Datasets First!
A Bottom-up Data Linking Paradigm

Konstantin Todorov

LIRMM / University of Montpellier / CNRS, France
konstantin.todorov@lirmm.fr

Abstract. Data linking is understood as the task of establishing typed links between entities across different RDF datasets via the help of automatic link discovery systems. Based on decades of research and practice in the Web community, the current paper speculates on the need of a paradigm shift when it comes to designing such systems. We depart from the premise that the current state-of-the-art focuses on genericness and automaticity, while not paying sufficient attention to the particular properties and nature of the underlying data and the ensuing linking problem types. We draw new research axes, upon which the data linking task will be redefined as automatic detection of the type of linking problem at hand based on the characteristics (profiles) of the candidate datasets.[1,2]

Linked data and its underlying technologies have been gaining popularity over the past years, due to the means they offer for data reuse and federation, increased visibility and sharing on the web and facilitated exchange of metadata. We define the problem of data linking as that of automatically establishing typed links between the entities of two or more RDF datasets or graphs. A variety of data linking systems have been proposed over the past 15 years within the Web community with interactions with government, cultural or research institutions as major linked data consumers and providers. As a result, vast amounts of linked data already exist on the Web (we refer, for example, to the LOD project). A number of benchmarks are developed and shared publicly in order to provide frameworks for the evaluation of data linking systems, driven by the well-known OAEI campaign, or the more industry-oriented EU HOBBIT project.[3]

Where are we now. State-of-the-art research into data linking [1] goes in two main directions: (1) proposing novel generic data linking systems and (2) developing methods for automatic link specification by (semi-)supervised machine learning techniques, in order to assist the configuration and tuning of established tools. Several of the most common systems, such as SILK [2] and LIMES [3], adopt a property-based link-discovery strategy: a set of predicates has to be

[2] The author thanks Stefan Dietze and Mathieu D'Aquin for their valuable feedback.

[3] http://oaei.ontologymatching.org, https://project-hobbit.eu/

selected before the system proceeds to compare their values by the help of (an aggregation of) similarity measures that also need to be selected and tuned. This configuration task can be demanding in terms of user involvement in real-world scenarios. Hence, a number of methods have been proposed to assist the users in the configuration process. Properties to compare can be selected by the help of key discovery tools. While many approaches exist (e.g., [4]), their use for data linking is not straightforward because they often produce a large number of keys that are valid on a single dataset with no assessment of their likelihood to discover links. On the other hand, automatic link specification learning approaches develop (semi-)supervised techniques to select and combine similarity measures and fix their thresholds. Systems like EAGLE [5] and WOMBAT [6] are included in LIMES, just as ActiveGenLink [7] is part of SILK.

Most existing linking approaches have in common the fact that they attempt to solve the problem from a generic stance by remaining vastly agnostic to the nature of the underlying data [1]. Many systems achieve good results on dedicated benchmarks [8], but fail to take into account the particularities of the various domains and/or data generation practices that raise very specific heterogeneity issues calling for a significant user input. In particular, the user is required to have an in-depth understanding of both their data *and* the internals of the linking system of choice in order to achieve satisfactory results, as shown in [9]. The quest to fully automate the linking task, on which recent research has departed, remains rather challenging, as investigated in [10]: a heavy machinery of learning link specification rules is being developed to only partially assist the users in the selection of parameters in the pre-processing step of the linking task.

Rethinking the data linking task. We argue that the current generic approach to develop data linking solutions has reached its limits and suggest that a paradigm shift in the way we look onto this task needs to take place. We propose to enable the development of data-centric approaches for bottom-up linking, rather than investing efforts in divising incremental generic solutions: time has come to step back and look at what we can learn from the large amount of existing cross-dataset links *and* linking systems.

We formulate the arguably outrageous hypothesis that there exist a finite number of identifiable and generalisable *types of linking problems*, defined as heterogeneity types that two to-be-linked datasets can manifest, e.g. differences in terminology, natural languages or structure, as presented in [10]. Additionally, we hypothesise that these linking problems can be detected automatically by the help of machine learning (ML) models trained on sufficiently large amounts of quality linked data. On the other hand, state-of-the-art linking tools are based on modules (we will call them *atomic* or *modular* solutions), that allow to handle separately many of these linking problem types (e.g., measure the string or semantic similarity of entities). On these bases, we redefine the data linking task as that of the *automatic identification via ML techniques of the linking problem type(s) that two datasets manifest and the application of an automatically generated combination of atomic linking solutions that are best fit for the datasets at hand*. We propose to lean upon the wealth of existing linked data sets, particularly those coming from real-world scenarios, in order to enable training and validation of ML models, while a number of RDF graph profiling and graph embedding methods will be applied in order to extract the necessary features for these models.

Proposed solution and challenges. Based on the hypotheses formulated above, we propose to direct future research and engineering effort into the development of a data-centric bottom-up linking framework that channelizes and consolidates existing disparate efforts. This will allow to build on the wealth of linked RDF graphs via their in-depth analysis and consolidation and take advantage of years of research and practice in the field (cf. Fig. 1). We identify a set of research axes and associated challenges on the way to realize this framework.

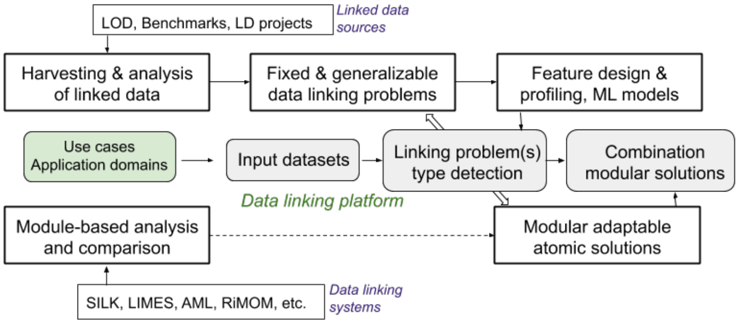
LOD, Benchmarks, LD projects
Linked data sources
Harvesting & analysis of linked data
Fixed & generalizable data linking problems
Feature design & profiling, ML models
Use cases Application domains
Input datasets
Linking problem(s) type detection
Combination modular solutions
Data linking platform
Module-based analysis and comparison
Modular adaptable atomic solutions
SILK, LIMES, AML, RiMOM, etc.
Data linking systems

Fig. 1: Conceptual overview of the proposed framework.

(1) Linked data harvesting and analysis. This axis consists in the consolidation of a large amount of already existing quality linked Web data, benchmarks, evaluation campaigns and linked data projects from a large variety of domains. An in-depth statistical analysis of these data will allow to identify a number of limited and generalisable linking problem types, discover correlations between application domains and data structure or quality, or between heterogeneity types and link density. This will inform the feature design in (2) and will generate training data for the automatic classification of pairs of datasets according to their linking problem type(s). *Challenge:* We need to ensure high quality of the links from which we will learn. Hence, we propose to rely on existing real-world benchmarks as a starting point, where datasets are often grouped according to specific heterogeneity criteria (terminology, logics, structure, etc) that can be mapped to the identified linking problem types. Alternatively, one can apply existing linking methods of high precision in a preprocessing step. Relying on a large variety of data sets and domains is important to guarantee representativity.

(2) Joint datasets feature design. This axis involves data linking-oriented graph profiling and feature extraction via state-of-the-art RDF graph profiling techniques [11] and joint graph embeddings methods (learning vector representations jointly on a pair of graphs). It will generate the set of features that describe jointly the linking candidate datasets and are indicative of the heterogeneities that they manifest, necessary to train the ML algorithms. Thus,

we aim to answer the question of what is it that discriminates between two pairs of datasets manifesting two different types of linking problems. *Challenge:* A large plethora of RDF dataset profiling methods and tools exist (reviewed in [11]), allowing to extract and represent the graphs in terms of a number of "profile features", such as their domains, connectivity, representative instances, quality, provenance, statistics, dynamicity, etc. Under the hypothesis that these features in combination account for describing the datasets from aspects that match the linking problems identified in (1), a significant challenge consists in identifying the set of features that are necessary and sufficient in order to design efficient linking problem classification ML models. In addition, from a practical viewpoint, the application of the methods that allow for the extraction of these features is not straightforward, as outlined in [11]. Finally, we will be interested in extracting *joint* profiles for a *pair* of datasets, which is not explicitly addressed in the literature. In that respect, profile features can be coupled with graph embeddings learned *jointly* on a pair of RDF graphs, which is a novel problem in the community [12].

(3) Learning and applying ML models. This axis will rely on the training data harvested in (1) and the features extracted in (2) in order to define and apply classification models for linking problem type detection. *Challenge:* We identify here the standard challenge of selection and tuning of ML model(s) from a set of supervised algorithms. In addition, the multitude of possible classes will lead us beyond the standard binary classification task.

(4) Filling in "a shelf" of automatic, adaptable, modular solutions for each of the linking problems identified in (1). We rely on the premise that a linking problem type is fine-grained enough so that a particular modular solution can be applied to it (for example, relying on lexical synset intersection for synonymy-type heterogeneity). These modular solutions will be identified by a comparative analysis of the modules and respective performances of a large spectrum of existing data linking systems, as this has been in part performed in [1]. *Challenge:* A significant effort will be involved in the association of state-of-the-art atomic solutions to the data linking problem types identified in (1). In the lack of training data, unsupervised ML models have to be divised, enhanced by a *human-in-the-loop* approach.

Conclusion. Instead of trying to fit a generic solution to any linking problem and dataset type, we suggest to enable a better understanding of the underlying data before applying a targeted solution best suited to the particular datasets at hand. We rely on the premise that the in-depth analysis of large amounts of linked data will allow to isolate a limited number of identifiable data linking problems that ML models based on datasets profiles will help detect automatically. The moment is appropriate to take this approach for reasons of, one the one hand, the large and growing availability of linked data in an ever greater number of domains and, on the other hand, the existence of a large plethora of data linking tools, result of decades of research and practice. We hypothesise that channelizing these decentralised endeavours will foster and facilitate the application of linked data technologies within and across an even larger variety of domains and will ultimately free the domain expert of the technological burden.

References

1. M. Nentwig, M. Hartung, A.-C. Ngonga Ngomo, and E. Rahm, "A survey of current link discovery frameworks," *Semantic Web*, vol. 8, no. 3, pp. 419–436, 2017.
2. A. Jentzsch, R. Isele, and C. Bizer, "Silk-generating rdf links while publishing or consuming linked data," in *ISWC*, 2010.
3. A.-C. N. Ngomo and S. Auer, "Limes - a time-efficient approach for large-scale link discovery on the web of data," in *IJCAI*, 2011.
4. D. Symeonidou, V. Armant, N. Pernelle, and F. Saïs, "Sakey: Scalable almost key discovery in rdf data," in *ISWC*, pp. 33–49, Springer, 2014.
5. A. N. Ngomo and K. Lyko, "EAGLE: efficient active learning of link specifications using genetic programming," in *ESWC*, pp. 149–163, 2012.
6. M. A. Sherif, A.-C. N. Ngomo, and J. Lehmann, "W ombat–a generalization approach for automatic link discovery," in *ESWC*, pp. 103–119, Springer, 2017.
7. R. Isele and C. Bizer, "Active learning of expressive linkage rules using genetic programming," *Web Semantics*, vol. 23, pp. 2–15, 2013.
8. M. Achichi, M. Cheatham, Z. Dragisic, J. Euzenat, *et al.*, "Results of the ontology alignment evaluation initiative 2017?," in *OM at ISWC*, CEUR-WS, 2017.
9. M. Achichi, P. Lisena, K. Todorov, R. Troncy, and J. Delahousse, "Doremus: A graph of linked musical works," in *ISWC*, pp. 3–19, Springer, 2018.
10. M. Achichi, Z. Bellahsene, M. B. Ellefi, and K. Todorov, "Linking and disambiguating entities across heterogeneous rdf graphs," *J. of Web Semantics*, 2019.
11. M. Ben Ellefi, Z. Bellahsene, J. G. Breslin, E. Demidova, S. Dietze, J. Szymański, and K. Todorov, "Rdf dataset profiling–a survey of features, methods, vocabularies and applications," *Semantic Web*, 2018.
12. S. Wang, J. Arroyo, J. T. Vogelstein, and C. E. Priebe, "Joint embedding of graphs," *arXiv preprint arXiv:1703.03862*, 2017.