

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 Vandenbussche ^{a,*}, Ghislain A. Ateazing ^b, María Poveda-Villalón ^c and Bernard Vatan ^d

^a *Fujitsu (Ireland) Limited, Swords, Co. Dublin, Ireland*

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

^b *Multimedia Communication Department, EURECOM, Campus SophiaTech 450, route des Chappes, 06410 Biot, France*

E-mail: auguste.ateazing@eurecom.fr

^c *Ontology Engineering Group (OEG), Universidad Politécnica de Madrid, Madrid, Spain*

E-mail: mpoveda@fi.upm.es

^d *Mondeca, 35 boulevard de Strasbourg, 75010 Paris, France*

E-mail: bernard.vatant@mondeca.com

Abstract. One of the major barriers to the deployment of Linked Data is the difficulty 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 interconnection between vocabularies, version history, maintenance policy, along with past and current referent (individual or organization). The LOV goes beyond existing Semantic Web search engines and takes into consideration the value's property type, matched with a query, to improve 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

Started in March 2011, as part of the DataLift research project [23] and hosted by the Open Knowledge Foundation, the Linked Open Vocabularies (LOV) initiative is now an innovative observatory of the semantic vocabularies¹ ecosystem. It gathers and makes vis-

ible indicators not yet harvested before, such as interconnection between vocabularies, versioning history, maintenance policy along with past and current referent (individual or organization) if any. 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, comprehensive dump file and a SPARQL endpoint access. According to the categories of ontology libraries defined by D'Aquin and Noy [8], LOV falls under the categories “*curated ontology directory*” and “*application platform*”.

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

¹In 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 the LOV has highlighted a number of interesting research challenges: “*What are the solutions 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 that can then be reused by external publishers. This creates a dependency on the original vocabulary availability as it retains the semantics of the data. “*How to facilitate vocabulary search and reuse*” [5,17]. To be used by a broader community, reuse and design of vocabularies have to be facilitated by intuitive tools and methods. “*How can we harmonise the various curated vocabulary catalogues on the Web to ease their adoption?*” [2]. One of the barriers to Semantic Web adoption is the confusion related to understanding and finding an appropriate vocabulary in compliance with best practice.

This 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 4, we explain how the LOV is used to support the Data Publication and Ontology Engineering process. 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 and reach out conclusions in section 8.

2. LOV state

As of June 2015, the LOV database contains 511 vocabularies. Figure 1 depicts the evolution of the number of vocabularies inserted in the LOV dataset. The addition of new vocabularies to LOV has been fairly constant with two outstanding events: 1) the increase beginning of 2012 corresponds to the deployment of LOV version 2 making automatic 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 thinking about removing offline vocabularies but finally decided to keep them with a special flag making LOV the last semantic continuity for datasets referencing a dead vocabularies.

Looking at the creation date of the vocabularies registered in LOV, illustrated in Figure 2, it informs on the adoption of Semantic Web technologies. The apex of this bell curve is happening in 2011. The figure put as well this in the perspective of the W3C recommendation effort of RDF, RDFS and OWL languages. A decrease of vocabulary creation does not mean a decrease

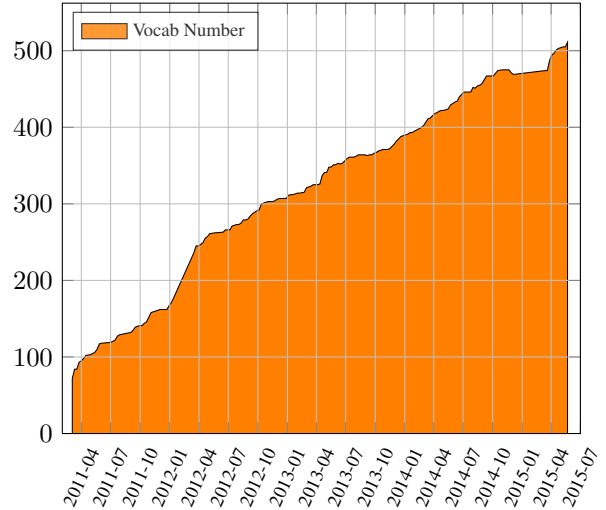


Fig. 1.: Evolution of the number of vocabularies in LOV from March 2011 to June 2015.

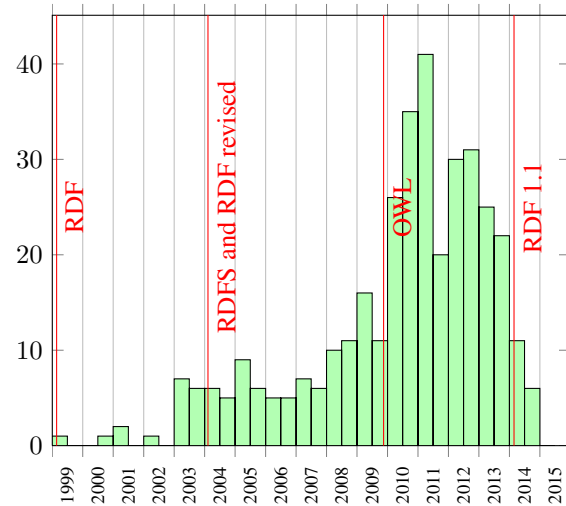


Fig. 2.: Vocabularies distribution by creation date.

in use of the technology but rather than the coverage of existing vocabularies now covers a large part of the semantic description need. This is as well confirmed by figure 3 that presents the last modification date of LOV vocabularies demonstrating a living ecosystem where vocabularies are not static. The median number of versions per vocabulary is 2. Where 98.23% of vocabularies makes use of RDFS in addition to RDF, only 45.99% of them makes use of OWL. Datasets usage are added to 127 vocabularies, from which 98 vocabu-

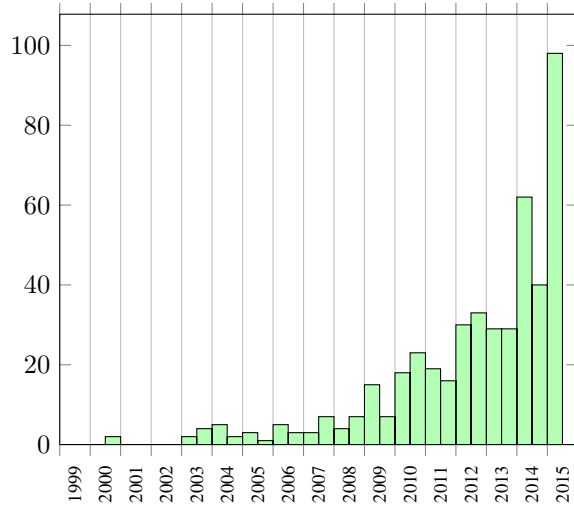


Fig. 3.: Vocabularies distribution by last modification date.

laries are used in at least 2 datasets and 29 of them are described in only one dataset.

Overall, LOV dataset contains almost 50,000 RDF vocabulary classes and properties with a median number of 9 and 17 respectively per vocabulary. Table 1 presents a breakdown of LOV dataset content by vocabulary element type. Out of 511 vocabularies, 66.14% use explicitly the English language for labels/comments. Table 2 presents the number and percentage of the top five languages detected in the LOV. Figure 4 shows the distribution of vocabularies per number of languages explicitly used: 27.98% vocabularies still does not provide any language information and only 14.68% of vocabularies are multilingual. 45 Languages are represented in LOV. We will discuss in section 8 the importance for publishers to provide multilingual vocabularies on the Web.

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.

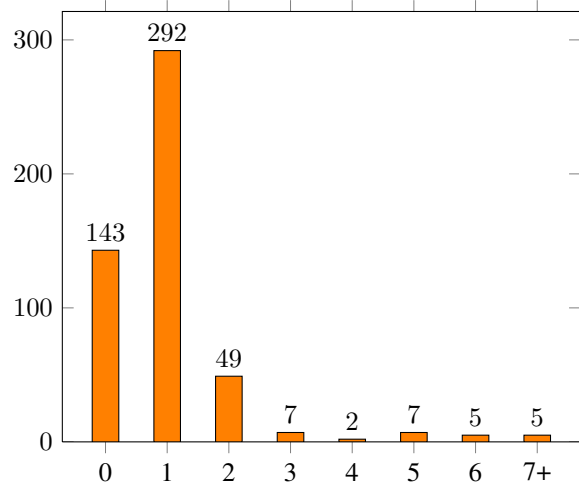


Fig. 4.: Distribution of the vocabularies per language count.

From January to June 2015, more than 1.4 million searches have been issued on LOV². A breakdown of the search per element type is provided in table 3. We can see that the new feature 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 includes keywords (and not only filters) concerns essentially vocabulary terms. Table 4 presents the top 10 terms searched over that time period. Although most of the searches are performed through the User Interface, an application ecosystem using LOV APIs is surfaced as shown in the figure 5.

Over the last four years, the Linked Open Vocabularies initiative has gathered a community of around 470 people interested in various domains among them: ontology engineering or data publication. LOV Google+

²This figures include API and UI searches as well as searches without keywords such as “all agents that participated 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 logs in total and with keyword.

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 logs.

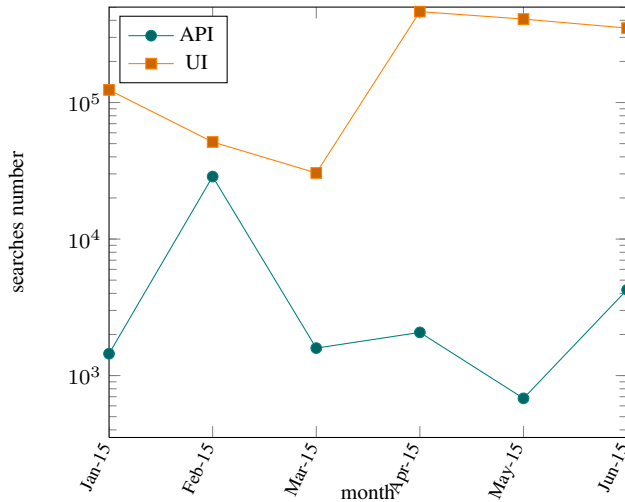


Fig. 5.: Evolution of the number of searches through UI and API methods from January and June 2015.

community³ is now an important place to discuss, report and announce general facts related to vocabularies on the Web. LOV dataset itself referenced 389 resources of type persons and 111 of type organization participating in vocabulary design and/or publication.

³<https://plus.google.com/communities/108509791366293651606>

3. System Components and Features

Figure 6 shows an overview of the LOV components architecture. It is composed of four main components: 1) *Tracking and Analysis*. Responsible to check for any vocabulary version update and analysing vocabularies specific features. 2) *Curation*. Ensure the high quality of the LOV dataset through methods for the community to suggest or edit the LOV catalogue. 3) *Data Access*. Consists in providing access to the data through a large range of methods and protocols to facilitate the use of LOV dataset and 4) *LOV User Interfaces and Application Program Interfaces*. Each components provides a set of features detailed in the following subsections.

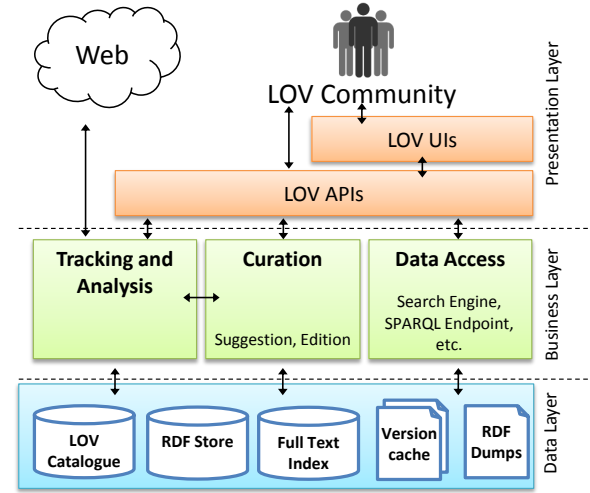


Fig. 6.: Overview of the Linked Open Vocabularies Architecture.

3.1. Tracking and Analysis

The *Tracking and Analysis* component takes care of dereferencing⁴ LOV vocabularies, storing a ver-

⁴URI is looked up over HTTP to return content in a processable format such as XML/RDF, notation 3 or turtle.

sion 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.

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 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 [26]. Some high level vocabularies can be reused for that purpose, such as Dublin Core to describe authors, contributors, publishers or Creative Commons⁵ 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 as $V1$ 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 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 in complement 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).

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).

Where `dul namespace` is `http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#` and `ontopic` is `http://www.ontologydesignpatterns.org/ont/dul/ontopic.owl#`

These relationships, with the exception of *Import* which is represented by `owl:imports`, are captured by the Vocabulary of a Friend⁶ (VOAF) created for LOV representation needs. Table 5 presents a breakdown of the occurrences of each relation in LOV.

Inter vocabulary relationship	Nb Relations
voaf:metadataVoc	2637
voaf:specializes	1269
voaf:extends	1031
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.

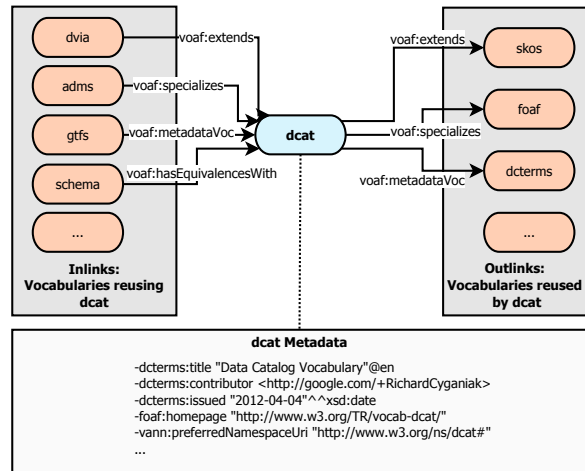


Fig. 7.: Metadata type, vocabulary inlinks and outlinks of DCAT vocabulary.

3.1.2. Vocabulary Term Level Analysis

At the Vocabulary Term level, the system extracts labels that will be used for full text search and language information.

⁵<http://creativecommons.org/ns#>

⁶<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://lsdis.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>

```

3.2. Curation

The vocabulary collection is maintained by curators in charge of validating⁷, inserting a vocabulary in the LOV ecosystem and assigning a detailed review.

3.2.1. Vocabulary Insertion

Compared to other vocabulary catalogues (cf. section 6), LOV relies on a semi-automated process for vocabulary insertion illustrated in figure 8. Whereas an automated process put 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 are coming from the community and from inter-vocabulary reference links. Our system provides a feature to suggest⁸ 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 basic vocabulary quality to be reused:

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

⁷Before a vocabulary is inserted, LOV curators contact the authors to make sure the vocabulary is published following the best practices and contains enough metadata

⁸<http://lov.okfn.org/dataset/lov/suggest/>

4. a vocabulary should refer and reuse relevant existing ones

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 [26].

3.2.2. Vocabulary Review

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 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 vocabularies specify at least one creator, contributor or editor. We augmented this information using manually gathered information, leading to the inclusion of data about the creator in over 85% of vocabularies in the 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). An automatic script is

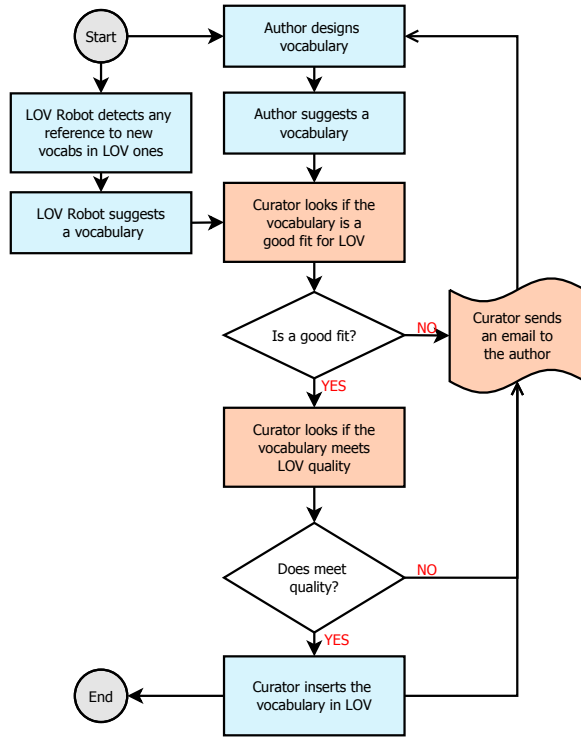


Fig. 8.: Curation workflow for vocabulary insertion.

in place to automatically check for vocabulary updates on a daily basis. 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. To the best of our knowledge, LOV provides for the first time such a vision.

3.3. Data Access

LOV system (code and data) is published under Creative Commons 4.0 license⁹ (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; 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.

⁹<https://creativecommons.org/licenses/by/4.0/>

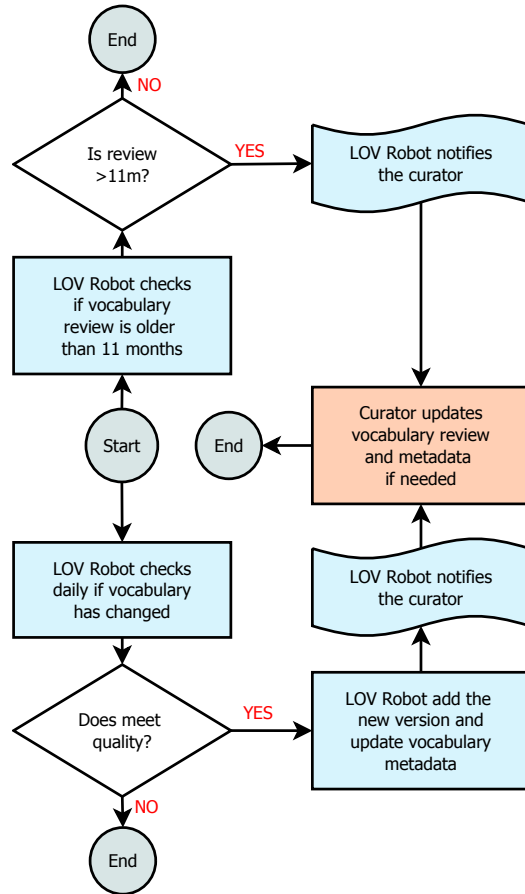


Fig. 9.: Curation workflow for systematic vocabulary review.

3.3.1. Search Engine

In [5], 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 resources 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

need 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.

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
σ_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 $\mathcal{D}, t_i \in D$

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

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

As highlighted in [5] and [22], 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 put the emphasis on the most used terms, result of 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 to RDF datasets, best practices about vocabularies publication makes their structure

consensual and stable. It becomes then intuitive to assign more importance of 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 assigns a heigher 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 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`;
- 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`; and
- 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`.

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

For every vocabulary in the LOV, terms (classes, properties, datatypes, instances) are indexed and a full

text search feature is offered¹⁰. 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 LOD dataset. While the factorisation of tf-idf and field normalisation factor is common for search engine ranking¹¹, 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 Semantic Web to the scoring equation.

$$\begin{aligned} \text{Score}(t, Q) &= \text{tf}(t, V) * \text{idf}(t, \mathcal{V}) \\ &\quad * \text{norm}(t, V) * \text{pop}(t, \mathcal{D}) \\ &\quad : \forall t \{ \exists q_i \in Q : t \in \sigma_V(q_i) \} \end{aligned} \quad (4)$$

3.3.2. Data Dumps

The system provides data dumps of the LOV vocabulary catalogue in RDF Notation 3 format¹² and the LOV catalogue along with the latest version of each vocabulary in RDF N-quads format¹³. 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 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 Endpoint¹⁴ 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 section 4.

3.4. LOV User Interfaces and Application Program Interfaces

LOV APIs give a remote access to the many functions of LOV through a set of RESTful services¹⁵. 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 type of 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 autocompletion or 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 autocompletion or obtain all details about a vocabulary; and
- Agents. This provides a software agent with a list of all agents references in the LOV catalogue, a mean to search for an agent, get autocompletion and details about an agent.

LOV APIs in a convenient manner to get access to the full functionality and data of the 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.

The LOV Website offers an intuitive navigation within the vocabularies catalogue. It allows users to explore vocabularies, vocabulary terms, agents, languages and to get the connection between these enti-

¹⁰<http://lov.okfn.org/dataset/lov/terms>

¹¹See `elasticsearch` documentation: <http://bit.ly/1e37sFL>

¹²<http://lov.okfn.org/lov.n3.gz>

¹³<http://lov.okfn.org/lov.nq.gz>

¹⁴<http://lov.okfn.org/dataset/lov/sparql>

¹⁵<http://lov.okfn.org/dataset/lov/apidoc/>

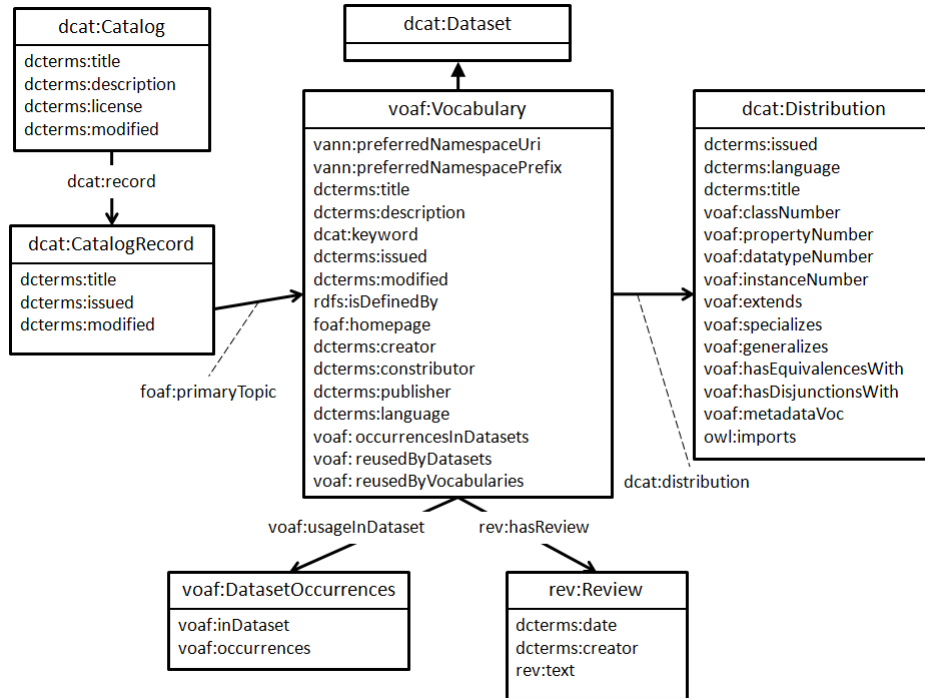


Fig. 10.: UML class diagram representation of LOV catalogue RDF schema model.

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.

ties. For instance, a user can look for experts in *geography* and *geometry* domains¹⁶. We use d3¹⁷ javascript library to display charts and make the navigation more interactive like star graph representation used to dis-

play incoming and outgoing links between vocabularies (cf. figure 12).

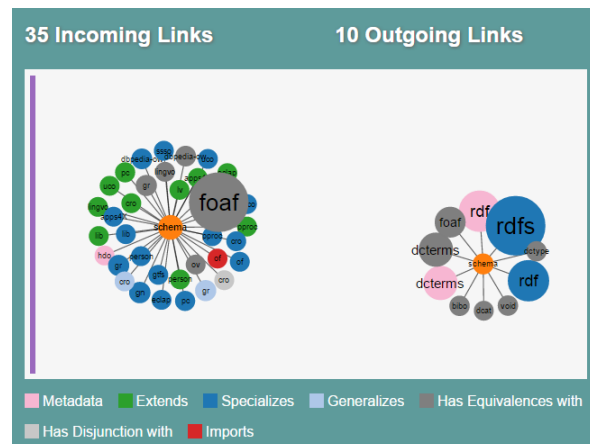


Fig. 12.: Schema.org vocabulary incoming and outgoing links graphical representation as displayed in the UI.

¹⁶<http://lov.okfn.org/dataset/lov/agents?&tag=Geography,Geometry>

¹⁷<http://d3js.org/>

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

Fig. 13.: Equivalent classes and properties between foaf and dcterms

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 methodology for supporting ontology development 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 [25].

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

Ontology Search. Main LOV's feature is the search of vocabulary terms. These vocabularies are categorized within the LOV 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 VOAf vocabulary¹⁸. 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 2 shows the SPARQL query to retrieve all the equivalent classes and properties between the vocabularies foaf and dcterms¹⁹.

Figure 13 shows the alignments between foaf and dcterms vocabularies by mean of owl:equivalentClass and owl:equivalentProperty.

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

```

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

```

5. LOV Adoption

The 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.

5.1. Derived tools and applications

Maguire et al. [16] uses the LOV search API to implement OntoMaton²⁰, a widget for bringing together ontology lookup and tagging within the collaborative environment provided by Google spreadsheets.

YASGUI (Yet Another SPARQL Query GUI)²¹ is a client-side JavaScript SPARQL query editor that uses the LOV API for property and class autocompletion together with <http://prefix.cc> for namespace prefix autocompletion [21].

Datalift²² platform [23], 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 as input a “raw RDF”, that is 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 owing easy discovery and interlinking.

OntoWiki²³ facilitates the visual presentation of a knowledge base as an information map, with different

¹⁸<http://purl.org/vocommons/voaf>

¹⁹The reader can run the query on LOV Endpoint: <http://goo.gl/sTIGQ6>. Prefixes are omitted for readability purpose. They can be found in LOV.

²⁰<https://github.com/ISA-tools/OntoMaton>

²¹<http://yasgui.laurensrietveld.nl>

²²<http://datalift.org/en/node/24>

²³<http://ontowiki.net/>

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 ProtégéLOV²⁴ tool which provides a technological support to help users in reusing terms based on the vocabularies in the LOV catalogue, mostly by retrieving the classes, object properties and data properties according to the panel in the editor during the modeling. It is a plug-in for Protégé editor tool [12] for improving the development of lightweight ontologies by reusing existing vocabularies at low fine grained level. The tool search for a term in LOV via APIs and provides three actions if the term exists : (i) replace the selected term in the current ontology, (ii) add the `rdfs:subClassOf` axiom and (iv) add the `owl:equivalentClass`.

5.2. Using LOV as a Research platform

LOV has served as the object of studies in [17] where Poveda-Villalón *et al.* analysed trends in ontology reuse methods. In addition, the LOV dataset has been used in order to analyze the occurrence of good and bad practices in vocabularies [18].

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. This requirement leads sometimes to a situation where the LOV use prefixes differently from the ones recommended by the vocabulary publishers.

The LOV query log covering the period between 2012-01-06 and 2014-04-16 is used in [5] to build a benchmark suite for ontology search and ranking. The CBRBench²⁵ benchmark uses eight ranking models of resources in ontologies and compares the results with

ontology engineers. We plan to start a collaboration with the authors to enhance the LOV search based on the study result.

In [14], the authors rate vocabularies according to some criteria beyond the `sameAs` links such as `subClassOf` and `equivalentClass` 'links' between vocabularies to foster interoperability, query federation and better interpretation of data.

RDFUnit²⁶ is a test-driven data debugging framework for the Web of Data. In [15], the authors provide an automatic test case for all available schema registered with the LOV. Vocabularies are used to encode semantics to domain specific knowledge to check the quality of data.

Governatori et al. [13] analyzes the current use of licenses in vocabularies on the Web based on the LOV catalogue to further propose a framework to detect incompatibilities between datasets and vocabularies.

6. Related work

In the literature, there bunch of work on vocabulary catalogues and proposals o reuse vocabulary based on ontology repositories. Reusing vocabularies requires searching for terms in existing specialized vocabulary catalogues or search engines on the web. While we refer the reader to [8] for a systematic survey of ontology repositories, we list below some existing catalogues relevant to find vocabularies:

- *Catalogues of generic vocabularies/schemas* similar to LOV catalogue. Example of catalogues falling in this category are vocab.org²⁷, ontologi.es²⁸, JoinUp Semantic Assets or the Open Metadata Registry.
- *Catalogues of ontologies for a specific domain* such as biomedicine with the BioPortal [27], geospatial ontologies with SOCoP+OOR²⁹, Marine Metadata Interoperability and the SWEET [20] ontologies³⁰. 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.

²⁴<http://labs.mondeca.com/protolov/>

²⁵<https://zenodo.org/record/11121>

²⁶<https://github.com/AKSW/RDFUnit>

²⁷<http://vocab.org/>

²⁸<http://ontologi.es/>

²⁹<http://socop.oor.net/>

³⁰<http://sweet.jpl.nasa.gov/2.1/>

- *Catalogues of ontology Design Patterns (ODP)* focused on reusable patterns in ontology engineering [19]. The submitted patterns are small pieces of vocabularies that can further be integrated or linked with 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 [11], Watson [7,9] and FalconS [6]. These search engines crawl for data schema from RDF documents on the Web. They offer a filtering based on ontology type (Class, Property) and a ranking based on the popularity. They don't look for ontology relations nor check if the definition of the ontology is available (usually known as dereferenciation)

LODStats [10] is another project based on a statement-stream-based approach for gathering comprehensive statistics for RDF resources in datahub. LODStats gathers up to 32 different statistical criteria and use the Vocabulary of Interlinked Datasets (VoID)[1] and the Data Cube vocabulary to represent these statistics. LODStats maintains a comprehensive statistics on the schema elements (i.e. classes, properties) defined and used in a dataset. LOV is similar to the approach presented by Demter et al. as it provides statistics at the schema level dimension. Another difference between LODStats and LOV is that our work do not only depends on automatic crawl of the CKAN dataset, but also on suggestions made by users with human curators on the loop for quality checking and review of the vocabulary. For example, while LODStats reports 2825 vocabularies, there are many duplicates representing in fact the same vocabulary URI (e.g., foaf has three different records³¹, while this is not the case in LOV.

A updated comprehensive empirical survey of Linked Data conformance is presented by Schmachtenberg et al. [24]. Their survey is based on a large-scale Linked Data crawl from March 2014 as well as detailed categorize datasets by topical domain to analyze the differences in the adoption of the best practices in different domains. Their results concerning the most used accessible vocabularies (e.g., foaf, dc-terms, skos, etc.) and the adoption of well-known vocabularies are inline with the findings of this paper. The general difference of our work with the one pre-

sented by Schmachtenberg et al. is that our approach is vocabulary-based while their approach is dataset-based. On the other hand, deferenceability of a vocabulary is a strong requirement of a vocabulary to be inserted into LOV, which is not always the case in their work (e.g., they define the notion of partly dereferencable for vocabularies).

7. Discussion

LOV main strength focuses mainly on giving access to up-to-date of highly curated vocabularies (subpart of semantic documents of the Web) submitted by the community, reviewed and validated by curators. In addition, LOV keeps track locally of all versions of the vocabularies. In contrast, Swoogle is designed to automatically discover Semantic Web Documents (SWDs), index their metadata. 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 1,562 vocabularies. Watson works similarly to Swoogle, crawling and indexing semantic documents 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 6 presents a summary of key features of LOV with respect to Swoogle, Watson and Falcons.

Apart from the core features for ontology development as search, ranking and reuse, LOV provides to users with useful functionalities and metadata information for ontology development processes and activities. This makes it possible to find the most suitable classes and properties to express data as Linked Data. For example, one could take advantage of LOV to localize ontologies or provide multilingual annotation as RDFs labels in many different languages are stored in LOV endpoint. This annotations can be used when translating terms into different languages or to provide multilingual search based on ontologies.

³¹<http://stats.lod2.eu/vocabularies?search=foaf>

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 6: Comparison of LOV with respect to Swoogle, Watson and Falcons; adapted from the framework presented by d’Aquin and Noy [8].

However, LOV itself can not replace the creation of term in an engineering process. Thus, LOV can be used as input for a vocabulary recommendation system such as the LOVER approach presented by Schaible et al.[22]. Furthermore, LOV does not currently support reasoning capabilities (such as transitive closure or schema matching techniques) on top of the vocabularies. Although such information would potentially be relevant to the LOV users, this out of scope of the current effort of explicitly represent the links between vocabularies using VOAF and maintaining version history.

8. 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.

The adoption and integration of LOV catalogue in applications for vocabulary engineering, reuse and data quality are significant. Linked Open Vocabularies

have a central role in vocabulary life-cycle on the Web of Data as highlighted by the W3C³². In the future, we see in particular the following directions for advancing 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. 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.

Multilingual vocabularies. 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 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

³²<http://www.w3.org/2013/data/>

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.

[TODO:add how to use/integrate the results of the new LOD cloud statistics in LOV vocabularies, or continue the work of harmonizing the 649 vocabs from new LOD Cloud stats]

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.

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 Atemezeng, Bernard Vatan, 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 Vatan. Requirements for vocabulary preservation and governance. *Library Hi Tech*, 31(4):657–668, 2013.
- [5] Anila Sahar Butt, Armin Haller, and Lexing Xie. Ontology search: An empirical evaluation. In *The Semantic Web-ISWC 2014*, pages 130–147. Springer, 2014.
- [6] 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.
- [7] Mathieu d’Aquin, Claudio Baldassarre, Laurian Gridinoc, Marta Sabou, Sofia Angeletou, and Enrico Motta. Watson: Supporting next generation semantic web applications. 2007.
- [8] 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.
- [9] Mathieu d’Aquin, Marta Sabou, Martin Džbor, 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*.
- [10] Jan Demter, Sören Auer, Michael Martin, and Jens Lehmann. LODStats – An Extensible Framework for High-performance Dataset Analytics. In *18th International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, pages 353–362, Galway, Ireland, 2012.
- [11] Timothy W. Finin, Li Ding, Rong Pan, Anupam Joshi, Pranam Kolar, 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.
- [12] Nuria García-Santa, Ghislain Auguste Atemezeng, and Boris Villazón-Terrazas. The protégé LOVPlugin: Ontology Access and Reuse for Everyone. In *The 12th Extended Semantic Web Conference (ESWC2015)*.
- [13] Guido Governatori, Ho-Pun Lam, Antonino Rotolo, Serena Villata, Ghislain Atemezeng, and Fabien Gandon. Checking licenses compatibility between vocabularies and data. In *Proceedings of the 5th International Workshop on Consuming Linked Data (COLD 2014) co-located with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Italy, October 20, 2014.*, 2014.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] María Poveda-Villalón, Mari Carmen Suárez-Figueroa, and Asunción Gómez-Pérez. The landscape of ontology reuse in linked data. 2012.
- [18] María Poveda-Villalón, Bernard Vatan, María Carmen Suárez-Figueroa, and Asunción Gómez-Pérez. Detecting good practices and pitfalls when publishing vocabularies on the web. 2013.
- [19] 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.

- [20] 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.
- [21] Laurens Rietveld and Rinke Hoekstra. Yasgui: Not just another sparql client. In *The Semantic Web: ESWC 2013 Satellite Events*, pages 78–86. Springer, 2013.
- [22] 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.
- [23] François Scharffe, Ghislain Atemezang, 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 Vatan. Enabling linked-data publication with the datalift platform. In *26th Conference on Artificial Intelligence (AAAI-12)*, 2012.
- [24] 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.
- [25] 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/>.
- [26] Pierre-Yves Vandenbussche and Bernard Vatan. Metadata recommendations for linked open data vocabularies. Technical report, 2012.
- [27] 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.