

LION'S DEN

Feeding the LINKLION

Mohamed Ahmed Sherif, Mofeed M. Hassan, Tommaso Soru, Axel-Cyrille Ngonga Ngomo, and Jens Lehmann

Department of Computer Science, University of Leipzig, 04109 Leipzig, Germany
{sherif,mounir,tsoru,ngonga,lehmann}@informatik.uni-leipzig.de

Introduction Over the last years, several tools have been developed with the aim of efficiently supporting the link discovery process [5,7]. This process consisting of two steps: (1) Discovering a Link Specifications (LS) for retrieving high-quality links (i.e. achieve high precision and recall). (2) Carry out the LS to compute the actual links. Several frameworks such as LIMES [3] and SILK [1] have been developed to create such links between the different knowledge bases (KB). While the importance of links between datasets is unequivocal, only few efforts have aimed at making LS available. Such a link repository would however enable a large number of applications, including transfer learning for LS, the provision of provenance and justification information for links, fuzzy inferences on Linked data sets and many more. The importance of links is further underlined by the community efforts have already led to the creation of link repositories such as LINKLION and sameAs.org. In view of the dispersed availability of LS in different formats (scripts, XML, RDF), we created LION'S DEN as a companion project to LINKLION. LINKLION is a store for the publication, retrieval and use of links between KB. The portal provides functionality for the upload and the storage of discovered links, as well as meta-information about these links. With LION'S DEN, we introduce an extension of such meta-information by letting the portal user upload files describing LS. We published the LION'S DEN dataset on the LINKLION link discovery portal so as to make them accessible and queryable via a SPARQL endpoint.¹

The LION'S DEN Dataset The dataset is now hosted within the LINKLION project at <http://linklion.org>. Currently, LION'S DEN contains 436 LS that are described by 15 457 triples including the ontology. Metadata on the LION'S DEN dataset is available on *DataHub*.²

Ontology To represent the LS in RDF and OWL, we developed the *Lion's Den* vocabulary dubbed LDEN³. LDEN was specified with the aim of supporting any type of LS regardless of the way it was created. In its current version, LDEN contains a set of ten classes. Each LS is an instance of the *LinkSpecs* class. The *LinkSpecs* class provides properties that allow referencing the five basic components of any LS which are the *source* and *target* datasets, the *metric* used for linking as well as the *acceptance*

¹ for more details see the extended paper in the project web site https://svn.aksw.org/papers/2016/ISWC_0M_LionDen/public.pdf

² <http://datahub.io/dataset/lionsden>

³ <http://www.linklion.org/lden/>

and *reviewing* criteria. In addition, the `LinkSpecs` class provides metadata such as the source LS's URL and creator, publisher, license and provenance information. Currently, our ontology contains three classes derived from the `LinkSpecs` class (`LimesSpecs`, `SilkSpecs` and `ScriptSpecs`), where each of the three classes contains special attributes related to the framework it represents.

Data Sources LION'S DEN original LS were collected from four different sources: (1) *The LATC project* provides the interlinking *24/7 Platform*⁴. (2) *LinkedGeoData*⁵ is a project to convert spatial information provided by *OpenStreetMap* to the Web of Data. (3) *DBpedia-links*⁶ is a repository that contains links, LS and link extraction scripts. (4) The LIMES⁷ Link discovery framework supports manual configuration for linking tasks through XML based specification files.

Conversion Process As the original configuration files for both SILK and LIMES were in XML format, we built a specialized XML to RDF converter for each of them. The source code of the dataset converters is available at the project repository⁸.

Provenance The LINKLION dataset reuses properties and classes from the PROV W3C recommendation⁹ to keep track of data provenance.

Use Cases Having the LS of LION'S DEN together with the links of LINKLION in a machine readable format and serving them from one portal offers a lot of opportunities, including, but not limited to: benchmarking link discovery algorithms, automatic linked data enrichment [6], key discovery [8], unification of LS, LS transfer learning [2] and Link Discovery over n Knowledge Bases [4].

References

1. R. Isele, A. Jentzsch, and C. Bizer. Efficient Multidimensional Blocking for Link Discovery without losing Recall. In *WebDB*, 2011.
2. A.-C. N. Ngomo, J. Lehmann, and M. Hassan. Transfer learning of link specifications. In *Seventh IEEE International Conference on Semantic Computing (ICSC)*, 2013.
3. A. N. Ngomo. A time-efficient hybrid approach to link discovery. In *Proceedings of the 6th International Workshop on Ontology Matching, Bonn, Germany, October 24, 2011*, 2011.
4. A.-C. Ngonga Ngomo, M. A. Sherif, and K. Lyko. Unsupervised link discovery through knowledge base repair. In *Extended Semantic Web Conference (ESWC 2014)*, 2014.
5. G. Papadakis, E. Ioannou, C. Niederèe, T. Palpanasz, and W. Nejdl. Eliminating the redundancy in blocking-based entity resolution methods. In *JCDL*, 2011.
6. M. Sherif, A.-C. Ngonga Ngomo, and J. Lehmann. Automating RDF dataset transformation and enrichment. In *12th Extended Semantic Web Conference, Portoroz, Slovenia, 31st May - 4th June 2015*. Springer, 2015.
7. J. Sleeman and T. Finin. Computing foaf co-reference relations with rules and machine learning. In *Proceedings of the Third International Workshop on Social Data on the Web*, 2010.
8. T. Soru, E. Marx, and A.-C. Ngonga Ngomo. ROCKER – a refinement operator for key discovery. In *Proceedings of the 24th International Conference on World Wide Web*, 2015.

⁴ https://www.assembla.com/wiki/show/silk/Link_Specification_Language

⁵ <http://linkedgeodata.org/>

⁶ <https://github.com/dbpedia/dbpedia-links/>

⁷ <https://github.com/AKSW/LIMES>

⁸ <https://github.com/AKSW/LionDen>

⁹ <http://www.w3.org/ns/prov#>