Inference and Serialization of Latent Graph Schemata Using ShEx

Daniel Fernández-Álvarez*, Jose Emilio Labra-Gayo† and Herminio García-González‡
Department of Computer Science
University of Oviedo
Oviedo, Spain
Email: *danifdezalvarez@gmail.com, †labra@uniovi.es, ‡herminiogg@gmail.com

Abstract—**Shape Expressions have recently been proposed as a high-level language to intuitively describe and validate the topology of RDF graphs. Current implementations of Shape Expressions are focused on checking which nodes of certain graph fit in which defined schemata, in order to get automatic typings or to improve RDF data quality in terms of completion and consistency. We intend to reverse this process, i.e., we propose to study the neighborhood of graph nodes that have already been typed in order to induce templates in which most of the individuals fit. This will allow to discover latent schemata of existing graphs, which can be used as a guideline for introducing coherent information in existing structures or for quality verification purposes. We consider that collaborative or general-purpose graphs are a specially interesting domain to apply this idea.**

Keywords–Inference, Graph schemata, Shape Expressions, RDF

I. Introduction

When tackling the task of adding knowledge to an existing RDF (Resource Description Framework) graph it is necessary to know the current topology of the data in order to be consistent with the ontologies already used. The success of this work is directly linked to the degree of coherence, completion and documentation of the targeted graph. Ontologies define the meaning and correct use for each class or property in terms of domain and range, but they are not able to declare how they should be combined in a concrete use in which a node is implementing several roles at a time or offering partial information. In some other contexts, such as XML world, several syntaxes, including RelaxNG[1] and XML Schema[2], cover those needs. Nowadays, there is not a standard syntax equivalent in RDF. However, there are some approaches under development, such as ShEx (Shape Expressions)[3].

Due to that lack of syntax to define how the neighborhood of specific nodes should look like, it is usual to use some SPARQL queries against certain key entities in order to get an approximate idea of local graph shapes and used ontologies. If we are manipulating small structures, maybe oriented to a very specific field of knowledge and possibly created by automatic processes, we may need few example queries. On the other hand, in cases of general-purpose collaborative graphs, finding correct and universal interfaces for certain type of data may be tricky and hard.

We can illustrate this idea using real examples extracted form DBpedia [4]. Precisely, we are going to check how the fact “Barack Obama and John F. Kennedy have studied at Harvard University” is represented. At the moment this paper is being written, Obama’s URI in the DBpdia is linked to Harvard’s one with the triple of Figure 1. John F. Kennedy is also linked with this node, but using a different property. In addition, the very same reality is expressed with the triples shown in Figure 2.

```
dbr:Harvard_people    dbp:name    dbr:B_Obama
```

Figure 1: Links of B. Obama with Harvard

```
dbr:Harvard_people    dbp:leadfigures    dbr:JFK
dbr:JFK   dbo:almaMater    dbr:Harvard
dbr:JFK    dbp:almaMater    ”Harvard”@en
dbr:JFK    dct:subject    dbc:Harvard_alumni
dbr:JFK     rdf:type    yago:HarvardAlumni
```

Figure 2: Links of JFK with Harvard

The information is actually contained in the graph. However, since the same notion has been expressed using too many different ways, it looks hard to design a single SPARQL query for tracking all those individuals that have studied at Harvard.

Our hypothesis is that it is possible to analyze the neighborhood of certain nodes that fit in a condition or few simple conditions, such as a link “dbo:profession dbr:Politician”, in order to detect a schema shared by all these nodes. With this, we could obtain latent topologies with certain degree of trustworthiness, that would be helpful for:

- Documentation: guideline to introduce new content.
- Verification of quality: the process of inferring an schema may produce a clear result with a high level of trustworthiness, that would be synonym of a highly coherent graph, or vice versa. Also, once a schema has been human-reviewed and accepted, it can be used to detect errors or inconsistencies across already typed entities.
- Discovering hidden entities: we may find nodes that perfectly fit in a defined shape but are not appropriately typed, which can make them “hidden” to certain SPARQL queries.

We think that our proposal can be applied to any kind of

```
<PoliticianShape> {
    foaf:name   xsd:string ,
    dbr:almaMater   @<UniversityShape>?,
    owl:sameAs    @<PoliticianShape>*
}
```

Figure 3: Politician Shape

graph, but would have special interest in collaborative, general-purpose graphs such as DBpedia or Wikidata [5]. These initiatives are thought to be a massive store of information, growing in unexpected directions with contributions from the community. Because of that, it may be hard to design an expected schema for every possible type of entity. In such structures, the schemata is not planned; there are latent and hidden forms that just emerge with community tendencies and self-moderation. Guiding users' efforts with induced graph topology based on their own actions can be a powerful tool to improve data quality of collaborative graphs.

In section II we will dig into ShEx syntax and possibilities. We will use section III to discuss some approaches for the task of schemata induction. In section IV we will explain the special interest of collaborative graphs. Finally, in section V we present the conclusions of our work.

II. ShEx to express graph topology

There are several proposals under development to describe constraints for RDF graphs topology. We are considering ShEx [3] and we may also consider SHACL (Shapes Constraint Language)[6]. Although they cover similar issues, we are planning to work with ShEx instead of SHACL because it presents a more readable, human-friendly syntax, it offers support for recursive or cyclic data models and it is more grammar-oriented. On the other hand, SHACL follows a more constraint-oriented approach. Nevertheless, core SHACL could also be a valid candidate for this task once its definition is more stable. A ShEx schema is composed of several expressions, called shapes, that specify which are the expected relations that a node of certain type (class) should include. ShEx has already been employed for documentation purposes [7], and some implementations for quality verification against defined shapes have been provided [8], [9].

If we come back to the example of USA presidents and we assume that the most usual way to link a politician with his university is the use of "dbo:almaMater", the resulting shape would look like the one in Figure 3. In order to provide some extra examples of ShEx expressibility, we have made some other assumptions: politician nodes use to have a name specified through "foaf:name" and they are linked to an unbound number of equivalent DBpedia entries of type politician through "owl:sameAs".

III. Technical discussion

XML shares some distinguishing features with RDF. Both of them can be employed to define data structures (tree-like in case of XML and graph-like in the case of RDF) with an unbounded number of possible node types. The XML community has already faced the described issues of schema specification, inference and verification. Syntaxes such as RelaxNg [1] and XML Schema [2] are handy to define the expected form and constraints of an XML document. At the same time, there are several tools that effectively check if a certain document fits in a given schema. ShEx syntax and their implementations have been thought to cover those needs for RDF and so, they could be applied in the same scenarios.

The problem of inferring a latent schema for an XML document and expressing it in some of the mentioned syntaxes has been studied in the past decade [10]. RDF world is yet a step back in that sense, since both ShEx and SHACL are recent proposals. However, the problem of exploring RDF graphs in order to induce latent or hidden structures is not new. Several works in the last years have provided techniques and frameworks that are able to find commonly used ontology elements across big RDF datasets, to discover logical axioms for type inference or even to induce common shapes of a class or type of element.

In [11], a framework for ontology learning is presented. This approach uses mining graph algorithms and machine learning techniques to extract, among other notions, which are the core or most usual properties associated with a certain class. Their main goal is to integrate ontologies of several datasets in order to find shared core elements.

In [12], an approach to extract graph schemata from large RDF datasets is presented. Association rule mining is used to induce non trivial axioms of logical descriptions relative to TBox (terminological box) knowledge. Those axioms are expressed with the EL profile of the Web Ontology Language OWL 2, which is based on the description logic $\mathcal{EL}^{++}$ [13]. Through this, the authors are able to extract graph schemata at ontology-level in a fully automatic manner.

In [14] a framework to discover common properties in clusters of individuals of an RDF graph is described. Each cluster, in an ideal situation, is identified with a class. The clusters are explored in order to detect properties widely used, which allows to elaborate descriptions of the clusters themselves and to detect domain and range restrictions when linking two instances of different classes in a general schema. This approach shares with ours the fact that is more class-centered (aka shape-centered) instead of ontology-centered. However, the results obtained are expressed in an ad-hoc syntax, less expressive compared with ShEx.

At this stage, we think we need further investigation in order to decide which are the techniques that may work better to achieve our goals. Several challenges will be faced, some of which linked to the targeted source, including graph size, adaptation to data model or noise management. However, we consider that the mentioned work proves that it is feasible to induce latent structures in RDF datasets, even when dealing with huge graphs such as DBpedia. The techniques that they employ, including association rule mining or instance clustering, may be appropriate approaches to cover most of our requirements for schema inference.

IV. Special case of collaborative graphs

General purpose and collaborative graphs are study cases where this proposal could be specially well exploited. Since they grow with unpredictable community contributions, the latent schemata may also vary in time depending on the users' agreement on the use of certain properties. Trying to limit the possible links between nodes by forcing them to fit in safe inferred shapes may be a wrong idea since it cuts the freedom philosophy that underlies this kind of initiatives. However, ShEx can be useful as a mechanism to guide this evolution.

In addition, the changeable and entropic nature of these graphs generates scenarios that support hypothesis which may make less sense in more constrained contexts. From a purist point of view, it may be desirable to obtain non-overlapped shapes of each existing class. For instance, a priori, it looks obvious that the shape of graduate should tell how to properly establish a relation between a person and his alma mater. Meanwhile, the shape of politician may indicate how to link someone to a political party. With this, if a user wants to add information about an entity that implements both roles at a time, such as B. Obama, he should look for two different shapes in order to discover the appropriate way to express these two pieces of information. Because of that, it may be interesting to discover "which information is associated in this context to entities of certain type" instead of "which information must be necessarily associated to a certain type". It could happen that most of the politicians have higher education. If a common property used to link politicians and universities is discovered and appears in the latent schema of the shape politician, the user who wants to add studies to certain politician would not need to query different shapes.

It even may be feasible to elaborate schema inference on users' demand to obtain a view of the state of a shape in nearly real-time. This could be done analyzing a representative set of entities of certain type. A periodical checking of the inferred schema, or an automatic update triggered by a significant number of modifications/additions would also reflect the nature of these graphs.

V. Conclusions

We propose to apply automatic schema inference over existing RDF graphs in order to discover latent structures. Our aim is to create automatic graph documentation and to provide the basis for a tool able to check data completion and coherence using ShEx syntax.

We consider collaborative general-purpose graphs, such as DBpedia or Wikidata, an specially interesting scenario to apply this idea, since it is hardly possible to design graph shapes a priori. The schemata just emerge and evolve with the community's efforts.

References

[1] E. van der Vlist, Relax NG: A Simpler Schema Language for XML. Beijing: O'Reilly, 2004.

[2] S. Gao, C. M. Sperberg-McQueen, H. S. Thompson, N. Mendelsohn, D. Beech, and M. Maloney, "W3c xml schema definition language (xsd) 1.1 part 1: Structures," W3C Candidate Recommendation, vol. 30, no. 7.2, 2009.

[3] E. Prud'hommeaux, J. E. Labra Gayo, and H. Solbrig, "Shape expressions: an rdf validation and transformation language," in Proceedings of the 10th International Conference on Semantic Systems. ACM, 2014, pp. 32–40.

[4] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and Others, "DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia," Semantic Web, vol. 6, no. 2, 2015, pp. 167–195.

[5] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledgebase," Communications of the ACM, vol. 57, no. 10, 2014, pp. 78–85.

[6] A. Ryman, "Z specification for the w3c editor's draft core shacl semantics," arXiv preprint arXiv:1511.00384, 2015.

[7] J. E. L. Gayo, E. Prud'hommeaux, H. R. Solbrig, and J. M. Á. Rodríguez, "Validating and describing linked data portals using rdf shape expressions." in LDQ@ SEMANTICS. Citeseer, 2014.

[8] https://www.w3.org/2013/ShEx/FancyShExDemo, accessed: 2016-01-10.

[9] http://rdfshape.weso.es, accessed: 2016-01-10.

[10] G. J. Bex, F. Neven, and S. Vansummeren, "Inferring xml schema definitions from xml data," in Proceedings of the 33rd international conference on Very large data bases. VLDB Endowment, 2007, pp. 998–1009.

[11] L. Zhao and R. Ichise, "Instance-based ontological knowledge acquisition," in The Semantic Web: Semantics and Big Data. Springer, 2013, pp. 155–169.

[12] J. Völker and M. Niepert, "Statistical schema induction," in Extended Semantic Web Conference. Springer, 2011, pp. 124–138.

[13] F. Baader, S. Brandt, and C. Lutz, "Pushing the el envelope," in IJCAI, vol. 5, 2005, pp. 364–369.

[14] K. Christodoulou, N. W. Paton, and A. A. Fernandes, "Structure inference for linked data sources using clustering," in Transactions on Large-Scale Data-and Knowledge-Centered Systems XIX. Springer, 2015, pp. 1–25.