A Domain-Agnostic Tool for Scalable Ontology Population and Enrichment from Diverse Linked Data Sources

© Efstratios Kontopoulos © Panagiotis Mitzias © Marina Riga © Ioannis Kompatsiaris

Information Technologies Institute,
GR-57001 Thessaloniki, Greece

skontopo@iti.gr pmitzias@iti.gr mriga@iti.gr ikom@iti.gr

Abstract. Ontologies are a rapidly emerging paradigm for knowledge representation, with a growing number of applications in data-intensive domains. However, populating enterprise-level ontologies with massive volumes of data is a non-trivial and laborious task. Towards tackling this problem, the field of ontology population offers a multitude of approaches for populating ontologies with instances in an automated or semi-automated way. Nevertheless, most of the related tools typically analyse natural language text and neglect more structured types of information like Linked Data. The paper argues that the rapidly increasing array of published Linked Datasets can serve as the input for large-scale ontology population in data-intensive domains and presents PROPheT, a novel software tool for ontology population and enrichment. PROPheT can populate a local ontology model with instances retrieved from diverse Linked Data sources served by SPARQL endpoints. As demonstrated in the paper, the tool is domain-agnostic and can efficiently handle vast volumes of input data. To the best of our knowledge, no existing tool can offer PROPheT's diverse extent of functionality.

1 Introduction

Ontologies constitute a knowledge representation paradigm for modelling domains, concepts and interrelations, effectively enabling the sharing of information between diverse systems [23].The rapidly emerging popularity of ontologies has led to their deployment in various *Data Intensive Domains* (*DID*s), like e. g. bioinformatics [7], e-commerce [11] and digital libraries [3]. Nevertheless, in order for ontologies to be further used at an enterprise level, massive volumes of data are required for populating the underlying models.

If performed manually, this task is extremely time-consuming and error-prone. *Ontology population* attempts to alleviate this problem, by introducing methods and tools for automatically augmenting an ontology with instances of concepts and properties. The schema of the ontology itself is not altered but only its set of concepts and relations. This process is part of *ontology learning*, which refers to the automatic (or semi-automatic) construction, enrichment and adaptation of ontologies [16].

The vast majority of ontology population tools and methodologies are aimed at textual input, typically extracting knowledge from natural language text [5], [20]. However, other more structured sources of information are very often neglected. Such an example is *Linked Data* [10], which builds upon standard Web technologies and is a standard for publishing interlinked structured data that are capable of responding to semantic queries. Linked Data are formalised using controlled vocabulary terms based on ontologies and can be publicly accessible via a SPARQL endpoint [4].

Proceedings of the XIX International Conference "Data Analytics and Management in Data Intensive Domains" (DAMDID/RCDL'2017), Moscow, Russia, October 10–13, 2017

This paper argues that the rapidly increasing array of published Linked Datasets [1] can serve as the input for large-scale ontology population in DIDs and presents PROPheT, a software tool for user-driven ontology population from Linked Data sources. The tool is domain-agnostic and can efficiently handle vast volumes of input data. To the best of our knowledge, no existing tool can offer PROPheT's extent of functionality.

The rest of the paper is structured as follows: Section 2 gives an overview of related work approaches. Section 3 presents PROPheT in detail, followed by a discussion on PROPheT's performance with regards to key challenges for accessing information served by SPARQL endpoints. Section 5 presents an illustrative use case that demonstrates the tool's versatility and scalability. Section 6 presents an evaluation of PROPheT, and the paper is concluded with final remarks and directions for future work.

2 Related Work

Ontology population has already been deployed in various domains, like e.g. e-tourism [22], web services [21] and clinical data [17], amongst others. Regarding the application of ontology population in DIDs, we only came across a recent (2016) work by Knoell et al. revolving around Big Data [15], indicating a potentially emerging interest in the area.

Overall, and as already mentioned in the introduction, state-of-the-art ontology population approaches in literature are mostly addressed to retrieving instances from textual corpora (i.e. natural language text, like e.g.

product catalogues) and mainly involve machine learning, text mining and natural language processing (NLP) techniques. Other indicative approaches besides the ones discussed above are presented in [5] and [20].

A less popular stream of ontology population research is aimed at retrieving instances from other types of content, like e.g. CAD files [8], or more structured content, like e.g. spreadsheets [9], [13], and XML files [19]. However, to the best of our knowledge, no other approach similar to PROPheT currently exists that is capable of populating an ontology with instances retrieved from Linked Data sources, rendering PROPheT into a highly novel tool.

3 The PROPheT Ontology Population Tool

PROPheT[1] is a novel software tool for ontology population and enrichment that can retrieve instantiations of concepts from SPARQL-served Linked Data sources in a scalable manner. The retrieved instances are filtered based on user preferences and are then inserted into a target ontology. As described in the following subsections, PROPheT provides various modes of instance retrieval, along with the capability for establishing user-defined mappings of the respective properties. The tool's mode of operation is purely user-driven, but relies on a step-by-step wizard-based interaction with the end-user, which greatly facilitates use of the software even by largely unfamiliarised users.

PROPheT's front-end (see main window in Figure 1) relies on Python and the PyQt application framework, while the back-end deploys RDFLib and SPARQLWrapper, two Python APIs for manipulating ontologies, along with an SQLite data store for storing settings and user preferences.

PROPheT is fully domain-independent in the sense that it can operate with *any* OWL ontology and *any* RDF Linked Dataset that is served via a SPARQL endpoint.

3.1 Motivation

PROPheT was developed within the recently finished PERICLES FP7 project on Digital Preservation[2]. One of the domains tackled by the project was cultural heritage, where we faced the non-trivial challenge of populating our domain ontologies with thousands of artefacts, each of which was associated with hundreds of metadata entries. In this affair, PROPheT was successfully deployed for populating the ontologies with instances retrieved from various Linked Data sources, like DBpedia and Freebase.

Nevertheless, though highly relevant [26], cultural heritage is not the only DID where populating ontologies from diverse sources poses a formidable challenge. Other domains share similar concerns, like e.g. the telecommunications and news industry [2], and health and biomedicine [6], [14]. This was our main motivation for turning PROPheT into a truly domain-agnostic tool, capable of performing ontology population and enrichment from Linked Data sources in virtually any domain, data-intensive or not.

3.2 Ontology Population

PROPheT offers the capability of class-based and instance-based ontology population. The former method, *class-based population*, retrieves instances from an external model and inserts them into a local ontology, based on a class name entered by the user. Since the exact class name has to be entered (e.g. `dbo:Artist` for the DBpedia class representing artists), this method has the peculiarity that the user needs to know the structure of the external ontology. PROPheT then submits appropriate SPARQL queries to the remote model's endpoint and retrieves a result set of instances belonging to the specified class. The user may then select the instances to populate an existing class in the local ontology.

The second method, *instance-based population*, has two different modes:

(a) Retrieval based on instance label, which is performed according to a label (`rdfs:label` property value) entered by the user. The match of the retrieved instances is based on an exact or partial match of the input text.

(b) Retrieval based on an existing instance, in which the user selects an instance already existing in the local ontology and PROPheT queries the endpoint for similar instances. More specifically, the tool finds classes in the remote ontology that include an instance with a similar `rdfs:label` property value with the input instance. The user may then select specific classes, view their extension (i.e. set of instances) and choose which instances to import into the local ontology model.

In all the cases described above, and after the set of preferred instances has been selected by the user to be populated into the ontology, PROPheTlaunches the ontology mapping process described next.

3.3 Ontology Mapping

In order for PROPheT to proceed with the ontology population with the selected instances, a user-driven ontology mapping is performed, in the sense that the properties of the retrieved instances have to be mapped to properties defined in the local model. In this context, PROPheT displays to the user a list of all datatype properties (`owl:DatatypeProperty`) for the selected instances, in order for him/her to manually define appropriate mappings to datatype properties already existing in the local ontology. This mapping between local and remote properties is mandatory for the property values to be inserted into the local ontology along with the instances. For example, the user might define that the retrieved property `dbo:birthDate` corresponds to the local property `ex:dateOfBirth`. Once defined by the user, PROPheT stores the mappings and offers suggestions when the same mappings occur again

[1] PROPheT is available at: http://mklab.iti.gr/project/prophet-ontology-populator

[2] http://www.pericles-project.eu/

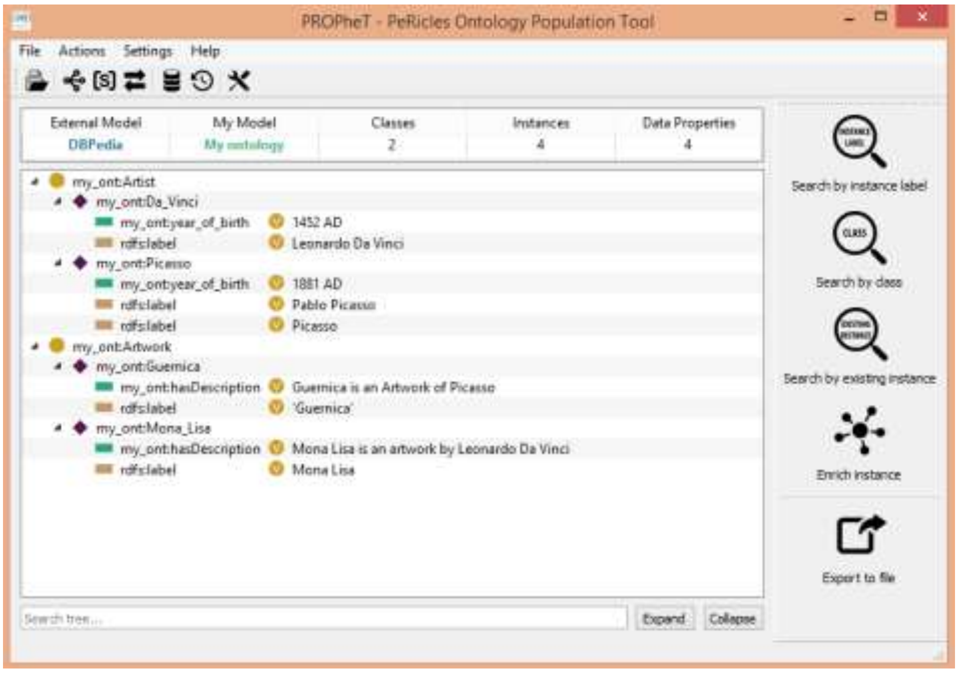

Figure 1 PROPheT's main window

3.4 Instance Enrichment

Besides the ontology population capabilities described above, PROPheT also offers the option of enriching instances already existing in the local ontology with properties and values from "similar" instances in remote ontologies; instance similarity here refers to similarity in the respective instance labels (i.e. `rdfs:label`).

The similar instances may belong to one or more different classes in the remote ontology, thus, the tool presents the user with the type (`rdf:type` property declaration) of each instance. Based on the content and semantics of the derived instances, the user may then decide which property-value pairs he/she will insert from the remote into the local ontology.

3.5 Ontology Enrichment

The local model may also be semantically enriched by establishing links between properties in the local and the remote ontologies via `owl:equivalentProperty` declarations added into the local model. Similar links between classes are represented via `owl:sameAs` and/or `rdfs:seeAlso` declarations added to the local ontology.

4 Challenges and PROPheT's Performance

The availability and scalability of the SPARQL endpoints serving Linked Data is not always guaranteed, since maintaining such heavyweight query services implies significant server-side costs coupled with various potential technical problems on the level of the infrastructure itself [4]. Key parameters for evaluating a SPARQL endpoint are [4]:

- *Discoverability*, referring to how an endpoint can be located and what are the available metadata;
- *Interoperability*, with regards to the supported SPARQL version(s);
- *Efficiency*, which relates to the time needed to respond to the query;
- *Reliability*, based on the uptime of the endpoint on a constant basis.

A useful tool for monitoring the above parameters of SPARQL endpoints is *SPARQLES* [24], while the recent *Linked Data Fragments* (*LDF*) paradigm promises to alleviate the burden from endpoints, by redistributing the load between clients and servers [25].

Taking the above challenges into consideration, and in order to demonstrate PROPheT's scalability, we experimented with timing the retrieval and population of instances from the following well-known SPARQL endpoints into a local custom ontology model:

- *DBpedia*[3], the Linked Data version of WikiPedia;
- *OpenDataCommunities*[4], the official Linked Data platform of the UK Department for Communities and Local Government (DCLG) that provides a selection of official statistics and data outputs on a variety of themes related to DCLG;
- *DBLP*[5], which provides open bibliographic information on major computer science journals and proceedings;
- The *Nobel Prize Linked Data dataset*[6] that contains the authoritative information about Nobel prizes and

[3] http://dbpedia.org/sparql
[4] http://opendatacommunities.org/sparql
[5] http://dblp.l3s.de/d2r/sparql
[6] http://data.linkedmdb.org/sparql

Nobel Laureates since 1901;

- *Eurostat statistics*[7] converted to RDF and re-published using Linked Data principles.

Table 1 illustrates the resulting retrieval and population times for all selected endpoints. PROPheT's performance is impacted by three parameters: (a) the software's efficiency in querying and handling data, (b) the endpoints' speed in serving the requested data, and (c) the volume of data (in the form of datatype property values) that the retrieved instances are attached to.

Table 1 Instance retrieval and population times

Ontology	No of instances	Retrieval time (sec)	Population time (sec)
DBpedia	10	6,0	2,5
	100	19,0	8,3
	1,000	171,0	54,0
	10,000	648,0	250,0
Open Data Communities	10	4,5	3,0
	100	18,0	6,7
	1,000	104,0	44,0
	10,000	510,0	210,0
DBLP	10	3,5	1,8
	100	10,0	5,0
	1,000	62,0	32,0
	10,000	316,0	192,0
Nobel Prize	10	3,7	2,0
	100	10,0	5,7
	1,000	56,0	31,0
	10,000	270,0	170,0
Eurostat	10	5,0	2,5
	100	15,7	7,7
	1,000	92,0	48,0
	10,000	440,0	225,0

Since parameter (a) remains constant within the experiments, it becomes obvious that any differentiation in times heavily depends on parameters (b) and (c). Considering the facts that DBpedia reportedly contains the largest volume of property values, that most of the rest endpoints had almost equal number of properties and that Eurostat's selected instances had no datatype properties, it is clear that an endpoint's response time (second parameter) has a great impact on the ontology population process from Linked Data sources.

5 A Use Case Scenario

This section intends to demonstrate PROPheT's functionality by presenting a use case scenario in a data-intensive domain. Thus, consider a government institution monitoring pollution in rural environments, which requires a directory of cities and towns worldwide, enriched with related information, such as population, postal codes, etc.

Initially, a local ontology schema needs to be deployed, incorporating the necessary classes (e.g. *Town*, *City*, etc.) and properties (e.g. *hasPopulation*, *hasPostalCode*, etc.). This schema will be loaded in PROPheT to be populated.

Next, the user will need to register the sources that serve the desired data (SPARQL endpoint URIs). For the domain of the specific use case, there are several established SPARQL-served ontologies that contain instances of cities and towns, such as *ENVO*[8], an ontology of environmental features and habitats, and *LinkedGeoData*[9]. Specifically, ENVO's class *City* (ENVO_00000856) and LinkedGeoData's classes *City* and *Town* contain related instances.

Table 2 Instance retrieval and population times

Ontology	No of instances	Population time (sec)
LinkedGeoData	10,000	120
ENVO	10,000	204
LinkedGeoData	10,000	158

Taking advantage of PROPheT's class-based instance extraction wizard, the user can respectively populate two (or more) different classes of the local schema with resources from two (or more) data sources. For the purposes of this case study, PROPheT flawlessly managed to retrieve and populate more than 30K instances, along with data property values. Specifically, 10K instances from ENVO's *City* and 10K instances from LinkedGeoData's *City* were populated in the local model's class *City*. Also, another 10K instances from LinkedGeoData's class *Town* were imported to the local model's *Town*. Indicatively, Table 2 displays the population times (in seconds) for the instances mentioned above. Population times in the second batch of LinkedGeoData instances is slightly higher, since the local ontology already contained 20K instances populated during the previous two phases.

Alternatively, supposing that the user cannot predefine the classes of interest in the external models, a different course will be followed. First, a single instance of the desired set will be located and imported. For example, using PROPheT's feature "*Search by instance label*", the user will find a certain city of interest, e.g. Amsterdam, and import it into the local model. Next, with the use of "*Search for similar instances*", the software will discover all the classes where Amsterdam is assigned to. Browsing the resulting list of classes, the user will now locate the classes of interest (e.g. class *City*) and select more instances to be populated.

Consequently, by utilizing the "*Enrich Instance*" function, the user can semantically enrich the major cities' instances (e.g. London, Paris, Amsterdam) with data regarding air pollution levels, residing in different endpoints.

To conclude, the aforementioned use case demonstrates PROPheT's ability to populate various classes of an ontology with data retrieved from more than one endpoints

[7] http://eurostat.linked-statistics.org/sparql
[8] http://www.obofoundry.org/ontology/envo.html
[9] http://linkedgeodata.org

.

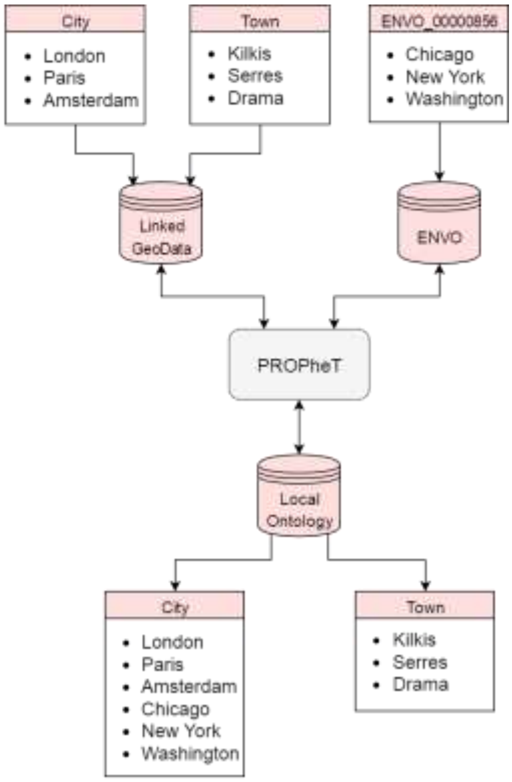

Figure 2 Use case diagrammatic overview

Figure 2 illustrates a diagrammatic overview of the use case described in this section. The population-related features permit different approaches for searching and browsing the available information, offering, thus, great flexibility to the user.

6 PROPheT Evaluation

We recently conducted a user evaluation on PROPheT with very positive results [18]. As indicated by the resulting evaluation of the participants, the following aspects of the tool were the most positive ones: attractiveness (93.5%), user-friendliness (93.5%), ease of usage (100%), innovativeness (87.5%), and efficiency (93.5%); the numbers in parentheses correspond to the respective percentages indicating acceptance on behalf of the users. The current section now presents a qualitative evaluation of the tool, based on the categorisation criteria for ontology population tools proposed in [20].

Elements extracted: Refers to the capacity of an ontology populating system to extract the various ontological aspects, like e.g. objects and relations. PROPheT offers the capability of extracting from external sources both objects (i. e. class instances) and relations (i.e. data property values), and inserts them into a local ontology model. Additionally, PROPheT also appends properties for semantically enriching the local model via `owl:equivalentProperty`, `owl:sameAs` and `rdfs:seeAlso`.

Initial requirements: This criterion refers to the system's initial requirements in terms of resources or background knowledge. PROPheT's only requirement is that a local OWL ontology is already available, in order to be populated with objects retrieved from Linked Data sources. No domain-dependant resources are needed, since PROPheT can flexibly adapt to any thematic domain. No specialised software should be installed in the host machine either; PROPheT is distributed as a standalone bundle.

Learning approach: Refers to the system's approach in extracting knowledge and whether this approach is specialised to a domain. Ontology population tools typically employ Machine Learning techniques (see Section 2), via statistical methods to identify terms or via automated pattern extraction. PROPheT, on the other hand, deploys a purely user-driven, step-by-step ontology population and enrichment approach, which is suitable even for users with only fundamental familiarity with the pertinent notions.

Degree of automation: A fully automated ontology population system is of course desirable, but it seldom is possible to achieve, as the involvement of a domain expert or an ontology engineer is very often needed. The PROPheT approach is mainly user-driven, requiring the involvement of an end user for performing ontology population and enrichment through a step-by-step wizard-based graphical user interface. Thus, although the tool requires user intervention at each step, the process is achieved in a highly user-friendly fashion, as demonstrated by our recent user evaluation of the tool [18] that indicated very positive feedback on behalf of the users.

Consistency maintenance and redundancy elimination: This criterion refers to the system's capability to maintain the consistency of the ontology, which is highly crucial, and to reduce redundancy, which is not equally vital but can facilitate the process of querying the ontology and can limit its size and complexity. Consistency maintenance in PROPheT is ensured by the integrated specialised APIs for manipulating ontologies and SPARQL queries. On the other hand, the problem of instance redundancy (i.e. two or more instances in the ontology referring to the same real object) is handled by PROPheT in a way that instances with the same name-identifier cannot be populated multiple times in the ontology, i.e. values of populated data properties are linked to one single instance. Moreover, we are currently investigating adding more complex handling mechanisms, such as heuristics or machine learning methods to identify similar resources.

Domain portability: This is an important aspect for all ontology populating systems and refers to their capability to be ported to multiple thematic domains or not. PROPheT is a totally domain-agnostic tool that is able to operate equally successfully in any domain, as long as the external sources are served through SPARQL endpoints. Thus, no domain-specific knowledge is incorporated into the system.

Corpora modality: A system that is able to process various modalities demonstrates its ability to accommodate and exploit diverse knowledge sources. In this context, PROPheT can only process input from Linked Data sources through SPARQL endpoints, but can easily be extended to process third-party ontologies as well, retrieving instances and enriching the local ontology with additional property values found in these models.

7 Conclusions and Future Work

The paper argued that, with the rapidly emerging advent of the use of ontologies in data-intensive domains, the process of ontology population becomes increasingly relevant. Most proposed solutions are typically aimed at analysing natural language text, often overlooking other sources of more structured information, like e.g. Linked Data. In this context, we presented PROPheT, a domain independent software tool for ontology population and enrichment from Linked Data sources. Through wizard-based user-driven processes, the tool facilitates the semi-automatic retrieval of instances and their insertion into a local OWLontology. An advanced mapping process enables the dynamic definition of matching classes and properties between source and target models. PROPheT's rich functionality and versatility cannot be matched by any other ontology population tool found in literature, making PROPheT a truly innovative system for populating and enriching ontologies.

Nevertheless, there are still a few areas of improvement for the tool. In its current implementation, PROPheT is only limited to handling datatype and not object properties; the latter are significantly more complex to tackle. Additionally, the tool cannot currently handle direct or indirect imports of ontologies. A further improvement could be considering additional semantic enrichment associations, like e.g. `skos:narrower` and `skos:broader` from SKOS [12]. And, finally, the process of suggesting similar instances or classes to the user during the population and enrichment steps could be suggested by the tool itself, according to appropriate similarity metrics. We are currently working on a revised version of the software, which will integrate the improvements mentioned above.

Acknowledgements

This research received funding by the European Commission Seventh Framework Programme under Grant Agreement Number FP7-601138 PERICLES. We would also like to thank the anonymous reviewers for their valuable remarks, thanks to which the paper has been significantly improved.

References

[1] Abele, A., McCrae, J.P., Buitelaar, P., Jentzsch, A., Cyganiak, R.: Linking Open Data Cloud Diagram (2017). http://lod-cloud.net/.

[2] Belam, M.: What is the Value of Linked Data to the News Industry? The Guardian (2010, January). https://www.theguardian.com/help/insideguardian/2010/jan/25/news-linked-data-summit

[3] Buckingham Shum, S., Motta, E., Domingue, J.: ScholOnto: An Ontology-based Digital Library Server for Research Documents and Discourse. International J. on Digital Libraries, 3 (3), pp. 237-248 (2000)

[4] Buil-Aranda, C., Hogan, A., Umbrich, J., Vandenbussche, P.Y.: Sparql Web-querying Infrastructure: Ready for Action? Int. Semantic Web Conf., pp. 277-293. Springer (2013)

[5] Buitelaar, P., Cimiano, P.: Ontology Learning and Population: Bridging the Gap Between Text and Knowledge, 167, Ios Press (2008)

[6] Callahan, A., Cruz-Toledo, J., Dumontier, M.: Ontology-based Querying with Bio2RDF's Linked Open Data. J. of Biomedical Semantics, 4 (1), S1 (2013)

[7] Gene Ontology Consortium. The Gene Ontology (GO) Database and Informatics Resource. Nucleic Acids Research, 32 (suppl 1), D258-D261 (2004)

[8] Häfner, P., Häfner, V., Wicaksono, H., Ovtcharova, J. Semi-automated Ontology Population from Building Construction Drawings. KEOD, pp. 379-386 (2013)

[9] Han, L., Finin, T., Parr, C., Sachs, J., Joshi, A.: RDF123: From Spreadsheets to RDF. Proc. of the 7th Int. Semantic Web Conf. pp. 451-466, Springer (2008)

[10] Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web: Theory and Technology, 1 (1), pp. 1-136 (2011)

[11] Hepp, M.: Goodrelations: An Ontology for Describing Products and Services Offers on the Web. Int. Conf. on Knowledge Engineering and Knowledge Management, pp. 329-346. Springer (2008)

[12] Isaac, A., Summers, E.: SKOS Simple Knowledge Organization System. Primer, World Wide Web Consortium (W3C) (2009)

[13] Jupp, S., Horridge, M., Iannone, L., Klein, J., Owen, S., Schanstra, J., … Stevens, R.: Populous: A Tool for Building OWL Ontologies from Templates. BMC Bioinformatics, 13 (Suppl 1), S5 (2012). http://doi.org/10.1186/1471-2105-13-S1-S5.

[14] Jupp, S., Malone, J., Bolleman, J., Brandizi, M., Davies, M., Garcia, L., ... Wimalaratne, S.M.: The EBI RDF Platform: Linked Open Data for the Life Sciences. Bioinformatics, 30 (9), pp. 1338-1339 (2014)

[15] Knoell, D., Atzmueller, M., Rieder, C., Scherer, K.P.: BISHOP-Big Data Driven Self-Learning Support for High-performance Ontology Population. LWDA, pp. 157-164 (2016)

[16] Maedche, A., Staab, S.: Ontology Learning from the Semantic Web. IEEE Intelligent Systems, 16 (2), pp. 72-79 (2001)

[17] Mendes, D., Rodrigues, I.P., Baeta, C.F.: Development and Population of an Elaborate Formal Ontology for Clinical Practice Knowledge Representation. KEOD, pp. 286-292 (2013)

[18] Mitzias, P., Riga, M., Kontopoulos, E., Stavropoulos, T.G., Andreadis, S., Meditskos, G., Kompatsiaris, I. User-Driven Ontology Population from Linked Data Sources. Int. Conf. on Knowledge Engineering and the Semantic Web, pp. 31-41. Springer International Publishing (2016)

[19] Modica, G., Gal, A., Jamil, H.M.: The Use of Machine-Generated Ontologies in Dynamic Information Seeking. Cooperative Information Systems, pp. 433-447, Springer (2001)

[20] Petasis, G., Karkaletsis, V., Paliouras, G., Krithara, A., Zavitsanos, E.: Ontology Population and Enrichment: State of the Art. Knowledge-driven Multimedia Information Extraction and Ontology Evolution, pp. 134-166. Springer-Verlag (2011)

[21] Reyes-Ortiz, J.A., Bravo, M., Pablo, H.: Web Services Ontology Population through Text Classification. Computer Science and Information Systems (FedCSIS), 2016 Federated Conference, IEEE, pp. 491-495 (2016)

[22] Ruiz-Martınez, J.M., Minarro-Giménez, J.A., Castellanos-Nieves, D., Garcıa-Sánchez, F., Valencia-Garcia, R.: Ontology Population: An Application for the E-tourism Domain. Int. J. of Innovative Computing, Information and Control (IJICIC), 7 (11), pp. 6115-6134 (2011)

[23] Uschold, M., Gruninger, M.: Ontologies: Principles, Methods and Applications. The Knowledge Engineering Review, 11 (02), pp. 93-136 (1996)

[24] Vandenbussche, P.Y., Umbrich, J., Matteis, L., Hogan, A., Buil-Aranda, C.: SPARQLES: Monitoring Public SPARQL Endpoints. Semantic Web (Preprint), pp. 1-17 (2016)

[25] Verborgh, R., Vander Sande, M., Hartig, O., Van Herwegen, J., De Vocht, L., De Meester, B., ..., Colpaert, P.: Triple Pattern Fragments: A Low-cost Knowledge Graph Interface for the Web. Web Semantics: Science, Services and Agents on the World Wide Web, 37, pp.184-206 (2016)

[26] Wacker, M.: Linked Data for Cultural Heritage, edited by Ed Jones and Michele Seikel. Chicago: ALA Editions (2016)