DACOC3 - DBpedia Archivo Challenging Ontology Consistency Check Collection

Johannes Frey[1], Denis Streitmatter[1] and Sebastian Hellmann[1]

[1]*Knowledge Integration and Linked Data Technologies (KILT/AKSW) Group & DBpedia Association, Institute for Applied Informatics (InfAI), Leipzig University, Germany*

Abstract

DBpedia Archivo is an online ontology interface and open augmented archive, containing more than 1,400 ontologies. It uses several fully automated ontology discovery mechanisms that run each week and have turned Archivo into one of the most exhaustive, and recent ontology archives. Archivo daily checks for new ontology versions and performs automated tests to evaluate the fitness for use of an ontology. As part of a 4-star quality rating, a logical consistency check is applied. However, several ontologies contained in Archivo cause problems with current reasoner implementations when verifying consistency, leading to timeouts and other runtime failures. In this paper, we present an approach to create a collection of such challenging ontologies and report key characteristics of these ontologies, that can be easily consumed by reasoning applications in order to evaluate their performance and stability.

Keywords

ontology consistency check, ontology reasoning challenge, ontology evaluation, augmented ontology archive

1. Introduction

DBpedia Archivo‘s initial vision was to create a fully automated, persistent ontology archive that can serve as a backbone for the Semantic Web [1] and brings a convenient and stable interface to ontology consumers [2].

Launched in May 2020, Archivo has meanwhile become one of the most exhaustive and recent ontology archives, providing alternative, persistent, and unified access to over 1,400 ontologies[1] in more than 3,700 versions. As of September 2021 growth has not reached a plateau, yet and it is steadily growing at a pace of around 12.6 ontologies per week (6 month average). While more than 1240 ontologies were archived automatically via web-scale discovery mechanisms, Archivo also performed over 160 successful ontology inclusions suggested by the community (i.e. submitting the ontology URL manually at https://archivo.dbpedia.org/add). This fact and around 90 ontology downloads on an average day (plus 640 daily downloads from major bots) show that Archivo is already being adopted by the community.

SemREC'21: Semantic Reasoning Evaluation Challenge, ISWC'21, Oct 24 – 28, Albany, NY

frey@informatik.uni-leipzig.de (J. Frey); streitmatter@informatik.uni-leipzig.de (D. Streitmatter); hellmann@informatik.uni-leipzig.de (S. Hellmann)
https://aksw.org/JohannesFrey (J. Frey)
0000-0003-3127-0815 (J. Frey)

[1] https://archivo.dbpedia.org/list

DBpedia Archivo has the potential to create a Unified Semantic Ontology Space (USOS), a holistic view over all available ontologies. Instead of soft and fuzzy principles or publishing guidelines, it uses hard, implementable criteria to evaluate ontologies in preparation of a well-defined, measurable standard in the future, which will not only help consumers to find usable and useful ontologies for their need, but will also ultimately yield better and reliable ontologies for (industrial) applications.

While Archivo is liberal w.r.t. the requirements towards an ontology in order to be indexed and archived by it, it augments the ontology with reports of SHACL-based quality tests and other Feature and Evaluation Plugins. Moreover, a 4-star rating gives a summary on the fitness for use of an ontology. As a crucial aspect of fitness for use, we consider logical consistency of an ontology. Four stars will only be awarded to an ontology which parses without errors via a proper Linked Data deployment, is containing a valid dct:license statement, and passes the consistency check.

However, evaluating the consistency for all ontologies in Archivo in a reliable way poses several challenges. We experienced loading and runtime errors as well as timeouts for many Archivo ontologies. By evaluating the current state of affairs w.r.t. consistency evaluation for different reasoners, we can contribute to both the ontology and reasoner developer community. We see Archivo as a good foundation and the experiments in this work as a first step to create flexible real world ontology benchmarks in the future.

The remainder of the paper is structured as follows. In the next section, we briefly sketch how Archivo discovers ontologies. In Section 3, the approach to determine challenging ontologies for DACOC3 is explained. Section 4 describes how the collection of ontologies can be accessed. Section 5 concludes and discusses future work.

2. Archivo Ontology Discovery

We devised four generic approaches to discover OWL and SKOS ontologies to be archived in Archivo; first, by vocabulary usage analysis of all RDF assets on the DBpedia Databus[2] via VoID Mods (dataset class and property usage analysis); Second, by querying already existing ontology repositories/registries (currently Linked Open Vocabularies [3] and prefix.cc). Moreover, we discover (transitive) dependencies/imports in ontologies from previous iterations of Archivo crawls. Finally, users can issue automated inclusion requests for missing ontologies via a Web interface. These approaches allow Archivo to have a good coverage of meaningful and relevant ontologies of the Semantic Web, while preventing uploading of incorrect ontologies (ontology hijacking or spamming) by users.

3. Challenging Ontologies Selection

In order to determine challenging ontologies, we selected the latest (as of September 13, 2021) snapshot version of each of the 1403 ontologies contained in Archivo and performed a consistency check with multiple reasoners. As input files we used the parsed NTriples files (using

[2] https://databus.dbpedia.org/

Table 1
Archivo Ontologies Consistency Check Processing Issues: Reported are the number of ontologies, that could not be loaded by OWL API, leaded to another error or exception when processed by the respective reasoner, or were subject to a timeout.

	HermiT	Openllet	ELK
loading error	143	143	143
other error	60	6	0
timeout	2	48	1

Raptor RDF Syntax Library in version 2.0.14) from Archivo. We loaded the ontology using OWL API 5.1.8. Subsequently, we used ELK 0.5.0, Openllet 2.6.5, and HermiT 1.4.3.517 to perform a consistency check (including imported ontologies) and measured the execution time for each of the tools. The experiment was run on a Ubuntu 20.04.3 server, with 64 AMD Opteron 6376@2.6 GHz CPUs, 256GB RAM and using Java OpenJDK version 11.0.12. After a timeout of 10 minutes the consistency check was aborted. The experiment code is available on GitHub[3].

When performing the experiment, the consistency check aborted for over 10 % of the ontologies. In Table 1, we have further analyzed the processing issues. The loading via OWL API failed for 143 ontologies. When passing the OWL API model to the reasoners, additional 60 ontologies failed for HermiT and 6 respectively for Openllet. In case of the latter another 48 ontologies hit the 10 minute computation timeout, while for HermiT two and for ELK only one ontology exceeded the timeout. The complete results of the experiment are available as interactive spreadsheet[4].

Since we wanted to select ontologies that are challenging for multiple reasoners, we filtered for ontologies which lead to issues with at least two of the three reasoners. The resulting three DACOC3 ontologies are shown with their respective computation times in Table 2. While ELK is parallelized, it is worth mentioning, that ELK does not support all types of OWL EL statements and prints warnings that the consistency report might not be accurate. This is likely to explain the huge time difference between ELK and the other reasoners that timed out for ExtruOnt. When having a look at the characteristics of the selected ontologies in Table 3, we see that ExtruOnt [4] (an ontology from engineering / industry 4.0 domain to represent extruders) is a rather small ontology, which, however has multiple `owl:imports` statements. The Unified Phenotype Ontology (uPheno) is an umbrella ontology which consists of imports and two metadata statements only, and integrates multiple phenotype ontologies, leading to timeouts for all tested reasoners. In contrast COSMO [5] (an upper level ontology to enable broad semantic interoperability) has no imports, but a high number of axioms, classes, and properties, which causes a timeout in Openllet but an error due to a malformed float literal (incorrect decimal point ;) in HermiT.

[3] https://github.com/yum-yab/archivo-consistency-mod

[4] https://docs.google.com/spreadsheets/d/1xLZqrbLtZV1qPLLotz1VpOK6zm3C0PkB8yvV3ow2Yo4/edit#gid=154794120

Table 2
DACOC3 Computation Times: Reported is the computation time for a consistency check in milliseconds for three popular reasoners. A timeout (T/O) of 10 minutes (600,000 ms) has been used.

Ontology Title	HermiT	Openllet	ELK
ExtruOnt	T/O	T/O	421
COSMO ontology	Error	T/O	2,309
Unified Phenotype Ontology (uPheno)	T/O	T/O	T/O

Table 3
DACOC3 Ontology Characteristics

Ontology Title	Bytes	Triples	Axioms	Classes	Properties	Imports
ExtruOnt	134,012	1,071	324	35	18	yes
COSMO ontology	58,913,939	365,940	272,100	24,391	1,442	no
uPheno	275	2	0	0	0	yes

4. Collection Access

DACOC3 is represented as DBpedia Databus Collection[5]. This allows persistent access to the DACOC3 ontologies via stable Databus download URLs / identifiers. The persistent download URLs can be copied from the Collection web view or programmatically retrieved via simple bash snippets (also documented on the Collection view). Additionally, the Databus Client[6] can be leveraged to download and convert the ontology files into various RDF formats. Finally, the ontologies can be also retrieved via the Archivo API[7] supporting HTTPS, CORS, timestamp versioning, and NTriples, Turtle, as well as the OWL format.

5. Conclusion and Future Work

We presented an approach to determine challenging real world ontologies using DBpedia Archivo. We identified three challenging ontologies for DACOC3 originated from different domains and having different characteristics. Besides in-depth debugging of the three ontologies in the different reasoners, the entire experiment results allow to further analyze errors during consistency evaluation in order to study and improve fault tolerance, stability and incorrect behavior of these reasoners in the future.

Moreover, we plan to integrate the consistency check performance analysis into Archivo in the future (making them accessible via Databus Mods and SPARQL) to give feedback to ontology developers but also allow researchers to have a fine grained view on the USOS similar to OOSP [6]. With regard to (transitive) ontology dependencies, we would like to implement a transparent proxying tool for reasoners and other semantic tools, that allows reliable and deterministic repeatability of experiments accessing ontologies (including the

[5] https://databus.dbpedia.org/jfrey/collections/dacoc3
[6] https://github.com/dbpedia/databus-client
[7] https://archivo.dbpedia.org/api

ones with *owl:imports* statements), by retrieving the correct, persistent ontology snapshots via Archivo instead of the original URL destination.

Acknowledgments

This work was partially supported by grants from the Federal Ministry for Economic Affairs and Energy of Germany (BMWi) to the projects LOD-GEOSS (03EI1005E) and PLASS (01MD19003D).

References

[1] J. Frey, S. Hellmann, Fair linked data - towards a linked data backbone for users and machines, in: WWW Companion, 2021. doi:10.1145/3442442.3451364.

[2] J. Frey, D. Streitmatter, F. Götz, S. Hellmann, N. Arndt, Dbpedia archivo: A web-scale interface for ontology archiving under consumer-oriented aspects, in: Semantic Systems, volume 12378 of *LNCS*, Springer, 2020, pp. 19–35. doi:10.1007/978-3-030-59833-4_2.

[3] P. Vandenbussche, G. Atemezing, M. Poveda-Villalón, B. Vatant, Linked open vocabularies (LOV): A gateway to reusable semantic vocabularies on the web, Semantic Web 8 (2017) 437–452. URL: https://doi.org/10.3233/SW-160213. doi:10.3233/SW-160213.

[4] V. Ramírez-Durán, I. Berges, A. Illarramendi, Extruont: An ontology for describing a type of manufacturing machine for industry 4.0 systems, Semantic Web 11 (2020) 1–23. doi:10.3233/SW-200376.

[5] D. A. Quartel, M. W. Steen, S. Pokraev, M. J. Van Sinderen, Cosmo: A conceptual framework for service modelling and refinement, Information Systems Frontiers 9 (2007) 225–244.

[6] O. Zamazal, V. Svátek, OOSP: ontological benchmarks made on the fly, in: G. Cheng, K. Gunaratna, A. Thalhammer, H. Paulheim, M. Voigt, R. García (Eds.), International Workshop on Summarizing and Presenting Entities and Ontologies, volume 1556 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2015. URL: http://ceur-ws.org/Vol-1556/paper1.pdf.