Article

Delta: A Modular Ontology Evaluation System

Haridimos Kondylakis ^{1,*} , Astyrakakis Nikolaos ², Papatsaroucha Dimitra ² , Koumarelis Anastasios ², Kritikakis Emmanouel ², Kalkanis Kyriakos ², Skepasianos Iraklis ², Klados Stylianos ² and Nikos Papadakis ²

¹ FORTH-ICS, 70013 Heraklion, Greece

² Department of Electrical & Computer Engineering, Hellenic Mediterranean University, 71500 Heraklion, Greece; mtp235@edu.hmu.gr (A.N.); mtp259@edu.hmu.gr (P.D.); mtp246@edu.hmu.gr (K.A.); mtp251@edu.hmu.gr (K.E.); mtp252@edu.hmu.gr (K.K.); mtp238@edu.hmu.gr (S.I.); mtp244@edu.hmu.gr (K.S.); npapadak@hmu.gr (N.P.)

* Correspondence: kondylak@ics.forth.gr

Abstract: Ontologies are widely used nowadays. However, the plethora of ontologies currently available online, makes it really difficult to identify which ontologies are appropriate for a given task and to decide on their quality characteristics. This is further complicated by the fact that multiple quality criteria have been proposed for ontologies, making it even more difficult to decide which ontology to adopt. In this context, in this paper we present Delta, a modular online tool for analyzing and evaluating ontologies. The interested user can upload an ontology to the tool, which then automatically analyzes it and graphically visualizes numerous statistics, metrics, and pitfalls. Those visuals presented include a diverse set of quality dimensions, further guiding users to understand the benefits and the drawbacks of each individual ontology and how to properly develop and extend it.

Keywords: ontologies; ontology evaluation; semantic web

Citation: Kondylakis, H.; Nikolaos, A.; Dimitra, P.; Anastasios, K.; Emmanouel, K.; Kyriakos, K.; Iraklis, S.; Stylianos, K.; Papadakis, N. Delta: A Modular Ontology Evaluation System. *Information* **2021**, *12*, 301. <https://doi.org/10.3390/info12080301>

Academic Editor: Willy Susilo

Received: 2 July 2021

Accepted: 26 July 2021

Published: 28 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

According to Gruber [1], an ontology is an explicit specification of a conceptualization. Ontologies now play a fundamental role in the era of the semantic web, as they are used for many purposes. For example, ontologies play a pivotal role in modeling data and enabling complex query answering [2], summarization [3], and decision support [4] on them. Further, they have been used economics for tracking supply chain [5] and in construction [6], education [7], and governance [8] for data interoperability, flexible data exchange, distributed data management, and the development of reusable tools. Ontologies have been even used for sharing critical health information [9], for providing health recommendations to patients [10–12], and regulating access to health information [13].

However, creating ontologies is a time-consuming and costly process with many challenges involved [14] and, in many cases, their development and maintenance require dedicated groups of domain experts (e.g., CIDOC-CRM [15]). Due to this difficulty, more and more systems are now proposed for automatic ontology construction [16].

Independent of the way ontologies have been constructed, ontology reuse tries to minimize the ontology development effort, in many cases by the adoption of already available ontologies or ontology components, focusing on reducing human labor.

However, identifying the ontologies of an appropriate quality for reuse is not an easy task. Multiple criteria have already been proposed for evaluating an ontology, which are usually scattered over different systems and tools. As such, actually evaluating an ontology easily becomes a challenge for the interested end-user.

All these aforementioned issues led us to develop Delta, a modular and extensible system that can be used as a starting point to enable end-users to easily evaluate an ontology. They may use a plethora of statistics and metrics, in a way in which it is easy to understand the idea behind each metric, employing a usable and self-explainable graphical user interface, further guiding users to easily grasp and understand the intuition behind

them. We show that existing metrics can be easily added, along with services for pitfall detection, while our platform can also be directly linked with other external systems. Our platform can help users both in the ontology selection process, highlighting the benefits and the drawbacks of each individual ontology and allowing users to easily compare them, or in the ontology development process, highlighting neglected aspects and identifying common pitfalls. The system is also available online [17].

The remainder of this paper is structured as follows: Section 2 reports on related works. Then, in Section 3, we present the architecture of our system. Section 4 reports on the evaluation performed on seven well-known ontologies as a proof of concept. Finally, Section 5 concludes this paper and presents directions for future work.

2. Related Work

Currently, multiple ontology evaluation methods have been proposed by the research community. In this section, we present an overview of the area in terms of the metrics proposed and the tools currently available.

For evaluating ontologies, multiple frameworks [18] and metrics have already been developed. For example, metrics developed to date include accuracy, completeness, conciseness, adaptability, clarity, computational efficiency, and consistency [19]. In addition, coherence metrics have already been proposed for modular ontologies [20] such as number of root classes, number of leaf classes, and median depth legacy tree of all leaf nodes, trying to evaluate the modules of which an ontology is composed. Pathak et al. [21] acknowledge essential properties that modules require to be satisfied, such as size, correctness, and completeness, whereas Schlicht and Stuckenschmidt [22] propose a collection of structural tests for ontology modules such as size, redundancy, connectedness, and relative distance. More recent work focuses on evaluating structural metrics based on public repository data [23] and identifies the most reliable ontology structural metrics in terms of stability and classification outcome.

For a classification of the different metrics in various categories, the interested reader is forwarded to relevant surveys in the area such as [24–27]. Besides specific metrics, there are also multiple tools evaluating the ontologies.

OntoQA [28] is a metric suite developed for ranking ontologies. The metrics evaluate the ontology on two levels, the schema level, which evaluates structural characteristics of the ontology, and the instance level, which evaluates the instances available in the ontology. Those metrics are used in combination with Swoogle in order to rank an ontology.

OntoVal [29], is a portable and domain-independent web tool for evaluating ontologies by non-technical domain experts. It presents the ontology in a textual way, making it readable for users with little to no knowledge on semantics, and allows users to give feedback and evaluate the correctness of the artifacts being developed. The evaluation data is automatically added and processed in order to finally present a detailed report for each evaluated ontology.

Ontometrics [30], provides an online platform for ontology metrics calculation. It includes a web interface for uploading ontologies and for calculating a fixed set of ontology quality metrics for them. The tool is extended in [31], where the authors argue that “most ontology evaluation tools today are no longer available or outdated and propose ontology metrics as a service”.

OntoKeeper [32] is a java-based web application that analyzes ontologies using semi-otic metrics developed by [33] such as lawfulness, richness, interpretability, consistency, clarity, accuracy, authority, etc. OntoKeeper exploits WordNet to identify word senses for each term available in the ontology, and also offers a review of natural language-generated statements for the experts through an intuitive interface.

Finally, OOPS! [34] is a tool to identify common pitfalls in ontology development. The tool enables end-users to upload their ontology, and then more than 40 common pitfalls are searched for and reported back to the users. The tool also provides a service for integrating the pitfall scanner into external applications as well.

Despite all the available work, to our knowledge, there is no tool implementing all proposed metrics and/or extensible to include new metrics that constantly appear. In addition, some of these tools focus only on the ontology development process (e.g., OOPS!), whereas others on the ontology selection (e.g., OntoQA). Further, those metrics are difficult to be understood by non-expert users and, as such, the message they convey should be presented using a self-explainable and intuitive graphical user interface. We also argue that building open, extensible platforms, providing all functionality as services, has the potential to significantly contribute to building high-quality ontologies and tools that will be exploitable by the research community. On the other hand, the lack of systematic approaches to evaluate the quality of the ontologies, the lack of golden standards, and the shortage of available tools for ontology evaluation are well documented [35], and we hope that our approach will be a step for improving the situation in the research community.

3. System Architecture

The high-level architecture of the platform is shown in Figure 1 and consists of three layers: the graphical user interface (GUI), the backend, and the data storage layer. In this sections, we will describe each one of those layers in detail.

Figure 1. The high-level architecture of the Delta framework.

3.1. The Graphical User Interface

The Graphical User Interface (GUI) of our framework contains the web application wherein end-users can upload their ontology for evaluation. As soon as the ontology is uploaded, users can see their ontology in the list with other uploaded ontologies, enabling users to quickly select and compare their ontology with others—the user can also delete his ontology from the list if needed. Then, they can select to visualize ontology statistics, evaluation metrics, or pitfalls for the selected ontologies. According to the selections made, the system presents many individual boxes, one for each metric, as shown in Figure 2. Each available value is rendered within bar charts, in which the user can immediately make a comparison about each value. Currently, the upload of new ontologies is disabled in order to avoid spamming the server with many ontologies.

The UI was developed with Vue.js frontend JavaScript framework, with a template from Creative-Tim under the coverage of MIT open-source license. The visualization with bar charts was implemented using the Chart.js library for JavaScript frameworks. In addition, the UI was developed to be compatible and responsive with every size of the screen including tablets, laptops, televisions, and mobile phones.

Figure 2. The graphical user interface.

3.2. The Backend

The backend of our framework includes extensive methods for calculating statistics, metrics, and pitfalls and also includes a RESTful API, which, upon request, feeds the frontend with data. The backend is also able to call external services, for example, tools such as OOPS!, or for accessing data or metrics from external providers.

For calculating the statistics, the metrics, and the pitfalls, the necessary data are retrieved using the SPARQL endpoint of the data storage layer. In the following section, we briefly describe the statistics, metrics, and pitfalls that are currently implanted, and then we focus on the generalizability aspects of our platform which make it unique.

3.2.1. Ontology Statistics

Measuring ontology statistics can offer valuable information about the quality of the ontology (for example an ontology with small numbers of properties can be identified as a taxonomy, etc.) and determine the level of detail at which a concept is described semantically (e.g., a class with many data properties is better described than another with only one data property). In addition, statistics such as the number of classes can assign in computing several other metrics about ontologies, such as ontology size, etc., which describe the quality of the ontology. In our framework, currently, the following statistic information is supported: number of classes, number of individuals, number of object properties, number of data-type properties, and number of axioms.

3.2.2. Ontology Metrics

Besides simple statistics, more complex ontology metrics are essential in the process of evaluating an ontology. They contribute to the extraction of data on the qualitative aspects of an ontology, and can significantly help in the process of comparison between different ontologies. Metrics can be also divided into categories according to the quality attribute that is extracted from the ontology, that is, structural, functional, and usability metrics [27]. As proof of concept in our framework, we implemented the provided API the metrics shown in Table 1.

Table 1. Proof of concept metrics.

Evaluation Metrics	
1	Size
2	Appropriateness of module size
Schema Metrics	
3	Attribute richness
4	Class/Relation Ratio (CRR)
5	Equivalence Ratio (ER)
Graph Metrics	
6	Average population (AP)

Size: *Size* refers to the number of entities in an ontology module, $|M|$. This can be further subdivided into class size $|C|$, object property size $|OP|$, data property size $|DP|$, and individual size $|I|$.

$$Size(M) = |M| = |C| + |OP| + |DP| + |I|$$

Appropriateness of module size: This metric refers to the appropriateness of module size to map the size of the ontology module to some appropriateness values in (0,1) based on an ideal ontology size. The optimal size of the value is 1. The value x is the number of axioms in the module.

$$Appropriate(x) = \frac{1}{2} - \frac{1}{2} \cos\left(x * \frac{x}{250}\right)$$

Attribute Richness: This metric is defined as the average number of attributes for all classes $|att|$ divided by the number of classes $|C|$, in an attempt to identify how much information classes contain. Each entity in an ontology has several axioms that describe it. These are referred to as attributes, or slots, and are indicative of attribute richness.

$$AR(M) = \frac{|att|}{|C|}$$

Class/Relation Ratio (CRR): CRR refers to the ratio derived from the number of classes divided by the number of relationships in the ontology.

$$CRR = \frac{|C|}{|Relationships|}$$

Average Population: The average population metric can be calculated through the division of the number of instances by the number of classes in ontology. The value $|I|$ is the number of instances, and the value $|C|$ is the number of classes in ontology.

$$AP = \frac{|I|}{|C|}$$

Equivalence Ratio (ER): This metric describes the ratio between similar classes and all classes in ontology.

$$ER = \frac{|SameAsClasses|}{|C|}$$

It is true that usually just looking at a single metric can only provide some indication on the quality of the ontology, and usually the metrics should be considered all together, based also on the purpose that they should be fulfilled using the specific ontology under examination. However, some generic metrics, such as appropriateness and attribute rich-

ness, can directly give an indication of which ontology is better based on those measures. For example, regarding attribute richness, the more the better, and regarding appropriateness, the ontologies with values closest to 1 are assumed to be better. Nevertheless, in our case, the purpose of this paper is not to devise new evaluation metrics, but to show the generalizability and the extensibility of our tool exploiting already defined and available metrics.

3.2.3. Pitfalls

A usual approach for evaluating any type of software or hardware is to check for common errors that other developers have noticed in similar situations. This eliminates the possibility that the newly proposed implementation will include commonly known errors, while it also ensures that the proposed methodology is an actual improvement of any previous version of it. Recent findings have introduced several common errors found in the literature and refer to them as “Pitfalls”. According to [34], pitfalls can be categorized into the following categories, based on their impact:

- Critical—the most severe;
- Important;
- Minor—with the lowest impact.

In this work, the pitfalls that were evaluated for each ontology are missing annotations and unconnected ontology elements:

- **Missing Annotations:** This pitfall detects an ontology element that fails to provide human-readable annotations often attached to it. Consequently, ontology elements lack annotation properties that label them or define them;
- **Creating Unconnected Ontology Elements:** This pitfall identifies ontology elements (classes, object properties, and datatype properties) that are created isolated, with no relation to the rest of the ontology.

Although the number of pitfalls detected by OOPS! is significantly larger than our engine, (a) we are able to directly call their service and (b) we implemented the detection of the aforementioned pitfalls to show that our engine supports pitfalls as well as its generalizability, as we do not have to rely on external services that might or might not work.

Finally, our framework is able to detect inconsistencies available in the ontology. Specifically, for detecting inconsistencies, we implemented a service calling an existing reasoner. There are two types of inconsistencies when a change takes place; structural inconsistency, which appears when there is no confirmation in the ontology language’s structure rules, and logical inconsistency, in which the ontology is not conformed to the logical theory. Both of these types of inconsistencies can be detected using Pellet, an open-source reasoner with a large and active community, which we utilize.

3.2.4. Extensibility

Our infrastructure has been specifically implemented to easily facilitate new metrics. Anyone interested can check the available APIs, and then, for a new metric to appear on the main web page, we only need a JAR file implementing the functionality and the corresponding documentation for the service.

There are also specific guidelines that can be provided for the output of the metric service, as it should produce both textual output and the corresponding graphics, compliant with a specific template that is readily integrated with the dashboard.

Further, our framework is also extensible with external services. For example, it can directly use the service offered by the OOPS! application for detecting the most common pitfalls as they are recognized by that engine.

3.3. Data Storage

The third layer of our framework is the data storage layer. In this layer, as long as the user loads an ontology, the ontology is stored in a triple store. Then, the metrics services can issue the necessary SPARQL queries for calculating the various measures needed.

For the current version of the platform, we selected Virtuoso Openlink as the triple store. Virtuoso is a Relational Database Management System (RDBMS) where every ontology is uploaded. Moreover, Virtuoso provides an API from where one can request certain information. From that API, each statistic, metric, and pitfall can be requested under specific SPARQL queries. On the other hand, metrics are computed based on the corresponding statistics values that the SPARQL queries have resulted in.

Nevertheless, Virtuoso can also be replaced by another triple store, as long as the triple store provides a SPARQL endpoint.

4. Application of Our Framework on Ontology Evaluation

In this section, we use our system to evaluate seven known ontologies and we present our results from their evaluation. The ontologies selected are the following:

- Pizza Ontology (PO): Pizza ontology is a well-known ontology focusing on pizzas and their topics. It includes 100 classes and 20 properties;
- Music Ontology (MO): The Music Ontology provides a model for publishing structured music-related data. It contains 54 classes and 153 properties;
- Friend of a Friend (FOAF): FOAF ontology describes persons, their activities, and their relations to other people and objects. It consists of 22 classes and more than 80 properties;
- Dublin Core (DC): Dublin Core is a set of fifteen core elements for describing resources. It has been formally standardized as an ISO and can be used for multiple purposes, from simple resource description, to combining metadata vocabularies of different metadata standards, to providing interoperability for metadata vocabularies in the linked data cloud and Semantic Web implementations. It consists of 24 classes and 86 properties;
- CIDOC-CRM: CIDOC-CRM v6.2 is a theoretical and practical tool for information integration in the field of cultural heritage. It is the outcome of over 20 years of development and maintenance work, of a working group, and is currently an ISO standard. It consists of 84 classes and more than 270 properties;
- Basic Formal Ontology (BFO). The Basic Formal Ontology is the upper-level ontology upon which OBO Foundry ontologies are built. It consists of 36 classes with only 24 annotation properties as usually is used with the Relations Ontology (RO) that includes the properties;
- Ontology for Biomedical Investigations (OBI). The ontology for Biomedical Investigations is an integrated ontology for the description of life-science and clinical investigations and is available at the OBO foundry. It consists of 4319 classes and more than 170 properties.

According to the statistics regarding the size of each ontology, shown in Figure 3, the OBI ontology has the largest number of classes, individuals, and properties, with CIDOC-CRM ranking second, and BFO and DC ranking last. At the same time, CIDOC-CRM appears to rank the highest when coming down to the number of object properties presented in the ontology's graph, while OBI ranks second and FOAF ranks third. On the contrary, regarding the number of axioms, the largest number seems to be acquired by OBI, followed by CIDOC-CRM. In general, FOAF and DC are the ontologies with the fewest classes.

Figure 3. The size of the ontologies under examination.

Based on the appropriateness, as shown in Figure 4, the ontology with the optimal module size is PO, followed by CIDOC-CRM and MO. DC and FOAF rank last, as they are the smallest ontologies.

Figure 4. Appropriateness of module size for the ontologies under examination.

According to the attribute richness metric calculated for the seven ontologies, shown in Figure 5, DC and FOAF have the most attributes per class, followed by CIDOC-CRM. As such, the classes of these ontologies contain more information when compared to the other ontologies. Surprisingly, the ontology with the smallest attribute richness is OBI, mainly as it includes a large number of classes and a small number of properties.

The large discrepancy between the number of classes and the number of properties is also shown in the class/relations ratio shown in Figure 6, where we can see that OBI has the largest classes per relations ratio. Most of the rest ontologies (MO, FOAF, DC, and CIDOC-CRM) have almost the same class/relations ratio, except PO, which ranks second, and BFO, which ranks third.

Figure 5. Attribute richness for the ontologies under examination.

Figure 6. Class/relations ratio for the ontologies under examination.

Figure 7 shows the average population of the instances for the seven ontologies. As all ontologies under examination besides DC contain only a few instances, the results matched our expectations.

Next, based on the Equivalence Ratio metric, visualized in Figure 8, it is notable, that FOAF and OBI have the largest number of equivalent classes, followed by the PO, whereas the other ontologies have no equivalent classes.

When checking the ontologies using the items implemented in our own pitfall scanner, the PO, the MO, and the FOAF ontologies seem to have isolated, unconnected elements, whereas DC, BFO, OBI, and CIDOC-CRM did not show any such problems. Further, all ontologies have elements with missing annotations; however, this is a common pitfall in all ontologies and does not significantly affect the quality of the ontology. Finally, using the Pellet reasoner, all ontologies do not include contradictory statements.

According to the aforementioned observations using the statistics, the metrics, and the pitfalls, it is evident that the CIDOC-CRM and PO have the most appropriate size, whereas OBI and CIDOC-CRM are the richest. The CIDOC-CRM has also a strong focus on properties, as it has the largest number of properties. Furthermore, the information in classes is more detailed in the OBI ontology, whereas DC, FOAF, and CIDOC-CRM are

described in more detail in terms of attribute richness, while the largest number in terms of average population is for DC. The small number of average populations is expected as, by nature, the ontologies include only a small number of instances. In all ontologies, human-readable annotations are missing; however, this is a minor pitfall, as mentioned before. On the other hand, the problem of the unconnected elements appears in many ontologies.

Figure 7. Average population for the ontologies under examination.

Figure 8. Equivalence ratio for the ontologies under examination.

As already mentioned, the metrics presented show a complementary view on the ontologies of interest. In order to select the most appropriate, a single metric is not enough and multiple aspects should be considered, whereas some metrics, such as appropriateness of size and attribute richness, show which ontology is better based on that measure alone. However, only looking at these measures is not enough. For example, the PO has been constructed as a tutorial, and there have been several concerns expressed regarding its quality (i.e., lagging behind design principles and not committing to any foundational ontology [36]). However, as seen in Figure 4, it has the optimal ontology size.

Nevertheless, our modular framework is extensible, therefore the various metrics already available in the literature can be easily added, and besides the web site we also have available the code of our application (https://github.com/dpapatSa/semantic_project,

accessed on 28 July 2021). We also have to acknowledge that, besides the aspects that can be captured using various metrics, selecting the most appropriate ontology also has to do with the purpose of that ontology and the domain of interest.

5. Conclusions

In this paper, we presented Delta, a fully automated ontology evaluation framework, which evaluates ontologies based on a plethora of evaluation metrics. The main aspects of the aforementioned framework are (1) an easy to use web graphical user interface (GUI), wherein the end-users/developers can deploy their ontologies for evaluation; (2) multiple individual visual reports based on the metrics selected; (3) the number of metrics implemented into the framework; and, most importantly, (4) an extensible and modular architecture that can be used as a starting point to facilitate numerous evaluation metrics. We used our framework to evaluate several common ontologies and we report our findings.

Our future steps for the framework are to implement more metrics and to combine them with ontology summarization approaches [37], enabling users to actively explore their ontology through summaries, to visualize the most important nodes, and further help them understand the contents and the structure of the various ontologies.

Author Contributions: Conceptualization, N.P.; investigation, H.K.; methodology, H.K.; Software, A.N., P.D., K.A., K.E., K.K., S.I. and K.S.; supervision, N.P.; writing—original draft, A.N., P.D., K.A., K.E., K.K., S.I. and K.S.; writing—review and editing, H.K. and N.P. All authors have read and agreed to the published version of the manuscript.

Funding: “Advanced Research Activities in Biomedical and Agro alimentary Technologies” (MIS 5002469), which is implemented under the “Action for the Strategic Development on the Research and Technological Sector”, funded by the Operational Program “Competitiveness, Entrepreneurship, and Innovation” (NSRF 2014–2020) and co-financed by Greece and the European Union (European Regional Development Fund).

Data Availability Statement: The source code and the datasets used in this paper can be found online [17].

Acknowledgments: The first version of the framework presented in this work was implemented in the context of the “Semantic Web” lesson, taught at the Department of Electrical and Computer Engineering, Hellenic Mediterranean University, Crete, Greece.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gruber, T.R. A translation approach to portable ontology specifications. *Knowl. Acquis.* **1993**, *5*, 199–220. [\[CrossRef\]](#)
2. Akbar, Z.; Mustika, H.F.; Rini, D.S.; Manik, L.P.; Indrawati, A.; Fefirenta, A.D.; Djarwaningsih, T. An ontology-driven personalized faceted search for exploring knowledge bases of capsicum. *Futur. Internet* **2021**, *13*, 172. [\[CrossRef\]](#)
3. Pappas, A.; Troullinou, G.; Roussakis, G.; Kondylakis, H.; Plexousakis, D. Exploring importance measures for summarizing RDF/S KBs. In *European Semantic Web Conference*; Springer: Cham, Switzerland, 2017; pp. 387–403.
4. Kondylakis, H.; Bucur, A.; Crico, C.; Dong, F.; Graf, N.; Hoffman, S.; Koumakis, L.; Manenti, A.; Marias, K.; Mazzocco, K.; et al. Patient empowerment for cancer patients through a novel ICT infrastructure. *J. Biomed. Inform.* **2020**, *101*, 103342. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Kim, H.M.; Laskowski, M. Toward an ontology-driven blockchain design for supply-chain provenance. *Intell. Syst. Account. Financ. Manag.* **2018**, *25*, 18–27. [\[CrossRef\]](#)
6. Pauwels, P.; Terkaj, W. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcowl ontology. *Autom. Constr.* **2016**, *63*, 100–133. [\[CrossRef\]](#)
7. Katis, E.; Kondylakis, H.; Agathangelos, G.; Vassilakis, K. Developing an ontology for curriculum and syllabus. In *European Semantic Web Conference*; Springer: Cham, Switzerland, 2018; pp. 55–59.
8. Tzagkarakis, E.; Kondylakis, H.; Vardakis, G.; Papadakis, N. Ontology based governance for employee services. *Algorithms* **2021**, *14*, 104. [\[CrossRef\]](#)
9. Spoladore, D.; Mahroo, A.; Trombetta, A.; Sacco, M. DOMUS: A domestic ontology managed ubiquitous system. *J. Ambient Intell. Humaniz. Comput.* **2021**, 1–6. [\[CrossRef\]](#)
10. Stratigi, M.; Kondylakis, H.; Stefanidis, K. Fairness in group recommendations in the health domain. In Proceedings of the IEEE 33rd International Conference on Data Engineering (ICDE), San Diego, CA, USA, 19–22 April 2017; pp. 1481–1488.

11. Stratigi, M.; Kondylakis, H.; Stefanidis, K. Fairgreys: Fair group recommendations by exploiting personal health information. In Proceedings of the International Conference on Database and Expert Systems Applications, Regensburg, Germany, 3–6 September 2018; pp. 147–155.
12. Kondylakis, H.; Koumakis, L.; Psaraki, M.; Troullinou, G.; Chatzimina, M.; Marias, E.K.K.; Tsiknakis, M. Semantically-enabled personal medical information recommender. In Proceedings of the International Semantic Web Conference (Posters & Demos), Bethlehem, PA, USA, 11–15 October 2015.
13. Kondylakis, H.; Flouris, G.; Fundulaki, I.; Papakonstantinou, V.; Tsiknakis, M. Flexible access to patient data through e-consent. In Proceedings of the 5th EAI International Conference on Wireless Mobile Communication and Healthcare, London, UK, 14–16 October 2015; pp. 263–266.
14. Spoladore, D.; Pessot, E. Collaborative ontology engineering methodologies for the development of decision support systems: Case studies in the healthcare domain. *Electronics* **2021**, *10*, 1060. [\[CrossRef\]](#)
15. CIDOC-CRM. Available online: <http://www.CIDOC-CRM.org/> (accessed on 16 June 2021).
16. Al-Aswadi, F.N.; Chan, H.Y.; Gan, K.H. Automatic ontology construction from text: A review from shallow to deep learning trend. *Artif. Intell. Rev.* **2019**, *53*, 3901–3928. [\[CrossRef\]](#)
17. Delta Ontology Evaluation System. Available online: <http://83.212.77.24/> (accessed on 1 July 2021).
18. Goldstein, A.; Fink, L.; Ravid, G. A Framework for evaluating agricultural ontologies. *Sustainability* **2021**, *13*, 6387. [\[CrossRef\]](#)
19. Raad, J.; Cruz, C. A Survey on ontology evaluation methods. In Proceedings of the International Conference on Knowledge Engineering and Ontology Development, Lisbon, Portugal, 12–14 November 2015.
20. Yao, H.; Orme, A.M.; Etzkorn, L. Cohesion metrics for ontology design and application. *J. Comput. Sci.* **2005**, *1*, 107–113. [\[CrossRef\]](#)
21. Pathak, J.; Johnson, T.M.; Chute, C.G. Modular ontology techniques and their applications in the biomedical domain. In Proceedings of the 2008 IEEE International Conference on Information Reuse and Integration, Las Vegas, NV, USA, 13–15 July 2008; pp. 351–356.
22. Schlicht, A.; Stuckenschmidt, H. Towards structural criteria for ontology modularization. In Proceedings of the WoMO, Aachen, Germany, 5 November 2006.
23. Franco, M.; Vivo, J.M.; Quesada-Martínez, M.; Duque-Ramos, A.; Fernández-Breis, J.T. Evaluation of ontology structural metrics based on public repository data. *Brief. Bioinform.* **2020**, *21*, 473–485. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Brank, J.; Grobelnik, M.; Mladenic, D. A Survey of ontology evaluation techniques. In Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD), Ljubljana, Slovenia; 2005; pp. 166–170.
25. Hlomani, H.; Stacey, D. Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey. *Semant. Web J.* **2014**, *1*, 1–11.
26. Aruna, T.; Saranya, K.; Bhandari, C. A Survey on ontology evaluation tools. In Proceedings of the 2011 International Conference on Process Automation, Control and Computing IEEE, Coimbatore, India, 20–22 July 2011; pp. 1–5.
27. Lourdusamy, R.; John, A. A review on metrics for ontology evaluation. In Proceedings of the International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 19–20 January 2018; pp. 1415–1421.
28. Tartir, S.; Arpinar, I.B.; Moore, M.; Sheth, A.P.; Aleman-Meza, B. OntoQA: Metric-based ontology quality analysis. In Proceedings of the IEEE ICDM Workshop on KADASH, Houston, TX, USA, 27 November 2005.
29. Avila, C.V.S.; Maia, G.; Franco, W.; Rolim, T.V.; Franco, A.O.R.; Vidal, V.M.P. OntoVal: A tool for ontology evaluation by domain specialists. In Proceedings of the 38th International Conference on Conceptual Modeling, Salvador, Brazil, 4–7 November 2019; pp. 143–147.
30. Lantow, B. Ontometrics: Putting metrics into use for ontology evaluation. *IC3K* **2016**, *2*, 186–191.
31. Reiz, A.; Dibowski, H.; Sandkuhl, K.; Lantow, B. Ontology metrics as a service (OMaaS). In Proceedings of the KEOD, Budapest, Hungary, 2–4 November 2020. [\[CrossRef\]](#)
32. Amith, M.; Manion, F.; Liang, C.; Harris, M.; Wang, D.; He, Y.; Tao, C. OntoKeeper: Semiotic-driven ontology evaluation tool for biomedical ontologists. *IEEE BIBM* **2019**, 1614–1617. [\[CrossRef\]](#)
33. Burton-Jones, A.; Storey, V.C.; Sugumaran, V.; Ahluwalia, P. A semiotic metrics suite for assessing the quality of ontologies. *Data Knowl. Eng.* **2005**, *55*, 84–102. [\[CrossRef\]](#)
34. Poveda-Villalón, M.; Gómez-Pérez, A.; Suárez-Figueroa, M.C. OOPS! (Ontology Pitfall Scanner!): Supporting ontology evaluation on-line. *IJISWIS* **2014**, *10*, 7–34.
35. Amith, M.; He, Z.; Bian, J.; Lossio-Ventura, J.A.; Tao, C. Assessing the practice of biomedical ontology evaluation: Gaps and opportunities. *J. Biomed. Inform.* **2018**, *80*, 1–13. [\[CrossRef\]](#) [\[PubMed\]](#)
36. Schulz, S.; Boeker, M.; Ramos, J.A.V.; Jansen, L. Pizza & wine: The need for educational tools for foundational ontologies. In Proceedings of the JOWO, Bozen-Bolzano, Italy, 21–23 September 2017.
37. Troullinou, G.; Kondylakis, H.; Stefanidis, K.; Plexousakis, D. Exploring RDFS kbs using summaries. In Proceedings of the International Semantic Web Conference, Monterey, CA, USA, 8–12 October 2018; pp. 268–284.