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

Abstract. Link discovery is one of the key tasks towards implementing the vision of the Web of Data. Over the last years, large numbers of link specifications (LS) have thus been created. Having a central repository of such specifications has significant advantages, such as empowering novel axes of research such as transfer learning and explicating why certain links between datasets exist. We thus present LION's DEN, an RDF repository of LS. LION's DEN is intended to be an open community-driven dataset that allows data publishers to also publish their interlinking schemes. As such, it is designed to be integrated into LINKLION, a portal for links of the Web of Data. In this paper, we present the design approach that led to the ontology underlying the dataset. In addition, we present the currently 436-specification strong dataset as well as use cases for the dataset.

1 Introduction

One of the main benefits of Linked Data is that it allows providing and managing integrated data sources [1]. Linked Data sets have thus gained significant momentum over the last years and are used within applications in question answering [13], federated querying [9], automatic linked data enrichment [10] and large-scale inferences [14]. Over the last years, several tools and libraries have been developed with the main aim of efficiently supporting the whole of the link discovery process [2,8,11]. In general, this process can be modeled as consisting of two steps: Once provided with a source and target set of instances, the first step consists of discovering a link specification for retrieving high-quality links. This step is of crucial importance as the precision and recall of the link discovery process obviously depend on the tue quality of the link specification used. Once a specification has been decided upon, it has to be carried out to compute the actual links. Several frameworks such as LIMES [6] and SILK [3] have been developed and employed successfully to create links between the different knowledge bases on the Linked Data Web.

While the importance of links between datasets is unequivocal, only few efforts that 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 [4] and `sameAs.org` (which focuses on owl:sameAs links). However, despite some small-scale efforts, there is no repository which stores LS. Several projects can however be regarded as sources for LS. One of the few projects in this direction was carried out in the context of DBpedia[1]. In compliance to the Linked Data principles, many LS were created to generate links between DBpedia and other datasets and made available at the project page. Another effort was the LATC project[2] – an EU-project that aimed to ease the publishing of LS and the correspondent generated links for data owners. Based on the given LS, its 24/7 interlinking platform computes, updates and assesses links between datasets. Repositories of link discovery tools are also sources of LS although those are scattered across different repositories or private computers. For each tool, the need to test its performance requires the existence of a set of LS. These sets of LS are mostly available online along with their related interlinking tool.

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 knowledge bases [4]. 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.

2 Ontology

To represent the LS in RDF and OWL, we developed the *Lion's Den* vocabulary dubbed LDEN[3]. LDEN was specified with the aim of supporting any type of LS regardless of the way it was created. To this end, in its current version, LDEN (as shown in Figure 1) 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* and *reviewing* criteria. The SourceDataset and The TargeteDataset classes contain the properties of the source and target datasets. The Metric class stores the metric expression used for linking. The Acceptance and Review classes provide attributes for the relation generated by the LS such as owl:sameAs and the acceptance resp. review thresholds. 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 classed contains special attributes related to the framework it represents. For example, the LimesSpecs class contains (in addition to the inherited attributes form its base class LinkSpecs) some LIMES-related attributes such as lden:executionPlan and lden:granularity. For keeping track of SILK-based LS, we use the SilkSpecs class and for free scripts

[1] http://dbpedia.org/

[2] http://latc-project.eu/

[3] http://www.linklion.org/lden/

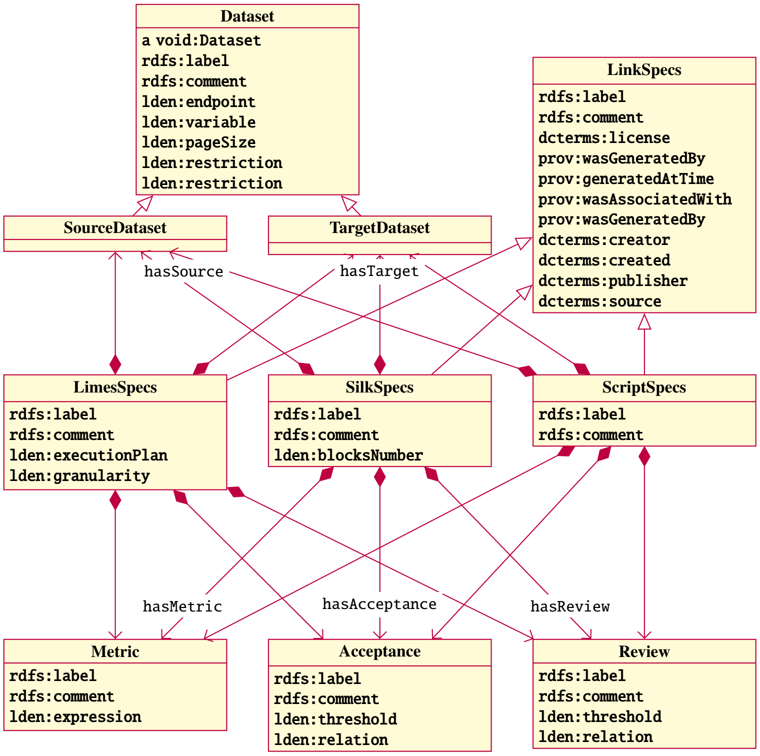


Fig. 1. Ontology of Lion's Den

we use the `ScriptSpecs` class. More LS generated from other frameworks can be embedded in our dataset by simply deriving new classes from the `LinkSpecs` class.

3 Dataset Generation

Our extraction process was governed by the set of LS at our disposal. In the following, we thus first present the datasets we converted according to the ontology presented above. Then we present the extraction process itself as well as its results.

3.1 Data Sources

Lion's Den original LS were collected from four different sources (see Figure 2 (b)):

1. *The LATC project* provides the interlinking *24/7 Platform*. This was a cloud based platform to generate RDF links between datasets in the Linked Open Data cloud. The platform contains a total of 176 LS mostly in Silk-LSL[4] format.

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

2. *LinkedGeoData*[5] is a project to convert spatial information provided by *OpenStreetMap* to the Web of Data. *LinkedGeoData* is linked to *DBpedia* using SILK based on a set of manually created LSs. In our dataset, we convert a set of 46 LSs between different classes of *LinkedGeoData* and *DBpedia*.
3. *DBpedia-links*[6] is a repository that contains links, LS and link extraction scripts. Lion's Den includes 43 LSs from *DBpedia-links* in from of SILK-LSL.
4. The Limes [6] Link discovery framework supports manual configuration for linking tasks through XML based specification files. Lion's Den includes 167 LSs from its release examples[7].

3.2 Conversion Process

First, we collected the original LSs from the aforementioned sources. 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 converters worked as follows: for each LS file, a unique identifier URI was generated.Using a Java DOM[8] parser, the conversion process started by reading each XML LS file into a DOM model.[9] By iterating over the generated DOM model, the converters were able to extract necessary data to include into the Lion's Den ontology representation of the specifications. For each piece of the extracted data, the converters was able to generate one triple as: the unique specification URI as *subject*, the extracted piece data as *object* and the respective `lden` property as *predicate*. Next, the *(subject, predicate, object)* triples are accumulated into one *Jena*[10] model out of each XML LS files. Afterwards, all the generated Jena models are accumulated together into one dataset. The source code of the dataset converters is available at the project repository[11]. The technical details of the Lion's Den dataset can be seen in Table 1.

4 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*.[12] Table 1 summarizes the metadata for the Lion's Den.

[5] `http://linkedgeodata.org/`

[6] `https://github.com/dbpedia/dbpedia-links/`

[7] `https://github.com/AKSW/LIMES`

[8] `http://www.w3.org/DOM/`

[9] Even with the known slow performance of the DOM parser, we were able to achieve high conversion speeds due to the small sizes of LS files.

[10] `https://jena.apache.org`

[11] `https://github.com/AKSW/LionDen`

[12] `http://datahub.io/dataset/lionsden`

Table 1. Technical details of the Lion's Den dataset.

Name	Lion's Den
Example Resource	http://www.linklion.org/lden/dailymed-drugbank_spec
Ontology	https://github.com/AKSW/LionDen/blob/master/LionDenDatahub/lden.owl.ttl
Dataset dump	https://github.com/AKSW/LionDen/blob/master/LionDenDatahub/dump.ttl
Sparql Endpoint	http://www.linklion.org:8890/sparql
Dataset graph	http://www.linklion.org/lden
Namespace	http://www.linklion.org/lden/
Prefix	lden:
Ver. Date	2015-04-30
Ver. No	1.0
License	Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0)
DataHub	http://datahub.io/dataset/lionsden

Sustainability Strategy: We manage updates through the LinkLion portal[13] interface where users are able to upload LS in RDF format. Here, the portal will check the uploaded specification for completeness. Should the LS files be in another format, the interface allows to create a new LS by hand through the LinkLion's portal user interface. In case of any missing information, the user who uploads the LS will be contacted for supplying the missing data. Then, the portal converts the LS to the LDEN ontology and the resulting triples will be added to the dataset. In case of new LS format found, there is no need to alter lden, only a new converters will be deployed to convert the new LS file format to the lden. As LinkLion is a backbone of many of our funded projects, we will be continuously monitoring the performance of the portal in terms of data quality, response time, and uptime rate. Being hosted at the Universität Leipzig Rechenzentrum[14] computing center, we can rely on long-time storage facilities and network monitoring.

Interlinking and Provenance: The LinkLion dataset reuses properties and classes from the PROV W3C recommendation[15] to keep track of data provenance. In particular, prov:wasGeneratedBy and prov:generatedAtTime are present for each mapping (i.e., run of a link discovery algorithm on a given pair of datasets). The former links a mapping to its algorithm, whereas the latter links the mapping to a date-time value. In turn, an algorithm is linked with its respective framework (incl. version) within a prov:wasAssociatedWith property. In addition, each dataset is an instance of the class void:Dataset and linked back to the original repositories or datasets through a void:uriSpace relation. Generated links are stored using the abstraction pattern, where the link itself is an instance of class llont:Link. Each link is connected to the source, link property, and target using the rdf:subject, rdf:predicate, and rdf:object properties, respectively. Being built upon LinkLion, Lion's Den keeps track of each original dataset. In addition to keeping the original LS source URI, the repository creator and publisher are stored using the dcterms vocabulary.

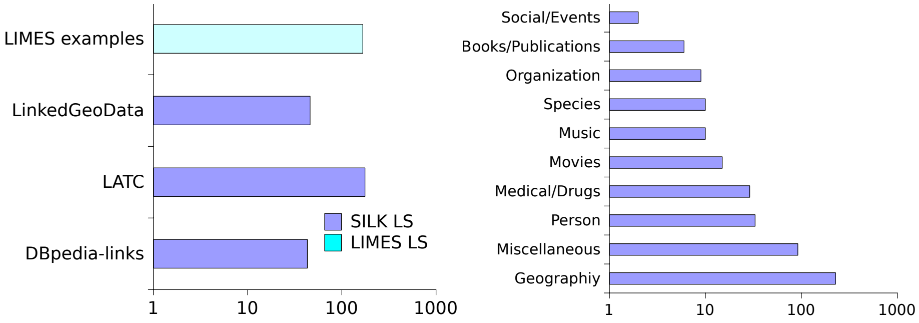
Statistics: Currently, The datasets cover several domains as Figure 2 (b) illustrates. Based on the interlinking information from the LS' sources, they generate more than one million links.

[13] http://www.linklion.org

[14] https://www.urz.uni-leipzig.de/

[15] http://www.w3.org/ns/prov#

Fig. 2. Lion's Den sources and statistics, where the *x*-axis represents the number of triples in log scale.



(a) Data Sources

(b) Statistics

Community Lion's Den is the consolidation of community-driven work towards easing link discovery. The Linked Data community continuously feeds new links to the dataset portal[16]. Moreover, Lion's Den provides a mailing list[17] which is not only for announcing new releases but also for collecting community feedback concerning the dataset. In addition to the mailing list we provide an issue tracker[18] for collecting issues concerning dataset conversion.

5 Relevance of the Dataset

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:

Gold Standard Creation and New Link Discovery Algorithms Evaluation: LinkLion stores the links generated by various algorithms in a concise dataset where each generated link is associated with its generator algorithm/framework/version. Feeding linking evaluation frameworks (for example EvaLink[19]) with existing links enable the creation of new linking gold standards. Then, the performance parameters (i.e. precision, recall, F-score and runtime) of stored LS of Lion's Den can be computed and added to the system. Even more, new link discovery algorithms can evaluate their results against the generated gold standard datasets.

LS analysis: Through Lion's Den, many link specification statistics can be generated including but not limited to: A sorted list from the most or least interlinked dataset (see next Listing), the most/least commonly used similarity functions, average complexity similarity expression per dataset.

[16] http://www.linklion.org/portal/

[17] https://groups.google.com/d/forum/lions_den

[18] https://github.com/AKSW/LionDen/issues

[19] https://github.com/AKSW/Evalink

```
  select ?se count(?se) as ?c where {
 ?ls lden:hasSource ?s .
 ?ls lden:hasTarget ?t .
 ?s  lden:endPoint ?se .
 ?t  lden:endPoint ?te .
  } order by DESC(?c)
```

Listing 1.1. SARQL query

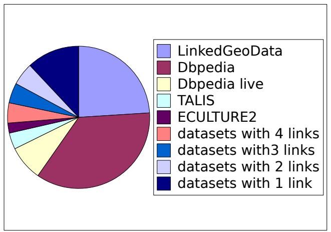


Fig. 3. Pie chart of the query result.

Fig. 4. SPARQL query to retrieve a sorted list from the most or least interlinked dataset from Lion's Den. and a pie chart of the results

Key Discovery: A *key* is a set of properties which can distinguish all instances that belong to a class [12]. Extracting the common set of properties used for linking a specific class instance can serve as a base in the process of discovering keys.

Unification of Link Specifications: As Lion's Den will continue to collect specifications generated by various tools and scripts, the similarities and differences between those will become more evident and easier to analyse. This may in turn lead to input in unification and standardisation efforts for LS as in other related areas, such as the W3C R2RML[20] standard for mapping of relational databases to RDF.

Link Specifications Transfer Learning: Transfer Learning is the process of applying the knowledge learned from previous experiences to a new problem. The objective here is either to solve the new problem or to improve an existing solution. For *Link Discovery*, a formalization framework of *Transfer Learning* was proposed in [5]. One of the most basic requirements for applying this form of machine learning is however the availability of a large number of specifications (previous experiences) from which new specifications (new problem) can be derived. Lion's Den would help addressing exactly this problem and support the development of a new subfield of Link Discovery.

Link Discovery over n Knowledge Bases: Another area of research that can largely profit from Lion's Den is link discovery of $n > 2$ knowledge basses. Current Link Discovery frameworks only provide specifications for two knowledge bases at a time. However, previous works, e.g. [7], have shown that by combining link discovery over several knowledge bases, the quality of the links can be improved. We will thus use Lion's Den to feed novel algorithms such as the aforementioned to achieve better precision and recall.

6 Conclusion and Future Work

We presented the Lion's Den dataset, which provides the community with a repository of LS and scripts for linking knowledge bases on the Web of Data. Through this repository, we enable a better overview and analysis of how the Web of Data is connected, which can spur further insights and opens up the research field of transfer learning

[20] http://www.w3.org/TR/r2rml/

of LS. We provide an ontology, machine-readable metadata and statistics as well as a long-term sustainability strategy. In future work, we will develop crawling mechanisms to automate the automatic gathering of specifications from the Web of Documents and the Web of Data. These specifications will be converted, deduplicated and added to Lion's Den automatically. Also, we will keep track of which of the LS are still working, this piece of data will be added using `lden:lastRunDate`. Moreover, we will use the specifications to fuel novel link discovery approaches such as link discovery over more than two input knowledge bases.

References

1. S. Auer, J. Lehmann, A.-C. N. Ngomo, and A. Zaveri. Introduction to linked data and its lifecycle on the web. In *Reasoning Web*, pages 1–90, 2013.
2. A. Hogan, A. Polleres, J. Umbrich, and A. Zimmermann. Some entities are more equal than others: statistical methods to consolidate linked data. In *Workshop on New Forms of Reasoning for the Semantic Web: Scalable & Dynamic (NeFoRS2010)*, 2010.
3. R. Isele, A. Jentzsch, and C. Bizer. Efficient Multidimensional Blocking for Link Discovery without losing Recall. In *WebDB*, 2011.
4. M. Nentwig, T. Soru, A.-C. Ngonga Ngomo, and E. Rahm. Linklion: A link repository for the web of data. In *Proceedings of Extended Semantic Web Conference (ESWC 2014)*, 2014.
5. A.-C. N. Ngomo, J. Lehmann, and M. Hassan. Transfer learning of link specifications. In *Seventh IEEE International Conference on Semantic Computing (ICSC)*, 2013.
6. 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.
7. 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.
8. G. Papadakis, E. Ioannou, C. Niederèe, T. Palpanasz, and W. Nejdl. Eliminating the redundancy in blocking-based entity resolution methods. In *JCDL*, 2011.
9. A. Schultz, A. Matteini, R. Isele, C. Bizer, and C. Becker. LDIF - linked data integration framework. In *COLD*, 2011.
10. 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.
11. 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.
12. 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.
13. C. Unger, L. Bühmann, J. Lehmann, A.-C. Ngonga Ngomo, D. Gerber, and P. Cimiano. Template-based question answering over rdf data. In *Proceedings of the 21st international conference on World Wide Web*, pages 639–648, 2012.
14. J. Urbani, S. Kotoulas, J. Maassen, F. van Harmelen, and H. Bal. OWL reasoning with webpie: calculating the closure of 100 billion triples. In *Proceedings of the ESWC 2010*, 2010.