openChart: Charting Quantitative Properties in LOD

Filip Zembowicz
Harvard University
414 Quincy Mailing Center
Cambridge, MA 02138
fzembow@fas.harvard.edu

David Opolon
MIT ESD
77 Massachusetts Avenue, E40-286
Cambridge, MA 02139
opolon@alum.mit.edu

Stephen Miles
MIT AutoID Labs
77 Massachusetts Avenue, 35-014
Cambridge, MA 02139
s_miles@mit.edu

ABSTRACT

In this paper, we discuss the development of openChart, a quantitative Linked Open Data charting tool. It targets novice semantic web users by generating SPARQL queries to present interesting information. We also acknowledge the problems encountered in development and suggest improvements.

Categories and Subject Descriptors

H.3.4 [**Semantic Web**]: Visualization, charting, search.

General Terms

Measurement, Documentation, Design, Human Factors

Keywords

Linked Open Data, Visualization, Charting

1. INTRODUCTION

The wealth of information in the Linked Open Data cloud (hereafter *LOD*) is large and growing, enabling comparisons between previously isolated datasets to be made [3]. However, exploring the linked data cloud is difficult for users unfamiliar with semantic web concepts such as SPARQL and RDF. Web-based visualization tools such as IBM's Many Eyes have shown promise in allowing collaborative data exploration [7]. We have developed a tool that allows users to similarly plot quantitative data found on the LOD cloud, with minimal knowledge of semantic web syntax. This tool enables users to explore, share, and expand upon the data found in the LOD cloud.

2. STRUCTURE

Finding data on the LOD cloud using openChart[1] consists of identifying an entity of interest, choosing two of its quantitative properties, and selecting a peer group with which to compare values. To enable entry into the semantic web, we use Wikipedia's autosuggest API to determine an entity's Wikipedia address. This is then matched with a corresponding semantic web resource using a SPARQL query on the DBpedia database, which is a central hub of the LOD cloud with many *owl:sameAs* linkages to other sources of data [2]. While other endpoints could be used with the openChart framework, DBpedia has a high number of links to other LOD sources, making it useful for a general purpose tool.

Following the identification of an entity of interest, for example *Bangladesh*, we find the quantitative properties from the RDF resource by using regular expressions to remove non-quantitative information. Two of these are selected by the user, for example *hdi* and *population density*. Then, peer groups are found through a SPARQL query that looks for distinct *rdfs:type* that contains objects with both of the quantitative properties. These peer groups may or may not contain the users' original search term—but the selection of one, for example *Country*, will display a scatter-plot of the two variables. This peer group feature is an important aspect of our application because it allows the user to branch out when navigating information on the semantic web rather than fixate on answering one question in particular.

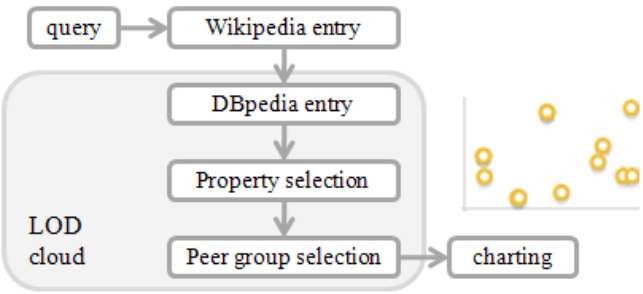

Figure 1. The openChart workflow

At all levels of the exploration, the data is locally cached in a MySQL server. The frontend is written using JavaScript and the jQuery library, while the backend is written in PHP 5 and the ARC2 library. The plotting is done using the Protovis JavaScript

3. RESULTS

3.1 Easy Exploration with Structured Queries

A focus on a simple user interface has made openChart an easy to use introduction for WWW users unfamiliar with Linked Open Data. By focusing on peer groups and not only information directly relevant to a users' query, the openChart tool emphasizes a broad exploration of available data rather than merely answering a specific question. Additionally, we incorporate a social component into openChart, where interesting relationships between concepts can be shared. This is new knowledge that is being created, and eventually will be integrated into the LOD cloud itself by defining such shared charts as RDF objects.

3.2 Identification of Errors in the Data[2]

An additional benefit of displaying data visually in openChart is the ability to quickly identify errors within the data contained in the LOD cloud. In isolation, it is often difficult to see errors in

LDOW2010, April 27, 2010, Raleigh, North Carolina, USA.

[1] The demonstration can be found at *http://openchart.mit.edu*

[2] An example may be seen: *http://hcs.harvard.edu/datavis/linkeddata/gallery/index.php?chart=19*

scale or other such mistakes—displaying them as outliers enables mistakes to be rapidly identified. These data points can then be flagged for review in order to improve the quality of the source data, or any scripts that are used to parse the data into the RDF format in the first place. Such flagging could be achieved by defining a quality ontology and publishing triples for user identified errors.

4. PROBLEMS ENCOUNTERED

4.1 Lack of Range Descriptors

When searching the LOD cloud through a SPARQL query, it would be economical to restrict SPARQL queries to retrieve only properties with ranges limiting them to numerical values. However, we found that many of the properties lack associated *rdfs:range* and/or *rdfs:domain* values. This resulted in a need to retrieve all results and then parse them using regular expressions, increasing the overhead of the application. Thus, we suggest that RDF authors take the time to specify *rdfs:range* and *rdfs:domain* values such as *xsd:integer* and *xsd:decimal* to facilitate statistical work using Linked Open Data.

4.2 Lack of Unit Descriptors

Another aspect often missing from data sources, especially from DBPedia, is units of measure. Particularly when comparing across endpoints, it is imperative that the units of measurements are understood, in order to prevent scaling errors when comparing data from different sources. We suggest that creators of RDF data take the time to include unit specifications, either through ontologies such as Quantities, Units, Dimensions and Data Types in OWL and XML [6], or by agreeing on standardized unit abbreviations and distributing unit-aware parsers.

5. FUTURE DEVELOPMENT

5.1 Automated Provenance

Since the data in openChart is coming from multiple sources, tracking the sources of a chart's data would be important in enabling the use of the charts in research. As a result, we plan to implement a feature by which the origins of the data contained within a chart will be displayed concurrently with the chart. Although RDF quadruples (such as [4]) would allow this to be easily implemented, methods that determine authorship based on particular endpoint characteristics could be implemented currently.

5.2 Integration with Existing LOD Browsers

There exist many existing browsers of semantic web data, such as Tabulator, which offer capabilities similar to our system [1]. Although openChart is easier to use than these programs, due to the restrictive nature of the queries permitted on our system, we are working to enable the switching back and forth between Tabulator and openChart, to allow more technical users to experience the full potential of the semantic web, using openChart as a starting point.

5.3 Publishing of Results

As mentioned previously, the information gleaned from openChart can be published for others to access. Statistical relationships can be described using the SCOVO ontology, which allows the specification of statistics with reference to a particular dataset over a range of time [5]. Care must be taken to ensure the completeness of the data, however, since the statistics generated only represent the data published to the LOD cloud. Two groups of statistics are generated – one describing the local cloud itself, such as describing the number of triples, and another describing the data contained therein.

6. ACKNOWLEDGMENTS

We would like to thank Tim Berners-Lee, K. Krasnow Waterman, Reed Stuyvesant, Ian Jacobi, Oshani Seneviratne, and everyone else who participated in and organized MIT's Linked Data week in January of 2010.

7. REFERENCES

[1] Berners-Lee et. al., *Tabulator: Exploring and Analyzing linked data on the Semantic Web*, Procedings of the The 3rd International Semantic Web User Interaction Workshop (SWUI06) workshop, Athens, Georgia, 6 Nov 2006.

[2] Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. DBpedia – A Crystallization Point for the Web of Data. *Journal of Web Semantics*: Science, Services and Agents on the World Wide Web, Issue 7, Pages 154–165, 2009.

[3] Bizer, C., Heath, T., and Berners-Lee, T. Linked Data - The Story So Far (in press). International Journal on Semantic Web and Information Systems, Special Issue on Linked Data.

[4] Cyganiak, R., Harth, A., and Hogan, A. *N-Quads: Extending N-Triples with Context*. (http://sw.deri.org/2008/07/n-quads/)

[5] Hausenblas, M., Halb, W., Raimond, Y., Feigenbaum, L., and Ayers, D. *SCOVO: Using Statistics on the Web of Data*.

[6] Masters, J, Hodgson, R., and Keller, P. *Quantities, Units, Dimensions and Data Types in OWL and XML* (http://qudt.org/)

[7] Viégas, F.B., Wattenberg, M., van Ham, F., Kriss, J., and McKeon, M. Many Eyes: A Site for Visualization at Internet Scale . Infovis, 2007.