OntoAnon: An Anonymizer For Sharing Ontology Structure Without Data

Achim Reiz^a, Robert Schlücker^a and Kurt Sandkuhl^a

^a Rostock University, 18051 Rostock, Germany

Abstract

The ontologies developed in enterprises often store sensitive information on products, processes, and the overall business. Companies are (understandably) reluctant to share them with persons outside their organization. For some use-cases, though, it is not necessarily the data but the structure that is of interest.

This paper presents "OntoAnon", an anonymizer for ontologies. The Python-based application allows the removal of sensitive information like class-and property names, as well as annotation contents, while preserving the ontology structure and used formalisms. This allows the developing knowledge engineers to use tools like OOPS and NEOntometrics without compromising sensitive data. Further, it allows researchers to collect insensitive but valuable data on enterprise ontologies, e.g., to study evolutional processes.

Keywords

Ontology, Anonymizer, OOPS, NEOntometrics, OWL, RDF

1. Introduction

Ontologies often contain sensitive data on business processes, products, or persons. The need to safeguard sensitive internal data within ontologies has emerged as a critical challenge. At the same time, online tools can aid the development of ontologies. Their recommendations are not necessarily based on content but can also rely on structure, like graph properties or axiom usage. Examples are NEOntometrics [1] and OOPS [2]. Due to security guidelines, these tools are often not allowed, as the ontologies would be shared with an untrusted third party.

As a result, the knowledge engineer misses out on the potentially helpful tools. Also, as researchers and developers of these tools, we miss out on valuable data. As the developer of NEOntometrics, the authors of this paper experienced it firsthand: While gathering data on open-source development processes is relatively easy and already allowed novel discoveries regarding ontology evolution [3], getting insights into internal enterprise development processes is often denied. However, investigating these internal data yields enormous research potential. For example, software evolution research showed discrepancies between open- and closed-source software developments [4]. Similar discrepancies in ontology evolution stand to reason: While open-source data showed no signs of stereotypical development processes, the statement is not yet proven for internal enterprise data. Tackling these research questions does not require access to the internal ontology data – the structural properties are sufficient.

Until today, however, no application allows for sharing structural accurate, anonymized ontologies. We aim to solve this lack with OntoAnon, an anonymization tool for ontologies that maintains structural integrity. OntoAnon removes the textual information in an ontology, like the elements' names and annotation contents, while preserving the structural attributes. It runs locally, has a simple graphical user interface (GUI), and creates a textual file for backtracing the translations. Thus, OntoAnon allows knowledge engineers to use tools like NEOntometrics or OOPS without sharing actual data. This also

CEUR Workshop Proceedings (CEUR-WS.org)

SEMANTICS 2023 EU: 19th International Conference on Semantic Systems, September 20-22, 2023, Leipzig, Germany EMAIL: achim.reiz@uni-rostock.de (A. 1); robert.schlücker@@uni-rostock.de (A. 2); kurt.sandkuhl@uni-rostock.de (A. 3)

ORCID: 0000-0003-1446-9670 (A. 1); 0000-0002-7431-8412 (A. 3) © 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

enables the developer of these tools to collect further valuable information on ontologies that normally would not be shared.

2. Related Work

Data anonymization and privacy-preserving data publishing already have a broad research foundation with many available approaches and techniques [5]. With the rise of knowledge graphs in various domains, the specifics of sharing graph data without compromising confidentiality got significant research attention:

Delanaux et al. [6] remove sensitive information with SPARQL-based anonymization policies. They define queries for information that either shall be preserved or removed, which are the basis for the anonymization algorithm. Thouvenot et al. [7] developed a technique for anonymization based on grouping and anatomization, thus altering the relations between critical data points. Hoang et al. [8] developed a time-ware algorithm for evolving graphs that regards insert, update, delete, and re-insert operations. The same authors also proposed a *k-ad* approach for adding additional edges to mask users [9].

However, most of today's applications view anonymization from the viewpoint of data, not structure. They are concerned with guarding privacy and less with sharing the most accurate view of the ontology structure. Furthermore, current approaches often lack a simple-to-use, easy implementation.

OntoAnon focuses on an easy sharing of ontology structure. It suppresses and substitutes all customcreated vocabulary while preserving the w3c standardized vocabulary. Compared to other approaches that target the removal of identity information, none of the data is no longer available after reading, but the graph itself keeps complete integrity.

3. OntoAnon

OntoAnon is a standalone Python application. It is open source and available on GitHub². The application can either be started by downloading and starting the bundled executable (currently only available for Windows), by installing the package via pip^3 and calling "OntoAnon" from the terminal, or starting the code manually, best by creating a virtual environment using the included *pipfile* and the software *pipenv*, which installs the only external requirement, the package *rdflib*⁴.

🖉 Ontology Anonymization Tool				×
Select a local ontology file and	a place to store the tran	slation dictionary.		
Ontology File Identify Form	at Anonymized File	Dictionary File	Namespa	aces
Ontology File:				
Ontology Format: Anonymized File:				
Dictionary File:				
Anonymize				

Figure 1: The GUI of OntoAnon

After startup, a GUI appears and queries the required input parameter for the ontology anonymization (cf. Figure 1). The first field, the *Ontology File*, is the location of the ontology that shall

² https://github.com/Uni-Rostock-Win/OntoAnon

³ https://pypi.org/project/OntoAnon/

⁴ http://rdflib.readthedocs.io/

be anonymized. *Identify Format* infers serialization, and *Anonymized File* points to the future anonymized ontology. The *Dictionary File* points to the path of the translation file. *Namespaces* allow the selection of vocabularies that shall be preserved. By default, all the standardized vocabularies like the ontology languages RDF(S), *OWL*, and elements from w3c ontologies like *FOAF*, *PROV*, or DC^5 are not anonymized. However, it is also possible to deselect ignored namespaces or add further ones.

A click on *Anonymize* starts the given process. The application now loads the graph, iterates through all triples, and replaces non-standardized vocabulary. In this way, the output graph has the same shape, individuals, and exact usage of the respective attributes without containing actual data. The anonymization process is depicted in Figure 2.



Figure 2: Flowchart for anonymization process. This routine is called for every element in the subjects, predicates, and objects.

After the translation process, the labels, URIs, and literals no longer contain any helpful information. Figure 3 exemplary presents the object property hierarchy and the object property attributes of *OBI_000304* of the *software ontology (swo)*¹⁰ loaded in protégé before and after completing the anonymization process.





The corresponding translation file shows how the namespaces, URIs, and literals have been renamed, allowing the user to trace back the anonymized terms to the original ones. It allows using structure-based tools like OOPS without sharing critical information and then applying the results to the original ontology.

An example of such a use case of the translations is given in Figure 4. It shows an excerpt of an OOPS analysis for the SWO run on the anonymized files. It identified a pitfall for *Subject97*. The

⁵ Standard Vocabulary: *BRICK, CSVW, DC, DCAT, DCMITYPE, DCTERMS, DCAM, DOAP, FOAF, ODRL2, ORG, OWL, PROF, PROV, QB, RDF, RDFS, SDO, SH, SKOS, SOSA, SSN, TIME, VANN, VOID, WGS, XSD, SWRL*

structural analysis results can be backtracked to the class *OBI_000304* using the translation file. The example shows that the structural recommendations and results of a tool like OOPS can be used without sharing the actual content.



Figure 4: An excerpt for the translation file (top). It allows using tools like OOPS (bottom).

4. Validation

To check whether the anonymization yields valid results, we tested five ontologies out of public repositories with varying sizes and checked whether the anonymized versions had the same structural attributes as the original files using the NEOntometrics web service [1].

The numerical comparison of structural properties showed no difference: OntoAnon does anonymize the given ontologies reliably. However, the execution time for larger ontologies is considerable. The largest tested ontology, *Foodon*, with 318.105 axioms, takes 252 seconds. The test machine, however, was a mid-sized business notebook⁶ not explicitly prepared for running the performance test. The input data and the test results are available online⁷.

Table 1

Tested and validated ontologies using NEOntometrics.

Ontology	Axioms	Classes	Execution Time	Successful			
biorealm-full ⁸	9,393	629	8 Seconds	\checkmark			
oba-base ⁹	78,606	21,754	159 Seconds				
Software Ontology ¹⁰	19365	1,971	13 Seconds				
UK Parliament ¹¹	3509	188	1 Seconds				
FoodOn ¹²	318,105	34,200	252 Seconds	\checkmark			

- ⁹ https://github.com/obophenotype/bio-attribute-ontology
- ¹⁰ https://github.com/allysonlister/swo

⁶ Lenovo Thinkpad L390 Yoga, 16GB Ram, i7-8565U

⁷ https://github.com/Uni-Rostock-Win/ontoAnon-Testdata

⁸ https://github.com/EnvironmentOntology/biorealm

¹¹ https://github.com/ukparliament/Ontology

¹² https://github.com/FoodOntology/foodon

5. Conclusion

Ontologies allow to formally describing of a domain to computers and humans. Thus they often contain sensitive information on products, processes, and business rules. These contents can hinder the sharing of ontologies, even though the interest may be more in the structural representations than in the contents. The proposed OntoAnon software offers a solution to this concern. It is a small, locally run, python-based software that removes the ontology content while preserving the structural integrity. The anonymized ontology allows using web-based software like NEOntometrics or OOPS without sacrificing data security. In this way, the authors hope that OntoAnon eases the collaboration between researchers and practitioners and allows more empirical insight into the structural developments and properties of enterprise ontologies.

While the presented application anonymizes a single ontology, a potential future development shall anonymize a whole git repository to enable the studying of evolutional development processes of enterprise ontologies and repositories. Furthermore, while the application already allows the selection and deselection of namespaces to be customized, a granular selection capability on subclasses and T-Box or A-Box could further aid possible sharing and usage scenarios.

6. References

- [1] A. Reiz, K. Sandkuhl, NEOntometrics: A Flexible and Scalable Software for Calculating Ontology Metrics, in: Proceedings of Poster and Demo Track and Workshop Track of the 18th International Conference on Semantic Systems co-located with 18th International Conference on Semantic Systems (SEMANTICS 2022), CEUR-WS, Vienna, 2022.
- [2] M. Poveda-Villalón, A. Gómez-Pérez, M.C. Suárez-Figueroa, OOPS! (OntOlogy Pitfall Scanner!), Semantic Web and Information Systems 10 (2014) 7–34. https://doi.org/10.4018/ijswis.2014040102.
- [3] A. Reiz, K. Sandkuhl, Debunking the Stereotypical Ontology Development Process, in: Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, Valletta, Malta, SCITEPRESS - Science and Technology Publications, 2022, pp. 82–91.
- [4] I. Herraiz, D. Rodriguez, G. Robles, J.M. Gonzalez-Barahona, The evolution of the laws of software evolution, ACM Comput. Surv. 46 (2013) 1–28. https://doi.org/10.1145/2543581.2543595.
- [5] V. Torra, Guide to Data Privacy, Springer International Publishing, Cham, 2022.
- [6] R. Delanaux, A. Bonifati, M.-C. Rousset, R. Thion, Query-Based Linked Data Anonymization, in: D. Vrandečić, K. Bontcheva, M.C. Suárez-Figueroa, V. Presutti, I. Celino, M. Sabou, L.-A. Kaffee, E. Simperl (Eds.), The Semantic Web – ISWC 2018, Springer International Publishing, Cham, 2018, pp. 530–546.
- [7] M. Thouvenot, O. Cure, P. Calvez, Knowledge Graph Anonymization using Semantic Anatomization, in: 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, IEEE, 2020, pp. 4065–4074.
- [8] A.-T. Hoang, B. Carminati, E. Ferrari, 2022. Time-Aware Anonymization of Knowledge Graphs. ACM Trans. Priv. Secur., 3563694. https://doi.org/10.1145/3563694.
- [9] A.-T. Hoang, B. Carminati, E. Ferrari, Cluster-Based Anonymization of Knowledge Graphs, in: M. Conti, J. Zhou, E. Casalicchio, A. Spognardi (Eds.), Applied Cryptography and Network Security, Springer International Publishing, Cham, 2020, pp. 104–123.