Link++: A flexible and customizable tool for connecting RDF data sources

Ali Masri[1,2], Karine Zeitouni[1], Zoubida Kedad[1], and Gabriel Kepeklian[2]

[1] DAVID Laboratory
University of Versailles Saint-Quentin-en-Yvelines
Versailles, France
{first.last}@uvsq.fr
[2] VEDECOM Institute
Versailles, France
{first.last}@vedecom.fr

Abstract. Existing interlinking tools focus on finding similarity relationships between entities of distinct RDF datasets by generating owl:sameAs links. These approaches address the detection of equivalence relations between entities. However, in some contexts, more complex relations are required, and the links to be defined follow more sophisticated patterns. This paper introduces Link++, an approach that enables the discovery of complex links in a flexible manner. Link++ enables the users to generate rich links by specifying a link pattern as well as rules and functions to discover them. When visiting the demo, attendees will be introduced to all the aspects of the system explaining the required steps to define custom functions, connection patterns and linking rules until finally obtaining custom connections.

Keywords: Linked Open Data, Data Interlinking, Semantic Web

1 Introduction

Data interlinking aims to find equivalence relations between entities of different datasets in the Web of data. Roughly speaking, a user defines two data sources and a linking rule which specifies how different entities can be related together. The results are a set of triples in the form x *owl:sameAs* y which are used to navigate from one data source to another in order to gain richer information.

Existing interlinking tools [2, 6, 1, 3, 5] support this task by providing a platform with a set of similarity functions that can be combined to form a designated rule. The interlinking engine processes the given datasets, applies an interlinking rule and returns the results.

In the same spirit as for data interlinking, our aim is to answer the need for more complex relations in some application domains. We want to enable the possibility of discovering complex relationships carrying more information in the form of properties that would be used for analysis purposes.

For instance, consider two transportation datasets representing a set of bus stops and a set of train stations respectively. Assume that our goal is to link bus

stops that can be reached from a train station, with the intention of developing a multi-modal trip planner that uses these links to compute trips. Using the existing tools, we are faced with two main limitations.

- The restriction to a predefined set of functions for composing linking rules. In our case, we lack of precise functions to calculate the closeness of a bus stop and a train station. Existing tools support geographical distances to calculate distance similarity which are not always reliable in real life, e.g., two geographically close stops that are separated by a river might be hard to reach from one another, therefore connecting them does not make sense.
- The representation of the generated output. Supporting complex relations requires more complex output patterns. Considering our example, interlinking based on the stations that can be reached from another gives the output *BusStop1 nextTo TrainStation132* which delivers no idea about the semantics of this relation. They are next to each others but how close are they? and what are the transportation modes that we can use? etc.

Our contribution is twofold, first we define connection patterns – a new way of customizing the output of a linking process. Second, we propose a platform that enables dynamic functions insertion and integration within the linking process.

In this work we introduce Link++, a tool that enables users to produce complex links by using their defined functions and connection patterns to support any type of relation between datasets.

The paper is structured as follows: In Section 2 we give an overall description of our system. We test the approach via a use case explained in Section 3 then we finally conclude in Section 4.

2 System design and implementation

To enable the discovery of complex links we introduce the notions of customized *Connection Patterns*. A connection pattern defines the content and the format of the link to be generated by defining the properties along with the associated properties. Figure 1 shows an overall view of the designed platform to generate customized connections.

TO begin, a configuration task is executed to define a connection pattern represented by a set of connection properties where each property is calculated by a function. Custom functions can be provided either by the user or by a common pre-coded functions library. In our implementation the functions are represented within a JAVA class and the library dependencies are regular .jar files. The configuration parameters are inputs of the connection generation component where the linking rule is defined. The linking rule is a file that describes the conditions to generate a connection pattern between entities. In short, if a rule between two entities is valid, a connection is instantiated and filled based on the given template (connection pattern). In the connection generation phase the system passes over the data sources and test the validity of a user-defined

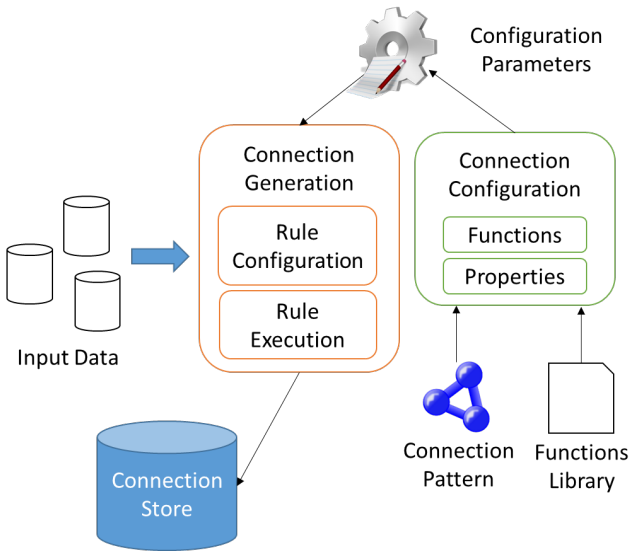

Fig. 1. An approach for flexible and customizable connection generation

linking rule. If a rule is valid the connection pattern is evaluated and the resulting connection is stored in a connection store. In our implementation, both the configuration file (connection pattern) and the linking rule are XML files described by a DTD and the connection is generated in Turtle format[3]. Figure 2 shows an example of a connection pattern. An executable version of the system

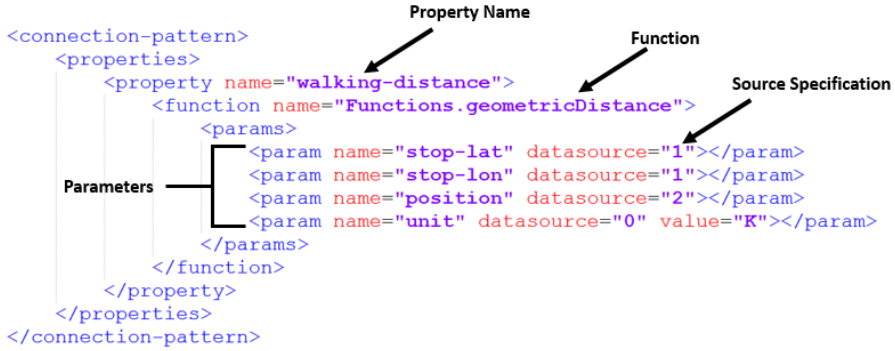

Fig. 2. An example of a connection pattern

[3] https://www.w3.org/TR/turtle/

can be found online via the link: https://github.com/alimasri/link-plus-plus.git in addition to a video tutorial on: https://youtu.be/u2gr7Wa4eT4.

3 Scenario

The scenario we present in this demo describes how we can use our tool to connect transportation points of transfer to provide data for multimodal trip planning solutions. Transportation companies work in isolation and provide specific planning solutions for their data. We have used our solution to expand the transportation view by creating chains of connections in cities. We have considered two data sources representing SNCF train stations[4] and VELIB[5] bike sharing stations in the Paris area in France. Both data sources are pre-processed and translated into RDF using the DataLift platform[4].

The specified linking rule aims to find for each train station the nearby bike sharing stations, which is defined to be the walking distance since it is more relevant for our case, and it can be calculated via any distance API over the web. In this scenario we choose the Google distance matrix API[6], however users are free to choose any existing API or create their own since this is completely independent from the process. For the connection pattern, we are interested in getting the source and destination (generated by default), the calculated walking distance and the estimated walking time. Both files are provided as inputs to our system and we have successfully generated an output file representing the needed connections as shown in a snapshot of our system in figure 3.

The output file can be used as an input for trip planning algorithms to create multimodal trips and to facilitate and optimize passengers trips. The reliability of the results depends on the chosen functions and this requires a careful selection from the user.

4 Conclusion

Current interlinking tools focus on discovering equivalence relationships between datasets. In the same spirit as these tools, we have proposed an approach which discovers more complex relations to support specific application requirements such as linking transportation data sources.

In this paper we introduced Link++, a flexible interlinking tool which provides user defined semantic relationships between entities. The system enables the users to define their own functions and connection patterns thus providing customized and flexible linking capability.

Future work will target the dynamic nature of the created links. For instance, in some application domains (e.g. transportation) the created links can be dynamic and vary in real time which indeed affect their correctness and reliability.

[4] `http://gtfs.s3.amazonaws.com/sncf_20131211_1451.zip`
[5] `http://opendata.paris.fr/explore/dataset/stations-velib-disponibilites-en-temps-reel/`
[6] https://developers.google.com/maps/documentation/distance-matrix/

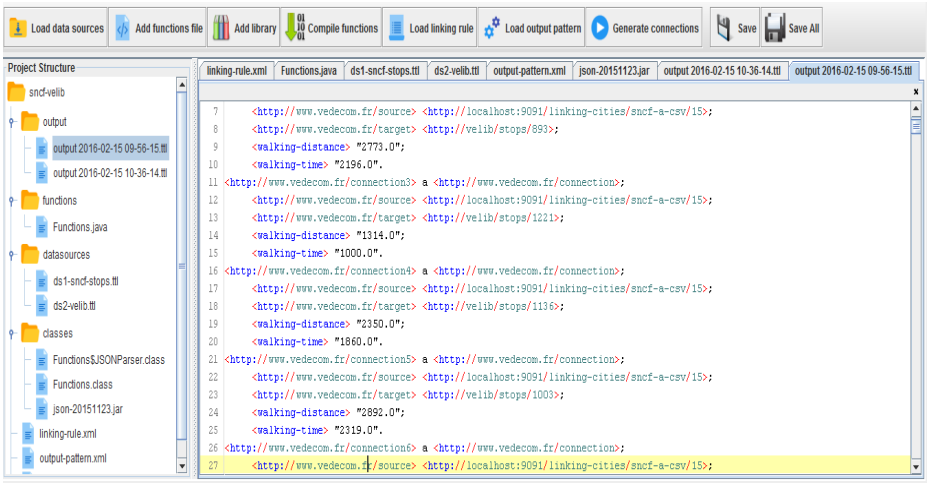

Fig. 3. Screen shot of the implemented system Link++

Moreover, we are interested in defining a functions library where users can share their functions and connection patterns allowing reusability and knowledge sharing.

Acknowledgments

We thank Mr. Bertrand Leroy for the fruitful discussion.

References

1. Afraz Jaffri, Hugh Glaser, and Ian C Millard. Managing uri synonymity to enable consistent reference on the semantic web. In *Proceedings of the Workshop on Identity, Reference, and the Web (IRSW)*, 2008.
2. Axel-Cyrille Ngonga Ngomo and Sören Auer. Limes: A time-efficient approach for large-scale link discovery on the web of data. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Three*, IJCAI'11, pages 2312–2317. AAAI Press, 2011.
3. Yves Raimond, Christopher Sutton, and Mark B Sandler. Automatic interlinking of music datasets on the semantic web. *Linked Data on the Web*, vol. 369, 2008.
4. François Scharffe, Ghislain Atemezing, Raphaël Troncy, Fabien Gandon, Serena Villata, Bénédicte Bucher, Fayçal Hamdi, Laurent Bihanic, Gabriel Képéklian, Franck Cotton, et al. Enabling linked data publication with the datalift platform. In *Proceedings of AAAI workshop on semantic cities*, 2012.
5. François Scharffe, Yanbin Liu, and Chuguang Zhou. Rdf-ai: an architecture for rdf datasets matching, fusion and interlink. In *Proceedings of IJCAI 2009 workshop on Identity, reference, and knowledge representation (IR-KR)*, 2009.
6. Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Silk-a link discovery framework for the web of data. *Linked Data on the Web*, vol. 538, 2009.