LOV: A gateway to reusable semantic vocabularies on the Web

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

[a] *Fujitsu (Ireland) Limited, Swords, Co. 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. One of the major barriers to the deployment of Linked Data is the difficulty data publishers have in determining which vocabularies to use for describing the semantic of data. This system report describes the Linked Open Vocabulary (LOV) as an innovative observatory of the vocabularies ecosystem. The LOV initiative gathers and makes visible indicators that have not previously been harvested, such as interconnection between vocabularies, version history, maintenance policy, past and current referent (individual or organization). Many applications of the LOV in vocabulary engineering, ontology ranking, data quality and vocabulary licenses show the benefits of such a set of vocabularies to aid the design and publication of data on the Web.

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

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 semantic vocabularies[1] ecosystem. It gathers and makes visible indicators not yet harvested before, such as interconnection between vocabularies, versioning history, maintenance policy and 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. It is the only catalog, to the best of our knowledge, that provides all types of search criteria (metadata search, within/across ontologies search), APIs, comprehensive dump file and a SPARQL endpoint access. According to the categories of ontology libraries defined in [5], LOV falls under the categories *"curated ontology directory"* and *"application platform"*.

---

[1]In this paper, "semantic vocabulary", "vocabulary" and "ontology" terms are used interchangeably. An explanation of their meaning is given in the following section.

The development of LOV has highlighted a number of interesting research challenges: *"What are the solutions for long-term vocabulary preservation on the Web?"*[**?**]. This is a particularly important problem in a distributed and uncontrolled environment where any individual can create and publish a vocabulary that can then be reused by external publishers, creating a dependency on its availability to get access to the semantic. *"How to facilitate vocabulary search and reuse"*[2,13]. To be used by a broader community, the reuse and design of vocabularies has to be facilitated by intuitive tools and methods. *"How can we harmonise the various curated vocabulary catalogues on the Web to ease their adoption?"*[1]. One of the barrier to Semantic Web adoption is the confusion in understanding and finding an appropriate vocabulary in compliance with the best practices.

The system report is structured as follows: In the next section, we describe the LOV architecture and the type of information we capture about vocabularies. This is followed by the main functionalities offered by LOV along with some usage analysis. In section 4, we explicate how LOV is used to support Data publication and Ontology Engineering process. Subsequently, we provide an overview of some applications and research projects based and motivated by LOV (Section 5). In Section 6, we report on related work and conclude in section 7.

2. A catalogue of vocabularies

A vocabulary consists of a collection of terms (classes and properties) expressed in W3C RDF, RDFS, OWL languages. A vocabulary is used to describe linked data on the Web. At the vocabulary level, LOV Bot extracts three types of information (Figure 1):

– Metadata associated to the vocabulary: this information is explicitly defined within the vocabulary to provide context, and useful data about the vocabulary. To be part of the LOV catalogue, a vocabulary must contain some minimal metadata information[**?**]. Some high level vocabularies can be reused in that purpose, such as Dublin Core to describe authors, contributors, publishers or cc[2] for the description of license using the Creative Commons rights expression language.

---

[2]http://creativecommons.org/ns#

– Inlinks vocabularies, making explicit the links to a given vocabulary based on the semantic relation of their terms.
– Outlinks vocabularies, making explicit the links from a given vocabulary based on the semantic relation of their terms.
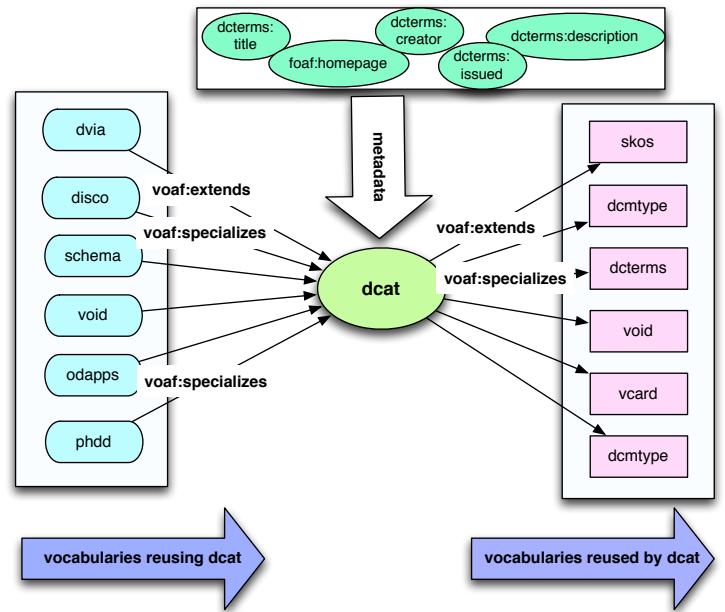

Fig. 1.: Sample type of relations and metadata used in the DCAT vocabulary.

At the Vocabulary Term level, LOV Bot extracts labels that will be used for full text search and language information. The information of language is then inserted into LOV database at the vocabulary level using the the dcterms:language property with the URI of the language in Library of Congress dataset using the ISO 639-2 code. For example, the URI associated to Japanese language is http://id.loc.gov/vocabulary/iso639-2/jpn. Currently, 91.25% (428 out of 469) of vocabularies use explicitly at least one language tag, with only 9 vocabularies with labels in a language different from English. Table 1 presents the number and percentage of the top five languages detected in LOV. We will discuss in section 7 the importance for publishers to provide multilingual vocabularies on the Web.

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

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

3. LOV functionalities

The intended purpose of LOV is to promote and facilitate the reuse of well documented vocabularies in the linked data ecosystem. To meet that goal, the LOV system offers the following features:

3.1. Vocabulary Documentation

The vocabulary collection is maintained by curators in charge of validating[3], inserting a vocabulary in the LOV ecosystem and assigning a detailed review (updated every year). When some metadata failed to be extracted automatically (such as creators of a vocabulary), curators enhance the description available on LOV system. The documentation assists any user in the task of understanding the semantics of each vocabulary term and therefore of the data using it. For instance, information about the creator and publisher is a key indication for a vocabulary user in case help or clarification is required from the author, or to assess the stability of that artifact. About 55% of vocabularies specify at least one creator, contributor or editor. We augmented this information using manually gathered information, leading to inclusion of data about the creator in over 85% of vocabularies in LOV. The database stores every different version of a vocabulary over time since its first issue. For each version, a user can access the file (even though the original online file is no longer available). An automatic script is in place to automatically check for vocabulary updates every day. To embrace the complexity of the vocabulary ecosystem and assess the impact of a modification, one needs to know in which vocabularies and datasets a particular vocabulary term is referenced. For the first time LOV provides such a vision.

Fig. 2.: Evolution of the number of vocabularies in LOV.

3.2. Data Access

LOV system (code and data) is published under Creative Commons 4.0 license[4] (CC BY 4.0). The evolution of the number of vocabularies in LOV is illustrated in figure **??**. Three options are offered for users and applications to access LOV data: 1) download data dumps of LOV catalogue in RDF Notation 3 format or the LOV catalogue and the latest version of each vocabulary in RDF N-quads format, 2) run SPARQL queries on LOV SPARQL Endpoint 3) use LOV system Application Program Interfaces (APIs) that provides the same services as user interaces.

3.3. Vocabulary Search Engine

For every vocabulary in LOV, terms (classes, properties, datatypes, instances) are indexed and a full text search feature is offered. Compared to other existing ontology search engines (cf. section 6), the Linked Open Vocabularies search engine ranking algorithm is not only based on term popularity in datasets but take as well into account its popularity within the LOV ecosystem and most importantly assigned a different score depending on which label property a searched term matched[2]. We distinguish four different elements on which a search term could match: 1) local name (URI without the namespace). Whereas a

---

[3]Before a vocabulary is inserted, LOV curators contact the authors to make sure the vocabulary is published following the best practices and contains enough metadata thus ensuring the overall LOV ecosystem quality.
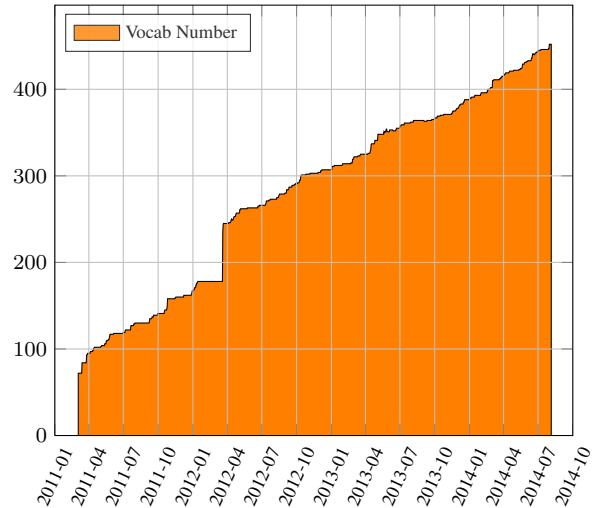
[4]https://creativecommons.org/licenses/by/4.0/

URI is not suppose to carry any meaning, it is a convention to use a compressed form of a term label to construct the local name. It becomes therefore an important artifact for term matching for which the highest score will be assigned. 2) primary labels. The highest score will also be assigned for matches on `rdfs:label`, `dce:title`, `dcterms:title`, `skos:prefLabel`. 3) secondary labels. We define as secondary labels: `rdfs:comment`, `dce:description`, `dcterms:description`, `skos:altLabel`. A medium score is assigned for matches on these properties. 4) tertiary labels. Finally all properties not falling in the previous categories are considered as tertiary labels for which a low score is assigned. As a result a term matching `rdfs:label` will have a higher score than if it matches `dcterms:comment`. Based on the different nature of these labels, we apply different indexing tokenizers and scoring methods.

Human users or agents can further narrow a search by filtering on term type (class, property, datatype, instance), vocabulary domain and vocabulary. The LOV log[5] 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 a composition 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 details the use of URIs, two words and at least three words for unique values in the LOV search log.

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

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

3.4. Community

Over the last four years, Linked Open Vocabularies initiative has gathered a community of data publishers,
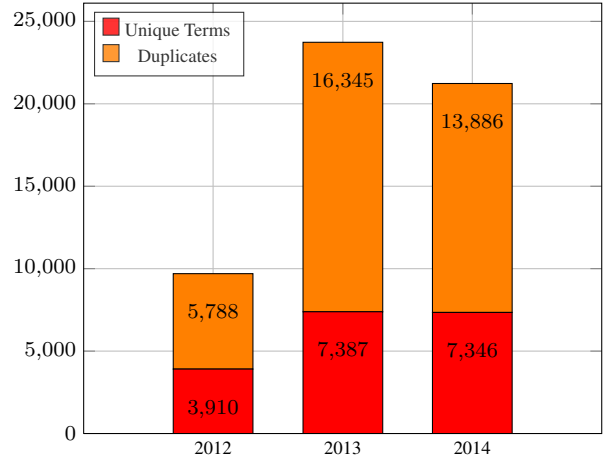

Fig. 3.: Unique and duplicate terms searched by agents/users according to LOV log in the period between 2012/01/06 until 2014/12/09.

ontology designers. LOV Google+ community[6] is now an important place to discuss, report and announce general facts related to vocabularies on the Web. Our system provides a way to suggest[7] the insertion of a new vocabulary. This feature allows a user to check what information the LOV Bot can automatically detect. From our experience with vocabulary publication, we published a handbook about Metadata recommendations for linked open data vocabularies[?].

4. LOV as a support for Data Publication and Ontology Engineering

LOV can be used in any methodology for the creation and reuse of ontologies. One of the most recent methodology for supporting collaborative development of ontologies is NeOn. 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 re-

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

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

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.

Ontology Mapping. In LOV, vocabularies rely on each other in seven different ways. These relationships are explicitly stated using VOAF vocabulary[8]. 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 SPARQL query to retrieve all the equivalent classes and properties between the vocabularies `foaf` and `dcterms`[9].

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.

```
1  SELECT DISTINCT ?elem1 ?alignment ?elem2 {
2    {?elem1 <http://www.w3.org/2002/07/owl#
        equivalentClass> ?elem2}
3    UNION {?elem1 <http://www.w3.org/2002/07/owl#
        equivalentProperty> ?elem2}
4    UNION {?elem2 <http://www.w3.org/2002/07/owl#
        equivalentClass> ?elem1}
5    UNION {?elem2 <http://www.w3.org/2002/07/owl#
        equivalentProperty> ?elem1}
6    FILTER(!isBlank(?elem2))
7    FILTER(!isBlank(?elem1))
8    ?elem1 ?alignment ?elem2.
9    ?elem1 rdfs:isDefinedBy <http://xmlns.com/
        foaf/0.1/>.
10   ?elem2 rdfs:isDefinedBy <http://purl.org/dc/
        terms/>.
11 } ORDER BY ?alignment
```

Ontology Localization. Labels in different languages are stored in the LOV endpoint. This annotations could be used when translating terms into different languages. This information could be extracted by querying the SPARQL endpoint[10] as shown in Listing 2 where all the labels defined for 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.

```
1  SELECT DISTINCT ?label2 ?element{
2    ?element rdfs:label ?label1 .
3    ?element rdfs:label ?label2 .
4    FILTER (?label1 != ?label2 ).
5    FILTER(REGEX(STR(?label1), "person", "i")).
6  } 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.

Figure 6 shows the activities within the overall Neon methodologies activity workflow that can benefit from LOV.

---

[8]http://lov.okfn.org/vocab/voaf
[9]The reader can run the query on LOV Endpoint: http://goo.gl/sTIGQ6. Prefixes are omitted for readability purpose. They can be found in LOV.

[10]Result of the query can be found at the following URL: http://goo.gl/JJCJ01

"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

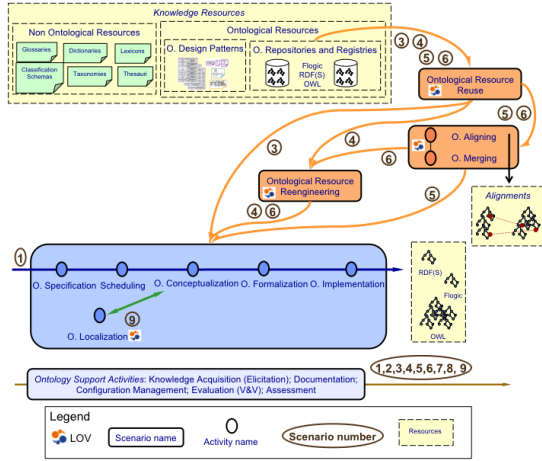Fig. 5.: Translations example for foaf:Person

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

5. LOV application and project ecosystem

LOV database as of today contains over 46,000 RDF vocabulary terms, with 28,000 properties and 18,000 classes, all accessible also via API[11]. 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`

5.1. Derived tools and applications

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

YASGUI (Yet Another SPARQL Query GUI)[13], a client-side JavaScript library that uses property, class and prefixes autocompletion using LOV API together with `http://prefix.cc` [16].

Datalift[14] 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[15].
⊙ **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 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.

5.2. 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.

Prefixes in LOV dataset is regularly mapped with namespaces in the prefix.cc service. In [1], the authors performs alignments of Qnames of vocabularies in both services, and provides different solutions to handle in case of clashes and disagreements on preferred namespaces. Both LOV and prefix.cc provide associations between prefixes and namespaces but following a different logic. The prefix.cc service supports polysemy and synonymy, and has a very loose control on its crowd-sourced information. In contrast, LOV has a much more strict policy forbidding polysemy and synonymy ensuring that each vocabulary in the LOV database is uniquely identified by a unique prefix identification allowing the usage of prefixes in

---

[11] `http://lov.okfn.org/dataset/lov/apidoc/`
[12] `https://github.com/ISA-tools/OntoMaton`
[13] `http://yasgui.laurensrietveld.nl`

[14] `http://datalift.org/en/node/24`
[15] `http://lov.okfn.org/dataset/lov/apidoc/\#lov2search`

various LOV publication URIs. This requirement leads sometimes to a situation where LOV uses prefixes different from the ones recommended by the vocabulary publishers.

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[16] 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[17] 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.

6. 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[18], ontologi.es[19], 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[20], Marine Metadata Interoperability and the SWEET

[15] ontologies[21]. 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 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 clas-

---

[16]https://zenodo.org/record/11121
[17]https://github.com/AKSW/Databugger
[18]http://vocab.org/
[19]http://ontologi.es/
[20]http://socop.oor.net/

[21]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].

sification 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.

7. Conclusion and Future work

The importance of the work presented in this paper is motivated by the difficulty for data publishers to determine which vocabularies to use to describe their data. This paper shows the benefit of using Linked Open Vocabularies, a vocabulary catalogue maintained by experts offering services such as searching, tracking, and accessing vocabularies published on the Web at any time. The adoption and integration of the LOV catalogue in different domains of vocabulary engineering, reuse and data quality are significant for the need of LOV as a central gateway about vocabulary lifecycle on the Web of Data.

This work leads to important improvements for data representation and vocabulary reuse. In particular, it would be very interesting to process not only one keyword by one but process them together. We could use the network of vocabularies defined in LOV to suggest not only terms result but graphs to represent several concepts together. Considering the relationships between vocabularies, it is then possible to tell if two vocabularies can natively be used together. Another perspective is to provide the vocabulary designer with

guidelines to determine of which type should be represented a concept (class, object/datatype property). Finally, it might be useful to automatically assign domains to vocabulary based on the metadata of a vocabulary.

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.

---

[22]https://plus.google.com/communities/
108509791366293651606

[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. D'Aquin, 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/.

[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.