TWC Data-Gov Corpus: Incrementally Generating Linked Government Data from Data.gov

Li Ding, Dominic DiFranzo, Alvaro Graves, James R. Michaelis, Xian Li,
Deborah L. McGuinness, James A. Hendler
Tetherless World Constellation, Rensselaer Polytechnic Institute
110 8th St, Troy, NY12180, USA,
{dingl,difrad,agraves,michaj6,dlm,hendler}@cs.rpi.edu, {lix15}@rpi.edu

ABSTRACT
The Open Government Directive is making US government data available via websites such as Data.gov for public access. In this paper, we present a Semantic Web based approach that incrementally generates Linked Government Data (LGD) for the US government. In focusing on the tradeoff between high quality LGD generation (requiring non-trivial human expert input) and massive LGD generation (requiring low human processing cost), our work is highlighted by the following features: (i) supporting low-cost and extensible LGD publishing for massive government data; (ii) using Social Semantic Web (Web3.0) technologies to incrementally enhance published LGD via crowdsourcing, and (iii) facilitating mashups by declaratively reusing cross-dataset mappings which usually are hardcoded in applications.

Categories and Subject Descriptors
H.4.m [Information Systems]: Miscellaneous

Keywords
Linked Government Data, Social Semantic Web, Data.gov

1. INTRODUCTION
The Data-gov project [1] investigates the role of semantic web technologies, especially linked data, in producing, enhancing and utilizing government data published on Data.gov and other websites. Simply opening up government data on the Web does not guarantee the data will be ready for mashups. Linked Government Data (LGD) [2] is introduced to establish links across distributed government datasets and thus facilitate data integration. The Tetherless World Constellation (TWC) Data-gov corpus, yielded by the Data-gov project at RPI, publishes LGD (over 5 billions RDF triples) converted from hundreds of Data.gov datasets covering a wide range of topics (e.g. US government spending, energy usage and public healthcare).

LGD publishing is non-trivial, as massive amounts of data in reasonable quality need to be produced quickly to facilitate further mashup and integration. To address this challenge, we designed an extensible LGD publishing framework with a focus on fast minimal LGD publishing and incremental LGD enhancement via crowdsourcing. This is viable because: (i) mashups can use LGD at different levels of quality (e.g. US budget data can be used without being linked to DBpedia[3]); (ii) enhancements of LGD may be contributed by both publishers and developers (e.g. developers can contribute cross-dataset mappings which are seldom provided by the original LGD datasets); and (iii) keeping enhancement incremental allows existing LGD applications to remain unchanged when new enhancement data are added.

The rest of this paper will describe how we design an incremental framework for LGD publishing and how this framework is used in linked data applications. We conclude with discussions on the tradeoffs inherent in our design.

2. LGD PUBLISHING FRAMEWORK
We publish LGD in RDF based on a few principles. In what follows, we explain these principles with examples.

2.1 P1: Let LGD Meet the Web
We choose RDF/XML as serialization format to enable LGD consumption by Semantic Web developers (RDF + SPARQL) and traditional Web developers (XML+ XQuery).

Example RDF data: http://data-gov.tw.rpi.edu/raw/353/data-353.rdf

We adopt linked data principles for assigning dereferenceable HTTP URIs to entities (e.g. datasets and dataset entries) and documents containing entity descriptions. We also keep the URI of an entity different from the URI of the document that contains the description of the entity, e.g.

Entity URI: http://data-gov.tw.rpi.edu/vocab/Dataset_353
Document URI: http://data-gov.tw.rpi.edu/vocab.php?instance=Dataset_353

2.2 P2: Keep LGD Raw Data Minimal
We generate LGD raw data by minimizing human involvement and preserving the structure and content of the original data. For example, many US government datasets published at Data.gov are organized as data tables. We can easily map a non-header table cell to an RDF triple where the row id is the subject, the column name is the predicate, and the cell content is the object. Here, each table entry (row) has a unique URI, and each header cell (column name) yields an auto-generated RDF property. Following is an example data table extracted from the Data.gov catalog (dataset 92). A corresponding RDF triple is generated in following way: the subject URI is auto-generated to uniquely identify the table row; the property URI is based an auto-generated namespace (determined by the ID of the specific dataset) and the local name trivially normalized from the corresponding column name; and the object preserves the original cell value (most values will be stored as literal strings unless they are confirmed http URIs).

URL	Title	Agency
http://www.data.gov/details/353	State Library Agency Survey: Fiscal Year 2006	Institute of Museum and Library Services

```
@prefix dgp92:  http://data-gov.tw.rpi.edu/vocab/p/92/
 <http://data-gov.tw.rpi.edu/raw/92/data-92#entry_00002>
   dgp92:url  <http://www.data.gov/details/353>;
   dgp92:agency “Institute of Museum and Library Services” .
```

2.3 P3: Keep LGD Editable and Extensible

The LGD raw data is fairly primitive and limited, so we build a Social Semantic Web (aka. Web 3.0) platform to help users collaboratively contribute LGD enhancements to the data. Currently, we adopt Semantic MediaWiki (SMW) to implement this platform.

Maintain extensible dataset description. Each LGD dataset has an editable wiki-page that links to the LGD raw data and the used properties. The page can be enriched by adding links to later generated LGD enhancement data.

```
dgv:Dataset_403
  dgtwc:complete_data  http://data-gov.tw.rpi.edu/raw/403/data-403.rdf ;
  dgtwc:more_data   http://data-gov.tw.rpi.edu/linked/401/agency_401.rdf;
  dgtwc:more_data   http://data-gov.tw.rpi.edu/linked/us_agencies.rdf.
```

Maintain extensible property definition. Every property created from a table column name can be dereferenced to the RDF version of a wiki page, and users can add the definition of the property (e.g. the triple below) directly through SMW editing interface.

```
[[rdfs:subPropertyOf::rdfs:label]]        -- SMW syntax on dgp92:title's wiki page
dgp92:title rdfs:subPropertyOf  rdfs:label .    -- the corresponding RDF triple
```

Maintain extensible cross-dataset mappings. Each named entity is maintained by a wiki page. In the LGD raw data, different string labels may refer the same entity. For example, Washington DC was referred by "DC", "Dist. of Col.", and "District of Columbia" in different Data.gov datasets, so we preserve such labels using skos:altLabel. These labels, if used within certain context, can uniquely identify entities; therefore, developers can use the labels to establish connections across different datasets. Meanwhile, users can manually contribute links to DBpedia via owl:sameAs statements and verify the statement by checking whether the corresponding Wikipedia page has been correctly embedded (see example below). Users can further annotate well-known types of names such as abbreviations and official names, to support screen display. Note that all well-known types of name are automatically considered as skos:altLabel of the entity.

```
Dgv:Washington,_D.C.
   dgtwc:abbreviation "DC" ;
   skos:altLabel  "Washington, D.C.", "DC", "Dist. Of Col.", "District of Columbia";
   owl:sameAs  dbpedia:EPA.
```

As shown in Figure 1, the RDF and HTML versions of the mappings can be synchronized and editable online using our platform; therefore, developers can curate and reuse declarative mappings online instead of hard-code data within their programs.

Figure1. Web 3.0 interface for curating and correcting dbpedia links

2.4 P4: Make Enhancement Data Incremental

In order to better understand and use LGD raw data, we need to improve data quality by using additional human and machine processes. We recommend users to generate LGD enhancement data based on the existing LGD raw data, and reuse data entry URIs from LGD raw data. For example, we can use the skos:altLabel data from the empirical cross-dataset mappings to generate LGD enhancement data by linking from LGD raw data entries to DBpedia URIs. In the SPARQL query below, state entries are retrieved from dataset 353 (State Library Agency Survey: Fiscal Year 2006, from data.gov) and dataset 10011 (user contributed linked data for US states and territories) while the linking is accomplished by string matching.

```
prefix dgp353: <http://data-gov.tw.rpi.edu/vocab/p/353/>
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix skos: <http://www.w3.org/2004/02/skos/core#>
construct {   ?s  dgp353:phys_st_link ?state_uri_dbpedia. }
from <http://data-gov.tw.rpi.edu/raw/353/data-353.rdf>
from <http://data-gov.tw.rpi.edu/wikidata/United_States_and_Territories>
where {
  {     ?s dgp353:phys_st ?state_abbrev .    }
  {      ?s1 skos:altLabel ?label .
         ?s1 owl:sameAs ?state_uri_dbpedia .
        filter ( regex(str(?state_uri_dbpedia),"http://dbpedia.org/","i" ) )    }
  filter ( str(?label) = str(?state_abbrev) ) }
```

(SPARQL endpoint at http://data-gov.tw.rpi.edu/joseki/sparql.html)

3. Consuming the TWC Data-Gov Corpus

In what follows, we show that linked data applications can be built using different amounts of links in LGD data.

3.1 Using LGD Raw Data Only

With just the LGD raw data converted from Data.gov, we built mashups that links data via existing Web data APIs to better exhibit the raw data, including the following applications:

Worldwide Earthquakes[1]. This uses raw data from the Department of Interior to get the latitude, longitude and descriptions about earthquakes observed in the past seven days. The raw data is linked by the demo application to a map-based user interface provided by Google Visulization API. This allows users to easily visualize the location and magnitude of earthquakes using a graphical interface.

Agency Budget and New York Times News[2]. This takes an agency's budget accounts data between 1976 and 2014 (2010-2014 are projected value), and then displays the raw data with relevant news retrieved from the New York Times (via their RESTful news data search API) using an annotated-timeline user interface provided by the Google Visualization API.

3.2 Using LGD Raw & Enhancement Data

Supreme Court Justices Decision Making[3]. This demo mainly visualizes the Supreme Court Justices' voting histories from the datasets (Supreme Court Database, SCDB) maintained by judicial scholars. With additional LGD enhancements maintained on SMW (linking a judge in SCDB to their DBpedia URI), we can obtain additional personal backgrounds of each Supreme Court judge (e.g. birth place and education). By mashuping attributes about the judges from SCDB, DBpedia and the demo, we can analyze how the education background of judges may affect their voting decisions. This cannot be easily done without combining LGD raw data and LGD enhancement data.

---

[1] http://data-gov.tw.rpi.edu/wiki/Demo:_Worldwide_Earthquakes

[2] http://data-gov.tw.rpi.edu/wiki/Demo:_Agency_Budget_and_New_York_Times_News

[3] http://data-gov.tw.rpi.edu/wiki/Demo:_Supreme_Court_Justices_Decision_Making

4. CONCLUSIONS

Our approach is highlighted by massive and fast production of extensible LGD. One concern with the minimal LGD generation is that the resulting LGD raw data may not be useful without enough high quality semantics, which are important for users to understand the datasets. This is especially true when datasets use a significant number of numerical codes (e.g. FIPS code) and when datasets do not fully conform to RDF modeling (e.g. a table entry mapping to multiple entities). However, the cost of building high quality LGD generated for thousands of US government datasets could also be prohibitively expensive. With our incremental framework, a lazy high quality LGD generation is possible (i.e. we only do low-cost LGD generation for all datasets and delay the LGD enhancement data generation until such data is needed and affordable). Being better than nothing, the minimal LGD raw data at least provides a starting point for developers.

Our work is also highlighted by supporting user contributed linked data (e.g. the cross-dataset mapping). Such data can greatly help developers to share links across datasets that are typically hardwired in their programs, and it can be used to lower the cost of enhancing the quality of LGD raw data. Our incremental data growth model also conforms to the expectation of Semantic Web developers. RDF data, once published, should be stable enough to support applications, and new enhancements to data should not affect any existing users of the raw data.

As an ongoing project, links to the linked data cloud are still limited due to our limited time to perform curation, and we are working on two complementary approaches: (i) leveraging automated mechanisms to bootstrap more link generation (may come with high error rate), and (ii) improving our Web 3.0 platform to distribute the cost of link creation via crowdsourcing.

5. ACKNOWLEDGMENTS

This work is funded in part by a grant from DARPA's Transformational Convergence Technology Office.

6. REFERENCES

[1] Li Ding and Dominic DiFranzo and Alvaro Graves and James Michaelis and Xian Li and Deborah L. McGuinness and Jim Hendler, Data-gov Wiki: Towards Linking Government Data, in AAAI Spring Symposium on Linked Data Meets Artificial Intelligence, 2010.

[2] Tim Berners-Lee. Putting government data online. http://www.w3.org/DesignIssues/GovData.html, 2009

[3] Sören Auer and Christian Bizer and Georgi Kobilarov and Jens Lehmann and Richard Cyganiak and Zachary Ives, Dbpedia: A Nucleus for a Web of Open Data. In *The Semantic Web*, pages 722-735. Springer, 2008