DBpedia Links: The Hub of Links for the Web of Data

Milan Dojchinovski
AKSW/KILT
InfAI, Leipzig University, Germany /
FIT, CTU in Prague
milan.dojchinovski@fit.cvut.cz

Dimitris Kontokostas
AKSW/KILT
InfAI, Leipzig University, Germany
kontokostas@informatik.uni-leipzig.de

Robert Rößling
AKSW/KILT
InfAI, Leipzig University, Germany
roessling@informatik.uni-leipzig.de

Magnus Knuth
Hasso Plattner Institute
University of Potsdam, Germany
magnus.knuth@hpi.uni-potsdam.de

Sebastian Hellmann
AKSW/KILT
InfAI, Leipzig University, Germany
hellmann@informatik.uni-leipzig.de

ABSTRACT

Links are the key enabler for retrieval of related information on the Web of Data. Currently, DBpedia is one of the central interlinking hubs in the Linked Open Data (LOD) cloud. With over 28 million of described and localized things it is one of the largest and open datasets. With the increasing number of linked datasets, there is need for proper maintenance of these links. In this paper, we describe the DBpedia Links repository, which maintains linksets between DBpedia and other LOD datasets. We describe the system for maintenance, update and quality assurance of the linksets.

Keywords

links, DBpedia, LOD, linking, resource

1. INTRODUCTION

Links are the most important Web component which make the World Wide Web and the Web of Data successful stories. By following those links agents can discover further information about resources. Currently, DBpedia acts as central interlinking hub for the emerging Web of Data. It provides localized descriptions for over 28 million things along with links to other data sources on the Web. As the number of data sources interlinked with DBpedia has significantly increased in the recent years, there is need for proper management of these links. In this paper, we describe DBpedia Links, a system which maintains links between DBpedia and other data sources.

We define guidelines for contributions which should be taken into consideration before accepting a linkset. Each individual contribution is going under a fully automated process of validation, re-generation and patching, before it is accepted as part of the DBpedia Links repository and ready for import in the official DBpedia endpoint.

The DBpedia Links resource can support various tasks which are dependent on cross-linked information. For example, *data fusion* tasks can be supported in generation of unified views for entities by incorporating information from different sources. It can also aid agents in navigating from one data source to another, or support *learning tasks*, such as learning entity resolution or link discovery, by providing rich structures and information for learning from multiple sources. Moreover, Web content can be enriched with annotations and references from DBpedia. Since DBpedia Links provides links to other sources, the annotation references can be substituted with links from other sources and in turn provide *multi-dataset enrichment*.

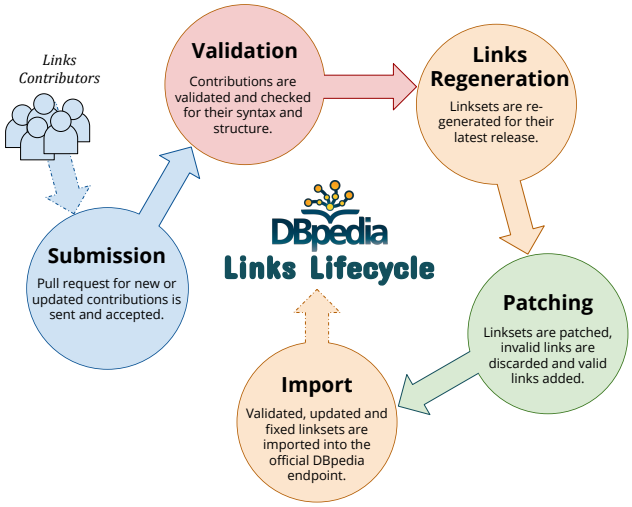

Figure 1: DBpedia Links Lifecycle.

The outline of the paper is as follows. Section 2 describes the related work. Section 3 describes the DBpedia Links management system and its services. It describes the lifecycle of a linkset, including its submission, regeneration, patching and quality assurance procedures. Section 4 provides information on the availability of the resource, its coverage and sustainability plans. Finally, Section 5 concludes the paper and provides future directions.

SEMANTICS 2016: Posters and Demos Track
September 13-14, 2016, Leipzig, Germany

2. RELATED WORK

Currently, there are several data management systems and repositories for managing, publishing and sharing collections of data. Datahub[1] is a data management platform where users can register, create, manage and search for datasets. META-SHARE[2], LingHub[3] and CLARIN[4] are other repositories which provide means for depositing, sharing and searching for language resources. All listed repositories provide sophisticated mechanisms for search, and description of datasets with metadata. Nevertheless, mechanisms for validation, re-generation, and update of RDF datasets are practically non-existent. Platform[5] is a W3C standard which defines a set of rules for read-write Linked Data on the Web. However, it does not define guidelines for management of the data and its re-generation. In this work, we fill-in the missing gaps and propose the DBpedia Links repository for management of links between DBpedia and other datasets. A particular tool that relates to our work is LinkLion[6] which aims at maintaining links, however, regeneration, patching and validation of the contributions are not considered.

3. THE DBPEDIA LINKS CONTRIBUTION SYSTEM AND ITS SERVICES

The DBpedia Links system defines and implements the entire links lifecycle as illustrated in Figure 1. The process starts with an initial submission of a links contribution. This step is initiated via a pull request to our git repository[7]. Upon acceptance of a pull request, a validation service in place verifies and validates the contribution (cf. subsection 3.3). Several quality checks are executed to assure the contribution meets the requirements. Next, the linkset is re–generated to its latest release. Follows a patching step which updates particular links from the linkset; invalid links are removed and new are added (cf. subsection 3.5). Finally, the validated, updated and fixed linkset is ready for import into the official DBpedia endpoint[8].

Bellow, we provide in–detailed description of the linkset contribution system, the contribution modalities, contribution structure requirements, patching mechanism, quality assurance policy and its supporting services.

3.1 Contribution Modalities

There are five different ways to contribute links via pull requests. Each contribution should provide the linkset in at least one of the following methods. The first and the most trivial approach is where the contributor provides (1) a download link to a dump with the links, or (2) submits the links file along with the contribution. If the linkset is supplied via a download link, then the link should be provided in the metadata file. Contributions can also provide (3) a script which can be used to re-generate the linkset. Usually, a script will download and unpack the files, but in some cases a script could implement a more advanced logic for generation of the links. Typically, this is a bash script. Links can be also contributed via (4) a SPARQL endpoint from where they can be retrieved. In this scenario, the user should provide a SPARQL endpoint and a CONSTRUCT query within the metadata file. In some cases, the linksets need to be updated with new links, or invalid or old links should be removed. Such updates can be submitted as (5) a patch, which provides all needed update information.

[1]https://datahub.io/, *visited on 11/08/2016.*
[2]http://www.meta-share.org/, *visited on 11/08/2016.*
[3]http://linghub.lider-project.eu/, *visited on 11/08/2016.*
[4]https://www.clarin.eu/, *visited on 11/08/2016.*
[5]https://www.w3.org/TR/2015/REC-ldp-20150226/, *visited on 11/08/2016.*
[6]http://www.linklion.org/, visited on 18/8/2016.
[7]https://github.com/dbpedia/links
[8]http://dbpedia.org/sparql/

3.2 Contribution Structure

Linkset contributions are packaged according to the following structure.

1. `metadata.ttl` – a metadata linkset file.
2. `README.md` – a brief description of the linkset.
3. `scripts/` – (optional) a folder containing a script that produces a linkset.
4. `name_links.nt` or `name_links.nt.bz2` – (optional) a file containing the links.
5. `patches/` – (optional) a folder containing patches including whitelists and/or blacklists of links.

Each contribution is a folder with a structure as described above. The contribution folder should be placed under the `dbpedia.org` folder, for links from the main DBpedia namespace. Or, in the `xxx.dbpedia.org` folder, for links from a subdomain of DBpedia.

The `metadata.ttl` file, which is an imminent part of each contribution, provides machine-readable description of the linkset contribution. Listing 1 shows an example of a metadata file.

Listing 1: Contribution metadata example.

```
<#1> a void:Linkset ;
 dbp:script <scripts/makeLinks.sh> ;
 dbp:endpoint "http://example.org/sparql" ;
 dbp:constructquery "CONSTRUCT {?b <http://www.w3.
     org/2002/07/owl#sameAs> ?o} where { ?o <http
     ://www.w3.org/2002/07/owl#sameAs> ?b. FILTER
     (REGEX(STR(?b),'http://dbpedia.org')) }" ;
 dbp:ntriplefilelocation <links.nt> ;
 dbp:approvedPatch <patches/pat_1/patch.ttl> ;
 dbp:approvedPatch <patches/pat_2/patch.ttl> ;
 dbp:optionalPatch <patches/opt/unofficial.ttl> ;
 dbp:outputFile <out.nt> ;
 dbp:updateFrequenceInDays "10" ;
 dct:author ex:dave-horn ;
 dct:description "These links link DBpedia with
     the RichData dataset." ;
 dct:license <http://creativecommons.org/
     publicdomain/zero/1.0/> .
```

Within the metadata file a user can specify the location of the links file (`dbp:ntriplefilelocation`). The location can be specified as a relative path to the file (if provided along with the contribution), or a file download link. The links should be provided in the N-triples format (one triple per line). The subject of each triple should be a DBpedia URI. In case of large linksets, linksets containing over 200K triples or over 20MB, the file should be compressed in the .bzip2 format. If a script for re-generation of the links is provided, then the user should specify the location of this script using the `dbp:script` property. All the scripts should be placed in the `scripts/` folder.

If a user provides the links via a SPARQL endpoint, it is necessary to specify a CONSTRUCT query (`dbp:constructquery`) and the SPARQL endpoint location (`dbp:endpoint`). Pagination using the `LIMIT` and `OFFSET` modifiers are handled by the system and they are not expected to be part of the CONSTRUCT query.

3.3 Validation, Quality & Compliance Assurance

In order to keep the quality of the links high, we apply a strict quality & compliance assurance policy. There are different levels of conformance checks to our contribution requirements.

Folder and File Structure Checking. We verify that every contribution has a correct file structure as defined in subsection 3.2. In particular, we verify an existence of a *metadata.ttl* and a *Readme.md* file descriptions for every linkset and patch.

Syntax Error Checking. All RDF files in the repository are parsed using the Raptor RDF syntax parsing and serializing utility[9] and checked for syntax errors.

DBpedia IRI as subject. For every contributed links file and patch, we check if the subject of the triple is a DBpedia IRI. This is required for link uniformity.

Metadata and Patch Structure. Within RDFUnit [3] we define constraints which validate whether the metadata and patch files are following the conventions as described in subsection 3.2. The constraints are defined using the Shapes Constraint Language (SHACL)[10] and they ensure that these files can be read without problems from the DBpedia links toolstack.

File existence. We verify that all the IRIs (local or remote) referenced from the *metadata.ttl* file exist.

Link Quality. Since checking the quality of the links is a demanding task, the contributors are urged to provide valid proof which confirms the link quality. The proof can be in form of an evaluation documented in a scientific paper, a website, or other type of report which provides valid evaluation of the quality of the links. The proof is then checked by the maintainers of the DBpedia Links.

Validation Service. The validation service implements quality checks and validates each contribution. The validation service is integrated with the Travis CI continuous integration service[11]. This assures that each pull request for linkset contribution is verified by an automated build, allowing the DBpedia Links community to detect problems early. It is executed upon every commit and every pull request.

3.4 Links Regeneration

In order to ensure up-to-date linksets, a links regeneration system via nightly builds has been implemented. However, since the links regeneration for some linksets can be computationally intensive, we implement a "conditional" links regeneration. The links regeneration process implements the following control mechanisms.

Owner–specified Frequency. A user can specify how frequently the links change and they need to be updated. The update frequency interval is explicitly set within the `metadata.ttl` file using the `dbp:updateFrequencyInDays` property. If no frequency is set, then the default value is set to 10 days. In other words, the linkset will be re-generated, if needed, in every 10th nightly build. If the linkset is not subject for change, the user can set the update frequency to 0 and the links will not be further re-generated.

Conditional Links Regeneration. Linksets which are provided as dumps are fetched only if they have been modified since their last retrieval. Using the `Last-Modified` response HTTP header the time of its last modification is checked and compared with the latest retrieval. Conditional requests according to the HTTP/1.1 specification[12] are executed to check for eventual changes of the linkset and fetch its latest release. If the links file has not been modified since the last generation, then it will not be re-generated.

Upon first commit. If a linkset is provided via a links regeneration script, a SPARQL endpoint, or a dump via a download link, then this linkset is generated upon the first commit. For every following nightly build the regeneration is performed according to the control mechanisms introduced above.

Listing 2: Patch description example.

```
<patch.ttl> a dbp:Patch ;
  dbp:whitelistFile <xx.wl.ttl> ;
  dbp:blacklistFile <xx.bl.ttl> ;
  dct:author <http://example.org/dave-horn> ;
  dct:description "Patch for the RichData." ;
  dct:license <http://creativecommons.org/
       publicdomain/zero/1.0/> .
```

3.5 Patching Linkset

Linksets are typically created by some heuristic, statistical, or machine-learning algorithms, which typically show a certain error rate in the sense of imperfect accuracy and completeness. In order to retroactively correct such errors, updates of linksets can be applied. An update can be removal of invalid triples or introduction of new triples. Users can contribute such updates by providing a so-called "patch" within the `patches/` folder. Each patch should be described with a simple metadata description file and provide a whitelist and a blacklist dump file in the N-triples format. Whitelists should contain triples which should be added to the linkset, while blacklists contain invalid triples which should be removed from the linkset. Listing 2 provides an example of a patch description file.

The patching service provides automatic generation of patching instructions for the linksets. The patching service takes as input a patch description (cf. Listing 2), and generates all necessary triple update instructions (cf. Listing 3). These update instructions are modelled using the Patch Request Ontology[13] [2]. We are aware about the Linked Data Patch format[14] and as soon as it becomes a W3C standard we will provide support for it.

Listing 3: Detailed patch example.

```
repo:MyPatch a pro:Patch ;
  pro:update [
    a guo:UpdateInstruction ;
    guo:target_subject dbpedia:Achille_Starace ;
    guo:delete [
      owl:sameAs <http://dati.camera.it/ocd/
          persona.rdf/p9970> ]
```

[9] http://librdf.org/raptor/

[10] https://www.w3.org/TR/shacl/, visited on 11/08/2016.

[11] https://travis-ci.org/dbpedia/links/builds

[12] https://tools.ietf.org/rfc/rfc7232.txt

[13] http://purl.org/hpi/patchr, visited on 11/08/2016.

[14] https://www.w3.org/TR/2015/NOTE-ldpatch-20150728/, visited on 11/08/2016.

Table 1: Linksets from dbpedia.org with `owl:sameAs` links.

Linkset	Triples
viaf.org	173 942
gadm.geovocab.org	38 793
www.geonames.org	86 547
lobid.org/manifestation	13 926
eunis.eea.europa.eu	11 235
www4.wiwiss.fu-berlin.de/bookmashup	8 903
lobid.org/organisation	7 012
www.bbc.co.uk:things	4 557
www4.wiwiss.fu-berlin.de/diseasome	2 301
dati.camera.it	1 533
rdfdata.eionet.europa.eu	307
worldbank.270a.info	214
transparency.270a.info	183
learning-provider.data.ac.uk	174

```
7     ] ;
8     pro:update [
9       a guo:UpdateInstruction ;
10      guo:target_subject dbpedia:Achille_Starace ;
11      guo:insert [
12        owl:sameAs <http://dati.camera.it/ocd/
              persona.rdf/pr11089> ]
13    ] ;
14    pro:appliesTo <http://example.org/void.ttl> ;
15    prov:wasGeneratedBy [
16      a prv:DataCreation ;
17      prv:involvedActor repo:Author ;
18      prv:performedAt "2016-04-20T21:32:52"^^xsd:
            dateTime  ] .
```

4. SUSTAINABILITY, AVAILABILITY AND COVERAGE

Sustainability. Each linkset contribution upon successful validation becomes part of the official DBpedia knowledge infrastructure. The contributed linksets will be distributed and announced along with the official DBpedia releases and localized DBpedia chapters. DBpedia is one of the most mature LOD datasets and the central hub within the LOD cloud [1]. Publishing linksets as part of DBpedia guarantees *long–term availability* through the DBpedia infrastructure, which assures high up-time and visibility for the linksets.

Availability. The links are collected and maintained online in a github repository. The repository also provides access to all the code and scripts which are used for validation, regeneration, update and summarization of the data.

In order to allow widest possible dissemination, all data and code (scripts) related to this effort is to be treated as Public Domain or CC0 Public Domain Dedication.

Dataset Coverage. Currently, the DBpedia Links repository maintains 14 linksets from 12 different resources. Table 1 summarizes linksets which provide links from the dbpedia.org domain and Table 2 lists the linksets with links from the DBpedia subdomains. There are 7 linksets from 5 different sources, providing over 27K RDF triples. The DBpedia Links repository maintains links between DBpedia and other datasets. In future, we would like to provide maintenance also for links between other datasets, however, we would like first to validate our approach on DBpedia links, collect and implement feedback from the community, and then expand to links between other datasets.

5. CONCLUSIONS AND FUTURE WORK

Links make the Web of Data a successful story. With the ever increasing popularity of the Web of Data, the number of links connecting different sources has significantly increased in the recent years. Along with these developments, DBpedia became the central LOD interlinking hub, which requires proper management of these links. In this paper, we described the DBpedia Links system, which defines and implements guidelines for management of these links. The systems is supported with several services for validation, regeneration, update and quality assurance of linksets. The entire links lifecycle is an automated process, which significantly reduces the costs for maintaining the links and it assures that the links are of high quality, they are up-to-date and properly described with metadata.

In our future work, we plan to provide support for link discovery frameworks, such as SILK [5] or LIMES [4]. Last but not least, we want to collect, describe, integrate and maintain additional links and scripts for generation of links to other datasets, such as Freebase, YAGO and GeoNames.

Table 2: Linksets from xxx.dbpedia.org with `owl:sameAs` links.

Linkset	Triples
lobid.org/manifestation	14 057
dati.camera.it	4 798
lobid.org/organisation	3 197
geonames.jp	2 332
test.rce.rnatoolset.net	1 111
wolterskluwer.de/courts	997
wolterskluwer.de/arbeitsrecht	777

Acknowledgements

This work was funded by grants from the EU's H2020 Programme for projects ALIGNED (GA 644055), FREME (GA 644771), and by grants from the Federal Ministry for Economic Affairs and Energy of Germany (BMWi) for the SmartDataWeb project (GA 01MD15010B).

6. REFERENCES

[1] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3), 2009.

[2] M. Knuth and H. Sack. Patchr: A framework for linked data change requests. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 11(1):30–45, 0 2015.

[3] D. Kontokostas, P. Westphal, S. Auer, S. Hellmann, J. Lehmann, R. Cornelissen, and A. Zaveri. Test-driven evaluation of linked data quality. In *Proceedings of the 23rd international conference on World Wide Web*, pages 747–758. ACM, 2014.

[4] A.-C. N. Ngomo and S. Auer. Limes-a time-efficient approach for large-scale link discovery on the web of data. *integration*, 15:3, 2011.

[5] J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Silk-a link discovery framework for the web of data. *LDOW*, 538, 2009.