Semantic Cities
AAAI Technical Report WS-12-13

Enabling Linked Data Publication with the Datalift Platform

François Scharffe
LIRMM
Universit de Montpellier
Montpellier, France

Ghislain Atemezing
Raphaël Troncy
Eurecom
Sophia Antipolis, France

Fabien Gandon
Serena Villata
INRIA Sophia-Antipolis
Sophia Antipolis, France

Bénédicte Bucher
Fayçal Hamdi
IGN
Paris, France

Laurent Bihanic
Gabriel Képéklian
Atos
Bezons, France

Franck Cotton
INSEE
Paris, France

Jérôme Euzenat
Zhengjie Fan
INRIA & LIG
Montbonnot, France

Pierre-Yves Vandenbussche
Bernard Vatant
Mondeca
Paris, France

Abstract

As many cities around the world provide access to raw public data along the Open Data movement, many questions arise concerning the accessibility of these data. Various data formats, duplicate identifiers, heterogeneous metadata schema descriptions, and diverse means to access or query the data exist. These factors make it difficult for consumers to reuse and integrate data sources to develop innovative applications. The Semantic Web provides a global solution to these problems by providing languages and protocols for describing and accessing datasets. This paper presents Datalift, a framework and a platform helping to lift raw data sources to semantic interlinked data sources.

Introduction

One decisive step in the transition towards a semantic and ubiquitous Web is the availability of linked and structured data. Structured data is already present in databases, in metadata attached to medias, and in millions of spreadsheets created everyday across the world. The recent emergence of linked data radically changes the way structured data is being considered. By giving standard formats for the publication and interconnection of structured data, linked data transforms the Web into a giant database. However, even if the raw data is there, even if the publishing and interlinking technology is there, the transition from raw published data to interlinked semantic data still needs to be done.

Datalift[1] ambition is to act as a catalyst for the emergence of the Web of Data (Heath and Bizer 2011). Made of large raw data sources interlinked together, the Web of Data takes advantage of Semantic Web technologies in order to ensure interoperability and intelligibility of the data. Adding data to the Web of Data consists of (i) publishing data as RDF graphs: a standard data format, (ii) linking these datasets together, by identifying equivalent resources in other data sources, (iii) describing the vocabulary used in published data through ontologies.

The Web of Data has taken recently a strong acceleration with the publication of large datasets by public institutions around the world and experiments being made to publish these data as Linked Data.

However, if isolated data publication initiatives using Semantic Web technologies exist, they remain limited for several reasons:

1. Similarly to the Web, the power of which comes from the interconnection of pages together through hyperlinks, the Web of Data will only make sense if the data it contains are interconnected. A few interlinking tools already exist but require too much manual intervention for reaching Web scale.
2. A large number of ontologies covering various domains are quickly appearing, raising the following problems: many ontologies overlap and require to be aligned together for proper interoperability between the data they describe. Selecting the appropriate ontology for describing a dataset is a tedious task. Once an ontology is selected, the data to be published eventually needs to be converted in order to be linked to the ontology. Solving these technical problems require expertise which leads to publication processes that are not suited for large amounts of heterogeneous data.
3. In order to ensure a publication space which is at the same time open and giving to each publisher its rights on the published data, it is necessary to provide methods for rights management and data access.
4. Finally, and again analogically with the Web, a critical

[1] Datalift: http://datalift.org

amount of published data is needed in order to create a snowball effect similar to the one that led the Web to take the importance it has nowadays.

The goal of Datalift is to address these four challenges in an integrated way. More specifically, to provide a complete path from raw data to fully interlinked, identified, and qualified linked data sets, we develop a platform for supporting the processes of:

- selecting ontologies for publishing data;
- converting data to the appropriate format (RDF using the selected ontology);
- interlinking data with other data sources;
- publishing linked data.

In the remainder of this paper, we detail this framework and its implementation: the Datalift platform.

Datalift framework

This section introduces the steps needed to bring raw structured data to interlinked, identified, and qualified RDF datasets.

Vocabulary Selection: Linked Open Vocabularies

Vocabularies selection is about the many dialects (RDFS and OWL ontologies) used in the growing linked data Web. Most popular ones form now a core of Semantic Web standards but many more are published and used. Not only linked data leverage a growing set of vocabularies, but vocabularies themselves rely more and more on each other through reusing, refining or extending, stating equivalences, declaring metadata.

The publisher of a dataset should be able to select the vocabularies that are the most suitable to describe the data, and the least possible terms should be created specifically for a dataset publication task. The problem is thus to provide means for a data publisher to be able to locate the vocabularies suited for the published data.

The Linked Open Vocabularies[2] (LOV) objective is to provide easy access methods to this ecosystem of vocabularies, and in particular by making explicit the ways they link to each other and providing metrics on how they are used in the linked data cloud, to improve their understanding, visibility and usability, and the overall quality.

LOV targets both vocabulary users and vocabulary managers.

- Vocabulary users are provided with a global view of available vocabularies, complete with precious metadata enabling them to select the best available vocabularies for describing their data, and assess the reliability of their publishers and publication process.
- Vocabulary managers are provided with feedback on the usability of what they maintain and publish, common best practices their publication should stick to in order to keep being reliably usable in the long run.

The LOV tool is further described later in Section .

Creating URIs

The Web uses URIs (Uniform Resource Identifiers) as a single global identification system. The global scope of URIs promotes large-scale network effects, in order to benefit from the value of Linked Data, government and governmental agencies need to identify their resources using URIs. We provide a set of general principles aimed at helping stakeholders to define and manage URIs for their resources. These principles are based on existing best practices as they are currently collected within the W3C Government Linked Data Working Group[3].

The first thing to do is to identify the type of entities the data to be published is about. For example, the National Institute of the Geographic and Forest Information (IGN) in France maintains a Large Scale Reference of the topography of the French territory with a 1-meter resolution. In this database, one can find information about departments, cities, monuments, roads and many other types of geographical entities. This analysis of the data will provide the top level objects that will be present in the URI scheme.

It is a good practice to have a base URI of the form http://{sector}.{domain} where *sector* refers to an activity (e.g. legislation, education, topography) and *domain* refers to an internet top-level domain name. From a base URI, we propose the following scheme depending on the nature of the resource the URI identifies:

- URI for ontologies or vocabularies: /ontology/shortname#class
- URI for real-world things: /id/type/id
- URI for data about real-world things: /data/type/id
- URI for datasets: /data/dataset/version

For example, the Eiffel tower will be identified by http://topolography.ign.fr/id/monument/PAICULOI0000000000142427 while the 7th district of Paris will be identified by http://topolography.ign.fr/id/arrondissement/SURFCOMM0000000013680725. These URIs must be persistent and dereferencable. Upon a retrieval request, a web server should redirect the /id/ URI to the /data/ URI and serve an HTML or RDF representation of the resource according to the instruction of the user agent.

Summarizing, the URIs to be minted should be short and human readable, and incorporate as much as possible existing identifiers where available. While IRIs can also be used for identifying resources, the internationalization of the identifiers yields numerous problems that softwares cannot deal with correctly at the moment.

Converting data formats to RDF

Once URIs are created and a set of vocabulary terms able to represent the data is selected, it is time to convert the source dataset into RDF. Many tools exist to convert various structured data sources to RDF.[4]. The major source of

[2] http://labs.mondeca.com/dataset/lov

[3] http://www.w3.org/2011/gld/

[4] http://www.w3.org/wiki/ConverterToRdf

structured data on the Web comes from of spreadsheets, relational databases and XML files.

Our approach is in two steps. First, a conversion from the source format to raw RDF is performed. Second, a conversion of the raw RDF into “well-formed” RDF using selected vocabularies is performed using SPARQL Construct queries. The first step can be automated in most of the cases.

Most tools provide spreadsheet conversion to CSV, and CSV to RDF is straightforward, each line becoming a resource, and columns becoming RDF properties. The W3C RDB2RDF working group[5] prepares two recommendations for mapping relational databases to RDF: the DirectMapping approach automatically generates RDF from the tables names for classes and column names for properties. This approach allows to quickly obtain RDF from a relational database but without using any vocabulary. The other approach, R2RML, provides a mapping language allowing to assign vocabulary terms to the database schema. In the case of XML, a generic XSLT transformation can be performed to produce RDF from a wide range of XML documents.

Access rights

In the Web of Data, providers expose their content publicly, knowing that it is not safe. This may prevent further publication of datasets, at the expense of the growth of the Web of Data itself. Moreover, the mobile, ubiquitous Web is continuously evolving, enabling new scenarios in consuming and contributing to the Web of Data. We must therefore not ignore the mobile context in which data consumption takes place. (Costabello et al. 2012) propose a framework which relies on two complementary lightweight vocabularies, `S4AC`[6] for access control, and `PRISSMA`[7] for modeling the mobile context. Context is seen as an encompassing term, an information space defined as the sum of three different dimensions: the *User* model, the *Device* features and the *Environment* in which the request is performed. Access Policies protect a named graph, thus targeting single triples, if needed. Each Access Policy is associated to a privilege level and includes a set of context-aware Access Conditions, i.e. constraints that must be satisfied, conjunctively or disjunctively, to access the protected resources. Access Conditions are implemented as SPARQL 1.1 ASK queries. At runtime, Access Policies are associated to the actual mobile context used to evaluate the set of Access Conditions.

The access control evaluation procedure includes the following main steps: (1) the mobile consumer queries the SPARQL endpoint. Contextual information is sent along with the query and saved as named graph using SPARQL 1.1 Update Language statements[8]. (2) The client query is filtered by the Access Control Manager instead of being directly executed on the SPARQL endpoint. (3) The Access Control Manager selects the set of policies affecting the client query and after their evaluation returns the set of accessible named graphs. (4) The client query is executed only on the accessible named graphs and (5) the result of the query is returned to the consumer.

Dataset publication

When the RDF dataset is ready it can be published on a web server and a made available as Linked Data. Various RDF triple stores are available providing access mechanisms to the data. Publication of the dataset also includes attaching metadata information using the VoID vocabulary[9], providing a Semantic Sitemap[10] and referencing it on The Data Hub[11].

Linking datasets

Data linking (Ferrara, Nikolov, and Scharffe 2011) is a set of techniques for identifying similar instances between datasets. This can be done by comparing the property values of the instances. String matching the most basic technique to compare whether two values are similar or not. Machine learning is also used to find out the most efficient comparison pattern, as shown in (Hu, Chen, and Qu 2011; Isele and Bizer 2011; Ngonga Ngomo et al. 2011). That is, which property values should be compared and what comparison functions should be used. There are two machine learning branches: supervised method and unsupervised method. Supervised method uses a training dataset to find out the most suitable matching pattern (Hu, Chen, and Qu 2011; Nikolov et al. 2008). Now more work focus on exploring unsupervised method for interlinking instances, for it saves time on collecting training sets (Araújo et al. 2011) . Ontology alignment is also used to decrease the comparison scale. For it is composed of correspondences on classes and properties, it is a heuristics to show from which classes to find similar instances, and comparing which property’s values for judging similarity or not. Besides, graph structure and statistical techniques are also used for matching. Usually, these techniques are combined to fulfill the interlinking process. One of the main problem lies in the number of comparison to perform in order to evaluate the similarities between two datasets. If there are M instances in the source dataset and N instances in the target dataset. There should be $M*N$ times comparisons. In order to reduce this number, several strategies are proposed. The fundamental strategy is to use keys to index entities so that entities may be looked up for from their keys. However, keys like item IDs or administrative codes often cannot be compared as they appear in only one of the two datasets. It is thus needed to be able to compute various keys on the two dataset to try finding a corresponding one. It is a combination of all these techniques, together with a proper user interface to set the tool parameters that will lead to efficient data linking.

Implementation

This section describes the Datalift platform as it is currently implemented. The platform is designed to be

[5] http://www.w3.org/2001/sw/rdb2rdf/

[6] http://ns.inria.fr/s4ac

[7] http://ns.inria.fr/prissma

[8] http://www.w3.org/TR/sparql11-update

[9] http://vocab.deri.ie/void

[10] http://sw.deri.org/2007/07/sitemapextension/

[11] http://thedatahub.org/

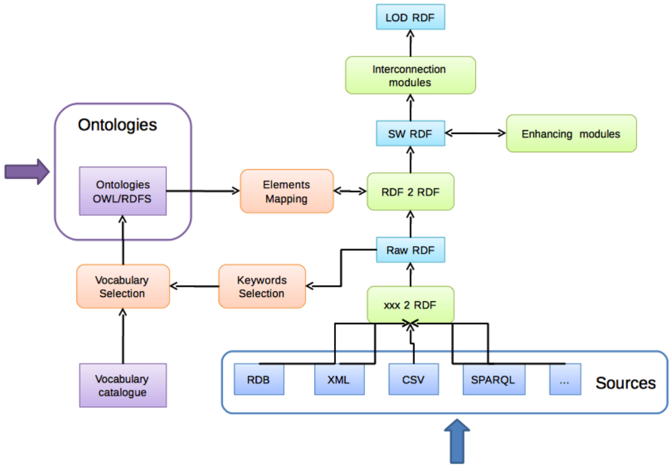

Figure 1: Datalift Data Workflow

modular, each module performing tasks of the following workflow. The platform can be downloaded at https://gforge.inria.fr/projects/datalift.

Datalift workflow

The actual workflow of the data lifting process is depicted on Figure 1.

The user begins the data lifting process by submitting a structured data source in one of the supported formats, may it be a CSV or XML file, a relational database or an existing RDF source. The system converts this file to RDF by following a set of rules to apply depending on the submitted file format. This conversion does not take into account vocabularies, namespaces nor links to existing datasets. We call it raw RDF conversion. From this point, we have a unique format: RDF, to work on data sources. Further modifications can be applied as follows. The Selection step asks the user to input a set of vocabularies terms that will be used to describe the lifted data. Once the terms are selected, they can be mapped to the raw RDF and then converted to properly formatted RDF. Technically, a set of SPARQL construct queries is performed to modify the source RDF graph and generate the target RDF using the selected vocabularies. Other modifications can be performed using enhancing modules. Enhancements include replacing a characters string with a URI from another dataset (for example ”Montpellier” becomes http://dbpedia.org/resource/Montpellier) or adding metadata to the published dataset. The data can then be published on the platform SPARQL endpoint. The last step in the process aims at providing links from the newly published dataset to other datasets already published as Linked Data on the Web. The interconnection modules give the possibility to achieve this linkage.

We next describe the platform architecture and the modules currently implementing this workflow.

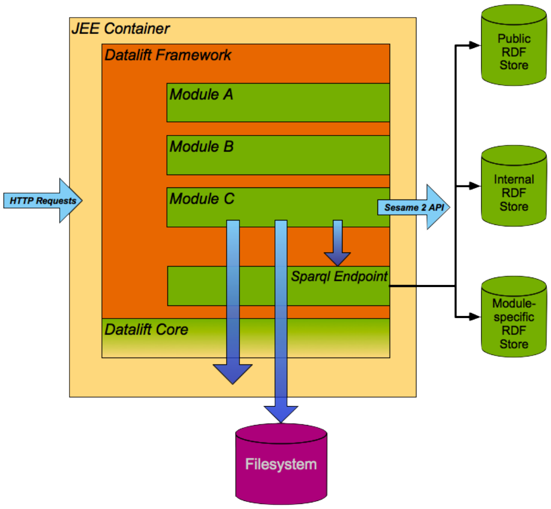

Figure 2: Datalift Architecture

Platform Architecture

The Datalift platform is a Java-based development and runtime environment for deploying RDF-based REST web services. It includes:

The Datalift framework : a set of Java APIs for developing Datalift components

The Datalift core : an implementation of the Datalift framework as a JEE web application

A set of RDF stores : (a.k.a. triple stores), to persist RDF data, accessed using the OpenRDF Sesame 2 API

Third-party components : (or modules) adding general purpose or customer-specific features to the platform.

The Datalift framework provides a high-level view of the underlying runtime environment to ease the development of RDF-based services, automating access to the RDF stores (regardless the actual RDF engine being used), enforcing authentication and access control, applying URI naming policy, etc. It also provides a file system abstraction to support deployment on distributed (grid) storage.

Once RDF data are produced and persisted in one of the RDF stores, the platform makes them available for both direct access (as a web resource accessible through an HTTP(S) URL) and querying (SPARQL 1.1). The provided SPARQL endpoint includes advanced features such as query routing between the available RDF stores, access control, data filtering, data enrichment (e.g. to attach licensing information to the query results), etc. Whereas Datalift components have full access to the RDF stores, including SPARQL Update, the public SPARQL endpoint only offers querying.

The Datalift core is one realization of the Datalift framework for deploying the platform as a JEE web application or a single-user standalone desktop application (actually a web application wrapped in a native executable) . It relies on Jersey (JAX-RS reference implementation), Apache Shiro (security framework) and Apache Velocity (HTML page template engines).

The well-supported OpenRDF Sesame 2 API allows Datalift components to transparently interface with many RDF store products, both open-source (OpenRDF Sesame) and commercial (OpenLink Virtuoso, Ontotext OWLIM, Franz AllegroGraph, etc.). A central configuration file manages the definition of the RDF stores and their visibility (public or private). Public stores are accessible for SPARQL querying or direct RDF resource access without requiring user to first authenticate.

Each Datalift component is packaged as a JAR file and can include REST web services, dynamic HTML pages, static resources, third-party libraries as well as implementations of various internal services (RDF store connector, URI mapping policy, native data parser, etc.). Components are deployed by simply dropping them in a (configurable) directory. Each of them is executed in a sandbox (dedicated Java class loader) to avoid conflict between module using different versions of the same third-party library.

An example Datalift component included in the default distribution is the Project Manager. this module provides a user-friendly interface for lifting datasets from various native formats (CSV, XML, relational databases, etc.) into RDF by applying a sequence of data transformations: direct mapping to ”raw” RDF, conversion to a selected ontology, publication of the transformed data, interlinking with public datasets (e.g. DBpedia), etc.

Modules

The modules developed in the platform are described below.

LOV Module. LOV provides functionalities for search and quality assessment among the vocabularies ecosystem, but it also aims at promoting a sustainable social management for this ecosystem.

The LOV dataset contains the description of RDFS vocabularies or OWL ontologies used or usable by datasets in the Linked Data Cloud. Those descriptions contain metadata either formally declared by the vocabulary publishers or added by the LOV curators. Beyond usual metadata using Dublin Core, voiD, or BIBO, new and original description elements are added, using the VOAF vocabulary[12] to state how vocabularies rely on, extend, specify, annotate or otherwise link to each other. Those relationships make the LOV dataset a growing ecosystem of interlinked vocabularies supported by an equally growing social network of creators, publishers and curators.

To be included in the LOV dataset, a vocabulary has to satisfy the following requirements:

- To be expressed in one of the Semantic Web ontology languages : RDFS or some species of OWL
- To be published and freely available on the Web
- To be retrievable by content negotiation from its namespace URI
- To be small enough to be easily integrated and re-used, in part or as a whole, by other vocabularies.

Apart from the dataset, the LOV tool provides the following features:

- The ”LOV Aggregator” feature aggregates all vocabularies in a single endpoint/dump file. Last version of each vocabulary is checked on a daily basis. This endpoint is used to extract data about vocabularies and is used to generate statistics (”LOV Stat” feature) or to support research (”LOV Search” feature). While a vocabulary is aggregated, for each vocabulary elements (class or property), an explicit link rdfs:isDefinedBy to the vocabulary it belongs to is added.
- The ”LOV Search” feature gives you the possibility to search for an existing element (property, class or vocabulary) in the Linked Open Vocabularies Catalogue. Results ranking is based on several metrics: element labels relevancy to the query string, element labels matched importance, number of element occurrences in the LOV dataset, number of Vocabulary in the LOV dataset that refer to the element, number of element occurrences in the LOD
- LOV Stats are computed on all LOV vocabularies aggregated in LOV Aggregator. It provides some metrics about vocabulary elements. ”LOV Distribution” metric is about the number of vocabularies in LOV that refers to a particular element. ”LOV popularity” is about the number of other vocabulary elements that refers to a particular one. ”LOD popularity” is about the number of vocabulary elements occurring in the LOD.
- The ”LOV Suggest” feature gives the possibility to submit a new vocabulary in order to include it in the LOV catalogue. After validating a vocabulary URI, the user correct the vocabulary before submitting it. Some recommendations for vocabulary metadata description may be of help.

In the current version of the Datalift platform LOV, is provided as a standalone tool. We are currently working on making it more tightly integrated so that users can select the terms relevant to describe their datasets directly form the LOV interface.

Data convertion modules. The platform actually provides two modules for converting CSV and relational databases to RDF; A third module allows to convert raw RDF to well formed RDF usng SPARQL Construct queries. The XML conversion module is currently being integrated into the platform.

Security module. The proposed Access Control Manager (Costabello et al. 2012) is designed as a pluggable component for SPARQL endpoints. The Datalift version has been implemented as a module of the Datalift platform[13], using the Sesame[14] RDF store. Each client query over the protected datastore is associated to the RDF graph modeling the client mobile context. The mobile context is sent by the requester to the data server at (Step 1). The module saves

[12] http://labs.mondeca.com/vocab/voaf/

[13] A demo of the module may be found at http://dl.dropbox.com/u/15116330/s4acTest.mov

[14] http://www.openrdf.org/

current mobile context in a local cache (Step 2). The selection of the Access Policies returns the set of Access Policies concerned by the query (Step 3). We select all the Access Policies which have the identified Access Privilege. The Access Control Manager appends a `BINDINGS` clause to the Access Conditions included in the selected policies. This is done to bind the `?context` variable to consumer's actual context. Access Conditions are executed on the protected datastore. The resources protected by verified Access Policies are added to the set of accessible named graphs, conjunctively or disjunctively (Step 3). Finally (Step 4), the "secured" client query is sent to the SPARQL endpoint with the addition of the `FROM` clause. Query execution is therefore performed only on the accessible named graphs, according to the contextual information of the requester.

Interlinking module. The interlinking module provides means to link datasets published through the Datalift platform with other data sets available on the Web of Data. Technically the module helps to find equivalence links in the form of "owl:sameAs" relations. The technical details of the module are as follows.

1. an analysis of the vocabulary terms used by the published data set and a potential data set to be interlinked is performed.
2. when the vocabulary terms are different, check if alignments between the terms used by the two data sets are available. We use the alignment server provided with the Alignment API[15] for that purpose.
3. translate correspondences found into SPARQL graph patterns and transformation functions combined into a SILK script.
4. run SILK (Bizer et al. 2009) to interlink datasets.

Until now, SILK is successfully embedded in the Datalift platform. An interface is built for the user to upload the SILK script and other parameters for running the script. Later versions of the module module will provide an interface to assist the user in finding candidates datasets for interlinking. Another interface is designed to display and validate the resulting links.

Conclusion

We have presented the Datalift project aiming at providing a framework and an integrated platform for publishing datasets on the web of Linked Data. We have described the framework, the platform architecture, and the modules actually available, providing a complete path from raw structured data to Linked Data.

While the datalift platform is used today with data publishers associated with the project, it remains an expert tool. Users of the platform needs significant knowledge of the Semantic Web formalisms (RDF, RDFS/OWL, SPARQL) in order to perform the lifting process. We are actually working on making the platform easier to use by

- providing graphical user interfaces for the lifting process;
- providing more automation to vocabulary selection, and mapping of vocabulary terms to the source data schema;
- automating the interconnection step by generating Silk scripts automatically;
- offering templates for the URI design ;

We are finally working on developing end-user applications that make use of the lifting process performed by the platform and show the added value of combining datasets to reveal new insights on the data. In particular we are working with with local and national government agencies to show how the publication and interlinking of datasets enable the development of innovative application for citizens.

Aknowledgements

The Datalift project is funded by the French National Research Agency under grant ANR-10-CORD-009

References

Araújo, S.; Hidders, J.; Schwabe, D.; and de Vries, A. 2011. Serimi - resource description similarity, rdf instance matching and interlinking. *CoRR* abs/1107.1104.

Bizer, C.; Volz, J.; Kobilarov, G.; and Gaedke, M. 2009. Silk - a link discovery framework for the web of data. In *CEUR Workshop Proceedings*, volume 538, 1–6.

Costabello, L.; Villata, S.; Delaforge, N.; and Gandon, F. 2012. Linked data access goes mobile: Context-aware authorization for graph stores. In *Proceedings of 5th Linked Data on the Web Workshop (LDOW2012)*.

Ferrara, A.; Nikolov, A.; and Scharffe, F. 2011. Data linking for the semantic web. *Int. J. Semantic Web Inf. Syst.* 7(3):46–76.

Heath, T., and Bizer, C. 2011. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool.

Hu, W.; Chen, J.; and Qu, Y. 2011. A self-training approach for resolving object coreference on the semantic web. In *Proceedings of WWW 2011*, 87–96. ACM.

Isele, R., and Bizer, C. 2011. Learning linkage rules using genetic programming. In *OM2011, CEUR Workshop Proceedings*, volume 814.

Ngonga Ngomo, A.-C.; Lehmann, J.; Auer, S.; and Höffner, K. 2011. Raven - active learning of link specifications. In *OM2011, CEUR Workshop Proceedings*, volume 814.

Nikolov, A.; Uren, V. S.; Motta, E.; and Roeck, A. N. D. 2008. Handling instance coreferencing in the knofuss architecture. In *IRSW 2008, CEUR Workshop Proceedings*, volume 422, 265–274.

[15] http://alignapi.gforge.inria.fr/