

# Linked Open Vocabularies (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 Vandebussche<sup>a,\*</sup>, Ghislain A. Atemezing<sup>b</sup>, María Poveda-Villalón<sup>c</sup> and Bernard Vatant<sup>d</sup>

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

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

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

*E-mail: ghislain.atemezing@mondeca.com*

<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.vatant@mondeca.com*

**Abstract.** One of the major barriers to the deployment of Linked Data is the difficulty that data publishers have in determining which vocabularies to use to describe the semantics of data. This system report describes Linked Open Vocabularies (LOV), a high quality catalogue of reusable vocabularies for the description of data on the Web. The LOV initiative gathers and makes visible indicators that have not been previously harvested such as the interconnections between vocabularies, version history along with past and current referent (individual or organization). LOV goes beyond existing Semantic Web vocabulary search engines and takes into consideration the value's property type, matched with a query, to improve vocabulary terms scoring. By providing an extensive range of data access methods (SPARQL endpoint, API, data dump or UI), we try to facilitate the reuse of well-documented vocabularies in the Linked Data ecosystem. We conclude that the adoption in many applications and methods of LOV shows the benefits of such a set of vocabularies and related features to aid the design and publication of data on the Web.

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

## 1. Introduction

The last two decades has seen the emergence of a “Semantic Web” enabling humans and computer systems to exchange data with unambiguous, shared meaning. This vision has been supported by World Wide Web Consortium (W3C) Recommendations such as the Resource Description Framework (RDF), RDF-Schema and the Web Ontology Language (OWL). Thanks to a major effort in publishing data following Semantic Web and Linked Data principles [5], there

are now tens of billions of facts spanning hundreds of linked datasets on the Web covering a wide range of topics. Access to the data is facilitated by portals (such as Datahub<sup>1</sup> or UK Government Data<sup>2</sup>) or by publishing the data directly (such as New York Times<sup>3</sup>).

Despite the enormous volumes of data now available on the Web, Linked Data suffers from low community interest in vocabulary<sup>4</sup> management in favour of the

---

<sup>1</sup><http://datahub.io/>

<sup>2</sup><http://data.gov.uk/>

<sup>3</sup><http://data.nytimes.com/>

<sup>4</sup>We use the terms “semantic vocabulary”, “vocabulary” and “ontology” interchangeably.

---

\*Thanks to Amélie Gyrard, Thomas Francart, Thérèse Rogez and Anthony McCauley for their help on the project.

data itself. A vocabulary consists of classes, properties and datatypes that define the meaning of data. RDF Vocabularies are themselves expressed as Linked Data. When a vocabulary is not published or not available any more, humans and machines do not have access to the definition of the terms used to qualify the data. This breaks the semantic interoperability, one of the fundamentals of the Semantic Web.

Started in March 2011, as part of the DataLift research project [24] and hosted by the Open Knowledge Foundation, the Linked Open Vocabularies (LOV) initiative is now an innovative observatory of the semantic vocabularies ecosystem. It gathers and makes visible indicators not yet harvested before, such as the interconnections between vocabularies, versioning history along with past and current referent (individual or organization). The intended purpose of LOV is to promote and facilitate the reuse of well documented vocabularies in the Linked Data ecosystem. The number of vocabularies indexed by LOV is constantly growing (511 as of June 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, a comprehensive dump file and SPARQL endpoint access. According to the categories of ontology libraries defined by D’Aquin and Noy [9], LOV falls under the categories “*curated ontology directory*” and “*application platform*”.

This report is structured as follows: In the next section, we provide statistics on the usage of LOV. In section 3, we describe the components and features that constitute LOV. Thereafter, in section 4, we explain how LOV is used to support Data Publication and Ontology Engineering processes. Subsequently, we provide an overview of some applications and research projects based and motivated by the LOV system (section 5). In section 6, we report on related work. Discussion about the limitations and further development of LOV is presented in section 7. We finally reach our conclusions in section 8.

## 2. LOV state

The LOV dataset consists of 511 vocabularies as of June 2015. Figure 1 depicts the evolution of the number of vocabularies inserted in the LOV dataset since March 2011. The addition of new vocabularies to LOV has been fairly constant with two outstanding events: 1) an increase beginning of 2012 corresponding to the deployment of LOV version 2 which automates most

of the vocabulary analyses; and 2) a small decrease and plateau beginning of 2015 corresponding to the deployment of LOV version 3. At that time we were considering removing offline vocabularies but finally decided to keep them with a special flag making LOV the last semantic continuity for datasets referencing unreachable vocabularies.

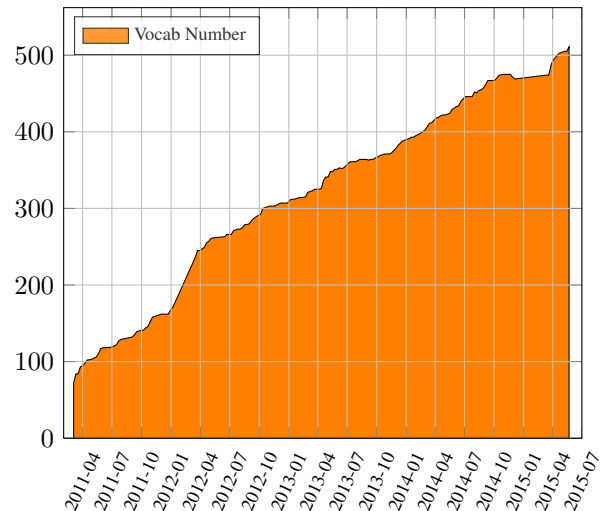


Fig. 1.: Evolution of the number of vocabularies in LOV from March 2011 to June 2015.

By observing the vocabularies contained in LOV as a whole, we can extract some interesting facts about Semantic Web adoption and dynamics. In figure 2, we present a distribution of LOV vocabularies by creation date, noting the main W3C Recommendation languages used RDF, RDFS and OWL. The distribution follows a bell curve with the peak in 2011. A decrease of vocabulary creation does not necessarily mean a decrease in use of the technology but rather that the existing vocabularies now cover a large part of the semantic description needed. Figure 3 shows the last modification date of LOV vocabularies over time demonstrating a living ecosystem.

Overall, LOV vocabularies contain 20,000 classes and almost 30,000 properties with a median number per vocabulary of 9 and 17 respectively. Table 1 presents a breakdown of LOV dataset content by vocabulary element type.

Out of 511 vocabularies, 66.14% explicitly use the English language for labels/comments. Table 2 presents the number and percentage of the top five languages detected in LOV. Figure 4 shows the distribu-

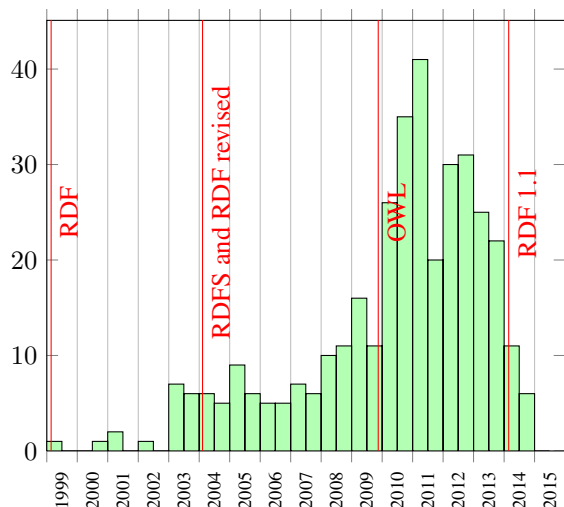


Fig. 2.: Vocabularies distribution by creation date.

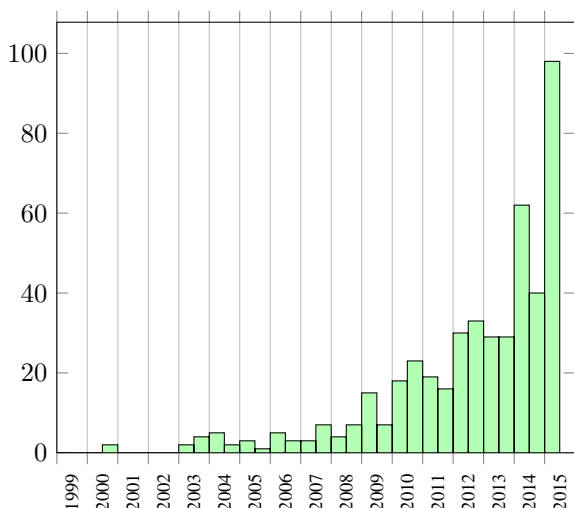


Fig. 3.: Vocabularies distribution by last modification date.

tion of vocabularies per number of languages explicitly used: 27.98% vocabularies still do not provide any language information and only 14.68% of vocabularies are multilingual. In total, 45 Languages are used by vocabularies in LOV. We will discuss the importance of providing multilingual vocabularies in section 8.

From January to June 2015, more than 1.4 million searches were conducted on LOV<sup>5</sup>. A breakdown of

<sup>5</sup>This figure includes searches from the API and UI as well as searches with and without keywords such as “all agents that partic-

Type	Count	Median per vocab
Classes	20,034	9
Properties	29,925	17
Instances	5,232	0
Datatypes	101	0

Table 1: LOV vocabulary elements statistics.

Language	Nb Vocabs	% Vocabs (out of 511)
English	338	66.14%
French	37	7.24%
Spanish	25	4.89%
German	19	3.72%
Italian	18	3.52%

Table 2: Top five languages and percentage detected in the LOV catalogue. Some vocabularies can make use of multiple languages.

searches per element type is provided in table 3. We can see that the new feature (from LOV version 3) of agent search (used for instance to identify experts of a domain in semantic web vocabulary design and publication) is the most used. Searches that include keywords (and not only filters) are concerned with vocabulary terms. Table 4 presents the top 10 terms searched from January to June 2015. Although most of the searches are performed through the User Interface, an application ecosystem using LOV APIs has surfaced as shown in the figure 5.

Vocabulary Term	Nb searches	% searches
set	7,092	8.79%
domain	2,518	3.12%
some	2,473	3.06%
status	1,486	1.84%
iso 639	1,389	1.72%
same	1,285	1.59%
state	1,235	1.53%
supports	1,145	1.42%
start	887	1.1%
space	864	1.07%

Table 4: Top 10 terms searched from January to June 2015 by users in LOV.

Over the last four years, the Linked Open Vocabularies initiative has gathered a community of around 480

ipated in vocabulary design and publication in the geo-location domain”.

Element Type	Nb searches	% searches	Nb searches with keyword	% searches with keyword
Term	205,682	14.19%	80,728	92.84%
Vocabulary	178,837	12.34%	5,918	6.81%
Agent	1,064,597	73.47%	306	0.35%

Table 3: Type of elements searched from January to June 2015 by users in LOV for all searches and those with keyword.

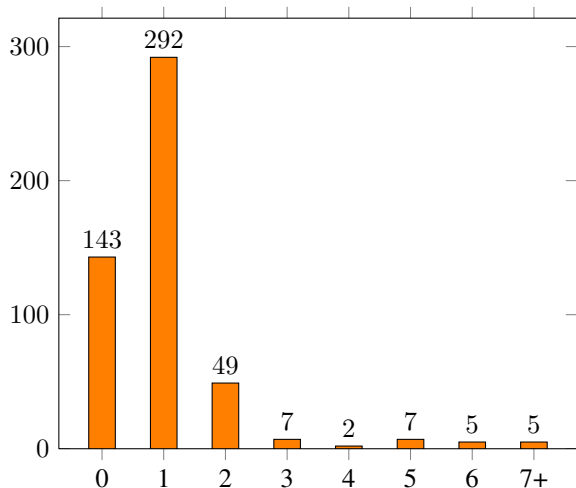


Fig. 4.: Distribution of the vocabularies per language count.

people interested in various domains including ontology engineering or data publication. the LOV Google+ community<sup>6</sup> is now an important place to discuss, report and announce general facts related to vocabularies on the Web. The LOV dataset itself referenced 389 resources of type persons and 111 of type organization participating in vocabulary design and/or publication.

### 3. System Components and Features

LOV architecture is composed of four main components (cf. figure 6): 1) *Tracking and Analysis*. Responsible for checking for any vocabulary version update and analysing vocabularies' specific features. 2) *Curation*. Ensuring the high quality of the LOV dataset through methods for the community to suggest or edit the catalogue. 3) *Data Access*. Provides access to the data through a large range of methods and protocols

<sup>6</sup><https://plus.google.com/communities/108509791366293651606>

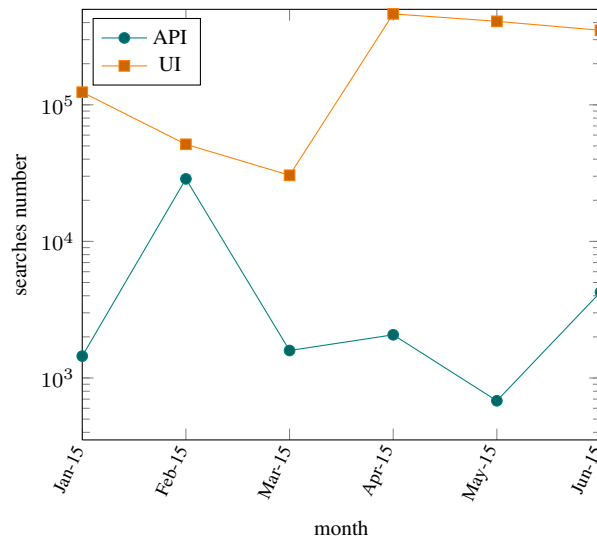


Fig. 5.: Evolution of the number of searches through UI and API methods from January to June 2015.

to facilitate the use of LOV dataset and 4) *LOV User Interfaces and Application Program Interfaces*. Each component provides a set of features detailed in the following subsections.

#### 3.1. Tracking and Analysis

The *Tracking and Analysis* component takes care of dereferencing<sup>7</sup> LOV vocabularies, storing a version locally (in notation 3 format) and extracting relevant metadata.

##### 3.1.1. Vocabulary Level Analysis

At the Vocabulary level, the system extracts three types of information for each vocabulary version (figure 7):

- Metadata associated to the vocabulary. This information is explicitly defined within the vocabu-

<sup>7</sup>URI is looked up over HTTP to return content in a processable format such as XML/RDF, notation 3 or turtle.

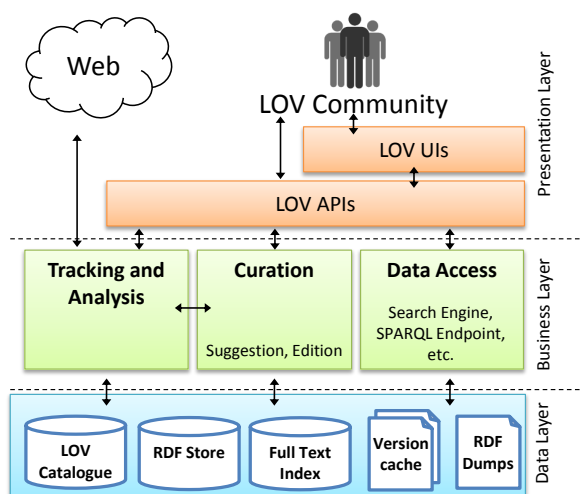


Fig. 6.: Overview of the Linked Open Vocabularies Architecture.

lary to provide context and useful data about the vocabulary. Some high level vocabularies can be reused for that purpose, such as Dublin Core<sup>8</sup> to describe authors, contributors, publishers or Creative Commons<sup>9</sup> for the description of a license.

- Inlinks vocabularies, making explicit the links from another vocabulary based on the semantic relation of their terms.
- Outlinks vocabularies, making explicit the links to another vocabulary based on the semantic relation of their terms.

There are many ways two vocabularies can be interlinked. Let's consider two vocabularies  $V1$  and  $V2$  such that  $V1$  contains a class  $c1$  and a property  $p1$  and  $V2$  contains a class  $c2$  and a property  $p2$ . Relationships between these two vocabularies can be of the following types (the lines and numbers in brackets correspond to real examples presented in listing 1):

**Metadata.** some terms from  $V2$  are reused to provide metadata about  $V1$  (lines 1 to 2).

**Import.** some terms from  $V2$  are reused with  $V1$  to capture the semantic of the data (lines 3 to 4).

**Specialization.**  $V1$  defines some subclasses or sub-properties (or local restrictions) of  $V2$  (lines 5 to 8).

**Generalization.**  $V1$  defines some superclasses or super-properties of  $V2$  (lines 9 to 11).

<sup>8</sup><http://purl.org/dc/terms/>

<sup>9</sup><http://creativecommons.org/ns#>

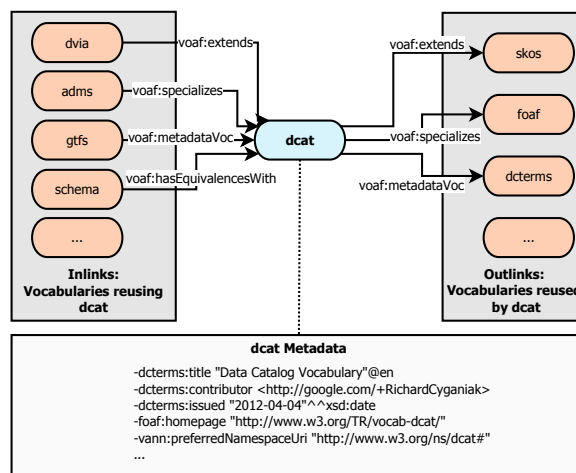


Fig. 7.: Metadata type, vocabulary inlinks and outlinks of DCAT vocabulary.

**Extension.**  $V1$  extends the expressivity of  $V2$  (lines 12 to 15).

**Equivalence.**  $V1$  declares some equivalent classes or properties with  $V2$  (lines 16 to 20).

**Disjunction.**  $V1$  declares some disjunct classes with  $V2$  (lines 21 to 23).

These relationships, with the exception of *Import* which is represented by `owl:imports`, are captured by the Vocabulary of a Friend<sup>10</sup> (VOAF). Table 5 presents a breakdown of the occurrences of each relation in LOV.

Inter vocabulary relationship	Nb Relations
voaf:metadataVoc	2,637
voaf:specializes	1,269
voaf:extends	1,031
owl:imports	373
voaf:hasEquivalencesWith	201
voaf:generalizes	57
voaf:hasDisjunctionsWith	16

Table 5: Inter-vocabularies relationship types and their number of occurrences in LOV.

### 3.1.2. Vocabulary Term Level Analysis

At the vocabulary term level, the system extracts two types of information:

- term type (class, property, datatype or instance defined in the namespace of the vocabulary)

<sup>10</sup><http://lov.okfn.org/vocommons/voaf/>

Listing 1: Examples of Inter-vocabulary relationships.

```

1 # Metadata
2 <http://www.w3.org/2004/02/skos/core> dct:title "SKOS Vocabulary"@en
3 # Import - V1 imports V2
4 <http://purl.org/NET/c4dm/event.owl> owl:imports <http://www.w3.org/2006/time>
5 # Specialization - c1 is subclass of c2
6 <http://open.vocab.org/terms/Birth> rdfs:subClassOf <http://purl.org/NET/c4dm/event.owl#Event>
7 # Specialization - p1 is subproperty of p2
8 <http://purl.org/spar/fabio/hasEmbargoDate> rdfs:subPropertyOf <http://purl.org/dc/terms/date>
9 # Generalization - c1 has for narrower match c2
10 <http://semanticweb.cs.vu.nl/2009/11/sem/Place> skos:narrowMatch
11 <http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing>
12 # Extension - p1 is inverse of p2
13 <http://vivoweb.org/ontology/core#translatorOf> owl:inverseOf <http://purl.org/ontology/bibo/translator>
14 # Extension - p1 has for domain c2
15 <http://xmlns.com/foaf/0.1/based_near> rdfs:domain <http://www.w3.org/2003/01/geo/wgs84_pos#SpatialThing>
16 # Equivalence - p1 is equivalent to p2
17 <http://lstdis.cs.uga.edu/projects/semdis/opus#journal_name> owl:equivalentProperty
18 <http://purl.org/net/nknouf/ns/bibtex#hasJournal>
19 # Equivalence - c1 is equivalent to c2
20 <http://www.loc.gov/mads/rdf/v1#Language> owl:equivalentClass <http://purl.org/dc/terms/LinguisticSystem>
21 # Disjunction - c1 is disjoint with c2
22 <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#TimeInterval>owl:disjointWith
23 <http://www.ontologydesignpatterns.org/ont/dul/ontopic.owl#SubjectSpace>

```

which is provided for indexing by the search engine so a user can filter a search based on this information.

- term natural language annotations with their predicate (e.g. `rdfs:label "Temperature"@en`). This information is provided for indexing by the search engine and will later be used (cf. section 3.3.1) in the ranking algorithm.

The information concerning vocabulary terms use in Linked Open Data, also named "popularity," is used in LOV search results ranking as explained in section 3.3.1. This information is not natively present in the vocabularies and can not be inferred from the LOV dataset. The LODStats project gathers comprehensive statistics about RDF datasets [11]. LOV regularly fetches LODStats raw data<sup>11</sup> using the Vocabulary of Interlinked Datasets (VoID) [1] and the Data Cube vocabulary. We pre-process LODStats data before inserting it to LOV. Indeed, There are many duplicates in LODStats representing in fact the same vocabulary URI (e.g., foaf has three different records<sup>12</sup>, and has to be mapped to a single entry in LOV.

<sup>11</sup>We keep synchronised the statistics available at: <http://stats.lod2.eu/rdfdocs/void>. Unfortunately this file has been unavailable since June 2014 which explains some differences between the statistics we use and LODStats.

<sup>12</sup><http://stats.lod2.eu/vocabularies?search=foaf>

### 3.2. Curation

The vocabulary collection is maintained by curators in charge of validating, inserting a vocabulary in the LOV ecosystem, and assigning a detailed review.

#### 3.2.1. Vocabulary Insertion

Compared with other vocabulary catalogues (cf. section 6), LOV relies on a semi-automated process for vocabulary insertion illustrated in figure 8. Whereas an automated process puts the emphasis on the volume, in our process, we focus on the quality of each vocabulary and therefore the quality of the overall LOV ecosystem. Suggestions come from the community and from inter-vocabulary reference links. Our system provides a feature to suggest<sup>13</sup> the insertion of a new vocabulary. This feature allows a user to check what information the LOV application can automatically detect and extract. LOV curators then check if the vocabulary falls in the scope of LOV and if it meets LOV vocabulary quality requirements to be reused:

1. a vocabulary should be written in RDF and be dereferenceable;
2. a vocabulary should be parsable without error (warning tolerated);
3. all vocabulary terms (classes, properties and datatypes) in a vocabulary should have an `rdfs:label`;

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

4. a vocabulary should refer to and reuse relevant existing ones;
5. a vocabulary should provide some metadata about the vocabulary itself (a minima a title).

If a suggested vocabulary meets these criteria it is then inserted in the LOV catalogue. During this process, LOV curators keep the authors informed and help them to improve their vocabulary quality. As a result of our experience in vocabulary publication, we are now able to publish a handbook about metadata recommendations for Linked Open Data vocabularies to help in publishing well documented vocabularies [27].

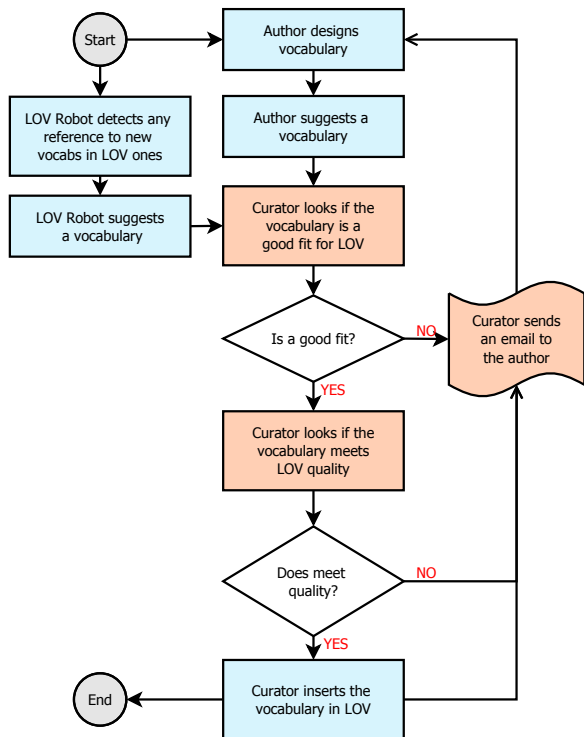


Fig. 8.: Curation workflow for vocabulary insertion.

### 3.2.2. Vocabulary Review

When automatic extraction of metadata fails, LOV curators enhance the description available in the system and notify the vocabulary authors. The documentation provided by the LOV application assists any user in the task of understanding the semantics 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 vocab-

ularies specify at least one creator, contributor or editor. We augment this information using manually gathered information, leading to the inclusion of data about the creator in over 85% of vocabularies in LOV. The database stores every version of a vocabulary since its first issue. For each version, a user can access the file (particularly useful when the original online file is no longer available). As illustrated in figure 9, a script is in place to automatically check for vocabulary updates on a daily basis. If a new version has been detected, the version is stored locally, the statistics about that vocabulary recomputed and a notification to the curators issued. Similarly we ensure that curated review for each vocabulary is less than one year old by sending curators a notification when a vocabulary review is older than eleven months. In both case, curators update the vocabulary review accordingly.

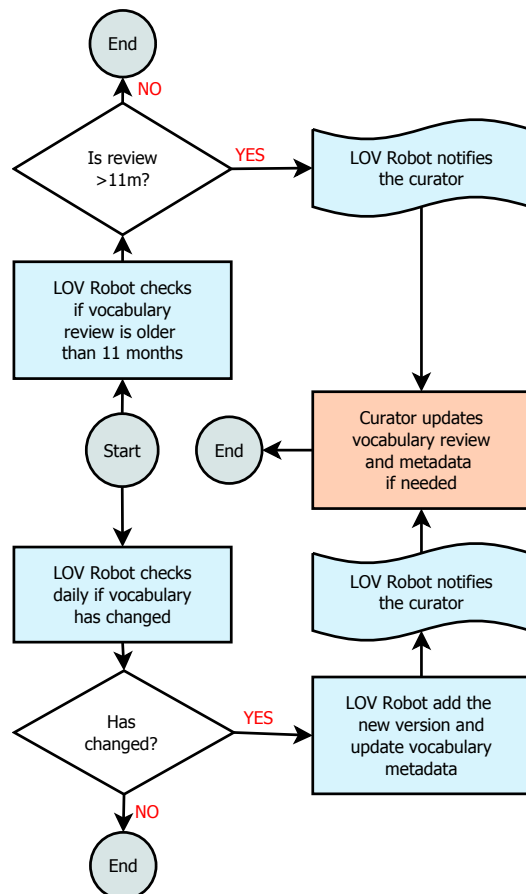


Fig. 9.: Curation workflow for systematic vocabulary review.



### 3.3. Data Access

LOV system (code and data) is published under Creative Commons 4.0 license<sup>14</sup> (CC BY 4.0). Four methods are offered for users and applications to access the LOV data: 1) query the LOV search engine to find the most relevant vocabulary terms, vocabularies or agents matching keywords and/or filters; 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 the LOV SPARQL Endpoint; and 4) use the LOV system Application Program Interface (API) which provides a full access to LOV data for software applications.

#### 3.3.1. Search Engine

In [6], Butt *et al.* compare eight different ranking methods grouped in two categories for querying vocabulary terms:

- Content-based Ranking Models: tf-idf, BM25, Vector Space Model and Class Match Measure.
- Graph-based Ranking Models: PageRank, Density Measure, Semantic Similarity Measure and Betweenness Measure

Based on their findings, we defined a new ranking method adapting *Term frequency inverse document frequency* (tf-idf) to the graph-structure of vocabularies. When compared to the other methods, tf-idf takes into account the relevance and importance of a resource to the query when assigning a weight to a particular vocabulary for a given query term. We reuse the augmented frequency variation of term frequency formula to prevent a bias towards longer vocabularies. Because of the inherit graph structure of vocabularies, tf-idf needs to be tailored so that instead of using a word as the basic unit for measuring, we are considering a vocabulary term  $t$  in a vocabulary  $V$  as the measuring unit. The equation 1 presents the adaptation of tf-idf to vocabularies (a definition of the variables used in this paper's equations is provided in table 6).

$$tf(t, V) = 0.5 + \frac{0.5 * f(t, V)}{\max \{f(t_i, V) : t_i \in V\}} \quad (1)$$

$$idf(t, \mathcal{V}) = \log \frac{N}{|\{V \in \mathcal{V} : t \in V\}|}$$

<sup>14</sup><https://creativecommons.org/licenses/by/4.0/>

Variable	Description
$\mathcal{V}$	Set of Vocabularies
$V$	A vocabulary: $V \in \mathcal{V}$
$N$	Number of vocabularies in $\mathcal{V}$
$t$	A vocabulary term URI (class, property, instance or datatype): $t \in V, t \in URI$
$Q$	Query string
$q_i$	Query term $i$ of $Q$
$\sigma_V$	Set of matched URIs for $Q$ in $V$
$\sigma_V(q_i)$	Set of matched URIs for $q_i$ in $V$ : $\forall t_i \in \sigma_V, t_i \in V, t_i$ matches $q_i$
$p$	A term predicate: $p \in URI$
$\mathcal{D}$	Set of Datasets
$D$	A Dataset: $D \in \mathcal{D}$
$M(t_i)$	Number of Datasets: $D$ in $\mathcal{D}, t_i \in D$

Table 6: Definition of the variables used in the equations.

As highlighted in [6] and [23], the notion of popularity of a vocabulary term across the LOD datasets set  $\mathcal{D}$  is significantly important. In equation 2 we introduce a popularity measure, function of the normalisation of the frequency  $f(t, \mathcal{D})$  of a term URI  $t$  in the set of datasets  $\mathcal{D}$  and the normalisation of the number of datasets in which a term URI appears  $M(t) : t \in \mathcal{D}$ . By using maximum in the normalisation we emphasise the most used terms, resulting in a consensus within the community. This measure will give a higher score to terms that are often used in datasets and across a large number of datasets.

$$pop(t, \mathcal{D}) = \frac{f(t, \mathcal{D})}{\max \{f(t_i, \mathcal{D}) : t_i \in \mathcal{D}\}} * \frac{M(t)}{\max \{M(t_i) : t_i \in \mathcal{D}\}} \quad (2)$$

When compared with RDF datasets, best practices about vocabulary publication makes their structure consensual and stable. It becomes then intuitive to assign more importance to a vocabulary term matching a query on the value of the property `rdfs:label` than `dcterms:comment`. The equation 3 extends the lucene based search engine `elasticsearch` inner field-length norm `lengthNorm(field)`, which attaches a higher weight to shorter fields, by combining it with a property-level boost `boost(p(t))`. Using this property-level boost we can assign a different score depending on which label property a query term matched. We



distinguish four different label property categories on which a query term could match:

- Local name (URI without the namespace). While a URI is not supposed 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`.
- 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`.
- Secondary labels. We define as secondary label the following 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`.
- 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`.

$$\begin{aligned} norm(t, V) = & lengthNorm(field) \\ & * \prod_{p \in V} boost(p(t)) \end{aligned} \quad (3)$$

For every vocabulary in LOV, terms (classes, properties, datatypes, instances) are indexed and a full text search feature is offered<sup>15</sup>. Human users or agents can further narrow a search by filtering on term type (class, property, datatype, instance), language, vocabulary domain and vocabulary.

The final score of  $t$  for a query  $Q$  (equation 4) is a combination of tf-idf, the importance of label properties of  $t$  on which query terms matched and the popularity of that term in the LOD dataset. While the

factorisation of tf-idf and field normalisation factor is common for search engine ranking<sup>16</sup>, we add a fourth parameter - the popularity - as it is fundamental in the Semantic Web. Indeed, the intention of LOV is to foster the reuse of consensual vocabularies that become de facto standards. The popularity metric provides an indication on how widely a term is already used (in frequency and in number of datasets using it). We therefore add this new factor specific to the Semantic Web to the scoring equation.

$$\begin{aligned} Score(t, Q) = & tf(t, V) * idf(t, \mathcal{V}) \\ & * norm(t, V) * pop(t, \mathcal{D}) \end{aligned} \quad (4)$$

$$: \forall t \{ \exists q_i \in Q : t \in \sigma_V(q_i) \}$$

### 3.3.2. Data Dumps

The system provides two data dumps, one containing the LOV vocabulary catalogue only in RDF Notation 3 format<sup>17</sup> and another one containing the LOV catalogue along with the latest version of each vocabulary and the statistics of use in LOD in RDF N-quads format<sup>18</sup> (keeping each vocabulary in a separate named graph). As illustrated in figure 10, the RDF model mainly reuses the Data Catalogue Vocabulary (DCAT) which allows the representation of the LOV catalogue as a `dcat:Catalog` composed of vocabulary entries (`dcat:CatalogRecord`) capturing information like the insertion date in LOV. Each entry point to the vocabulary itself is represented by a sub class of `dcat:Dataset` defined in the Vocabulary Of A Friend (VOAF). This artifact contains metadata extracted by the LOV application such as creators, first issued date, number of occurrences of the vocabulary in Linked Open Data. Each vocabulary is then linked to its various published versions represented by the `dcat:Distribution` entity on which information such as inter vocabulary links or languages can be found.

### 3.3.3. SPARQL Endpoint

The LOV SPARQL Endpoint<sup>19</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 the Jena

<sup>15</sup><http://lov.okfn.org/dataset/lov/terms>

<sup>16</sup>See [elasticsearch documentation: http://bit.ly/1e37sFL](http://bit.ly/1e37sFL)

<sup>17</sup><http://lov.okfn.org/lov.n3.gz>

<sup>18</sup><http://lov.okfn.org/lov.nq.gz>

<sup>19</sup><http://lov.okfn.org/dataset/lov/sparql>

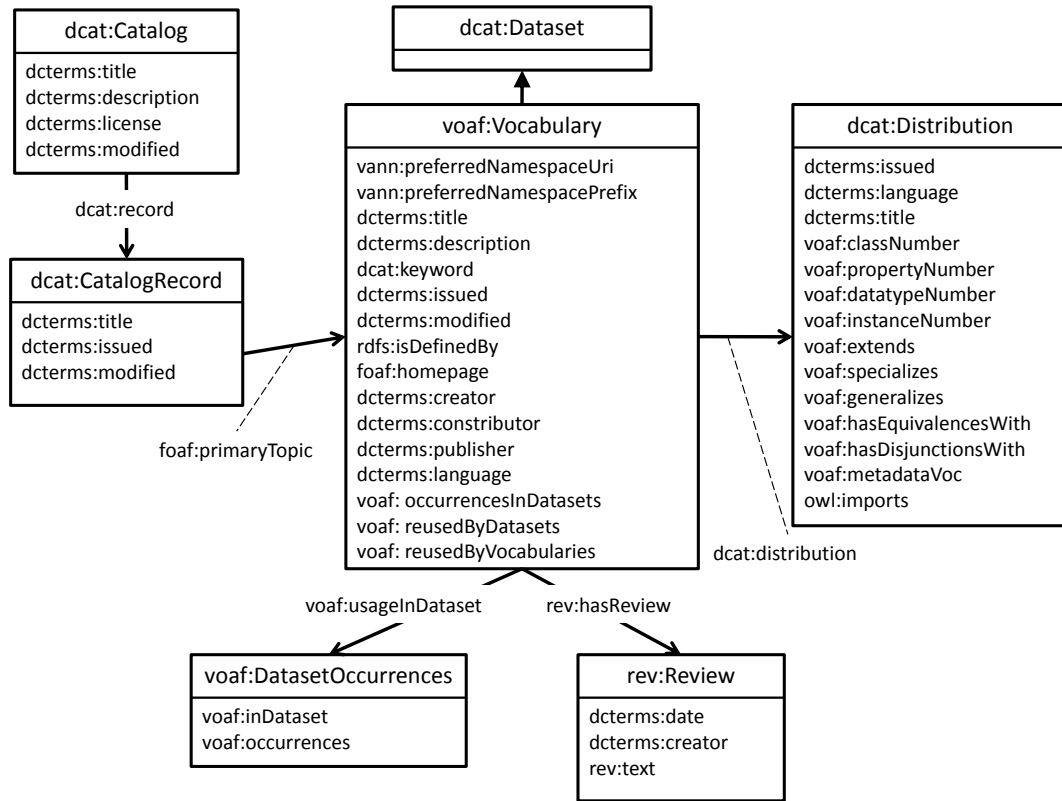


Fig. 10.: UML class diagram representation of LOV catalogue RDF schema model.

fuseki triple store to store the N-quads file containing the LOV catalogue and the latest version of each vocabulary. We believe this is the first service to allow users to query multiple vocabularies at the same time and to detect inter-vocabulary dependencies.

### 3.4. LOV Application Program Interfaces and User Interfaces

LOV APIs give a remote access to the many functions of LOV through a set of RESTful services<sup>20</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. More precisely the APIs give access, through three different services (cf. figure 11), to functions related to:

- Vocabulary terms (classes, properties, datatypes and instances). With these functions, a software application can query the LOV search engine,

ask for auto-completion or a suggestion for misspelled terms.

- Vocabularies. A client can get access to the current list of vocabularies contained in the LOV catalogue; search for vocabularies, get auto-completion or obtain all details about a vocabulary.
- Agents. This provides a software agent with a list of all agents references in the LOV catalogue, a means to search for an agent, get auto-completion and details about an agent.

LOV APIs are a convenient means to access 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 follow the requirements of, Ontology Design and Data Publication tools.

The LOV Website offers intuitive navigation within the vocabularies catalogue. It allows users to explore vocabularies, vocabulary terms, agents and languages, and to see the connections between these entities. For instance, a user can look for experts in *geography* and

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

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
GET	/api/v2/vocabulary/info	Info 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
GET	/api/v2/agent/info	Info Agent API v2

Fig. 11.: List of APIs to access LOV data.

geometry domains<sup>21</sup>. We use d3<sup>22</sup> javascript library to display charts and make the navigation more interactive; for example, we use the star graph representation to display incoming and outgoing links between vocabularies (cf. figure 12).

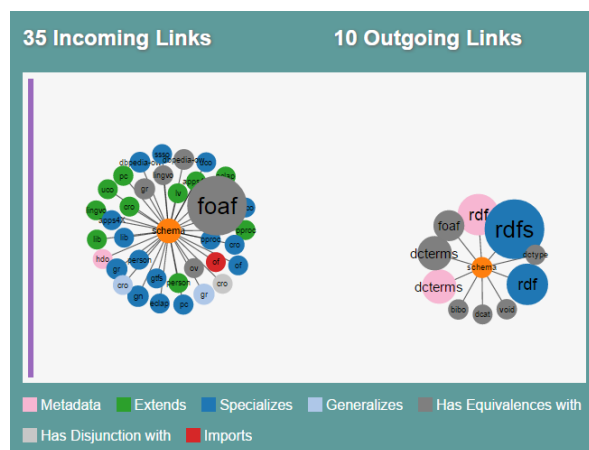


Fig. 12.: Schema.org vocabulary incoming and outgoing links graphical representation as displayed in the UI.

#### 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 mature methodologies for supporting ontology development is NeOn. Scenario-based, it supports the collaborative aspects of ontology development and reuse, as well as the dynamic evolution of ontology networks in distributed environments [26].

Based on the NeOn Methodology's glossary of activities for building ontologies, the LOV system is relevant in three activities:

**Ontology Search.** LOV's primary feature is the search of vocabulary terms. These vocabularies are categorised according to the domain they address. In this way, the LOV system 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 the 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 equivalent classes and equivalent properties among two ontologies. Listing 2 shows the SPARQL query to retrieve all the equivalent classes between the vocabularies foaf and schema.org<sup>23</sup>. Table 7 shows the alignments between foaf and schema.org vocabularies.

elem1	alignment	elem2
foaf:Document	owl:equivalentClass	schema:CreativeWork
foaf:Image	owl:equivalentClass	schema:ImageObject
foaf:Person	owl:equivalentClass	schema:Person

Table 7: Equivalent classes and properties between foaf and schema.org.

<sup>21</sup><http://lov.okfn.org/dataset/lov/agents?tag=Geography,Geometry>

<sup>22</sup><http://d3js.org/>

<sup>23</sup>The reader can run the query on LOV Endpoint: <http://bit.ly/1Lqybcu>.

Listing 2: SPARQL query asking for all the equivalent classes and properties between the vocabularies foaf and schema.

```

1 SELECT DISTINCT ?elem1 ?alignment ?elem2{
2 GRAPH ?g{
3   ?elem1 rdfs:isDefinedBy
4     <http://xmlns.com/foaf/0.1/>.
5   {?elem1 owl:equivalentClass ?elem2}
6   UNION {?elem2 owl:equivalentClass ?elem1}
7   FILTER (STRSTARTS (STR(?elem2),
8     "http://schema.org/"))
9   ?elem1 ?alignment ?elem2.
10  }}

```

## 5. LOV Adoption

LOV supports 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 project.

### 5.1. Derived tools and applications

Maguire et al. [17] use LOV search API to implement OntoMaton<sup>24</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>25</sup> is a client-side JavaScript SPARQL query editor that uses the LOV API for property and class auto-completion together with <http://prefix.cc> for namespace prefix auto-completion [22]. YASGUI is itself reused by LOV for its SPARQL Endpoint User Interface.

The Datalift<sup>26</sup> platform [24], 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. The Data2Ontology module takes as input a “raw RDF”, a dataset that has been converted directly from legacy format to triples. The goal is to help publishers reuse existing ontologies for converting their dataset with easy discovery and interlinking.

OntoWiki<sup>27</sup> facilitates the visual presentation of a knowledge base as an information map, with different views on instance data [3]. 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.

Furthermore, we can mention the ProtégéLOV<sup>28</sup>, a plug-in for the Protégé editor tool [13] that aims at improving the development of lightweight ontologies by reusing existing vocabularies at a low fine grained level. The tool searches for a term in LOV via APIs and provides three actions if the term exists : (i) replaces the selected term in the current ontology, (ii) adds the `rdfs:subClassOf` axiom and (iv) adds the `owl:equivalentClass`.

### 5.2. Using LOV as a Research platform

LOV has served as the object of studies in [18] where Poveda-Villalón *et al.* analysed trends in ontology reuse methods. In addition, the LOV dataset has been used in order to analyse the occurrence of good and bad practices in vocabularies [19].

Prefixes in the LOV dataset are regularly mapped with namespaces in the prefix.cc service. In [2], the authors perform alignments of Qnames of vocabularies in both services and provide different solutions to handle clashes and disagreements between preferred namespaces. Both LOV and prefix.cc provide associations between prefixes and namespaces but follow 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.

The LOV query log covering the period between 2012-01-06 and 2014-04-16 has been used in [6] to build a benchmark suite for ontology search and ranking. The CBRBench<sup>29</sup> benchmark uses eight ranking models of resources in ontologies and compares the results with ontology engineers. Our vocabulary term ranking method relies on and extends the outcome of this work.

In [15], the authors provide a 5 star rating for RDF vocabulary publication to foster interoperability, query federation and better interpretation of data on the Web similar to the 5 stars rating for Linked Open Data. Based on LOV insertion criteria, all vocabularies must

<sup>24</sup><https://github.com/ISA-tools/OntoMaton>

<sup>25</sup><http://legacy.yasgui.org/>

<sup>26</sup><http://datalift.org/>

<sup>27</sup><http://ontowiki.net/>

<sup>28</sup><http://labs.mondeca.com/protolov/>

<sup>29</sup><https://zenodo.org/record/11121>

be 5 stars using this ranking and must provide further quality attributes imposed by LOV to facilitate vocabulary reuse.

RDFUnit<sup>30</sup> is a test-driven data debugging framework for the Web of data. In [16], the authors provide an automatic test case for all available schema registered with LOV. Vocabularies are used to encode semantics to domain specific knowledge to check the quality of data.

Governatori et al. [14] analyse the current use of licenses in vocabularies on the Web based on the LOV catalogue in order to propose a framework to detect incompatibilities between datasets and vocabularies.

## 6. Related work

Reusing vocabularies requires searching for terms in existing specialised vocabulary catalogues or search engines on the Web. While we refer the reader to [9] for a systematic survey of ontology repositories, we list below some existing catalogues relevant to finding vocabularies:

- *Catalogues of generic vocabularies/schemas* similar to LOV catalogue. Example of catalogues falling in this category are vocab.org<sup>31</sup>, ontologi.es<sup>32</sup>, JoinUp Semantic Assets or the Open Metadata Registry.
- *Catalogues of ontologies for a specific domain* such as biomedicine with the BioPortal [28], geospatial ontologies with SOCoP+OOR<sup>33</sup>, Marine Metadata Interoperability and the SWEET [21] ontologies<sup>34</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.
- *Catalogues of ontology Design Patterns (ODP)* focused on reusable patterns in ontology engineering [20]. The submitted patterns are small pieces of vocabularies that can further be integrated or linked with other vocabularies. ODP does not provide a search function for specific terms as is the case with Swoogle or Watson.

- *Search Engines of ontology terms.* Among ontology search engines, we can cite: Swoogle [12], Watson [8,10] and FalconS [7]. These search engines crawl for data schema from RDF documents on the Web. They offer 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). While in Swoogle the ranking score is displayed, Watson shows the language of the resource and the size. However, none of these services provide any relationship between the related ontologies, nor any domain classification of the vocabularies. Table 8 presents a summary of key features of LOV with respect to Swoogle, Watson and Falcons.
- *Datasets and Vocabularies statistics.* One of the major project on which LOV relies, LODStats [11] makes a bridge between datasets and vocabularies. LODStats gathers up to 32 different statistical criteria based on a statement-stream-based approach for RDF datasets in Datahub<sup>35</sup>. LODStats maintains a comprehensive statistics on vocabularies terms (i.e. classes, properties) defined and used in a dataset. Schmachtenberg et al. [25] present a survey based on a large-scale Linked Data crawl from March 2014 to analyse the differences in best practices adoption across different application domains. Their results concerning the most used vocabularies (e.g., foaf, dcterms, skos, etc.) and the adoption of well-known vocabularies are inline with the findings of this paper.

## 7. Discussion and Future Work

With many users, both humans and applications, and several years of development, LOV is a mature system that offers a wide range of services for RDF vocabulary reuse.

While LOV is delivering quality vocabularies, it presents some limitations:

- LOV focuses on a subset of vocabularies for the description of RDF data. It does not include any *Value Vocabularies* such as SKOS thesaurus that would benefit from LOV services to encourage their reuse. This limitation is due to the rather

<sup>30</sup><https://github.com/AKSW/RDFUnit>

<sup>31</sup><http://vocab.org/>

<sup>32</sup><http://ontologi.es/>

<sup>33</sup><https://ontohub.org/socop>

<sup>34</sup><http://sweet.jpl.nasa.gov/>

<sup>35</sup><http://datahub.io/>

Feature	Swoogle	Watson	Falcons	LOV
Browsing ontologies	Yes	Yes	Yes	Yes
Ontology discovery method	Automatic	Automatic	Automatic	Automatic and manual
Scope	SWDs	SWDs	Concepts	Vocabularies
Ranking	LOD popularity	LOD popularity	LOD popularity	LOD/LOV popularity + label's property type
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 8: Comparison of LOV with respect to Swoogle, Watson and Falcons; adapted from the framework presented by d'Aquin and Noy [9]. SWD stands for Semantic Web Document

small team of curators (4 at the date of writing this paper). Although we automated all the processes and analyses we could, the support to vocabulary authors is far from negligible.

- LOV relies on third projects to get the valuable information of vocabulary usage in published datasets. At the moment, the popularity information coming from LODStats does not take into account the most recent interest (e.g. Schema.org) in publishing RDF data using markup language. As a consequence, the popularity measure is incomplete and does not represent all possible use of a vocabulary.

From the study of LOV as a dynamic ecosystem we can draw the following lessons learned:

- There is a need for vocabularies to support more languages. Labels are the main entry point to a vocabulary and their associated language is the key. Only 15% of LOV vocabularies make use of more than one language. 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.

- There is at the moment no solution for long-term vocabulary preservation on the Web [4]. This is a particularly important problem in a distributed and uncontrolled environment where any individual can create and publish a vocabulary. Third parties can reuse such vocabularies and therefore create a dependency on the original vocabulary availability as it retains the semantics of the data. This issue weakens the Semantic Web foundations.

In the future, we see in particular the following directions for advancing the LOV initiative:

- An area that is still largely unexplored is multi-term vocabulary search. During the ontology design process, it is common to have more than 20 concepts represented using existing vocabularies or a new one in case there is no corresponding artifact. While 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 a list of terms but graphs to represent several concepts together.
- We would like to provide more vocabulary based services such as vocabulary matching to help the authors adding more relationships to other re-

lated vocabularies. Vocabulary checking is another service the community is asking for. We could integrate useful applications to LOV such as: Vapour<sup>36</sup>, RDF Triple-Checker<sup>37</sup> or OOPS!<sup>38</sup>.

- 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.

## 8. Conclusion and Future work

In this system report we presented an overview of the Linked Open Vocabularies initiative, a high quality catalogue of reusable vocabularies for the description of data on the Web. 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 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.

The adoption and integration of the LOV catalogue in applications for vocabulary engineering, reuse and data quality are significant. LOV has a central role in vocabulary life-cycle on the Web of data as highlighted by the W3C<sup>39</sup>.

## 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; the Spanish project BabelData (TIN2010-17550) and Fujitsu Laboratories Limited. The Linked Open Vocabularies initiative is graciously hosted by the Open Knowledge Foundation. We would like to thank all the members of LOV community, all the editors and publishers of vocabularies who trust in LOV catalogue. A special thank to Phil Archer for proofreading this paper.

<sup>36</sup><http://validator.linkeddata.org/vapour>

<sup>37</sup><http://graphite.ecs.soton.ac.uk/checker/>

<sup>38</sup><http://oops.linkeddata.es/>

<sup>39</sup><http://www.w3.org/2013/data/>

## References

- [1] Keith Alexander and Michael Hausenblas. Describing linked datasets-on the design and usage of void, the vocabulary of interlinked datasets. In *In Linked Data on the Web Workshop (LDOW 09), in conjunction with 18th International World Wide Web Conference (WWW 09)*. Citeseer, 2009.
- [2] Ghislain Auguste Atemezang, Bernard Vatant, Raphaël Troncy, and Pierre-Yves Vandenbussche. Harmonizing services for LOD vocabularies: a Case Study. In *Proceedings of the Workshop on Semantic Web Enterprise Adoption and Best Practice Co-located with 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 22, 2013.*, Sydney, Australia, 2013.
- [3] 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.
- [4] Thomas Baker, Pierre-Yves Vandenbussche, and Bernard Vatant. Requirements for vocabulary preservation and governance. *Library Hi Tech*, 31(4):657–668, 2013.
- [5] Tim Berners-lee. Linked data. design issues for the world wide web. *World Wide Web Consortium*. <http://www.w3.org/DesignIssues/LinkedData.html>, 2006.
- [6] Anila Sahar Butt, Armin Haller, and Lexing Xie. Ontology search: An empirical evaluation. In *The Semantic Web-ISWC 2014*, pages 130–147. Springer, 2014.
- [7] 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.
- [8] Mathieu d’Aquin, Claudio Baldassarre, Laurian Gridinoc, Marta Sabou, Sofia Angeletou, and Enrico Motta. Watson: Supporting next generation semantic web applications. 2007.
- [9] Mathieu d’Aquin 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.
- [10] Mathieu d’Aquin, Marta Sabou, Martin Dzbor, Claudio Baldassarre, Laurian Gridinoc, Sofia Angeletou, and Enrico Motta. Watson: A gateway for the semantic web. In *Poster session of the European Semantic Web Conference, ESWC, 2007*.
- [11] Jan Demter, Sören Auer, Michael Martin, and Jens Lehmann. LODStats – An Extensible Framework for High-performance Dataset Analytics. In *18<sup>th</sup> International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, pages 353–362, Galway, Ireland, 2012.
- [12] Timothy W. Finin, Li Ding, Rong Pan, Anupam Joshi, Pranam Kolari, Akshay Java, and Yun Peng. Swoogle: Searching for knowledge on the semantic web. In *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, pages 1682–1683, 2005.
- [13] Nuria García-Santa, Ghislain Auguste Atemezang, and Boris Villazón-Terrazas. The protégé LOVPlugin: Ontology Access and Reuse for Everyone. In *The 12th Extended Semantic Web Conference (ESWC2015)*.
- [14] Guido Governatori, Ho-Pun Lam, Antonino Rotolo, Serena Villata, Ghislain Atemezang, and Fabien Gandon. Checking licenses compatibility between vocabularies and data. In *Proceedings of the 5th International Workshop on Consuming*



- Linked Data (COLLD 2014) co-located with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Italy, October 20, 2014.*, 2014.
- [15] 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.
- [16] 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.
- [17] 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*, 29(4):525–527, 2013.
- [18] María Poveda-Villalón, Mari Carmen Suárez-Figueroa, and Asunción Gómez-Pérez. The landscape of ontology reuse in linked data. 2012.
- [19] María Poveda-Villalón, Bernard Vatant, María Carmen Suárez-Figueroa, and Asunción Gómez-Pérez. Detecting good practices and pitfalls when publishing vocabularies on the web. 2013.
- [20] Valentina Presutti and Aldo Gangemi. Content ontology design patterns as practical building blocks for web ontologies. In Spaccapietra S. et al., editor, *ER '08: Proceedings of the 27th International Conference on Conceptual Modeling*, pages 128–141. Springer-Verlag, 2008.
- [21] Robert G. Raskin and Michael J. Pan. Knowledge representation in the semantic web for earth and environmental terminology (sweet). *Computers & Geosciences*, 31(9):1119 – 1125, 2005.
- [22] Laurens Rietveld and Rinke Hoekstra. Yasgui: Not just another sparql client. In *The Semantic Web: ESWC 2013 Satellite Events*, pages 78–86. Springer, 2013.
- [23] Johann Schaible, Thomas Gottron, Stefan Scheglmann, and Ansgar Scherp. Lover: support for modeling data using linked open vocabularies. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, pages 89–92. ACM, 2013.
- [24] François Scharffe, Ghislain Ateazing, 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.
- [25] Max Schmachtenberg, Christian Bizer, and Heiko Paulheim. Adoption of the linked data best practices in different topical domains. In *The Semantic Web—ISWC 2014*, pages 245–260. Springer, 2014.
- [26] 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/>.
- [27] Pierre-Yves Vandenbussche and Bernard Vatant. Metadata recommendations for linked open data vocabularies. Technical report, 2012.
- [28] 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.