VizCurator: a Visual Tool for Curating Open Data

Bahar Ghadiri Bashardoost
University of Toronto
ghadiri@cs.toronto.edu

Christina Christodoulakis
University of Toronto
christina@cs.toronto.edu

Soheil Hassas Yeganeh
University of Toronto
soheil@cs.toronto.edu

Oktie Hassanzadeh
IBM Research
hassanzadeh@us.ibm.com

Renée J. Miller
University of Toronto
miller@cs.toronto.edu

Kelly Lyons
University of Toronto
klyons@cs.toronto.edu

ABSTRACT

VizCurator permits the exploration, understanding and curation of open RDF data, its schema, and how it has been linked to other sources. We provide visualizations that enable one to seamlessly navigate through RDFS and RDF layers and quickly understand the open data, how it has been mapped or linked, how it has been structured (and could be restructured), and how deeply it has been related to other open data sources. More importantly, VizCurator provides a rich set of tools for data curation. It suggests possible improvements to the structure of the data and enables curators to make informed decisions about enhancements to the exploration and exploitation of the data. Moreover, VizCurator facilitates the mining of temporal resources and the definition of temporal constraints through which the curator can identify conflicting facts. Finally, VizCurator can be used to create new binary temporal relations by reifying base facts and linking them to temporal resources. We will demonstrate VizCurator using LinkedCT.org, a five-star open data set mapped from the XML NIH clinical trials data (clinicaltrials.gov) that we have been maintaining and curating for several years.

Categories and Subject Descriptors

H.5.0 [**Information Interfaces and Presentation**]: General; H.2.8 [**Database Management**]: Database Applications

Keywords

Linked Open Data, Data Curation, RDF, Visualization

1. INTRODUCTION

Organizations and individuals see tremendous value in publishing data as open linked data. There are a plethora of open data publishing tools that help create RDF data from relational data [10], semi-structured data (such as XML, JSON, or XBRL) [12], and even unstructured data [11]. Many data browsers help in visualizing and exploring the resulting RDF data [5, 7] and link discovery tools help in creating rich semantic links between open data sources [9].

With the growing quantity and complexity of open RDF data, there is a increasing need for tools to help data scientists assess the quality of this data and ensure the data maintains its quality and value over time, a process referred to as data curation [4]. VizCurator provides a suite of visualization and curation tools to help data curators inspect and explore their open data, understand how it has been structured, and how it has been linked. The unstructured, dynamic and heterogeneous nature of open data makes it necessary for a curator to be able to understand (and modify) the data's structure and to understand (and modify) how it has been linked.

VizCurator brings together some well-known, effective visualization techniques for RDF schema and data. It augments these with new curation services that can be invoked using clear visual cues. VizCurator is useful for any RDF data source but especially for data that has been translated or extracted from another (structured or semi-structured) format where the data may not match its schema well. Our demonstration will illustrate the following contributions.

- Large Scale Schema Understanding. In RDF, the schema may be as large as the data, so traditional schema browsing may not be effective. Automated link discovery can add to the visual clutter by creating many low-quality links to external sources. VizCurator provides new visualizations that help a curator understand what schema elements (entity types and relation types) may be of interest, in part by highlighting which are consistently or richly populated.
- Schema Refinement. Many data publishing tools create a default RDF Schema that may not fit the data or a given task for which the data is going to be used. Usually, data translated from a relational source contain complex N-ary relations that make RDF querying complicated. VizCurator provides new visualizations (and visual cues) to help a curator understand (and refine) the schema. For example, with VizCurator, users can browse N-ary relations and break down overly complicated or unintuitive relations into binary relations to simplify querying.
- Temporal Semantics. Given that the goal of curation is to maintain the value of data over time, VizCurator also helps a curator understand the temporal characteristics of data. The tool suggests when facts might be temporal (based on patterns in related data) and facilitates the definition of temporal constraints. Few data publish-

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2742845.

ing tools create temporal semantics, so this facility is an important new contribution of VizCurator.

Demo. We demonstrate VizCurator using a knowledge base (KB) of international clinical drug trials, LinkedCT [8], that we have been curating for several years and that is generated and maintained using xCurator [12], a framework for assisting data curators in transforming semi-structured data into linked open data. A schema is derived (automatically) from the semi-structured data, which is translated and published as RDF data. This translation creates relationships between entities (intra-linking entities within a single data source), and also automatically inter-links entities to external KBs such as Freebase[1] and OpenCyc[2]. Here, we present several use cases for how a curator might employ VizCurator to curate and improve (re-curate) a linked KB. Note that VizCurator can be used to curate any KB that uses the standard RDFS guidelines.

2. DISCOVERY VIA VISUALIZATION

When approaching a new data set or reviewing a newly generated open data set, curators need tools to help them understand the data and its quality (how complete it is, how well it has been linked to other data sources, and how consistently it is structured). Schemas are the primary abstraction for understanding data. But with open data, the schema may not match the data well and may not be the best structure for querying the information.

In order to understand and review the schema, curators often need to 1) interact with the RDF schema, 2) understand how the data has been linked (both intra-links between entities in the data set and inter-links to resources outside this data set), and 3) select or specify a desired resource as a starting point from which they can explore and discover both the schema and its underlying data. Below, we describe a carefully-selected set of interactive visualizations available through VizCurator to help a curator browse the schema and understand how well it matches the data.

2.1 Basic Schema Browsing

The most basic function of VizCurator enables curators to interact with the actual structure of a KB's schema. Similar to other RDF browsers and editors [6], we employ tree and node-link diagrams (most implemented using D^3 - Data Driven Documents [3]). These visualizations are familiar and work well for providing information about the general structure of the schema.

We provide a color-coded tree-based browser, which allows curators to navigate the RDF schema and browse the entity types, relations, and external links. In the first level of this tree, the curator can see all the entity types and each entity type can be expanded to provide more detailed, schema-related information: 1) literal relations, 2) non-literal relations, and 3) external links (relations with external KBs).

Sometimes curators are interested in a specific category of generated relations such as literal relations. In VizCurator, color coding is used to assist curators in quickly identifying three categories of relations. Nodes with a red component (*e.g.*, red, orange, purple, or black) indicate entity types with literal relations. Blue and yellow color components indicate non-literal relations and external links, respectively.

Curators are sometimes interested in the structural properties of the generated schema not the details of entity types. For example, they may want to understand how normalized the schema is (that is, how much redundancy there is in the data). VizCurator presents curators with a high-level graph representing entity types and their relationships in the form of a force-directed node-link network (Figure 1A).

In this network, each node represents a resource which can be either an entity type or a literal, and each edge represents a relation. To provide more insight into the structural properties, the weights of internal relations are proportional to the number of facts of the relations and the weights of the nodes are proportional to the number of incident relations. External links are given the maximum weight so that the resulting thicker edges are easily identifiable. Moreover, literals and external resources are represented as nodes with a degree of one.

Semi-structured data is often structured as a tree with substantial redundancy in data. For instance, consider a data source in which both `company` and `employee` entities include an `address`. Having a node-link network in the form of a tree indicates that the mapping to RDF has created two separate entity types for addresses (and perhaps redundantly created multiple nodes for the same address). However, curation tools such as xCurator, try to eliminate this redundancy by intra-linking the data (also called *deduplication*) when creating a mapping to RDF. The force-directed graph can help a curator understand possible redundancy in the data and the nature of deduplication in the mapping process.

2.2 Large Scale Schema Browsing

Although node-link networks are common and natural choices to visualize linked data, they can overwhelm the user with visual clutter for complex, large-scale data sets. One way to overcome this problem is to focus on high-level relationships among resources. To do this, we use edge-bundling graphs to reduce visual clutter. The Relation-mapping view of VizCurator (Figure 1B) is an edge bundling graph that depicts entities and their relations from another perspective. In a KB where each entity has a type, one can say that each binary relation has a type signature [11]. For example, for a relation `works_in` we can say that `works_in` $\subseteq$ `Person` $\times$ `Company`. In other words, the `works_in` relation has the domain `Person` and the range `Company`. In the Relation-mapping view, each edge represents a relation. Two different colors are used to distinguish between the domain and the range of a relation. Hovering on the domain/range of a relation will make the relation's edge red/green, respectively. Since we use edge bundling, the entity types which are participating in more relations have thicker edges.

This graph is very important especially when the KB is generated from a relational database. Relational databases are normalized to reduce redundancy which makes the domain of the facts of the generated KB limited. For example, in Figure 1B the `clinical_study` domain has the most relations because most facts in our KB are about clinical studies. This might not be desirable and the curator might want to add more relations to help users query the KB more easily. In Section 3.1, we show how VizCurator can help the curator re-curate the KB to address this problem.

The curator can inspect external links using the Inter-linkage view (Figure 1C). The Inter-linkage view utilizes a

[1] https://www.freebase.com/

[2] http://sw.opencyc.org/

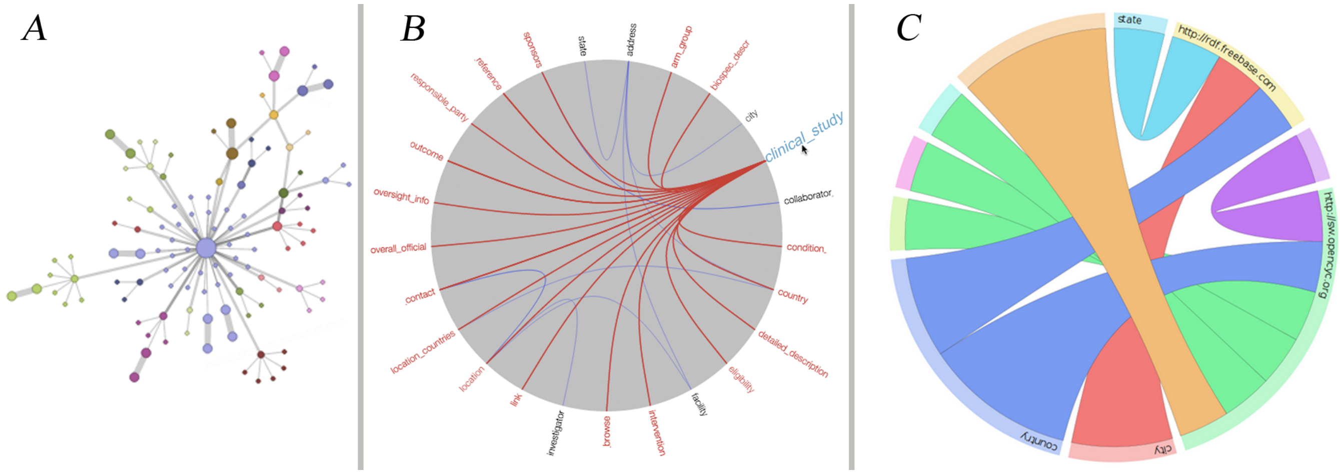

Figure 1: A: RDF schema as a node-link network; B: Relation-mapping view; C: Inter-linkage view

chord diagram to depict all the links to the external KBs, and the number of the links from an internal entity to an external KB. Using this view, the curator can selectively unlink generated linkages. This capability is important in helping a curator restrict access to trusted data sources for privacy or for performance.

2.3 Seamless Connection to Data Browsing

The need for data quality visualization in open linked data has been a recent source of interest. One requirement to address this is the ability to switch easily and seamlessly from exploring the RDF schema to viewing representative samples of the underlying data [2]. Using LodLive, a framework for exploring linked open data [5], within VizCurator enables a curator to seamlessly navigate through the KB at different levels of granularity (data level and schema level). More detailed information about the underlying data can be explored while navigating the schema tree in VizCurator by clicking on the desired resource to open a LodLive view. The LodLive view can also be accessed directly. If the curator is interested in a particular resource, she may specify the URI in the LodLive tab of VizCurator and start exploring from that point.

3. CURATION VIA VISUALIZATION

Open data publishing tools such as xCurator or RDB2-RDF tools map semi-structured or structured data to RDF. Often these mappings are straight forward, for example mapping each relational or XML attribute to a single entity type (or attribute) in RDF. Although these mapping rules can be useful for transforming 3-star data into 5-star data, often further curation is needed. Below we illustrate through use cases how VizCurator can be used to re-curate the schema and underlying data in KBs that are created using such mappings.

3.1 Extracting Binary from N-ary Relations

Bast et al. [1] consider complex N-ary relations as one of the major usability issues when working with a linked KB. These relations are necessary when one wants to depict a relation among more than two resources and are typical when the KB is automatically generated from a relational model. For example in our dataset, the generated schema has a central entity type called `clinical_study` and nearly eighty percent of the other entity types are only in relation with this entity type. Thus, to find clinical conditions which are investigated in the United States, a curator needs to first find the clinical studies which were located in United States and then among those find the conditions. Although VizCurator can help in understanding the schema by visualizing different aspects of it, for complex N-ary relations, more sophisticated visualizations may be needed. To address this problem, VizCurator helps the curator to link the most important entities together so that most of the important facts can be accessed using a single statement like:

```
select ?condition where {
    ?condition conditon_location "United States"}
```

In VizCurator's schema tree view, if an entity type is a hub in an N-ary relation a star will appear besides its name. By clicking on that star, the curator can view a heat map (Figure 2) in which, the rows and the columns represent different relations that have the hub entity type as their subject. Each cell represents a count of the entities in a relation. This intuitively means that the relations that have more facts are more important and these facts are more likely to be asked. Bast et al. [1] suggest that usually the two top frequent relations in a multiway relation should be merged. We have adapted their approach with the difference that we let the curator make an informed decision about which relations should be joined using a heat map visualization. If the curator decides to join two entities by creating a new relation, he/she can simply click on the cell. A dialogue will open and ask for the name of the new relation. After the curator enters the name `p`, the tool will link the data and create a new set of RDF triples such that:

$$\{\,\texttt{(s,p,o)}\,|\,(\,\texttt{(y,row,s)}\in \mathrm{KB} \wedge \texttt{(y,rdf:type,hub-entity)}\in \mathrm{KB}\,) \wedge (\,\texttt{(y,column,o)}\in \mathrm{KB} \wedge \texttt{(y,rdf:type,hub-entity)}\in \mathrm{KB}\,)\,\}.$$

The curator can create multiple binary relations per N-ary relation if desired.

3.2 Temporal Semantics

Time-dependent relations are an important part of shared curated KBs [11]. Thus, curators must be equipped with tools that help them manage time evolving facts. VizCurator can help a curator define time constraints for different relations, curate conflicting facts, and create new temporal relations in order to better describe a temporal statement.

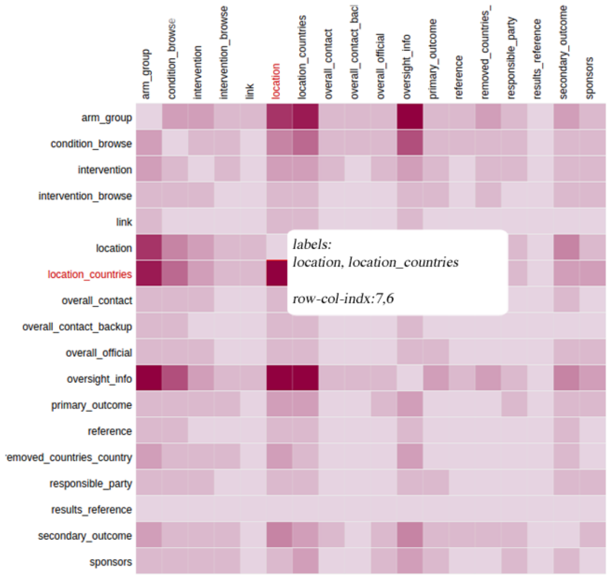

Figure 2: Heat map: darker cells are more populated

For each type in a KB, VizCurator selects types that use dates and potential temporal types. A potential temporal type is shown with a clock icon besides its name in the schema tree view. The curator can set a constraint by right clicking on a property of an entity type in the schema tree view. For example, for the `Completion_date` property of the `clinical_study` one can define a constraint: `UNIQUE AND AFTER http://www.linkedct.org/0.1#start_date`. The conflicting facts will be shown as warnings.

Sometimes a temporal relation is needed to better describe facts. For example, in our dataset each `clinical_study` has an `overall_status` that can be recruiting, active, etc. It is very useful to know approximately how long a `clinical_study` is in a specific state because a push for an update can be triggered if a fact stays in a state longer than a typical average time. VizCurator allows the curator to create new temporal relations and assign them to the facts by right clicking on a property of an entity type in the schema tree. A dialogue will open and ask for the name of the temporal property which can be one of `since, onDate, until`. The curator can set the object of that property to be either Null or a default value. VizCurator will then automatically create the necessary resources both in the RDFS and RDF layers. In order to do so, we adopted T-YAGO's approach [11] where we first reify the base facts and give them a new URI, and then use that URI in the new relation.

4. DEMONSTRATION PLAN

We will invite the audience to explore LinkedCT while highlighting different aspects of VizCurator.

Exploring the LinkedCT KB: The audience will be able to explore and discover the LinkedCT KB both in RDFS and RDF levels. As they explore, they can view how complex the structure of a trial is and what parts are used sparsely and what parts are coherent and used in almost all trials. They can drill down to components, such as sponsoring agencies and see the heterogeneity in the way information about these agencies is structured. Or, they can have a bird's-eye view of LinkedCT and inspect the linkage points between LinkedCT and other external KBs.

Identifying possible problems in the KB: To make things even more interesting, we will ask the audience to use VizCurator's cues to identify curation actions that can improve the structure of LinkedCT. These actions include finding N-ary relations and picking out the best binary candidates to be extracted from those relations, finding temporal resources, and inspecting the external links.

Re-curating the KB: The audience can create new binary relations from the N-ary relations and inspect the changes using VizCurator. Also, in this step, the audience will define temporal rules and find the conflicting triples.

5. ACKNOWLEDGMENTS

This research was partially funded by the GRAND NCE as part of the MEOW project, and by NSERC BIN.

6. REFERENCES

[1] H. Bast, F. Bäurle, B. Buchhold, and E. Haußmann. Easy Access to the Freebase Dataset. In *WWW*, WWW Companion '14, pages 95–98. International World Wide Web Conferences Steering Committee, 2014.

[2] C. Bizer, P. Boncz, M. L. Brodie, and O. Erling. The Meaningful Use of Big Data: Four Perspectives – Four Challenges. *SIGMOD Rec.*, 40(4):56–60, Jan. 2012.

[3] M. Bostock, V. Ogievetsky, and J. Heer. D^3 Data -Driven Documents. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2301–2309, 2011.

[4] P. Buneman, J. Cheney, W. C. Tan, and S. Vansummeren. Curated Databases. In *PODS*, PODS '08, pages 1–12. ACM, 2008.

[5] D. V. Camarda, S. Mazzini, and A. Antonuccio. LodLive, Exploring the Web of Data. In *I-SEMANTICS*, pages 197–200, 2012.

[6] A.-S. Dadzie and M. Rowe. Approaches to Visualising Linked Data: A Survey. *Semantic Web*, 2(2):89–124, 2011.

[7] V. Geroimenko and C. Chen. *Visualizing the Semantic Web: XML-based Internet and Information Visualization.* Springer, 2006.

[8] O. Hassanzadeh, A. Kementsietsidis, L. Lim, R. J. Miller, and M. Wang. LinkedCT: A Linked Data Space for Clinical Trials. *CoRR*, abs/0908.0567, 2009.

[9] A. Jentzsch, R. Isele, and C. Bizer. Silk - Generating RDF Links while Publishing or Consuming Linked Data. In *ISWC'10.* Citeseer, 2010.

[10] S. S. Sahoo, W. Halb, S. Hellmann, K. Idehen, T. Thibodeau Jr, S. Auer, J. Sequeda, and A. Ezzat. A Survey of Current Approaches for Mapping of Relational Databases to RDF. *W3C RDB2RDF Incubator Group Report*, 2009.

[11] G. Weikum and M. Theobald. From Information to Knowledge: Harvesting Entities and Relationships From Web Sources. In *PODS*, PODS '10, pages 65–76. ACM, 2010.

[12] S. H. Yeganeh, O. Hassanzadeh, and R. J. Miller. Linking Semistructured Data on the Web. In *ACM SIGMOD (WebDB Workshop 2011)*, 2011.