CEUR-WS.org/Vol-3064/DBpedia-Technology-tutorial.pdf

The DBpedia Technology Tutorial

Milan Dojchinovski[1,2], Jan Forberg[1], Johannes Frey[1], Marvin Hofer[1], Denis Streitmatter[1] and Kirill Yankov[1]

[1]Knowledge Integration and Language Technologies (KILT/AKSW), DBpedia Association/InfAI, Leipzig University, Leipzig, Germany
[2]Web Intelligence Research Group, FIT, Czech Technical University in Prague, Prague, Czech Republic

Abstract

DBpedia (https://www.dbpedia.org) is a crowd-sourced community effort which aims at extraction and publishing structured information from various Wikimedia projects. This structured information resembles an open knowledge graph, the DBpedia Knowledge Graph, which is publicly available for everyone on the Web. The DBpedia Knowledge Graph has been under development for many years and is being improved to this day. In this tutorial, participants gained general information on the DBpedia Knowledge Graph and the DBpedia community. The tutorial also provided information on the complete DBpedia Knowledge Graph lifecycle, i.e. from extraction and modelling to publishing and maintenance of the DBpedia KG. A particular focus was put on the DBpedia Infrastructure, i.e. the DBpedia's Databus publishing platform and the associated DBpedia services, i.e. DBpedia Spotlight, DBpedia Lookup, the DBpedia service endpoints and DBpedia Archivo.

Keywords

DBpedia, Linked Data, knowledge extraction, knowledge integration, open data

1. Introduction

Over the last decade, DBpedia has become one of the most widely used knowledge graphs and one of the central interlinking hubs in the Linked Open Data (LOD) cloud[1]. The ultimate goal of the DBpedia community project is to *i) build a large-scale, multilingual knowledge graph (KG) by providing structured information extracted from Wikipedia*, and to *ii) integrate and complement this knowledge with information from other sources.* Over the last few years, the DBpedia core team has significantly consolidated the knowledge and technology around DBpedia and introduced some novel technologies and concepts. These efforts have positively impacted the community around DBpedia, which has a constant growth. According to the bibliographic database Google Scholar, there are over 39,800 articles[2] citing DBpedia; using DBpedia or developing technology for DBpedia. The ultimate goal of the tutorial was to provide an overview of the latest advancements in the DBpedia technology stack and the DBpedia KG lifecycle, together with detailed discussions on the regular DBpedia releases, the DBpedia infrastructure and services, and concrete usage scenarios of the DBpedia knowledge graph and the technology behind it.

2. The Tutorial

The aim of this tutorial was, from a practical perspective, to explain in detail the process of replicating the DBpedia infrastructure, exploiting DBpedia in third-part applications and contributing and improving the DBpedia knowledge graph and services. Hands-on sessions were organized to internalize the DBpedia technology stack with practice. The topics covered in the tutorial are as follows:

- DBpedia KG in the Nutshell
- DBpedia Technology Stack
- DBpedia Databus Platform
- DBpedia Infrastructure Services
- Replication of the DBpedia Infrastructure
- Consumption of the DBpedia KG
- Use cases and Applications of DBpedia

The tutorial was organized in two parts. The first part was exclusively dedicated for beginners and people that have minimal knowledge about DBpedia (Section 3). It provided a general information on the DBpedia project and the community behind it . It also provided information on how the DBpedia knowledge graph is created and how the extracted knowledge is organized (Section 3.1) and where the knowledge can be found (Section 3.2). A particular topic covered in the first part was on the recently created DBpedia's Dutch National Knowledge Graph (Section 3.3).

The second part provides more detailed information on the DBpedia technology ecosystem. In particular, the DBpedia Databus platform (Section 4.1) as a platform for publishing DBpedia's KG data artifacts, the DBpedia release process (Section 4.2), the Databus Mods component

Language, Data and Knowledge Conference, September 01–04, 2021, Zaragoza, Spain

dojchinovski@informatik.uni-leipzig.de (M. Dojchinovski); forberg@infai.org (J. Forberg); frey@informatik.uni-leipzig.de (J. Frey); hofer@informatik.uni-leipzig.de (M. Hofer); streitmatter@informatik.uni-leipzig.de (D. Streitmatter); yankow@infai.org (K. Yankov)
http://www.dojchinovski.mk (M. Dojchinovski)

© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

[1]https://lod-cloud.net
[2]https://scholar.google.com/scholar?q=intext%3Adbpedia

to provide additional metadata for datasets (Section 4.3), DBpedia ID management , pre-fusion and cartridge creation process (Section 4.4) and the DBpedia Archivo; an ontology archive service (Section 4.5).

3. Part 1: Getting Started with DBpedia

3.1. DBpedia in a Nutshell

DBpedia is a community project which has been initiated in 2007 with the ultimate goal to extract and publish knowledge from various Wikipedia projects. In 2007, was run the first extraction and DBpedia KG for first time was available as Linked Data and offered for querying via a dedicated SPARQL endpoint[3]. In 2010, the DBpedia Ontology was made open for editing and already in 2011 big companies and organisations, such as IBM Watson, Yahoo!, BBC, Siemens and the Unicode Consortium, have started adopting DBpedia. In the period between 2012 and 2016 the DBpedia was internationalized and adopted to over 138 Wikipedia languages. In order to establish better coordination of the DBpedia efforts, the DBpedia Association was founded in 2014, hosted at the Institute for Applied Informatics (InfAI) at the Leipzig University. In 2020, DBpedia has published the largest ever release with over 21B facts.

DBpedia enables users to query Wikipedia and retrieve structured knowledge using the SPARQL query language. The queries can vary from simple queries such as *"Retrieve all persons, their names in English, their country of birth and country population"* to much more complex queries such as *"Retrieve all soccer players, who are born in a country with more than 10 million inhabitants, and played as a goalkeeper for a club that has a stadium with more than 30.000 seats."*

Over the years the DBpedia mission has also evolved. From the initial, still valid mission *"A crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web."*, to the new mission *"Global and unified access to knowledge graphs"*.

The DBpedia community is organized into DBpedia chapters, each with its own focus. *Language chapters* - maintaining particular DBpedia language (e.g. English - core language, German, Dutch, Czech, etc.), *regional chapters* - focused on regions or cities related information (see Section 3.3 for more information on the DBpedia's Dutch National Knowledge Graph), and *domain chapters* - focused on data related to libraries, museums, law, medicine, linguistics, etc.

The DBpedia Association, as the core management entity, brings together companies, non-profit organisations, start-ups and self-employed/tiny entities, which have particular interest in contributing to the DBpedia project. As of September 2021, the DBpedia Association consists of 32 DBpedia members: 41% industry and start-up, 37% non-profit and 22% tiny & self-employed. More information about the DBpedia Association membership can be found at https://www.dbpedia.org/members/membership/.

3.1.1. The Overarching DBpedia KG Release Process

The overall DBpedia KG release process consists of six phases, which are as follows:

- **Phase 1: Mappings and ontology definitions:** during this phase old mappings are being updated and new mappings introduced. The mappings are maintained at the DBpedia mappings server[4] and are used to homogenize the information extracted from Wikipedia.
- **Phase 2: Knowledge extraction:** in this phase the DBpedia Information Extraction Framework (DIEF)[5] is executed and structured information is extracted from each Wikipedia language edition. The extraction process can be monitored at the DBpedia release dashboard[6].
- **Phase 3: Data validation:** in this phase the data is parsed and validated against a strict set of rules. This is necessary in order to assure that the published data is of high quality and has a valid RDF syntax.
- **Phase 4: Release of data artifacts:** the data is published on the DBpedia Databus[7] and each data artifact is versioned and published with rich metadata.
- **Phase 5: ID management and fusion:** data partitions for different languages are fused and identifiers are assigned using the DBpedia's Global ID identifier system[8].
- **Phase 6: KG deployment:** finally, the core of the DBpedia knowledge graph (latest-core data collection[9]) is published as Linked Data and served via the main DBpedia SPARQL endpoint[10].

3.1.2. Main DBpedia Dataset Groups

The data published by DBpedia is organized into several dataset groups. The main four dataset groups are as follows:

[3] https://dbpedia.org/sparql

[4] http://mappings.dbpedia.org/

[5] https://github.com/dbpedia/extraction-framework

[6] https://release-dashboard.dbpedia.org

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

[8] https://global.dbpedia.org

[9] https://databus.dbpedia.org/dbpedia/collections/latest-core

[10] https://dbpedia.org/sparql

- **Mappings-based**[11]: data extracted using the predefined mappings and exported using the http://dbpedia.org/ontology/ properties. Currently, there are mappings defined for over 40 languages.
- **Generic**[12]: information from automatic extraction from Wikpiedia pages. The generic extraction exports information from unmpped information in infoboxes and other structured information found in the Wikipedia pages (e.g. categories, images, redirects, etc.). The data is exported using http://dbpedia.org/property/ properties.
- **Text**[13]: contains Wikipedia articles text, links in the text and the structure of the articles (i.e. sections, sub-sections and paragraphs)[1].
- **Wikidata**[14]: data from Wikidata modelled using DBpedia properties. The Wikidata extraction approach implements the mappings-based and the generic extraction techniques.

3.1.3. The DBpedia Ontology

The DBpedia ontology[15] is the backbone of the DBpedia KG. The DBpedia ontology is designed as a shallow cross-domain ontology which is used to model the information from Wikipedia. While it is primarily used to capture information from Wikipedia, it also provides mappings and links to other ontologies, e.g. schema.org, Wikidata, etc. The ontology is generated on-the-fly as changes are introduced in the mappings. Currently, the ontology contains over 700 classes and more than 3,000 properties. All versions of the ontology are published the on Databus platform[16] via the DBpedia Archivo service (see Section 4.5).

3.2. Getting Started with DBpedia

As the DBpedia KG is provided into many different files, one of the first steps of working with DBpedia data is retrieving the relevant DBpedia data files for a specific use case. A more NLP focused task, for instance, would require data containing Wikipedia abstracts and labels while a task involving geo-spacial information would require data containing that respective information.

There are multiple ways of retrieving DBpedia data. The simplest way is to get the data from the DBpedia download file server or the official DBpedia SPARQL endpoint. However, the recommended, most efficient, way of retrieval is via the DBpedia Databus (https://databus.dbpedia.org.

The DBpedia Databus is a metadata repository that holds pointers to the actual files along with means to verify both, the file content and correctness of the metadata. It allows publishers to hierarchically structure their data into so-called groups and artifacts. While a group is a collection of multiple artifacts, an artifact is a logical data entity that is associated with files of different formats, languages or versions that describe a common topic. This allows data consumers to retrieve specific information in a well-structured way by using group and artifact identifiers.

The latest version of the DBpedia KG is released on the DBpedia Databus every month. The files are then dascribed as Databus artifacts based on their contents and Databus groups based on their extraction method. For example, all files in the *generic* group have been extracted with the generic extraction module of the extraction framework. In addition to the structural information, DBpedia groups and artifacts provide extensive textual documentation to allow consumers to select the correct data for their task. The HTML interface of the Databus provides a faceted file browser that can help with the file selection (see Figure 1).

In order to streamline data selection and retrieval, the Databus provides a so-called *Databus Collection* feature. Since all the metadata published on the Databus is RDF, it is possible to query for specific information using SPARQL. In its essence, a Databus Collection is a SPARQL query annotated with label and description, with the query result being a list of file URIs. The collection can be published on the Databus to make it reusable and accessible to other Databus users. This enables the creation of predefined sets of files for specific use cases. For example, the DBpedia Databus provides collections for time-based snapshots of the DBpedia data as well as the so-called “Latest Core Collection”[17] - a dynamic collection that always points to the files of the latest DBpedia core data.

3.3. DBpedia‘s Blueprint for Creating National Knowledge Graphs

Linked Data provides a scalable, entity-centered access mechanism for resources on the Web of Data. However, its decentral nature introduces many challenges w.r.t. findability and interoperability (FAIR Linked Data) [2]. Linked Data, ideally, is sourced by a plethora of providers, where each source contributes their own in order to cover the long tail of information. In order to tackle these challenges, the DBpedia core team developed a solution, named as a “Blueprint for Creating National Knowledge Graphs”. The blueprint aims at sustainable, flexible and transparent creation of National Knowledge

[11] https://databus.dbpedia.org/dbpedia/mappings/

[12] https://databus.dbpedia.org/dbpedia/generic/

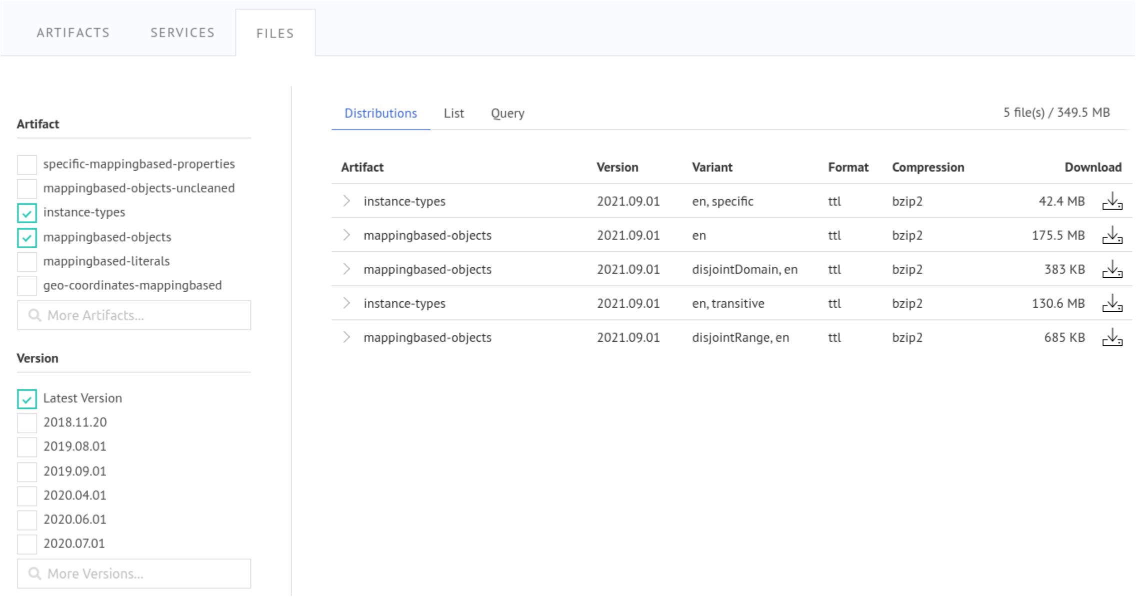
[13] https://databus.dbpedia.org/dbpedia/text/

[14] https://databus.dbpedia.org/dbpedia/wikidata

[15] https://www.dbpedia.org/resources/ontology/

[16] https://databus.dbpedia.org/ontologies/dbpedia.org/ontology--DEV

[17] https://databus.dbpedia.org/dbpedia/collections/latest-core



ARTIFACTS SERVICES FILES

Artifact

□ specific-mappingbased-properties
□ mappingbased-objects-uncleaned
☑ instance-types
☑ mappingbased-objects
□ mappingbased-literals
□ geo-coordinates-mappingbased

More Artifacts...

Version

☑ Latest Version
□ 2018.11.20
□ 2019.08.01
□ 2019.09.01
□ 2020.04.01
□ 2020.06.01
□ 2020.07.01

More Versions...

Distributions List Query 5 file(s) / 349.5 MB

Artifact	Version	Variant	Format	Compression	Download
instance-types	2021.09.01	en, specific	ttl	bzip2	42.4 MB
mappingbased-objects	2021.09.01	en	ttl	bzip2	175.5 MB
mappingbased-objects	2021.09.01	disjointDomain, en	ttl	bzip2	383 KB
instance-types	2021.09.01	en, transitive	ttl	bzip2	130.6 MB
mappingbased-objects	2021.09.01	disjointRange, en	ttl	bzip2	685 KB

Figure 1: Faceted file browsing on the DBpedia Databus.

Graphs (NKGs). The blueprint is based on the concept of cartridges, which are defined as unified, interoperable, modularized and materialized views of Linked Data sources managed on the DBpedia Databus.

The blueprint has been successfully instantiated on the DBpedia infrastructure and validated on a concrete complex scenario creating the Dutch National Knowledge Graph[18]. The DNKG integrates 6 different sources, including four authoritative sources such as National Library of the Netherlands (KB), Building and Addresses Register of Netherlands (BAG), Cultural Heritage Objects (CHO), Artists dataset from the Netherlands Institute for Art History (RKDA) in combination with data from Dutch DBpedia and Digital Bibliography & Library Project (DBLP). The system consists of loosely coupled components which support the overall data integration lifecycle, from data acquisition and mappings, to ID management, cartridge creation, and knowledge fusion. We validated the approach on a concrete pilot and created the first Dutch National Knowledge Graph, which integrates data from authoritative sources of the Netherlands and contains over 300 million RDF triples. Configuration files and built information are available in the DNKG-pilot[19] Github repository.

4. Part 2: Deep in the DBpedia Ecosystem

4.1. The Databus Platform

The Databus platform[20][3] is a registry of metada for datasets on the Web. The metadata describe the different properties of the dataset, its files, ownership, license information, etc.

The Databus stores the meta information as RDF documents, modelled using the DataID ontology[4]. DataID requires publishers of the datasets to specify the location, provenance, license and verification details. DataID also enforces versioning of the datasets. The consumers of the datasets can download the data and perform necessary verification of the publishers and the licenses automatically. There are two options for publishers to prove their identity: i) using WebIDs or ii) by using their Databus account.

The Databus datasets (i.e. data artifacts) have the following versioning identifier system: {GROUP}/{ARTIFACT}/{VERSION}, which is reflected in the following URI of the dataset:

https://databus.dbpedia.org/{PUBLISHER}/{GROUP}/{ARTIFACT}/{VERSION}/{FILE}

This hierarchical structure allows to perform fine-grained versioning and logically group related datasets together.

[18] https://databus.dbpedia.org/dnkg/fusion/dutch-national-kg/2020.10.02

[19] https://github.com/dbpedia/dnkg-pilot

[20] https://github.com/dbpedia/databus

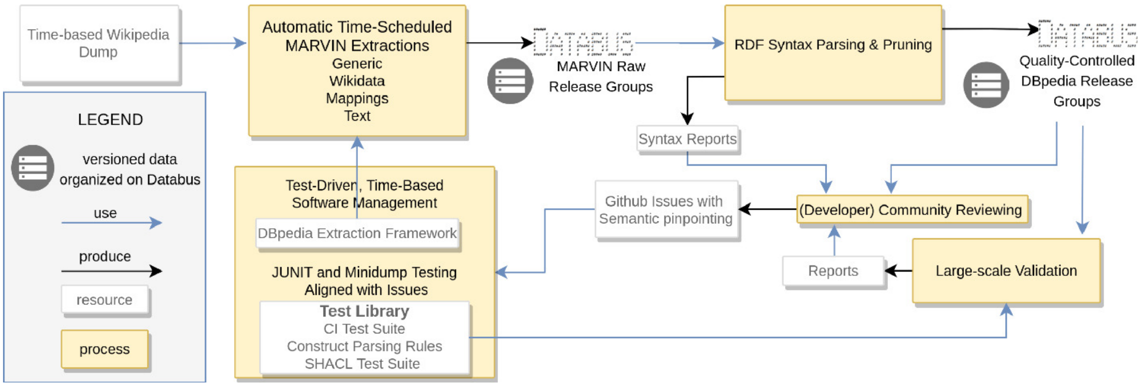

Time-based Wikipedia Dump
Automatic Time-Scheduled MARVIN Extractions
Generic
Wikidata
Mappings
Text
DATABUS
MARVIN Raw Release Groups
RDF Syntax Parsing & Pruning
DATABUS
Quality-Controlled DBpedia Release Groups
LEGEND
versioned data organized on Databus
use
produce
resource
process
Syntax Reports
Test-Driven, Time-Based Software Management
DBpedia Extraction Framework
JUNIT and Minidump Testing Aligned with Issues
Test Library
CI Test Suite
Construct Parsing Rules
SHACL Test Suite
Github Issues with Semantic pinpointing
(Developer) Community Reviewing
Reports
Large-scale Validation

Figure 2: The DBpedia release cycle.

Data consumers may not only download the data or view DataIDs, but also make custom extensions of the meta information (Mods) and compose files from the datasets into collections. To simplify the deployment process, the Databus is also delivered via Docker (see https://github.com/dbpedia/databus), which allows users to deploy local instance of the Databus platform.

The Databus also provides convenient tools for automation of the release cycle, data retrieval and derivation of datasets. Using an HTTP, API users can upload, download and delete DataIDs in the JSON-LD format. As an alternative, users can use the HTML interface to enter all the necessary meta information to generate and publish DataID on the Databus. All published DataIDs on the Databus are also available and queryable via the dedicated SPARQL endpoint (http://databus.dbpedia.org/sparql).

There are additional services deployable with Docker: DBpedia Lookup[21], DBpedia Spotlight[22] and the Virtuoso triple store[23]. DBpedia Lookup is an entity retrieval Web service for RDF data that resolves keywords to DBpedia entity identifiers. DBpedia Spotlight is a service performing named entity extraction from texts. Using the DBpedia Virtuoso docker users can deploy a local Virtuoso triple store preloaded with the data from DBpedia (by specifying a Databus collection).

4.2. DBpedia Release Process

Since its inception in 2009 the release process was facing massive delays (from 12 to 17 months) with increasing development costs and lower productivity due to the sole focus on quality and the increased size and complexity. Nowadays, the release cycle is heavily backed by the Databus platform and was automated to publish regular (i.e., monthly) releases with over 21 billion triples with minimal publishing effort [5].

Figure 2 gives an overview of DBpedia's new release cycle. The new release cycle provides two parallel data releases: a "raw release" containing possible errors released by the MARVIN agent; and a "pruned and cleaned data release" that the DBpedia release agent[24] publishes.

One significant improvement in DBpedia's release cycle is the capability of using a subset of Wikipedia articles to test changes and track issues in the DBpedia's information extraction framework (DIEF). This subset is referred to as *Minidump*. Based on the Minidump a broad range of tests can be applied to validate fixed issues and track improvements in the DIEF software. The test suite covers 7 different test methods. This includes software Unit tests, structural tests with SHACL and comprehensive consumer SPARQL tests.

The overall extraction process can be observed at the release dashboard[25]. The dashboard shows the current stage of an ongoing release, its completeness and offers links to relevant logs or configuration files.

4.3. Databus Mods

The Databus platform allows everyone to link and extend its core DataID metadata with any kind of additional information using the so called Databus Mods feature. Databus Mods are activities analyzing and assessing files published with the Databus DataID that provide additional metadata in the form of fine-grained information such as data summaries, statistics or descriptive metadata enrichments.

These activities create provenance metadata based on PROV-O to link any generated metadata to the persistent

[21] https://github.com/dbpedia/lookup
[22] https://hub.docker.com/r/dbpedia/dbpedia-spotlight
[23] https://github.com/dbpedia/virtuoso-sparql-endpoint-quickstart
[24] https://databus.dbpedia.org/dbpedia
[25] http://release-dashboard.dbpedia.org

Databus file identifiers, independent of its publisher. The generated metadata is provided in a SPARQL endpoint[26] and an HTTP file server, increasing (meta)data discovery and access.

Additionally, we propose the Databus Mods Architecture, which uses a master-worker approach to automate Databus file metadata enrichments. The master service monitors the Databus SPARQL endpoint for updates, distributes scheduled activities to metadata (worker) services, collects the generated metadata, and stores it uniformly. The worker services implement the proposed provenance metadata model and provide an HTTP interface for the master service to invoke a Mod Activity for a specific Databus file. The master can handle multiple workers, allowing scaling the system's throughput.

The Databus Mods Architecture implementation is provided in a public accessible GitHub repository[27], allowing users to deploy Mods reusing existing components. Further, the repository provides a maven library that can be used to create own Mod Workers in JVM (Java Virtual Machine)-like languages or validate the implementation of the so-called Mod API, which is necessary for the Mod Master to control a Mod Worker.

4.4. DBpedia ID Management, PreFusion, and Cartridge Creation

Although the concept of Linked Data offers many benefits, there are still some open challenges concerning interoperability of information [2]. One problem is to handle redundancy in identifiers for same entities and equivalent properties. Another issue is the instability of IRI identifiers due to domain changes or link rot.

In order to tackle these challenges, we developed the DBpedia Global ID Management [6], a central linking hub. DBpedia Global ID materializes the global view of links formed by several linksets and datasets available on the Web of Data. It computes `same-as` clusters by deriving connected components and selects a DBpedia Global ID as a representative for every cluster. These identifiers can be used as uniform identifiers for all of its equivalent identifiers.

The DBpedia Global ID management process involves 5 stages: 1) harvest links and identifiers from input sources, 2) assign stable Singleton IDs for external IDs (bijection), 3) compute connected components based on `owl:sameAs` or `owl:equivalentProperty` properties, 4) assign global IDs based on the cluster member, and finally, 5) release a cluster snapshot dump and update the related microservices. The final deployed microservices translate external IDs to Singleton & Global IDs and allow users to discover known references to other datasets for `same things` or `same properties`. A DBpedia Global IRI is of the form:

https://global.dbpedia.org/id/{Base58}

were the last segment represents the assigned cluster ID encoded with base58.

The Global ID management builds the foundation for DBpedia's FlexiFusion [6] approach, which offers a system based on a PreFusion data structure to integrate and fuse multiple data sources into one knowledge graph. The data structure provides a unified, interoperable, modularized view of a Linked Data source. A PreFusion dataset containing only one source is called a (Knowledge)"cartridge".

The Cartridge compilation process includes ID rewriting and prefusion. In the first stage, input data artifacts with partially transformed triples are selected from the Databus. Additionally, a snapshot version of the Global ID and property assignment is selected as input. In the second stage, the rewritten source datasets are transformed into the cartridge/prefusion format. Therefore, prefused entities are derived by grouping all triples, first by the same subject, and then, by their predicate value.

The prefusion/cartridge serialization is a JSON-LD-based format tracing down the origin of the triples from all input files (Databus file IDs). The serialization resembles a compact representation of values grouped by subject-predicate pair and allows unified access to all sources (rewritten global IDs and properties mapped to property clusters). In addition, the original input statement's provenance is recorded via iHash or as an subject-property-object record.

4.5. DBpedia Archivo

Currently, ontology consumers face many challenges in terms of access (e.g. wrong RDF deployment or missing/incorrect versioning) and quality (e.g. missing documentation/metadata or missing licenses) which impede the easy usage of ontologies.

DBpedia Archivo [7] addresses these issues by providing an augmented ontology archive, making every different version of every ontology persistently available. Archivo also tries to extend its index of distinct ontologies by crawling different sources (e.g. Linked Open Vocabularies[28], prefix.cc, URIs in ontologies, classes used in data on the Databus and suggestions from users[29]) for yet unknown ontologies. Additionally to the ontology itself in RDF (available as RDF-XML, Turtle, and N-Triples) Archivo runs multiple tests, mainly based on SHACL[30]

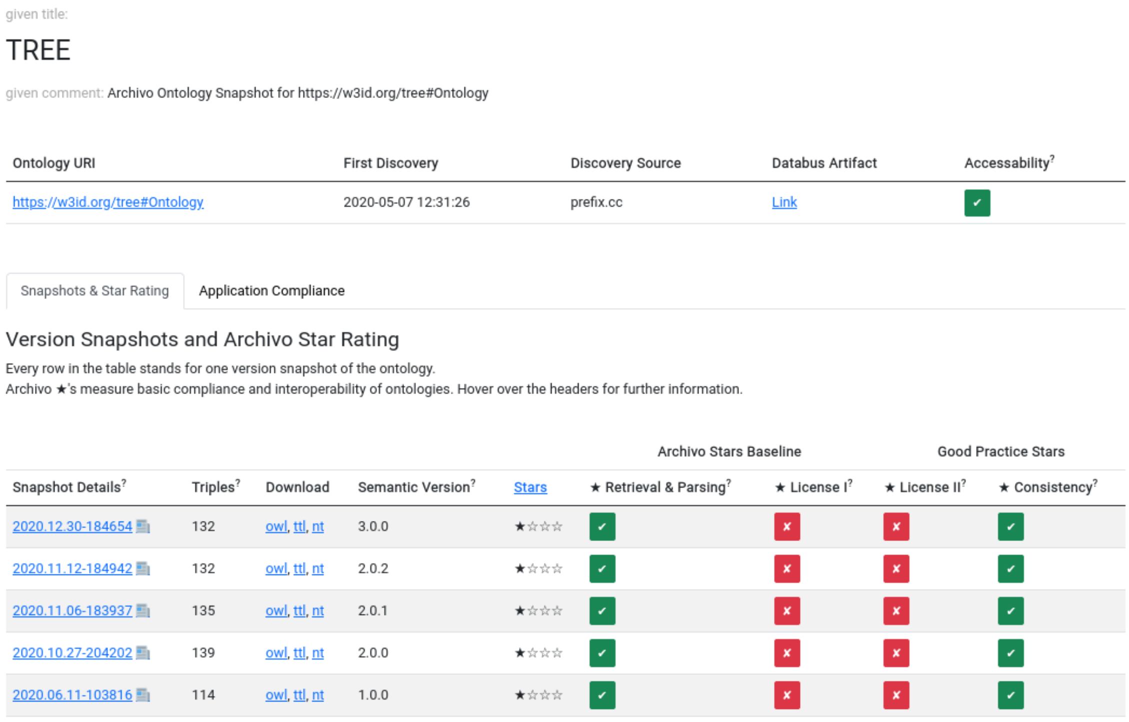
[26] https://mods.tools.dbpedia.org

[27] https://github.com/dbpedia/databus-mods

[28] https://lov.linkeddata.es/dataset/lov/

[29] https://archivo.dbpedia.org/add

[30] https://www.w3.org/TR/shacl/



given title:

TREE

given comment: Archivo Ontology Snapshot for https://w3id.org/tree#Ontology

Ontology URI	First Discovery	Discovery Source	Databus Artifact	Accessability?
https://w3id.org/tree#Ontology	2020-05-07 12:31:26	prefix.cc	Link	✓

Snapshots & Star Rating | Application Compliance

Version Snapshots and Archivo Star Rating

Every row in the table stands for one version snapshot of the ontology.
Archivo ★'s measure basic compliance and interoperability of ontologies. Hover over the headers for further information.

					Archivo Stars Baseline		Good Practice Stars	
Snapshot Details?	Triples?	Download	Semantic Version?	Stars	★ Retrieval & Parsing?	★ License I?	★ License II?	★ Consistency?
2020.12.30-184654	132	owl, ttl, nt	3.0.0	★☆☆☆	✓	✗	✗	✓
2020.11.12-184942	132	owl, ttl, nt	2.0.2	★☆☆☆	✓	✗	✗	✓
2020.11.06-183937	135	owl, ttl, nt	2.0.1	★☆☆☆	✓	✗	✗	✓
2020.10.27-204202	139	owl, ttl, nt	2.0.0	★☆☆☆	✓	✗	✗	✓
2020.06.11-103816	114	owl, ttl, nt	1.0.0	★☆☆☆	✓	✗	✗	✓

Figure 3: An overview page for the TREE ontology.

and available in Archivos' SHACL library[31] checking the fitness for usage. One part of the test is summarized in the Archivo Stars, which make the usability of an ontology visible at a glance:

- ⋆ The first star is earned by making an ontology available as RDF at its representational URI without any parsing errors.
- ⋆ The second star is granted for providing any kind of license statement associated with the URI of the ontology.
- ⋆ The third star is earned by providing said license statements with the property `dct:license` and the license being a URI.
- ⋆ The fourth star is granted for being logically consistent (currently tested with the Pellet reasoner).

In addition to these tests, Archivo runs services to augment the ontologies, for example, LODE[32] and pyLODE[33] for a human-readable HTML documentation. The compliance for such services can also be tested by adding a fitting SHACL shape to Archivo's SHACL library.

The most straightforward way of using Archivo is as an ontology backup or citing tool. For this, Archivo offers an easy way to address every version of the ontologies with a REST API:

http://archivo.dbpedia.org/download?o=
{ontology-URI}&v={version}&f={format}

The version and the format can be omitted, which defaults to the latest version in RDF/XML. Other use case is debugging of an ontology. For this, Archivo offers an overview[34] of the test results and feature plugins (see Figure 3).

Furthermore, Archivo can be used to research and analyse an (arbitrarily) huge set of ontologies. Since it is based on the Databus, all the tools described in previous sections can easily be used on Archivo's data. For example, a certain set of ontologies can be represented by a Databus collection (see Section 3.2) and then directly loaded into a local SPARQL endpoint (see Section 4.1) for analysis. All available access mechanisms are described at the Archivo API page[35].

[31] https://github.com/dbpedia/archivo/tree/master/shacl-library
[32] https://essepuntato.it/lode/
[33] https://github.com/RDFLib/pyLODE
[34] https://archivo.dbpedia.org/info
[35] https://archivo.dbpedia.org/api

5. Conclusions

The main goal of the tutorial was to provide general information about the DBpedia community project and give an overview of the latest advancements in the DBpedia technology stack. The tutorial was organized as an onsite event and it was part of the series of tutorials which have been initiated in 2020. We plan to continue organize events of similar character also in future and continue informing the community about the latest developments around DBpedia. Future events are regularly announced on the DBpedia blog (https://www.dbpedia.org/blog/).

Acknowledgments

This work was partially supported by grants from PLASS project (01MD19003D) and the NexusLinguarum COST Action (CA18209).

References

[1] M. Brümmer, M. Dojchinovski, S. Hellmann, DBpedia abstracts: A large-scale, open, multilingual NLP training corpus, in: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), European Language Resources Association (ELRA), Portorož, Slovenia, 2016, pp. 3339–3343. URL: https://aclanthology.org/L16-1532.

[2] J. Frey, S. Hellmann, FAIR Linked Data - Towards a Linked Data Backbone for Users and Machines, in: Companion of The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021, ACM / IW3C2, 2021, pp. 431–435. URL: https://svn.aksw.org/papers/2021/sci-k_fair-linked-data/public.pdf. doi:10.1145/3442442.3451364.

[3] J. Frey, F. Götz, M. Hofer, Managing and compiling data dependencies for semantic applications using databus client, in: 15th International Conference on Metadata and Semantics Research, Springer, 2021. URL: http://svn.aksw.org/papers/2021/databus-client/public.pdf.

[4] M. Freudenberg, M. Brummer, J. Rucknagel, R. Ulrich, T. Eckart, D. Kontokostas, S. Hellmann, The metadata ecosystem of dataid, in: Special Track on Metadata & Semantics for Open Repositories at 10th International Conference on Metadata and Semantics Research, 2016. URL: https://svn.aksw.org/papers/2016/MSOR_DataID2/public.pdf.

[5] M. Hofer, S. Hellmann, M. Dojchinovski, J. Frey, The new dbpedia release cycle: Increasing agility and efficiency in knowledge extraction workflows 16 (2020). URL: https://svn.aksw.org/papers/2020/semantics_marvin/public.pdf.

[6] J. Frey, M. Hofer, D. Obraczka, J. Lehmann, S. Hellmann, Dbpedia flexifusion the best of wikipedia > wikidata > your data., in: ISWC (2), volume 11779 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 96–112. URL: http://dblp.uni-trier.de/db/conf/semweb/iswc2019-2.html#FreyHO0H19.

[7] J. Frey, D. Streitmatter, F. Götz, S. Hellmann, N. Arndt, Dbpedia archivo - a web-scale interface for ontology archiving under consumer-oriented aspects, in: SEMANTICS 2020, 2020. URL: https://svn.aksw.org/papers/2020/semantics_archivo/public.pdf.