

# LOV: a gateway to reusable semantic 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 <sup>a,\*</sup>, Ghislain A. Ateazing <sup>b,\*\*</sup>, María Poveda-Villalón <sup>c</sup> and Bernard Vatat <sup>d</sup>

<sup>a</sup> *Fujitsu (Ireland) Limited, Swords, Co. Dublin, Ireland*

*E-mail: pierre-yves.vandenbussche@ie.fujitsu.com*

<sup>b</sup> *Multimedia Communication Department, EURECOM, Campus SophiaTech 450, route des Chappes, 06410 Biot, France*

*E-mail: auguste.ateazing@eurecom.fr*

<sup>c</sup> *Ontology Engineering Group (OEG), Universidad Politécnica de Madrid, Madrid, Spain*

*E-mail: mpoveda@fi.upm.es*

<sup>d</sup> *Mondeca, 35 boulevard de Strasbourg, 75010 Paris, France*

*E-mail: bernard.vatat@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 for the Web of data. 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 [20] hosted by the Open Knowledge Foundation, the Linked Open Vocabularies (LOV) initiative is now standing as an innovative observatory of the semantic vocabularies<sup>1</sup> 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 January 2015) thanks to a community effort. It is the only catalogue, to the best of our knowledge, that provides all types of search criteria: metadata search, ontology search, APIs, comprehensive dump file and a SPARQL endpoint access. According to the categories of ontology libraries defined in [7], LOV falls under the categories “*curated ontology directory*” and “*application platform*”.

---

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

\*\*Corresponding author. E-mail: auguste.ateazing@eurecom.fr

<sup>1</sup>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?”[3]. 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. This creates a dependency on the original vocabulary availability as it holds the semantic of data using it. “How to facilitate vocabulary search and reuse”[4,15]. To be used by a broader community, 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 along with some high level results that the system has collected. In section 3, 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 4). In section 5, we report on related work and conclude in section 6.

## 2. System Architecture

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, LOV performs the following three main activities: 1) collecting new vocabularies from the LOV Community; 2) tracking and analysis of the LOV vocabulary catalogue; and 3) giving access to the data using various indexes and publication methods to ease data consumption. To carry out these tasks, LOV is based on a number of components depicted in figure 1, relying on existing standards and open technologies.

### 2.1. LOV Community

Over the last four years, Linked Open Vocabularies initiative has gathered a community of around 350 people interested in various domain among them: ontology engineering or data publication. LOV Google+

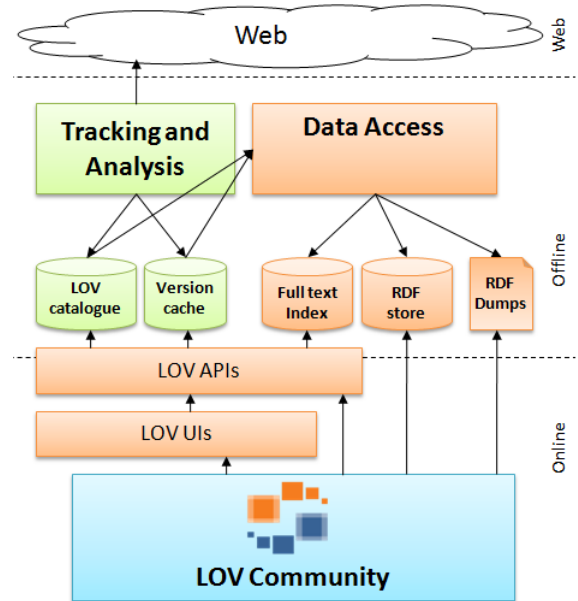


Fig. 1.: Overview of the Linked Open Vocabularies Architecture.

community<sup>2</sup> is now an important place to discuss, report and announce general facts related to vocabularies on the Web. Compared to other vocabulary catalogues (cf. section 5), LOV relies on a manual process for vocabulary insertion thus ensuring the quality of each vocabulary and therefore the quality of the overall LOV ecosystem. Suggestions are coming from the community and from inter-vocabulary reference links. Our system provides a feature to suggest<sup>3</sup> the insertion of a new vocabulary. This feature allows a user to check what information LOV can automatically detect and extract. From our experience in vocabulary publication, we published a handbook about Metadata recommendations for linked open data vocabularies to help that process [22]. We consider LOV community as a core component of the system.

### 2.2. Tracking and Analysis

The vocabulary collection is maintained by curators in charge of validating<sup>4</sup>, inserting a vocabulary in the LOV ecosystem and assigning a detailed review

<sup>2</sup><https://plus.google.com/communities/108509791366293651606>

<sup>3</sup><http://lov.okfn.org/dataset/lov/suggest/>

<sup>4</sup>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

(updated every year). Following this manual step, the *Tracking and Analysis* component takes care of dereferencing<sup>5</sup> the vocabulary, storing a version locally (in notation 3 format) and extracting relevant metadata. A vocabulary consists of a collection of terms (classes and properties) expressed in W3C RDF, RDFS, OWL languages.

At the Vocabulary level, the system extracts three types of information for each vocabulary version (figure 2):

- 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 [22]. Some high level vocabularies can be reused in that purpose, such as Dublin Core to describe authors, contributors, publishers or Creative Commons<sup>6</sup> for the description of license.
- Inlinks vocabularies, making explicit the links to another vocabulary based on the semantic relation of their terms.
- Outlinks vocabularies, making explicit the links from another vocabulary based on the semantic relation of their terms.

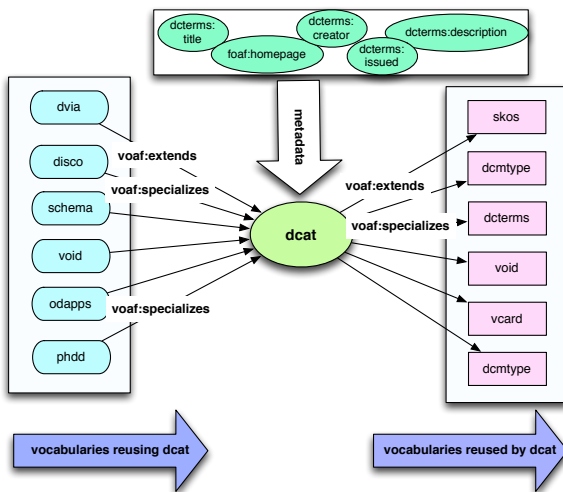


Fig. 2.: Metadata type, vocabulary inlinks and outlinks of DCAT vocabulary.

<sup>5</sup>URI is looked up over HTTP to return content in a processable format such as XML/RDF, notation 3 or turtle.

<sup>6</sup><http://creativecommons.org/ns#>

At the Vocabulary Term level, the system 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 `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 6 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.

When some metadata failed to be extracted automatically (such as creators of a vocabulary), LOV curators enhance the description available in the system. The documentation provided by LOV assists any user in the task of understanding the semantic of each vocabulary term and therefore of any 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.

### 2.3. Data Access

LOV system (code and data) is published under Creative Commons 4.0 license<sup>7</sup> (CC BY 4.0). Three methods are offered for users and applications to access LOV data: 1) query LOV search engine to find the most relevant vocabulary terms, vocabularies or agents matching keywords; 2) download data dumps of the LOV catalogue in RDF Notation 3 format or the LOV catalogue and the latest version of each vocabulary in RDF N-quads format; 3) run SPARQL queries on LOV SPARQL Endpoint; and 4) use LOV system Application Program Interfaces (APIs) which provide a full access to LOV data for software applications.

#### 2.3.1. Search Engine

For every vocabulary in LOV, terms (classes, properties, datatypes, instances) are indexed and a full text search feature is offered<sup>8</sup>. Compared to other existing ontology search engines (cf. section 5), 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 [4]. We distinguish four different label property categories on which a search term could match: 1) local name (URI without the namespace). While a 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. An example of local name matching the term “person” is `http://schema.org/Person`; 2) primary labels. The highest score will also be assigned for matches on `rdfs:label`, `dce:title`, `dcterms:title`, `skos:prefLabel` properties. An example of primary label matching the term “person” is `rdfs:label "Person"@en`; 3) secondary labels. We define as secondary label properties: `rdfs:comment`, `dce:description`, `dcterms:description`, `skos:altLabel`. A medium score is assigned for matches on these properties. An example of secondary label matching the term “person” is `dcterms:description "Examples of a Creator include a person, an organization, or`

`a service."@en`; and 4) tertiary labels. Finally all properties not falling in the previous categories are considered as tertiary labels for which a low score is assigned. An example of tertiary label matching the term “person” is `http://metadataregistry.org/uri/profile/RegAp/name "Person"@en`. As a result a term matching a value for the property `rdfs:label` will have a higher score than if it matches a value for the property `dcterms:comment`. Based on the different nature of these labels, we apply different indexing tokenizers and scoring methods.

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: Two words, three words and more and URI search number in LOV.

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<sup>9</sup> 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 has been an increase of more than 50% of search in LOV. Searched 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.

#### 2.3.2. Data Dumps

The system provides data dumps of the LOV vocabulary catalogue in RDF Notation 3 format<sup>10</sup> and the LOV catalogue along with the latest version of each vocabulary in RDF N-quads format<sup>11</sup>. As illustrated in figure 4, the RDF model mainly reuse the Data Catalogue Vocabulary (DCAT) which allows the representation of LOV catalogue as a `dcat:Catalog` composed of vocabulary entries (`dcat:CatalogRecord`) capturing information like the insertion date in LOV.

<sup>7</sup><https://creativecommons.org/licenses/by/4.0/>

<sup>8</sup><http://lov.okfn.org/dataset/lov/terms>

<sup>9</sup><http://lov.okfn.org/dataset/lov/stats/searchLog.csv>

<sup>10</sup><http://lov.okfn.org/lov.n3.gz>

<sup>11</sup><http://lov.okfn.org/lov.nq.gz>

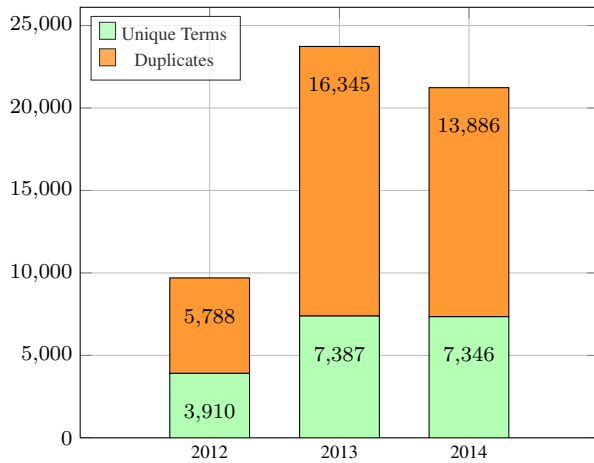


Fig. 3.: Unique and duplicate terms searched by agents/users according to LOV log in the period between 2012/01/06 until 2014/12/09.

Each entry points to the vocabulary itself represented by a sub class of `dcap:Dataset` defined in the Vocabulary of a Friend (VOAF). This artifact contains metadata extracted by LOV such as creators, first issued date, number of occurrences of the vocabulary in Linked Open datasets. Each vocabulary is then linked to its various published versions represented by the `dcap:Distribution` entity on which information such as inter vocabulary links or languages can be found.

#### 2.3.3. SPARQL Endpoint

The LOV SPARQL Endpoint<sup>12</sup> offers a complementary data access method and allows clients to pose complex queries to the server and retrieve direct answers computed over the LOV dataset. We use Jena fuseki triple store to store the N-quads file containing the LOV catalogue and the latest version of each vocabulary. This allows for the first time to query multiple vocabulary at the same time and to detect inter-vocabulary dependencies. An example of this use is explained in the ontology mapping in section 3.

#### 2.3.4. API and UI

LOV APIs give a remote access to the many functions of LOV through a set of Restful services<sup>13</sup>. The basic design requirements for these APIs is that they should allow applications to get access to the very same information humans do via the User Interfaces.

<sup>12</sup><http://lov.okfn.org/dataset/lov/sparql>

<sup>13</sup><http://lov.okfn.org/dataset/lov/apidoc/>

More precisely the APIs give access, through three different type of services (cf. figure 5), to functions related to: 1) vocabulary terms (classes, properties, datatypes and instances). With these functions, a software application can query the LOV search engine, ask for autocompletion or suggestion for misspelled terms; 2) vocabularies. A client can get access to the current list of vocabularies contained in the LOV catalogue; search for vocabularies or get autocompletion; and 3) agents. This provides a software agent with a list of all agents references in the LOV catalogue, a mean to search for an agent and get autocompletion. LOV APIs is a convenient manner to get access to the full functionality and data of LOV. It is particularly appropriate for dynamic Web applications using scripting languages such as Javascript. The APIs described above have been developed for, and following the requirements of, Ontology design and data publication tools.

Vocabulary Term (Class, Property, Datatype, Instance)		
GET	/api/v2/term/search	Search Term API v2
GET	/api/v2/term/autocomplete	Autocomplete Term API v2
GET	/api/v2/term/suggest	Suggest Term API v2
Vocabulary		
GET	/api/v2/vocabulary/list	List Vocab API v2
GET	/api/v2/vocabulary/search	Search Vocab API v2
GET	/api/v2/vocabulary/autocomplete	Autocomplete Vocab API v2
Agent		
GET	/api/v2/agent/list	List Agent API v2
GET	/api/v2/agent/search	Search Agent API v2
GET	/api/v2/agent/autocomplete	Autocomplete Agent API v2

Fig. 5.: List of APIs to access LOV data.

### 3. 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 mature 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 [8]:

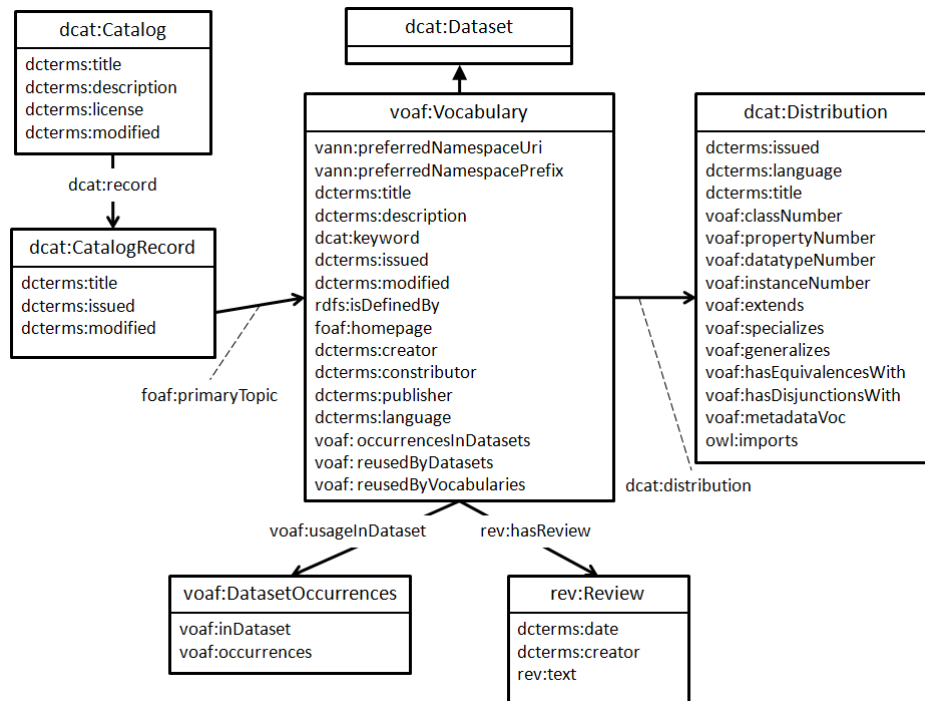


Fig. 4.: UML class diagram representation of LOV catalogue RDF schema model.

- A set of nine scenarios for building ontologies and ontology networks, emphasizing the reuse of ontological and non-ontological resources, the re-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<sup>14</sup>. 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<sup>15</sup>.

Figure 6 shows the alignments between foaf and dcterms vocabularies by mean of owl:equivalentClass and owl:equivalentProperty.

**Ontology Localization.** Labels in different languages are stored in the LOV endpoint. This annotations could be used when translating terms into dif-

<sup>14</sup><http://lov.okfn.org/vocab/voaf>

<sup>15</sup>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.



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

```
PREFIX owl:<http://www.w3.org/2002/07/owl#>
SELECT DISTINCT ?elem1 ?alignment ?elem2 {
  {?elem1 owl:equivalentClass ?elem2}
  UNION {?elem1 owl:equivalentProperty ?elem2}
  UNION {?elem2 owl:equivalentClass ?elem1}
  UNION {?elem2 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. 6.: Equivalent classes and properties between foaf and dcterms

ferent languages. This information could be extracted by querying the SPARQL endpoint<sup>16</sup> 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.

```
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 7. From that result, “Persona”@es and “Personne”@fr could be used as translations for the English term “Person” in Spanish and French respectively.

Figure 8 shows the activities within the overall Neon methodologies activity workflow that can benefit from LOV.

<sup>16</sup>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. 7.: Translations example for foaf:Person

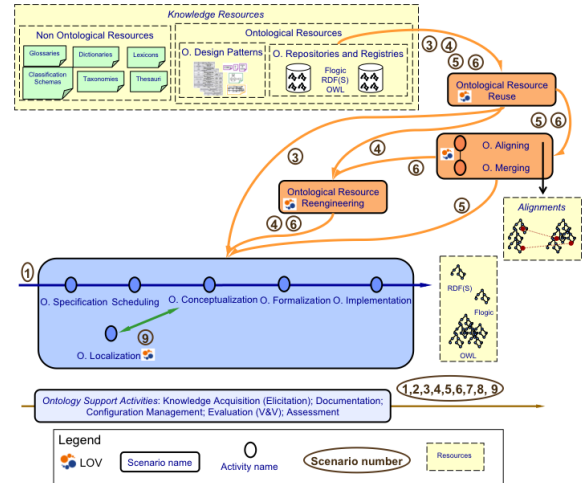


Fig. 8.: Meeting points between LOV and the NeOn methodology, derived from [8].

#### 4. LOV application and project ecosystem

As of today, LOV database contains over 46,000 RDF vocabulary terms, with 28,000 properties and 18,000 classes. The evolution of the number of vocabularies in LOV is illustrated in figure 9. LOV is supporting the emergence of a rich application ecosystem thanks to its various data access methods. We list below some tools using our system as part of their service and projects using LOV as a research artifact.

##### 4.1. Derived tools and applications

Maguire et al. [14] uses the LOV search API to implement OntoMaton<sup>17</sup>, a widget for bringing together ontology lookup and tagging within the collaborative environment provided by Google spreadsheets.

YASGUI (Yet Another SPARQL Query GUI)<sup>18</sup> is a client-side JavaScript SPARQL query editor that uses LOV API for property and class autocompletion together with <http://prefix.cc> for namespace prefix autocompletion [18].

<sup>17</sup><https://github.com/ISA-tools/OntoMaton>

<sup>18</sup><http://yasgui.laurensrietveld.nl>

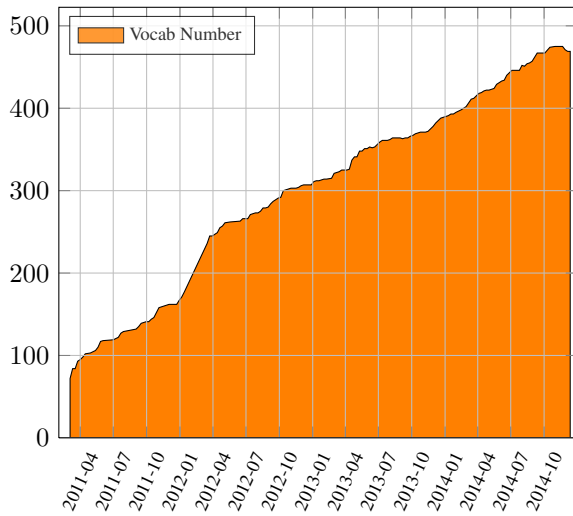


Fig. 9.: Evolution of the number of vocabularies in LOV from March 2011.

Datalift<sup>19</sup> platform [20], 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.

- 1-LOV component. This component is in charge to connect with the LOV catalogue to retrieve up-to-date ontologies using the LOV search API<sup>20</sup>.
- 2-Matching Workflow. Data2Ontology offers to map the data to LOV by automatically proposing a list of best matches.
- 3-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.

OntoWiki<sup>21</sup> facilitates the visual presentation of a knowledge base as an information map, with different

views on instance data [2]. It enables intuitive authoring of semantic content, with an inline editing mode for editing RDF content, similar to WYSIWIG for text documents. OntoWiki offers a vocabulary selection feature based on LOV.

#### 4.2. Using LOV as a Research platform

LOV vocabularies have served as object of study in [15] 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 [16] 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 various LOV publication URIs. This requirement leads sometimes to a situation where LOV uses prefixes differently 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 [4] to build a benchmark suite for ontology search and ranking. The CBR-Bench<sup>22</sup> benchmark uses eight ranking models of resources in ontologies and compare the results with ontology engineers. We plan to start a collaboration with the authors to enhance LOV search based on the study result.

In [11], 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.

Datbugger<sup>23</sup> is a test-driven data debugging framework for the Web of Data. In [12,13], the authors provide an automatic test case instantiations for all avail-

<sup>19</sup><http://datalift.org/en/node/24>

<sup>20</sup><http://lov.okfn.org/dataset/lov/apidoc/>

#lov2search

<sup>21</sup><http://ontowiki.net/>

<sup>22</sup><https://zenodo.org/record/11121>

<sup>23</sup><https://github.com/AKSW/Datbugger>



able 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. [10] 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.

## 5. Related work and Discussion

Reusing vocabularies requires searching for terms in existing specialized vocabulary catalogues or search engines on the web. While we refer the reader to [7] for a systematic survey of ontology repositories, we list below some existing catalogues relevant to find vocabularies [1]:

- *Catalogs of generic vocabularies/schemas* similar to LOV catalogue. Example of catalogues falling in this category are vocab.org<sup>24</sup>, ontologi.es<sup>25</sup>, JoinUp Semantic Assets or the Open Metadata Registry.
- *Catalogs of ontologies for a specific domain* such as biomedicine with the BioPortal [23], geospatial ontologies with SOCoP+OOR<sup>26</sup>, Marine Metadata Interoperability and the SWEET [17] ontologies<sup>27</sup>. 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 [21]. 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 [9], Watson [6,19] and FalconS [5]. These search engines crawl for data schema from RDF document on the Web. They offer a filtering based on ontol-

ogy 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, the term only 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 classification of the vocabularies. Table 3 compares 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.

## 6. Conclusion and Future work

In this system report, we presented an overview of the Linked Open Vocabularies initiative. The importance of this work is motivated by the difficulty for data publishers to determine which vocabularies to use to describe their data. The key innovations described in this article include: 1) the availability of a high quality vocabularies dataset through multiple accessing methods; 2) the vocabulary metadata curation by experts, making explicit for the first time the relationships between vocabularies and their version history; and 3) the consideration of property semantic in term search scoring.

<sup>24</sup><http://vocab.org/>

<sup>25</sup><http://ontologi.es/>

<sup>26</sup><http://socop.oor.net/>

<sup>27</sup><http://sweet.jpl.nasa.gov/2.1/>

Feature	Swoogle	Watson	Falcons	LOV
Browsing ontologies	Yes	Yes	Yes	Yes
Ontology Discovery Method	Automatic	Automatic	Automatic	Manual
Scope	SWDs	SWDs	Concepts	Ontologies
Ranking	LOD popularity	LOD popularity	LOD popularity	LOD/LOV popularity + property semantic score
Domain filtering	No	No	No	Yes
Comments and review	No	Yes	No	Only by curators
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 [7].

The adoption and integration of the LOV catalogue in applications for vocabulary engineering, reuse and data quality are significant. Linked Open Vocabularies has a central role in vocabulary life-cycle on the Web of Data. In the future, we see in particular the following directions for advancing the LOV initiative:

*From single to multi-term search.* An area which is still largely unexplored is multi-term vocabulary search. During the ontology design process, it is common to have more than 20 concepts to be represented using existing vocabularies or a new one in case there is no corresponding artifact. When we are able to search for relevant terms in LOV it is still the responsibility of the ontology designer to understand the complex relationships between all these terms and come up with a coherent ontology. We could use the network of vocabularies defined in LOV to suggest not only terms result but graphs to represent several concepts together.

*Multilingual vocabularies.* There is a need for vocabularies to support more languages. Labels are the main entrypoint to a vocabulary and their associated language is the key. Only 18% of LOV vocabularies use a different language than English. Multilingualism is important at least for two reasons: 1) the most obvious one is allowing users to search, query and navigate vocabularies in their native language; and 2) translating is a process through which the quality of a vocabulary can only improve. Looking at a vocabulary through the eyes of other languages and identifying the difficulties of translation helps to better outline the initial concepts and if necessary refine or revise them.

Hence multilingualism and translation should be native, built-in features of any vocabulary construction, not a marginal task.

*Query extension and rewriting.* Another research perspective is SPARQL query extension and rewriting based on Linked Vocabularies. Using the inter-vocabulary relationships we could transform the query to use the same semantic (same vocabulary terms) as the data source(s) to query.

## 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+<sup>28</sup>, all the editors and publishers of vocabularies who trust in the LOV catalogue.

## References

- [1] Ghislain Auguste Atemezang, Bernard Vatan, Raphaël Troncy, and Pierre-Yves Vandenbussche. Harmonizing services for

<sup>28</sup><https://plus.google.com/communities/108509791366293651606>

- LOD vocabularies: a Case Study. In *Workshop on Semantic Web Enterprise Adoption and Best Practice (WaSABi)*, Sydney, Australia, 2013.
- [2] Sören Auer, Sebastian Dietzold, and Thomas Riechert. Ontowiki—a tool for social, semantic collaboration. In *The Semantic Web-ISWC 2006*, pages 736–749. Springer, 2006.
- [3] Thomas Baker, Pierre-Yves Vandenbussche, and Bernard Vatant. Requirements for vocabulary preservation and governance. *Library Hi Tech*, 31(4):657–668, 2013.
- [4] Anila Sahar Butt, Armin Haller, and Lexing Xie. Ontology search: An empirical evaluation. In *The Semantic Web-ISWC 2014*, pages 130–147. Springer, 2014.
- [5] 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.
- [6] M. D’Aquin, C. Baldassarre, L. Gridinoc, M. Sabou, S. Angeletou, and E. Motta. Watson: Supporting next generation semantic web applications. 2007.
- [7] Mathieu D’Áquin 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.
- [8] Maria del Carmen. Suárez-Figueroa. *NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse*. PhD thesis, Universidad Politécnica de Madrid, Spain, June 2010. <http://oa.upm.es/3879/>.
- [9] 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.
- [10] Guido Governatori, Ho-Pun Lam, Antonino Roto, Serena Villata, Ghislain Ateamezing, and Fabien Gandon. Checking licenses compatibility between vocabularies and data. In *Fifth International Workshop on Consuming Linked Data (COLD2014)*.
- [11] 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.
- [12] 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.
- [13] Dimitris Kontokostas, Patrick Westphal, Sören Auer, Sebastian Hellmann, Jens Lehmann, Roland Cornelissen, and Arapali 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.
- [14] 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.
- [15] María Poveda Villalón, Mari Carmen Suárez-Figueroa, and Asunción Gómez-Pérez. The landscape of ontology reuse in linked data. 2012.
- [16] 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.
- [17] 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.
- [18] Laurens Rietveld and Rinke Hoekstra. Yagui: Not just another sparql client. In *The Semantic Web: ESWC 2013 Satellite Events*, pages 78–86. Springer, 2013.
- [19] M. Sabou, M. D’Aquin, and C. Baldassarre. Watson: A gateway for the semantic web. In *Poster session of the European Semantic Web Conference, ESWC*, 2007.
- [20] François Scharffe, Ghislain Ateamezing, Raphaël Troncy, Fabien Gandon, Serena Villata, Bénédicte Bucher, Fayçal Hamdi, Laurent Bihanic, Gabriel Képékian, 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.
- [21] 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.
- [22] Pierre-Yves Vandenbussche and Bernard Vatant. Metadata recommendations for linked open data vocabularies. Technical report, 2012.
- [23] 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.