be used in several scenarios, including:

Automation of Ontology Testing. In order to automate as much as possible the translation of ontological requirements into axioms or tests it is needed to analyze how the requirements are constructed. The corpus presented in this work helps to identify LSPs in the requirements, as well as the set of OWL constructs associated to them. The OWL constructs associated to the LSPs can be used to automatically generate tests cases in order to verify whether the ontology satisfies the corresponding requirements, easing the testing process.

Improvement of the Requirements Specification Activity. The ontology requirements should be defined with the collaboration of domain experts, which are the responsible of identifying the needs the ontologies need to cover. However, there are considerable differences between what is defined in the requirements and what is implemented in the ontologies. For example, there are OWL constructs that are not associated to any ontological requirement, but are common in ontology implementations, such as axioms related to transitive or functional properties. In addition, due to the ambiguity of the requirements, some of them are associated to more than one possible implementation in the ontology. It should be necessary to improve the ontology elicitation activity in order to produce more precise requirements. This corpus can be used for a deeper analysis on

how requirements are specified, as well as for identifying ambiguous expressions that should be removed from the requirements definition in order to ease their formalization into axioms.

Ontology Reuse. The eXtreme Design methodology [12] is a collaborative, incremental and iterative method for pattern-based ontology design. This methodology is based on the application, exploitation, and definition of ODPs for solving ontology development issues. It uses competency questions in order to manually select the appropriate ODP to implement them in the ontologies. The dictionary of LSPs proposed here, can be used for the automatic suggestion of LSPs that can be associated to each competency question. Consequently, using this association between LSP and competency question, it can be proposed the ODPs needed to implement each competency question.

6 Related Work

After analyzing the literature, we conclude that in the state of the art there are no works which deal with the same motivation as the one presented here. However, there are similar initiatives. On the one hand, there are several works related to the analysis of lexico-syntactic patterns (LSPs) from ontological requirements. On the other hand, there are works which deal with ontology testing and which use small corpora of requirements in order to support their transformation between natural language and SPARQL queries.

Regarding the analysis of patterns, Daga et al. [4] provide a set of LSPs extracted from ontological requirements which have a direct correspondence with one or more ontology design patterns (ODPs). These LSPs are also described in Montiel-Ponsoda [9]. They help users in the development of ontologies by using a system that permits an automatic detection of the ontological relations expressed as requirements. However, the analysis on LSPs carried out by the authors is not exhaustive, and even though they support sentences written in natural language, they do not support ontological requirements written as competency questions, which are very common in the specification of requirements. CORAL corpus extends the LSPs presented by Daga et al. [4].

Concerning the use of corpora about ontological requirements in order to propose testing approaches, Ren et al. [13] introduce a work in which they use natural language processing to analyze competency questions written in controlled natural language. From this controlled natural language they create competency question patterns that could be automatically tested in the ontology. However, these competency questions are extracted from only one project and from a tutorial. Because of the limitation of data, this work can only support a small set of patterns related to ontological requirements. Additionally, another example of work which tries to translate natural language sentences into a formal language is the one presented by Lopez et al. [7]. In this work, the authors show a subset of the sentences used to automatically generate SPARQL queries. However, the set of sentences is not published and, therefore, cannot be reused. Finally,

the dataset presented by Wisniewski et al.[8] provides a corpus of 234 competency questions related to 5 different ontologies[9]. This dataset includes competency questions and their formalization into manually extracted SPARQL-OWL queries. Nevertheless, the dataset provided by Wisniewski et al. does not include any annotation or information related to the lexico-syntactic analysis needed to translate the competency questions into SPARQL queries and only the list of competency questions is available in a reusable format, namely, CSV.

The corpus presented is this work aims to solve these limitations of data by providing a openly available and reusable corpus of ontological requirements written in the form of competency questions and sentences. This corpus includes ontological requirements extracted from different ontologies and written by different authors, avoiding bias in writing the requirements. Additionally, it provides a dictionary of LSPs which is used to annotate these requirements and determine how each ontological requirement could be implemented.

7 Conclusions

The work presented here provides (1) a dictionary of 29 LSPs, from which 12 are new ones and 17 are LSPs already defined in the state of the art, and (2) an corpus of 834 ontological requirements annotated with these LSPs. This annotation determines the associated OWL constructs needed to implement each requirement in an ontology based, in most cases, on ODPs. Both the dictionary of patterns and the annotated corpus are provided in the machine-readable format RDF, in CSV and in HTML as Zenodo and Datahub resources. Additionally, CORAL is accessible from Google Dataset Search.

From the dictionary of LSPs, it was confirmed that there are several polysemous expressions that could result in different ontology structures. Therefore, the requirements associated with these polysemous patterns could also result in different ontology models. This result illustrates the need of improvement of the specification of requirements, which should aim to avoid ambiguities and to be more precise in order to identify the appropriate needs to model the ontologies. With this, it should be possible to reduce errors during the modelling of ontologies and also ease the automation from requirements into tests.

During the process of collecting requirements, significant difficulties to find available real-world functional requirements were found. We consider that publishing this kind of data is helpful not only of reusability, but also for ontology testing to verify completeness and conformance between ontologies.

We believe that the proposed corpus can promote research in several problems, such as ontology testing and ontology requirements specification, while it can also facilitate research in other topics such as in natural language processing. Future work will be directed to a further evaluation of how CORAL requirements are annotated and to automate the requirements annotation.

[8] https://github.com/CQ2SPARQLOWL/.

[9] Please note that we consider 4 of the 5 ontologies analyzed by them. The only ontology left was not considered due to the fact that their requirements were published recently (November 26, 2018). It will be included in future releases (see Sect. 2.2).

References

1. Aguado de Cea, G., Gómez-Pérez, A., Montiel-Ponsoda, E., Suárez-Figueroa, M.C.: Natural language-based approach for helping in the reuse of ontology design patterns. In: Gangemi, A., Euzenat, J. (eds.) EKAW 2008. LNCS (LNAI), vol. 5268, pp. 32–47. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87696-0_6
2. Baader, F., Horrocks, I., Sattler, U.: Description logics. Found. Artif. Intell. **3**, 135–179 (2008)
3. Bezerra, C., Freitas, F., Santana, F.: Evaluating ontologies with competency questions. In: 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), vol. 3, pp. 284–285. IEEE (2013)
4. Daga, E., et al.: NeOn D2. 5.2 Pattern Based Ontology Design: Methodology and Software Support (2010)
5. Dennis, M., van Deemter, K., Dell'Aglio, D., Pan, J.Z.: Computing authoring tests from competency questions: experimental validation. In: d'Amato, C., et al. (eds.) ISWC 2017. LNCS, vol. 10587, pp. 243–259. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68288-4_15
6. Gangemi, A., Presutti, V.: Ontology design patterns. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. IHIS, pp. 221–243. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-92673-3_10
7. Lopez, V., Pasin, M., Motta, E.: AquaLog: an ontology-portable question answering system for the semantic web. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 546–562. Springer, Heidelberg (2005). https://doi.org/10.1007/11431053_37
8. Malone, J., Brown, A., Lister, A.L., Ison, J., Hull, D., Parkinson, H., Stevens, R.: The software ontology (SWO): a resource for reproducibility in biomedical data analysis, curation and digital preservation. J. Biomed. Semant. **5**(1), 25 (2014)
9. Montiel-Ponsoda, E.: Multilingualism in Ontologies - Building Patterns and Representation Models. LAP Lambert Academic Publishing, Germany (2011)
10. Panov, P., Soldatova, L.N., Džeroski, S.: Generic ontology of datatypes. Inf. Sci. **329**, 900–920 (2016)
11. Parkkila, J., Radulovic, F., Garijo, D., Poveda-Villalón, M., Ikonen, J., Porras, J., Gómez-Pérez, A.: An ontology for videogame interoperability. Multimedia Tools Appl. **76**(4), 4981–5000 (2017)
12. Presutti, V., Daga, E., Gangemi, A., Blomqvist, E.: eXtreme design with content ontology design patterns. In: Proceedings of Workshop on Ontology Patterns (2009)
13. Ren, Y., Parvizi, A., Mellish, C., Pan, J.Z., van Deemter, K., Stevens, R.: Towards competency question-driven ontology authoring. In: Presutti, V., d'Amato, C., Gandon, F., d'Aquin, M., Staab, S., Tordai, A. (eds.) ESWC 2014. LNCS, vol. 8465, pp. 752–767. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07443-6_50
14. SmartM2M: SAREF extension investigation Technical report (TR 103 411)
15. Suárez-Figueroa, M.C., et al.: NeOn D5.1.1 NeOn Modelling Components (2007)
16. Suárez-Figueroa, M.C., Gómez-Pérez, A.: First attempt towards a standard glossary of ontology engineering terminology. In: Proceedings of 8th International Conference on Terminology and Knowledge Engineering (TKE 2008), p. 1 (2008)
17. Suárez-Figueroa, M.C., Gómez-Pérez, A., Fernandez-Lopez, M.: The NeOn methodology framework: a scenario-based methodology for ontology development. Appl. Ontol. **10**(2), 107–145 (2015)

18. Suárez-Figueroa, M.C., Gómez-Pérez, A., Villazón-Terrazas, B.: How to write and use the ontology requirements specification document. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2009. LNCS, vol. 5871, pp. 966–982. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-05151-7_16
19. Zemmouchi-Ghomari, L., Ghomari, A.R.: Translating natural language competency questions into SPARQLQueries: a case study. In: The First International Conference on Building and Exploring Web Based Environments, pp. 81–86 (2013)