

# PROPheT – Ontology Population and Semantic Enrichment from Linked Data Sources

Marina Riga, Panagiotis Mitzias, Efstratios Kontopoulos, Ioannis Kompatsiaris

Information Technologies Institute, Thessaloniki, Greece  
{mriga, pmitzias, skontopo, ikom}@iti.gr

**Abstract.** Ontologies are a rapidly emerging paradigm for knowledge representation, with a growing number of applications in various domains. However, populating ontologies with massive volumes of data is an extremely challenging task. The field of ontology population offers a wide array of approaches for populating ontologies in an automated or semi-automated way. Nevertheless, most of the related tools typically analyse natural language text, while sources of more structured information like Linked Open Data would arguably be more appropriate. The paper 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. To the best of our knowledge, no existing tool can offer PROPheT's diverse extent of functionality.

**Keywords:** Ontologies, Ontology Population, Semantic Enrichment, Linked Data, DBpedia.

## 1 Introduction

Ontologies constitute a knowledge representation paradigm for modelling domains, concepts and interrelations in a structured, uniform and effective way, enabling the sharing of information between different systems [1]. The rapidly emerging popularity of ontologies has led to their deployment in various domains, like bioinformatics [2], e-commerce [3] and digital libraries [4]. Nevertheless, in order for ontologies to be more efficiently 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 potentially 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 that represent real data/objects [5]. The schema of the ontology itself is not altered but only the realisation of its set of concepts and the asserted relations on the newly introduced instances. This process is part of *ontology learning*, which refers to the automatic (or semi-automatic) construction, enrichment and adaptation of ontologies [6].

The vast majority of ontology population tools and methodologies are aimed at textual input, typically extracting knowledge from natural language text [7]. Nevertheless, the unstructured nature of free text drastically increases the efforts for utilising its content in already structured frameworks. Instead, other more structured sources of information could be used alternatively; such an example is *Linked Open Data (LOD, or often referred to simply as Linked Data)* [8], which builds upon established Web technologies and is a standard for publishing interlinked structured data that are capable also 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 [9]. A popular Linked Dataset is *DBpedia*<sup>1</sup>, the Linked Data version of Wikipedia.

This paper argues that the rapidly increasing array of published Linked Datasets [10] can serve as the input for scalable ontology population and presents *PROPheT (PERICLES<sup>2</sup> Ontology Population Tool)*, a novel 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. Upon user request, PROPheT locates realisations of concepts in Linked Data sources and appropriately inserts them into a local schema, preserving its initial structure and semantic representations. To the best of our knowledge, no existing tool can offer PROPheT's extent of functionality.

The work presented here constitutes an extension to previous work of ours [11], including the following new materials over the previous paper: (a) an extended related work section, now featuring a number of approaches for ontology population from DBpedia (section 2); (b) a more thorough account of PROPheT's technical implementation details (section 3.2); (c) two revised use cases, demonstrating the tool's functionality in diverse scenarios (section 4).

The rest of the paper is structured as follows: Section 2 gives an overview of related work approaches. Section 3 presents PROPheT's functionalities and operational workflow, followed by two illustrative use cases that demonstrate the tool's versatility and scalability in section 4. Section 5 reports on evaluating 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 [12], web services [13] and clinical data [14], amongst others. Another recent work deploys ontology population in a Big Data setting [15], indicating a potentially emerging interest in the area. Overall, state-of-the-art ontology population approaches are mainly addressed to extracting and retrieving possible instances from natural language text, like e.g. product catalogues or university homepages, corpora, other Web sources, etc., and typically involve machine learning, text mining and natural lan-

---

<sup>1</sup> <http://wiki.dbpedia.org/>

<sup>2</sup> The tool has been developed in the context of the PERICLES FP7 project: <http://www.pericles-project.eu/>

guage processing techniques. Other representative approaches besides the ones discussed above are presented in [5] and [7].

Another, albeit less popular, direction of ontology population research is aimed at retrieving instances from other types of input, like e.g. CAD files [16], or more structured content, like e.g. spreadsheets [17, 18], and XML files [19].

Regarding ontology population from DBpedia, a recent attempt is presented in [20], where the authors manually map a local ontology to DBpedia classes and run a series of SPARQL queries that retrieve the respective instances; no details are given regarding the specifics of the population process. A similar approach for semantic annotation of news items is presented in [21], while the authors in [22] present a methodology for ontology enrichment based on input from DBpedia and (the now obsolete) Schema.org<sup>3</sup>.

PROPhET's similarity to these approaches lies in the use of a LOD source as input for ontology population and enrichment. In this sense, PROPhET could easily be used as the underlying ontology population tool in [20-22]. Nevertheless, no other ontology population tool can currently instantiate new concepts from a LOD source so flexibly, regardless the domain of interest or the content of the source. PROPhET can handle any kind of LOD as an external knowledge source for extracting concepts of interest and for populating them to corresponding resources into the domain ontology.

### 3 The PROPhET Ontology Population Tool

PROPhET<sup>4</sup> is a novel software tool for ontology population and semantic enrichment that can retrieve instantiations of concepts from Linked Data sources. In this sense, the tool is fully domain-independent and capable to operate with any OWL ontology and any RDF LOD dataset served via a SPARQL endpoint. The retrieved instances are filtered by the user and are then inserted, together with their accompanied/selected properties and values, into a target ontology. As described in the following subsections, PROPhET provides various modes of instance retrieval, and allows establishing user-defined mappings of the respective properties. Through its step-by-step wizard-based interaction mode, the tool is extremely easy to use even by unfamiliarised users.

#### 3.1 Technical Infrastructure

PROPhET's front-end (see Fig. 1) is implemented in Python along with PyQt<sup>5</sup>, while specialised Python APIs (RDFLib<sup>6</sup>, SPARQLWrapper<sup>7</sup>) are deployed for handling local and remote ontologies. An SQLite database was also set up in the back-end for storing dynamic data (e.g. settings, user preferences) that are created during the tool's operation.

---

<sup>3</sup> <http://schema.org/>

<sup>4</sup> PROPhET is available at: <http://mklab.itit.gr/project/prophet-ontology-populator>

<sup>5</sup> <https://riverbankcomputing.com/software/pyqt>

<sup>6</sup> <https://github.com/RDFLib/rdfliib>

<sup>7</sup> <https://github.com/RDFLib/sparqlwrapper>

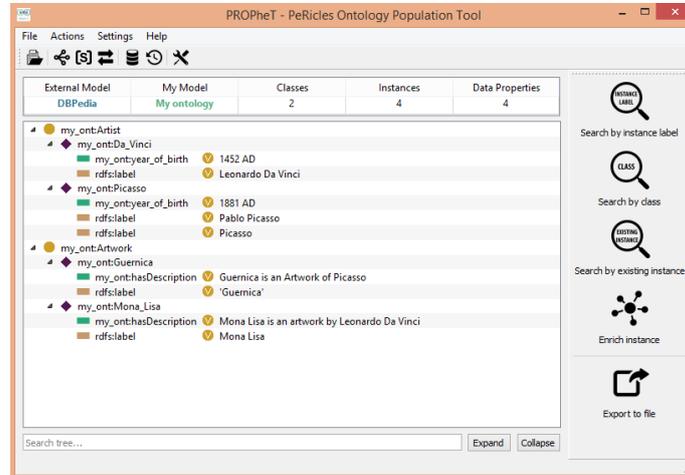


Fig. 1. PROPheT’s main window.

### 3.2 Ontology Population

PROPheT offers the capability of class-based and instance-based ontology population. *Class-based population* retrieves instances from an external source, based on a given class name<sup>8</sup>, and inserts them into a local ontology. PROPheT submits appropriate SPARQL queries to the remote endpoint in order to first retrieve a result set of instances belonging to the specified class (declared with a unique *classURI* value in Table 1), and then to derive additional info for each instance, such as its label(s) (*rdfs:label*), data properties and related values defined in the remote ontology. The total number of fetched instances can be bound according to a maximum number of results (*query\_limit*) specified by the user. The user may then select the instances to populate under an existing class in the local ontology.

Table 1. An example SPARQL query submitted to the endpoint during class-based population.

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?instances WHERE {
  ?instances rdf:type ?classURI
}
ORDER BY ?instances
LIMIT ?query_limit
```

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

1. Retrieval instances based on their *rdfs:label* property value, where the match of the retrieved instances is based on specific parameters. More specifically, the user may input the corresponding label field to search for instances defined in the exter-

<sup>8</sup> E.g. *dbo:Artist* for the DBpedia class representing artists.

nal source, together with additional search options, such as: (i) the exact or partial match (i.e. contains term) of typed text with the label of retrieved instance(s), (ii) the exact match of language code<sup>9</sup> selected by the user with that specified in label(s) of retrieved instance(s), and (iii) the ability for the search execution to be performed as case sensitive or insensitive. Detailed examples of corresponding SPARQL queries are presented in Table 2. Retrieved results can be of any class (`rdf:type`), thus, the user may select any of the derived instances to be populated under a specific class in the local ontology.

**Table 2.** Example SPARQL queries submitted to the endpoint during instance-based population, when performing searching by instance label.

<p><b>Case A. Exact match and language code specified.</b></p> <pre>PREFIX rdf:&lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX rdfs:&lt;http://www.w3.org/2000/01/rdf-schema#&gt; SELECT DISTINCT ?instance WHERE {   ?instance rdf:type ?class .   ?instance rdfs:label 'example_label'@language_code . }</pre>
<p><b>Case B. Contains word(s), case insensitive and language code not specified.</b></p> <pre>PREFIX rdf:&lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX rdfs:&lt;http://www.w3.org/2000/01/rdf-schema#&gt; SELECT DISTINCT ?instance WHERE {   ?instance rdf:type ?class .   ?instance rdfs:label ?label .   FILTER regex (?label, 'example_label', 'i') }</pre>
<p><b>Case C. Contains word(s), case sensitive and language code specified.</b></p> <pre>PREFIX rdf:&lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt; PREFIX rdfs:&lt;http://www.w3.org/2000/01/rdf-schema#&gt; SELECT DISTINCT ?instance WHERE {   ?instance rdf:type ?class .   ?instance rdfs:label ?label .   FILTER regex (?label, 'example_label') .   FILTER langMatches( lang(?label), 'language_code' ). }</pre>

- Retrieval based on instances similar to an existing instance. More specifically, PROPheT detects the classes in the remote ontology that include an instance with a similar `rdfs:label` property value (exact match) with the input instance. The user may then select specific classes and choose which instances to import into the local ontology, following a similar approach as class-based population but performed for multiple classes results simultaneously.

<sup>9</sup> Language code can be a two-letter abbreviation as standardized by ISO 639-1. Selection is feasible in PROPheT via the provided list box.

In all the above cases, after the set of preferred instances has been selected by the user to be populated into the ontology, PROPheT launches the ontology mapping process described in subsection 3.4.

### 3.3 Instance Enrichment

PROPheT also offers the option of semantically enriching instances already existing in the local ontology with properties and values from similar instances in remote ontologies, i.e. instances with similar labels. The similar instances may belong to one or more different classes in the remote ontology, thus, the tool presents the user with the `rdf:type` 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.

### 3.4 Ontology Mapping

In order for PROPheT to proceed with populating the ontology with the selected instances, the properties of the retrieved instances have to be mapped to properties defined in the local model. PROPheT displays a list of all datatype properties for the selected instances, so that the user can define suitable mappings to datatype properties already existing in the local ontology; for example, mapping the retrieved property `dbo:birthDate` to the local property `ex:dateOfBirth`. PROPheT stores the mappings in a linked SQLite database and offers suggestions when the same mappings occur again in future occasions.

When ontology mapping is finalised, the instances and their related properties and values can directly be populated as new triples in the local ontology.

### 3.5 Semantic 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 `rdfs:seeAlso` declarations added to the local ontology.

## 4 Use Cases

This section presents two use case scenarios: the former demonstrates PROPheT's versatility in performing ontology population and semantic enrichment from diverse sources, while the latter illustrates the tool's scalability in data-intensive domains.

### 4.1 Use Case 1: Ontology Population from Different LOD Sources

Suppose that Alice, an avid movie enthusiast, has developed an ontology of actors and films and wishes to initially populate it with an instance of the movie "*The Godfa-*

ther” retrieved from LinkedMDB<sup>10</sup>. She loads her model in PROPheT and registers LinkedMDB as the current source. Since the name of the movie is specified a priori, she searches for existing instances through the “Search by Instance Label” method. One result is retrieved<sup>11</sup> and Alice adds this instance to her local ontology.

She then wishes to retrieve additional information on the specific movie from another LOD source, DBpedia. Through PROPheT’s “Enrich existing instance” function, Alice retrieves a set of instances that may belong to different classes, but they all share the same *rdfs:label* with the newly populated instance. At this point, she may select any pairs of datatype properties/values she wants to add to her local instance of “The Godfather” movie. After manually mapping the relevant pairs of properties, the data is inserted into the corresponding fields in Alice’s ontology.

In case the user wishes to further populate her model with similar resources, she can employ PROPheT’s methods “Search by Class” or “Search by Existing Instance” for any LOD endpoint. For instance, if the former method is selected, Alice should type e.g. *dbo:Film* for DBpedia, or *movie:film* for LinkedMDB. A set of instances will be retrieved, and Alice may then proceed with the selection and mapping process as described previously. If, on the other hand, “Search by Existing Instance” is selected, PROPheT will search for alternative classes that contain instances with the same label. Alice can now select one or more classes from which instances will be retrieved and proceed with the selection of instances to be populated (see Fig. 2).

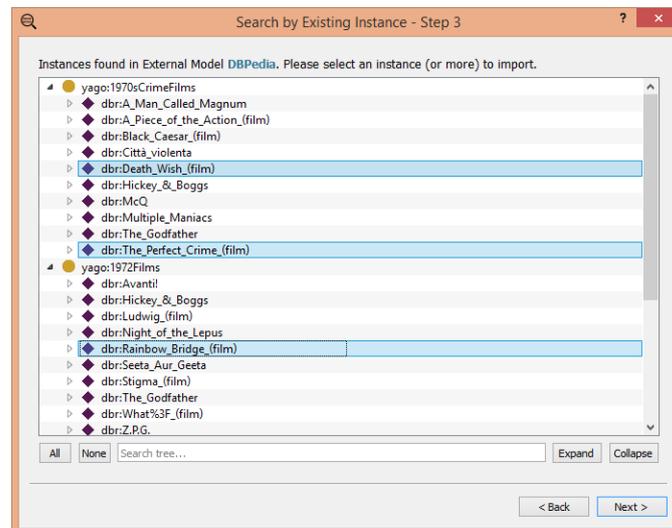


Fig. 2. Selection of movie instances from different classes.

<sup>10</sup> <http://www.linkedmdb.org/>

<sup>11</sup> The retrieved entry in LinkedMDB is <http://data.linkedmdb.org/resource/film/43338>

## 4.2 Use Case 2: Ontology Population in a Data-intensive Domain

Bob, an employee at a government institution monitoring pollution in rural environments, wishes to create a directory of cities and towns worldwide, including related information, such as population, postal codes, etc., along with the respective pollution levels. Bob deploys a local ontology schema incorporating the necessary classes (e.g. *Town*, *City*, etc.) and properties (e.g. *hasPopulation*, *hasPostalCode*, etc.) and loads it into PROPheT. This ontology now needs to be populated with instances of cities and towns.

Bob then registers the sources that serve the desired data. Two suitable candidates are *ENVO*<sup>12</sup> and *LinkedGeoData*<sup>13</sup>. Specifically, ENVO’s class *City* (ENVO\_00000856) and LinkedGeoData’s classes *City* and *Town* contain relevant instances. Using PROPheT’s class-based instance extraction wizard, Bob populates his ontology with 10K instances from ENVO’s *City* and 10K instances from LinkedGeoData’s *City*, along with an additional 10K instances from LinkedGeoData’s class *Town*. Data property values were also mapped and added. Table 3 displays the population times (in seconds) for the 30K instances.

**Table 3.** Instance retrieval and population times.

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

Finally, through PROPheT’s “*Enrich Instance*” function, Bob can semantically enrich the major cities’ instances (e.g. London, Paris, Amsterdam) with data from different endpoints regarding air pollution levels.

## 5 PROPheT Evaluation

We conducted a user evaluation of the tool, which resulted in very encouraging conclusions by the participants, who distinguished the following aspects of the tool as 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. More information on the user evaluation is presented in [23].

Furthermore, we also conducted a qualitative evaluation of PROPheT, based on the criteria for ontology population tools proposed in [7]. The key findings of the evaluation are presented in Table 4, while more information is given in [11].

Finally, considering the fact that the availability and scalability of SPARQL endpoints serving Linked Data is not always guaranteed [9], and in order to demonstrate PROPheT’s scalability, we experimented with timing the retrieval and population of

<sup>12</sup> <http://www.obofoundry.org/ontology/envo.html>

<sup>13</sup> <http://linkedgeo.org>

instances from several well-known SPARQL endpoints into a local custom ontology model. Our findings are presented in more detail in [11].

**Table 4.** PROPheT’s qualitative evaluation.

<b>Criterion</b>	<b>PROPheT’s evaluation</b>
Elements extracted	Objects and relations.
Initial requirements	Availability of a local OWL ontology – no domain-dependant resources or specialised software is required.
Learning approach	Step-by-step ontology population and enrichment; SPARQL querying of Linked Data endpoints.
Degree of automation	Retrieval is automated; selection is user-driven, but highly user-friendly.
Consistency maintenance	Integrated specialised APIs ensure consistency.
Redundancy elimination	The same instance, i.e. those carrying the same URI, cannot be populated multiple times.
Domain portability	Totally domain-agnostic.
Corpora modality	Limited to LOD sources with a SPARQL endpoint.

## 6 Conclusions and Future Work

The paper argued that, with the rapidly emerging advent of the use of ontologies in various 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 automatic retrieval of instances and their insertion into a local OWL ontology, without the need for technical details of the applied queries in the Linked Data endpoints or of the SPARQL query language’s syntax. An advanced mapping process enables the dynamic definition of matching classes and properties between source and target models.

The tool’s rich functionality and versatility outweighs any other ontology population tool found in literature, making PROPheT a truly innovative system for populating and enriching ontologies in various domains where populating ontologies from diverse sources poses a formidable challenge. Indicative paradigms include cultural heritage [24]<sup>14</sup>, telecommunications and news [25], health and biomedicine [26, 27]. 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.

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. In this context, we are plan-

<sup>14</sup> PROPheT is already being successfully deployed in the DigiArt EU-funded project on cultural heritage (<http://digiart-project.eu/>).

ning adopting the approach presented in [28]. Additionally, the ability of simultaneous querying in multiple selected endpoints or the handling of direct/indirect imports of ontologies would enrich the size and content correspondingly of the retrieved results, in one single query. A further improvement could be considering additional semantic enrichment associations, like e.g. `skos:narrower` and `skos:broader` from SKOS [29]. 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.

## Acknowledgements

This research received funding by the European Commission Seventh Framework Programme under Grant Agreement Number FP7-601138 PERICLES.

## References

1. Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. *The knowledge engineering review*, 11(02), 93-136. doi:10.1017/S0269888900007797
2. Gene Ontology Consortium. (2004). The Gene Ontology (GO) database and informatics resource. *Nucleic acids research*, 32(suppl 1), D258-D261. doi:10.1093/nar/gkh036
3. Hepp, M. (2008, September). Goodrelations: An ontology for describing products and services offers on the web. In *International Conference on Knowledge Engineering and Knowledge Management* (pp. 329-346). Springer Berlin Heidelberg. doi:10.1007/978-3-540-87696-0\_29
4. Buckingham Shum, S., Motta, E., & Domingue, J. (2000). ScholOnto: an ontology-based digital library server for research documents and discourse. *International Journal on Digital Libraries*, 3(3), 237-248. doi:10.1007/s007990000034
5. Buitelaar, P. and Cimiano, P. (2008). *Ontology learning and population: bridging the gap between text and knowledge*, Vol. 167, Ios Press.
6. Maedche, A. and Staab, S. (2001). Ontology learning from the semantic web. *IEEE Intelligent Systems*, Vol. 16(2), pp. 72-79. doi:10.1007/978-1-4615-0925-7
7. Petasis, G., Karkaletsis, V., Paliouras, G., Krithara, A., & Zavitsanos, E. (2011, January). Ontology population and enrichment: State of the art. In *Knowledge-driven multimedia information extraction and ontology evolution* (pp. 134-166). Springer-Verlag. doi:10.1007/978-3-642-20795-2\_6
8. Heath, T., & Bizer, C. (2011). Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1), 1-136. doi:10.2200/s00334ed1v01y201102wbe001
9. Buil-Aranda, C., Hogan, A., Umbrich, J., & Vandenbussche, P. Y. (2013, October). Sparql web-querying infrastructure: Ready for action?. In *International Semantic Web Conference* (pp. 277-293). Springer Berlin Heidelberg. doi:10.1007/978-3-642-41338-4\_18
10. Abele, A., McCrae, J. P., Buitelaar, P., Jentzsch, A., & Cyganiak, R. *Linking Open Data cloud diagram 2017*. <http://lod-cloud.net/>. Last accessed: January'18.
11. Kontopoulos, E., Mitzias, P., Riga, M., Kompatsiaris, I. (2017). A Domain-Agnostic Tool for Scalable Ontology Population and Enrichment from Diverse Linked Data Sources. In: Kalinichenko, L.A., Manolopoulos, Y., Skvortsov, N.A., and Sukhomlin, V.A. (eds.) *Data*

*Analytics and Management in Data Intensive Domains: Collection of Scientific Papers of the XIX International Conference DAMDID / RCDL'2017*. pp. 234–240. Moscow: FRC CSC RAS, Moscow, Russia.

12. Ruiz-Martinez, J. M., Minarro-Giménez, J. A., Castellanos-Nieves, D., García-Sánchez, F., & Valencia-García, R. (2011). Ontology population: an application for the e-tourism domain. *International Journal of Innovative Computing, Information and Control (IJICIC)*, 7(11), 6115-6134.
13. Reyes-Ortiz, J. A., Bravo, M., & Pablo, H. (2016, September). Web services ontology population through text classification. In *Computer Science and Information Systems (FedCSIS), 2016 Federated Conference on* (pp. 491-495). IEEE. doi:10.15439/2016f332
14. Mendes, D., Rodrigues, I. P., & Baeta, C. F. (2013, July). Development and Population of an Elaborate Formal Ontology for Clinical Practice Knowledge Representation. (2013). *Proceedings of the International Conference on Knowledge Engineering and Ontology Development*. doi:10.5220/0004548602860292
15. Knoell, D., Atzmueller, M., Rieder, C., & Scherer, K. P. (2016). BISHOP-Big Data Driven Self-Learning Support for High-performance Ontology Population. Proc. *LWDA 2016 (FGWM Special Track)*, Potsdam, Germany, University of Potsdam, (pp. 157-164).
16. Häfner, P., Häfner, V., Wicaksono, H., & Ovtcharova, J. (2013). Semi-automated Ontology Population from Building Construction Drawings. *Proc. Int. Conf. on Knowledge Engineering and Ontology Development*, (pp. 379-386). doi:10.5220/0004626303790386
17. Han, L., Finin, T., Parr, C., Sachs, J. and Joshi, A. (2008). RDF123: from Spreadsheets to RDF. *Proceedings of the 7th Int. Semantic Web Conference*, pp. 451-466, Springer Berlin Heidelberg. doi:10.1007/978-3-540-88564-1\_29
18. Jupp, S., Horridge, M., Iannone, L., Klein, J., Owen, S., Schanstra, J., ... Stevens, R. (2012). Populous: a tool for building OWL ontologies from templates. *BMC Bioinformatics*, 13(Suppl 1), S5. doi:10.1186/1471-2105-13-S1-S5.
19. Araujo, C., Henriques, P. R., & Martini, R. G. (2017). Automatizing ontology population to drive the navigation on Virtual Learning Spaces. *2017 12th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1-6). doi:10.23919/cisti.2017.7975754
20. Gavankar, C., Kulkarni, A., Fang Li, Y., & Ramakrishnan, G. (2012, December). Enriching an academic knowledge base using linked open data. In *24<sup>th</sup> International Conference on Computational Linguistics* (p. 51).
21. Agarwal, S., & Singhal, A. (2013). Autonomous ontology population from DBpedia based on context sensitive entity recognition. In *Fourth international joint conference on advances in engineering and technology, AET* (pp. 580-589). doi: 10.1145/2797115.2797127
22. Tiddi, I., Mustapha, N. B., Vanrompay, Y., & Aufaure, M.-A. (2012). Ontology Learning from Open Linked Data and Web Snippets. *Lecture Notes in Computer Science*, 434–443. doi:10.1007/978-3-642-33618-8\_59
23. Mitzias, P., Riga, M., Kontopoulos, E., Stavropoulos, T. G., Andreadis, S., Meditskos, G., & Kompatsiaris, I. (2016, September). User-Driven Ontology Population from Linked Data Sources. In *International Conference on Knowledge Engineering and the Semantic Web* (pp. 31-41). Springer International Publishing. doi:10.1007/978-3-319-45880-9\_3
24. Wacker, M. (2017). *Linked Data for Cultural Heritage*, edited by Ed Jones and Michele Seikel. Chicago: ALA Editions, 2016.
25. Belam, M. (2010, January). What is the value of Linked Data to the news industry? The Guardian. <https://www.theguardian.com/help/insideguardian/2010/jan/25/news-linked-data-summit>. Last accessed: January'18.

26. Callahan, A., Cruz-Toledo, J., & Dumontier, M. (2013). Ontology-based querying with Bio2RDF's linked open data. *Journal of biomedical semantics*, 4(1), S1. doi:10.1186/2041-1480-4-s1-s1
27. Jupp, S., Malone, J., Bolleman, J., Brandizi, M., Davies, M., Garcia, L., ... & Wimalaratne, S. M. (2014). The EBI RDF platform: linked open data for the life sciences. *Bioinformatics*, 30(9), 1338-1339. doi:10.1093/bioinformatics/btt765
28. Booshehri, M., & Luksch, P. (2015). An Ontology Enrichment Approach by Using DBpedia. *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics - WIMS '15*. doi:10.1145/2797115.2797127
29. Isaac, A., & Summers, E. (2009). *SKOS Simple Knowledge Organization System*. Primer, World Wide Web Consortium (W3C).