LOV: A semantic repository to access vocabularies on the Web

Editor(s): Name Surname, University, Country
Solicited review(s): Name Surname, University, Country
Open review(s): Name Surname, University, Country

Pierre-Yves Vandenbussche [a,*], Ghislain A. Atemezing [b,**], María Poveda-Villalón [c] and Bernard Vatant [d]

[a] *Fujitsu Laboratories, Dublin, Ireland*
E-mail: Pierre-Yves.Vandenbussche@ie.fujitsu.com
[b] *Multimedia Communication Department, EURECOM, Campus SophiaTech 450, route des Chappes, 06410 Biot, France*
E-mail: auguste.atemezing@eurecom.fr
[c] *Ontology Engineering Group (OEG), Universidad Politécnica de Madrid, Madrid, Spain*
E-mail: mpoveda@fi.upm.es
[d] *Mondeca, 35 boulevard de Strasbourg, 75010 Paris, France*
E-mail: bernard.vatant@mondeca.com

Abstract. The abstract should be clear, descriptive, self-explanatory and no longer than 200 words. It should also be suitable for publication in abstracting services. Do not include references or formulae in the abstract.

Keywords: LOV, Linked Open Vocabulary, Ontology search, Linked Data

1. Introduction

Started in March 2011, in the framework of the DataLift research project [18] hosted by the Open Knowledge Foundation, the Linked Open Vocabularies (LOV) initiative is now standing as an innovative observatory of the vocabularies ecosystem. It gathers and makes visible indicators not yet harvested before, such as interconnection between vocabularies, versioning history and maintenance policy, past and current referent (individual or organization) if any. The number of vocabularies indexed by LOV is constantly growing (469 as of December 2014) thanks to a community effort and it is the only catalog, to the best of our knowledge, that provide all types of search criteria (metadata search, within/across ontologies search), both an API and a SPARQL endpoint access. According to the categories of ontology libraries defined in [5], LOV falls under the category of *"curated ontology directory"* and *"application platform"*.

The development of LOV has highlighted a number of interesting research issues such as *"where to find the best domain vocabulary to reuse?"*, and *"is it possible to create a curated catalogue of vocabularies that are links?"*. In this paper, **[TODO: add what are the contributions]**

2. LOV functionalities

[PIERRE-YVES TO: give some input here, and current figures]

[*] Thanks to Amélie Gyrard and Thomas Francart for their reviews on vocabularies

[**] Corresponding author. E-mail: auguste.atemezing@eurecom.fr

LOV high-curated vocabularies are suitable for ontology search and reuse activities during the process of creating and publishing a vocabulary. Below are the relevant features of LOV achieving the aforementioned activities:

Domain filtering. Each vocabulary is inserted into LOV according to its domain and/or scope. This information is guided by the scope of the vocabulary, such as City, Science, Library, Metadata, Media, etc. This feature helps in disambiguating the results of the querying service and to classify vocabularies.

Content aware Search. If the searched term matches a `rdfs:label` it will have a higher score than if it matches `dcterms:comment`.

Links between vocabularies. One of the key feature of LOV design is the explicit links between vocabularies,

Scope of LOV. The intended use is to promote and facilitate the reuse of vocabularies in the linked data ecosystem.

Vocabulary curation. The collection of the vocabularies is maintained by curators in charge of validating and inserting vocabularies in the LOV ecosystem, by taking care of the versions of the vocabulary and giving some reviews. The vocabulary is then automatically enriched with more information about the datasets using it, and relations to other vocabularies.

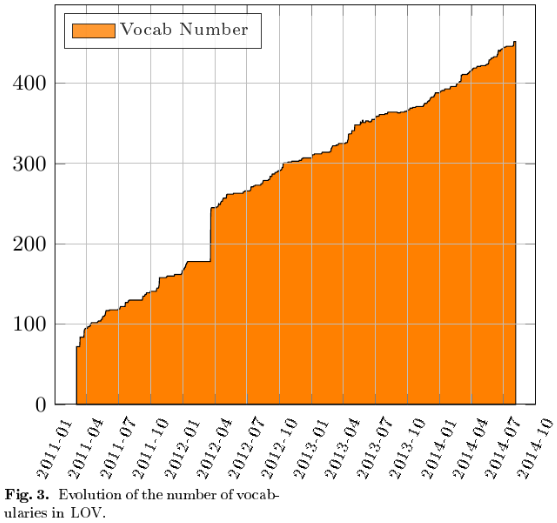


Fig. 1. Graph evolution of vocabularies inserted into LOV from 2011 to 2014.

3. Metadata information of a vocabulary

A vocabulary to be accepted and inserted into LOV, a minimal metadata information is required in the vocabulary. Those metadata should be described reusing existing meta-vocabularies, such Dublin core, lexvo[1] and provenance vocabulary. Three levels of metadata are present in any vocabulary in LOV:

- Metadata associated to the vocabulary: this information is embed in within the vocabulary to provide context, and useful data about the vocabulary. Four minimal metadata is asked to the publisher of a vocabulary to be LOV-able: title, description, dereferencable URI. The bot spots more information
- Inlinks vocabularies, using links to a given vocabulary
- Outlinks vocubularies, that is vocabularies reused to create a given vocabulary.

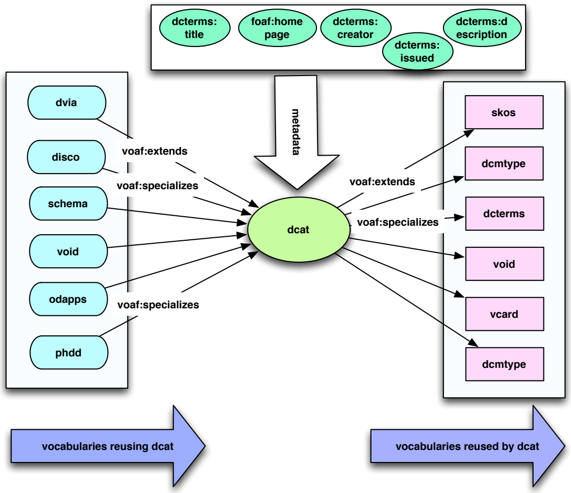


Fig. 2. Sample type of relations and metadata used in the DCAT vocabulary.

Languages tag retrieved for each vocabulary is inserted into LOV database by using the `dcterms:language` property with the URI of the language in lexvo dataset using the ISO 639-3 code. For example, the URI associated to Japanese language is `http://www.lexvo.org/page/iso639-3/jpn`. Currently, 91.25% (428 out of 469) of vocabularies use a language tag associated, with only 9 vocabularies with

[1] `http://lexvo.org/ontology#`

labels in a language different from English. Table **??** presents the number and percentage of the top five languages detected in LOV. This results suggest that publishers should make more effort to provide multilingual vocabularies on the Web,

Table 1
Top five languages and percentage detected in LOV catalogue.

Language	**Number**	**%**
English	419	97.89%
French	42	9.81%
Spanish	28	6.54%
German	21	4.90%
Italian	20	4.67%

4. LOV as an ontology-based search engine

Users or agents use keywords to search properties or classes within vocabularies in the LOV catalogue. The log[2] of search terms between 2012/01/06 and 2014/12/09 presents a total of 54,657 terms, with 36,019 (65.90%) duplicate terms and 18,643 unique terms (34.10%). Figure 3 depicts the number of terms in the log grouped by year. From 2012 to 2013, there have been an increase of more than 50% of terms for searching vocabularies in LOV. Searching terms are mostly single words (e.g., currency). However, terms can be composite of two words (e.g., family tree), three words (e.g., semantic sensor network) or an URI (e.g., http://www.aktors.org/ontology/portal). Table 2 shows details for the use of URIs, two words and at least three words for unique values in the LOV log.

Table 2
Patterns used other than one term in searching vocabularies in LOV.

Pattern	**2012**	**2013**	**2014**	**Total**
URI	38	54	63	155
t_1 t_2	200	480	466	1146
$(t_1\ t_2\ t_3)^*$	93	233	249	575

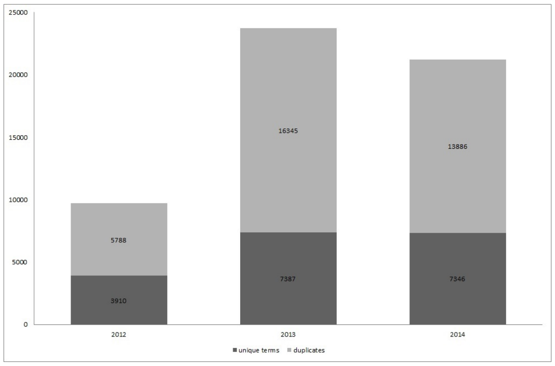


Fig. 3. Unique and duplicate terms searched by agents/users according to LOV log in the period between 2012/01/06 until 2014/12/09.

5. Life-cycle of a vocabulary in LOV

[TODO: ADD some text here]
Review –insertion –publication –archive version –Daily checking –

[GHIS TO: add a figure of tracking the vocabulary evolution]

Criteria to remove a vocabulary When it is orphan, no metadata, no labels.

6. Building ontologies with LOV

The NeOn Methodology is a scenario-based methodology that supports the collaborative aspects of ontology development and reuse, as well as the dynamic evolution of ontology networks in distributed environments. The key assets of the NeOn Methodology are [6]:

- A set of nine scenarios for building ontologies and ontology networks, emphasizing the reuse of ontological and non-ontological resources, the reengineering and merging, and taking into account collaboration and dynamism.
- The NeOn Glossary of Processes and Activities, which identifies and defines the processes and activities carried out when ontology networks are collaboratively built by teams.
- Methodological guidelines for different processes and activities of the ontology network development process, such as the reuse and re-engineering

[2] `http://lov.okfn.org/dataset/lov/stats/searchLog.csv`

of ontological and non-ontological resources, the ontology requirements specification, the ontology localization, the scheduling, etc.

LOV is a catalog and API that can fits well within the Neon methodology for building vocabularies and ontologies. Based on the Neon Methodology's glossary of activities for building ontologies, LOV is relevant in four activities:

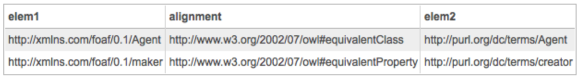
Ontology Search. Main LOV's feature is the search of vocabulary terms. These vocabularies are categorized within LOV according to the domain they address. In this way, LOV contributes to ontology search by means of (a) keyword search and (b) domain browsing.

Ontology Assessment. LOV provides a score for each term retrieved by a keyword search. This score can be used during the assessment stage and includes a unique term statistical feature[3] which provides for each term registered in LOV the following information: (a) "LOV distribution" that represents the number of vocabularies in LOV that refer to a particular element; (b) "LOV popularity" that shows the number of other vocabulary elements that refers to a particular one; and (c) "LOD distribution" that refers to the number of datasets in LOD which use a particular vocabulary; and (d) "LOD popularity" that refers to the number of vocabulary element occurrences in the LOD.

Ontology Mapping. In LOV, vocabularies rely on each other in seven different ways. These relationships are explicitly stated using VOAF vocabulary. This data could be useful to find alignments between ontologies, for example one user might be interested in finding equivalent classes for a given class or all the equivalent classes among two ontologies. Listing 1 shows the retrieved data when asking for all the equivalent classes and properties between the vocabularies foaf and dcterms by means of the related VOAF query[4]:
Figure 4 shows the alignments between foaf and dcterms vocabularies by mean of `owl:equivalentClass` and `owl:equivalentProperty`.

Listing 1: SPARQL query asking for all the equivalent classes and properties between the vocabularies foaf and dcterms.

```
    SELECT DISTINCT ?elem1 ?alignment ?elem2 {
        {?elem1 <http://www.w3.org/2002/07/owl#equivalentClass> ?elem2}
        UNION {?elem1 <http://www.w3.org/2002/07/owl#equivalentProperty> ?elem2}
        UNION {?elem2 <http://www.w3.org/2002/07/owl#equivalentClass> ?elem1}
        UNION {?elem2 <http://www.w3.org/2002/07/owl#equivalentProperty> ?elem1}
        FILTER(!isBlank(?elem2))
        FILTER(!isBlank(?elem1))
        ?elem1 ?alignment ?elem2.
        ?elem1 rdfs:isDefinedBy <http://xmlns.com/foaf/0.1/>.
        ?elem2 rdfs:isDefinedBy <http://purl.org/dc/terms/>.
      } ORDER BY ?alignment
```



elem1	alignment	elem2
http://xmlns.com/foaf/0.1/Agent	http://www.w3.org/2002/07/owl#equivalentClass	http://purl.org/dc/terms/Agent
http://xmlns.com/foaf/0.1/maker	http://www.w3.org/2002/07/owl#equivalentProperty	http://purl.org/dc/terms/creator

Fig. 4. Equivalent classes and properties between foaf and dcterms

Ontology Localization. Labels in different languages are stored in the LOV endpoint for the ontology terms that provide such information. This annotations could be used when translating terms into different languages. This information could be extracted by querying the SPARQL endpoint[5] as shown in Listing 2 where all the labels defined for

[3] `http://lov.okfn.org/dataset/lov/stats/`

[4] `http://goo.gl/sTIGQ6`. Prefixes are omitted for readability purpose. The reader can find the correct namespace for a prefix in LOV.

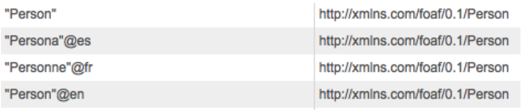
[5] `http://goo.gl/JJCJ01`

the terms that have at least one *rdfs:label* containing strictly “person":

Listing 2: SPARQL query asking all the labels defined for the terms containing person.

```
    SELECT DISTINCT ?label2 ?element {
                ?element rdfs:label ?label1 .
                ?element rdfs:label ?label2 .
                FILTER (?label1 != ?label2 ).
                FILTER(REGEX(STR(?label1), "person", "i")).
    } ORDER BY ?element
```

An excerpt of the query result is shown in Figure 5. From that result, “Persona”@es and “Personne”@fr could be used as translations for the English term “Person” in Spanish and French respectively.



"Person"	http://xmlns.com/foaf/0.1/Person
"Persona"@es	http://xmlns.com/foaf/0.1/Person
"Personne"@fr	http://xmlns.com/foaf/0.1/Person
"Person"@en	http://xmlns.com/foaf/0.1/Person

Fig. 5. Translations example for foaf:Person

Figure 6 shows the activities in which LOV can support within the overall Neon methodologies activity workflow.

7. The LOV APIs

[PIERRE-YVES TO: write the way to use the API, with a snapshot of some results] LOV database as of today contains over 46,000 RDF vocabularies elements, with 28,000 properties and 18,000 classes, all accessible also via API[6]. For example, using the LOV Search API, an application can search for all classes with the term “Catalog” in any literal value by making the following call: `http://lov.okfn.org/dataset/lov/api/v2/search?q=Catalog&type=class`

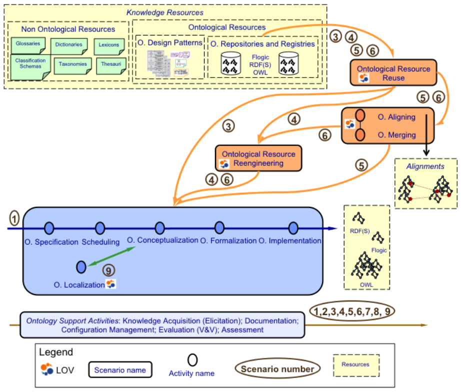


Fig. 6. Meeting points between LOV and the NeOn methodology, derived from [6].

7.1. Derived tools and applications

Maguire et al. [12] uses the LOV search API to implement OntoMaton[7], a widget for bringing together ontology lookup and tagging within the collaborative environment provided by Google spreadsheets.

YASGUI (Yet Another SPARQL Query GUI)[8], a client-side JavaScript library that uses property, class and prefixes autocompletion using LOV API together with `http://prefix.cc` [16].

Datalift[9] platform [18], a framework for “lifting” raw data into RDF, comes with a module to map data objects and properties to ontology classes and predicates available in the LOV catalogue. Data2Ontology module takes an input a “raw RDF”, that is a dataset that has been converted directly from legacy format to triples. The goal is to help to publishers reusing existing ontologies for converting their dataset for easy discovery and interlinking. It consists of three main components assisting the publisher in selecting properties suitable for the dataset to be published.

- **LOV component:** This component is in charge to connect with the LOV catalogue to retrieve up-to-date ontologies using the LOV search API[10].
- **Matching Workflow:** Data2Ontology offers to map the data to LOV by automatically proposing a list of best matches.
- **SPARQL Generator:** This module receives as input the desired mappings and creates the SPARQL

[6] `http://lov.okfn.org/dataset/lov/apidoc/`

[7] `https://github.com/ISA-tools/OntoMaton`

[8] `http://yasgui.laurensrietveld.nl`

[9] `http://datalift.org/en/node/24`

[10] `http://lov.okfn.org/dataset/lov/apidoc/\#lov2search`

CONSTRUCT query needed to implement the mapping. The query can further be modified before the execution to generate a new dataset in the lifting process with Datalift.

7.2. *Using LOV as a Research platform*

LOV vocabularies have served as object of study in [13] where trends in ontology reuse techniques were analyzed in 2012. In addition, LOV dataset has been used in order to analyze the occurrence of good and bad practices in vocabularies as described in [14] in 2013.

LOV query log covering the period between 06/01/2012 and 16/04/2014 is used in [2] to build a benchmark suite for ontology search and ranking. The CBR-Bench[11] benchmark uses eight ranking models of resources in ontologies and compare the results with ontology engineers.

In [9], the authors rate vocabularies according to some criteria beyond the sameAs links but subClassOf and equivalentClass 'links' between vocabularies to foster interoperability, query federation, ease the interpretation of data, and so forth.

Databugger[12] is a test-driven data debugging framework for the Web of Data. In [10,11], the authors provide an automatic test case instantiations for all available schemata registered with LOV. In this case, the vocabularies of LOV are used to encode semantics to domain specific knowledge to check the quality of data.

Giovanni et al. [8] analyzes the current use of licenses in vocabularies on the Web based on LOV catalogue to further propose a framework to detect incompatibilities between datasets and vocabularies.

[MARIA TO: Maria: something about LOVER, was it really relate?].

8. Related work and Discussion

Reusing vocabularies requires searching for terms in existing specialized vocabulary catalogs or search engines on the web. While we refer the reader to [5] for a systematic survey of ontology repositories, we list below some existing catalogs relevant to find vocabularies [1]:

- *Catalogs of generic vocabularies/schemas* similar to the LOV catalog. Example of catalogs falling in this category are vocab.org[13], ontologi.es[14], JoinUp Semantic Assets or the Open Metadata Registry.
- *Catalogs of ontologies for a specific domain* such as biomedicine with the BioPortal [20], geospatial ontologies with SOCoP+OOR[15], Marine Metadata Interoperability and the SWEET [15] ontologies[16]. The SWEET ontologies include several thousand terms, spanning a broad extent of Earth system science and related concepts (such as data characteristics), with the search tool to aid finding science data resources.
- *Catalogs of ontology Design Patterns (ODP)* focused on reusable patterns in ontology engineering [19]. The submitted patterns are small pieces of vocabularies that can further be integrated or linked with others vocabularies. ODP is more targeted on reusable successful solution to a recurrent modeling problem. However, it does not provide a search function for specific terms as it is the case with Swoogle or Watson.
- *Search Engines of ontology terms*. Among ontology search engines, we can cite: Swoogle [7], Watson [4,17] and FalconS [3]. These search engines crawl for data schema from RDF document on the Web. They offer a filtering based on ontology type (Class, Property) and a ranking based on the popularity. They don't look for ontology relations nor check if the definition of the ontology is available (usually known as dereferenciation)

LOV focuses only on vocabularies (subpart of semantic documents of the web) submitted by any user, reviewed and validated by curators. In addition, LOV keeps track of different versions of the vocabularies in the server that can be retrieved for comparing the differences between along the time evolution. In contrast, Swoogle is designed to automatically discover Semantic Web Documents (SWDs), indexes their metadata and answers queries about it. Thus, the result of a search query retrieved any semantic document. For example, a query of the term *person* gives $16,438$ results while in LOV, only the term appears in 134 vocabularies. Watson works similarly to Swoogle, crawling and

[11] https://zenodo.org/record/11121

[12] https://github.com/AKSW/Databugger

[13] http://vocab.org/

[14] http://ontologi.es/

[15] http://socop.oor.net/

[16] http://sweet.jpl.nasa.gov/2.1/

Feature	Swoogle	Watson	Falcons	LOV
Browsing ontologies	Yes	Yes	Yes	Yes
Scope	SWDs	SWDs	Concepts	ontologies
Metrics	Ranking	Ranking	Ranking	LOD popularity
Domain filtering	No	No	No	Yes
Comments and review	No	Yes	No	Only by curators
Ranking	Doc. based	Doc. based	Doc. based	Metric-based
Web service access	Yes	Yes	Yes	Yes
SPARQL endpoint	No	No	No	Yes
Read/Write	Read	Read & Write	Read	Read
Ontology directory	No	No	No	Yes
Application platform	No	No	No	Yes
Storage	Cache	-	-	Dump & endpoint
Interaction with Contributors	No	-	No	Yes

Table 3

Comparison of LOV, with respect to Swoogle, Watson and Falcons; based on part of the framework defined in [5].

indexing semantic document at a small scale, explicitly distinguishing for each document (resource), concepts, properties and individuals if available. While in Swoogle the ranking score is displayed, Watson shows the language of the resource and the size. Falcons is a keyword-based search system for concepts and objects on the Semantic Web, and is equipped with entity summarization for browsing. It is notable that Falcons limits the search only to ontologies and a recommendation feature is provided according to users' preferences. However, it does not provide any relationships between the related ontologies, nor any domain classification of the vocabularies. Table 3 lists some key features of LOV with respect to Swoogle, Watson and Falcons.

LOV search engine is to the best of our knowledge, the only purpose-built ontology search engine available on the Web with an up-to-date index.

9. Conclusion and Future work

[TODO: add summary]
Next version v3 functionality coming soon :)

Acknowledgments

This work has been partially supported by the French National Research Agency (ANR) within the Datalift Project, under grant number ANR-10-CORD-009 and the Spanish project BabelData (TIN2010-17550) and Fujitsu Laboratories. The Linked Open Vocabularies initiative is graciously hosted by the Open Knowledge Foundation.Thanks to all the members of the LOV community in Google+[17], all the editors and publishers of vocabularies who trust in the LOV catalogue.

References

[1] Ghislain Auguste Atemezing, Bernard Vatant, Raphaël Troncy, and Pierre-Yves Vandenbussche. Harmonizing services for LOD vocabularies: a Case Study. In *Workshop on Semantic Web Enterprise Adoption and Best Practice (WaSABi)*, Sydney, Australia, 2013.

[2] Anila Sahar Butt, Armin Haller, and Lexing Xie. Ontology search: An empirical evaluation. In *The Semantic Web–ISWC 2014*, pages 130–147. Springer, 2014.

[3] Gong Cheng, Weiyi Ge, and Yuzhong Qu. Falcons: searching and browsing entities on the semantic web. In *Proceedings of the 17th international conference on World Wide Web*, pages 1101–1102. ACM, 2008.

[4] M. DÁquin, C. Baldassarre, L. Gridinoc, M. Sabou, S. Angeletou, and E. Motta. Watson: Supporting next generation semantic web applications. 2007.

[5] Mathieu DÁquin and Natasha F. Noy. Where to publish and find ontologies? a survey of ontology libraries. *Web Semantics: Science, Services and Agents on the World Wide Web*, 11(0):96–111, 2012.

[6] Maria del Carmen. Suárez-Figueroa. *NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse*. PhD thesis, Universidad Politecnica de Madrid, Spain, June 2010. http://oa.upm.es/3879/.

[17] `https://plus.google.com/communities/108509791366293651606`

[7] T. Finin, L. Ding, R. Pan, A. Joshi, P. Kolari, A. Java, and Y. Peng. Swoogle: Searching for knowledge on the semantic web. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 20, page 1682. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

[8] Guido Governatori, Ho-Pun Lam, Antonino Rotolo, Serena Villata, Ghislain Atemezing, and Fabien Gandon. Checking licenses compatibility between vocabularies and data. In *Fifth International Workshop on Consuming Linked Data (COLD2014)*.

[9] Krzysztof Janowicz, Pascal Hitzler, Benjamin Adams, Dave Kolas, and Charles Vardeman II. Five stars of linked data vocabulary use. *Semantic Web*, 5(3):173–176, 2014.

[10] Dimitris Kontokostas, Patrick Westphal, Sören Auer, Sebastian Hellmann, Jens Lehmann, and Roland Cornelissen. Databugger: A test-driven framework for debugging the web of data. In *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*, WWW Companion '14, pages 115–118. International World Wide Web Conferences Steering Committee, 2014.

[11] Dimitris Kontokostas, Patrick Westphal, Sören Auer, Sebastian Hellmann, Jens Lehmann, Roland Cornelissen, and Amrapali Zaveri. Test-driven evaluation of linked data quality. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 747–758, Republic and Canton of Geneva, Switzerland, 2014. International World Wide Web Conferences Steering Committee.

[12] Eamonn Maguire, Alejandra González-Beltrán, Patricia L Whetzel, Susanna-Assunta Sansone, and Philippe Rocca-Serra. Ontomaton: a bioportal powered ontology widget for google spreadsheets. *Bioinformatics*, page bts718, 2012.

[13] María Poveda Villalón, Mari Carmen Suárez-Figueroa, and Asunción Gómez-Pérez. The landscape of ontology reuse in linked data. 2012.

[14] María Poveda-Villalón, Bernard Vatant, Mari Carmen Suárez-Figueroa, and Asunción Gómez-Pérez. Detecting good practices and pitfalls when publishing vocabularies on the web. 2013.

[15] R. G. Raskin and M. J. Pan. Knowledge representation in the semantic web for earth and environmental terminology (sweet). *Computers & Geosciences*, 31(9):1119 – 1125, 2005.

[16] Laurens Rietveld and Rinke Hoekstra. Yasgui: Not just another sparql client. In *The Semantic Web: ESWC 2013 Satellite Events*, pages 78–86. Springer, 2013.

[17] M. Sabou, M. Dzbor, and C. Baldassarre. Watson: A gateway for the semantic web. In *Poster session of the European Semantic Web Conference, ESWC*, 2007.

[18] François Scharffe, Ghislain Atemezing, Raphaël Troncy, Fabien Gandon, Serena Villata, Bénédicte Bucher, Fayçal Hamdi, Laurent Bihanic, Gabriel Képéklian, Franck Cotton, Jérôme Euzenat, Zhengjie Fan, Pierre-Yves Vandenbussche, and Bernard Vatant. Enabling linked-data publication with the datalift platform. In *26th Conference on Artificial Intelligence (AAAI-12)*, 2012.

[19] Presutti Valentina and Gangemi Aldo. Content ontology design patterns as practical building blocks for web ontologies. In Spaccapietra S. et al., editor, *Proceedings of ER2008*, 2008.

[20] Patricia L Whetzel, Natalya F Noy, Nigam H Shah, Paul R Alexander, Csongor Nyulas, Tania Tudorache, and Mark A Musen. Bioportal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. *Nucleic acids research*, 39(suppl 2):W541–W545, 2011.