Proceedings of the I-SEMANTICS 2012 Posters & Demonstrations Track, pp. 31-37, 2012.

LODGrefine – LOD-enabled Google Refine in Action

Mateja Verlic

Zemanta d. o. o.
mateja.verlic@zemanta.com,
http://www.zemanta.com

Abstract. As a part of LOD2 project we developed several extensions for Google Refine - a simple, yet very powerful open-source tool for working with messy data, to make LOD a first-class citizen in this tool, which is currently tightly coupled with Freebase and has no support for DBpedia. LODGrefine is a version of Google Refine with integrated extensions developed by Zemanta and DERI, adding support for reconciliation with DBpedia, export to RDF, augmentation of data with columns from DBpedia and extraction of entities from full text. Use of LODGrefine will be demonstrated in three use cases.

Keywords: Semantic Web, data cleaning tools, Google Refine, LOD

1 Introduction

Data cleansing and linking are very important steps in the life cycle of linked data [1]. They are even more important in the process of creating new linked data. Data comes from different sources and it is published in many formats, either as XML, CSV, HTML, as a dump from relational databases, or in custom formats like JSON, obtained from different web services. By linking these different bits of data from various sources we can extract information otherwise hidden and in some cases even gain new knowledge.

Unfortunately, these steps are not always trivial for an average user, e.g. a statistician working with statistical government data; on the contrary, they pose a problem even for more skilled researchers working in the field of semantic web. If data is available online this doesn't necessary mean it is ready to be used in semantic applications. In most cases such assumptions are wrong; it is very likely that data has to be cleaned, because *everyone* can publish data on the Web. Taking care of quality of Web data is still one of the main challenges for Semantic Web applications [2]. Data cleansing, especially if done manually, is a very tedious and time consuming task, mostly due to the lack of good tools. Commercial products such as PoolParty [7] provide a wide range of functionalities (thesaurus management, text mining, data integration), but they may not be the best solution when dealing with smaller datasets (in comparison to

huge datasets in big companies) and by less proficient users trying to convert data stored in Excel files or flat files.

A good and publicly available cleansing/linking tool should at least be able to: assist user in detecting inconsistent data, quickly perform transformations on a relatively large amount of data, export cleansed data into different formats, be relatively simple to use, and be available for different operating systems. Fortunately, there is one open-source (BSD licensed) solution available, which meets all the criteria mentioned above and even more. It was created especially for dealing with messy data, it is modular based and extendable, it works on all three major operating systems and it already provides functionalities to reconcile data against Freebase. This tool is Google Refine (GR) [4].

GR provides means to reconcile and extend data with data from Freebase, but not from DBpedia. By providing a LOD-friendy version of this tool (LODGrefine) supporting DBpedia we've made an important step towards making LOD a first-class citizen in this powerful, yet easy to use tool. LODGrefine has preserved all of the GR's cleansing and reconciliation functionalities and added new ones to make it even more useful for Semantic Web community.

2 From Google Refine to LODGrefine

GR is currently one of most powerful and user-friendly open-source tools for cleansing and linking data with Freebase. Support for faceted browsing and good filtering possibilities are its main assets, it works fast even when dealing with large amounts of data and it has a built-in support for Google Refine Expression Language (GREL), a special scripting language, which is easy to learn and use to transform data. The most important features of GR are the reconciling and extending data.

It is a server-client web application intended to run locally by one user. Instead of using a database to store imported data, it uses memory data-store, which is built up-front and optimized for GR operations. Its data cleansing and reconciliation abilities are tightly integrated with Freebase (Fig. 1) and making it support a different triplestore offering a SPARQL endpoint, e.g DBpedia, was not possible without implementation.

2.1 LOD extensions

Due to the modular nature of GR architecture it was not required to change the code of GR itself to make it LOD-enabled. We implemented extensions with additional functionalities. Maali and Cyganiak, researchers at Digital Enterprise Research Institute already developed RDF Refine extension [6] for GR, which can be used to reconcile data against any SPARQL endpoint or RDF dump and to export data as RDF based on a user-defined RDF schema.

Extensions (dbpedia-extension) developed by Zemanta complements functionalities of RDF Refine with ability to extend reconciled data with new

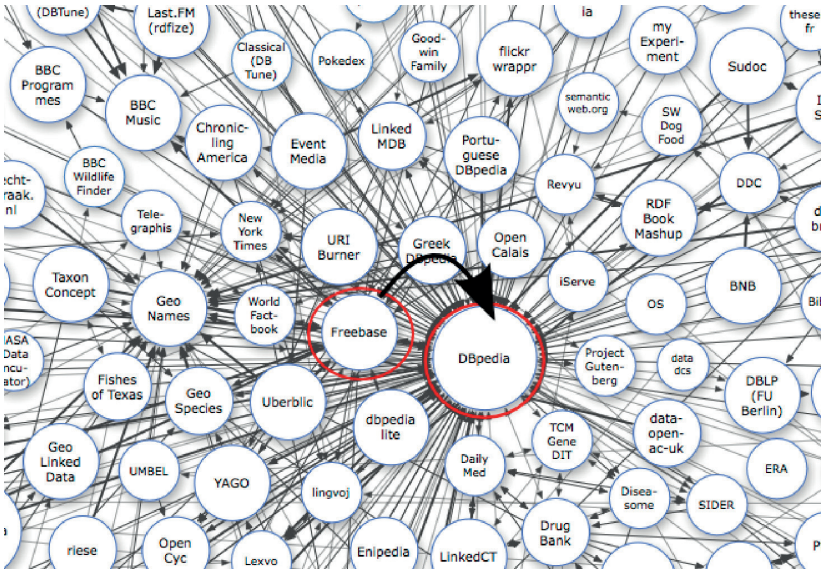

Fig. 1. With LODGrefine we closed the gap between Freebase and DBpedia in the LOD cloud [3].

columns based on data from DBpedia. It also supports extraction of entities from unstructured text using Zemanta API [9]. For example, if we extend reconciled data with description or biography property from DBpedia, we can extract different types of entities from this text and add in new columns to use it in RDF schema, which maps data in columns to nodes in linked graph.

Both extensions are free to use, their code is shared under the BSD License on Github (RDF Refine[1] [5], dbpedia-extension[2] [8]) and binary versions can be obtained from their home pages.

2.2 LODGrefine

To simplify the process of obtaining and installing LOD extensions we decided to integrate them into the latest version of GR (2.5-r2407) and name the LOD-enabled version LODGrefine. Although GR itself does not need any special installation, it is enough to unpack it and run it, the location of extensions depends on the operating system and it is more convenient, especially for first time users, if extensions are already integrated in the tool. Furthermore, we created a Debian package for LODGrefine, which will be integrated into LOD2 Stack, a stack of tools for managing the life-cycle of Lined Data.

[1] https://github.com/fadmaa/grefine-rdf-extension
[2] https://github.com/sparkica/dbpedia-extension

LODGrefine is available under Apache License 2.0 and can be freely downloaded either in binary format or as source code [8].

3 LODGrefine in action - use cases

For demonstration we prepared three use cases – examples of how LODGrefine can be used to clean data from different sources and domains and how to transform it into Linked Data.

3.1 100 best novels

In first example we will demonstrate how to convert data from a website first to a LODGrefine project, reconcile it, augment it additional columns from DBpedia and then export it as Linked Data.

In this example we will transform a list of 100 best novels from Modern library web page[3]. The list contains two rows for each novel - first row contains the title and the second one the author, but we need data in columns - one column for title and one for author. Fortunately, LODGrefine has an option to import line based text files and it can read text from clipboard. With some minor changes of default settings our data is imported in columns in few seconds instead of minutes or even hours. With GREL functions we convert titles from uppercase to titlecase and remove '*by*' preceding authors names.

Fig. 2. Reconciled and extended data. Third and fourth column contain entities extracted from autor's biography in the last column obtained from DBpedia.

[3] http://www.modernlibrary.com/top-100/100-best-novels/

Next step is reconciling author names with DBpedia using RDF extension to entity type dbo:Person[4]. After reconciliation data is ready to be extended with *has abstract* property from DBpedia using Zemanta extension (Fig. 2).

The last step of converting online list of novels into Linked Data is configuring RDF schema alignment skeleton, with which we specify how RDF data will be generated (Fig. 3). At any time we can preview the Turtle representation of generated RDF data to see whether schema we defined produced expected results. After the schema has been configured, data can be exported into one of the RDF representations supported by LODGrefine - RDF/XML or Turtle (fig. 4).

In this example we demonstrated how easy it can be to transform data from a website into Linked Data using LODGrefine. We also demonstrated its most important functionalities.

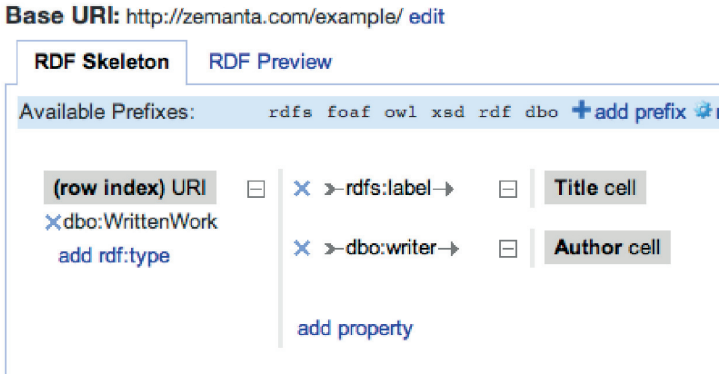

Fig. 3. RDF alignment schema for describing novels.

3.2 CKAN datasets

The Comprehensive Knowledge Archive Network (CKAN) is well known system for storage and distribution of data. It is widely used for storing government data and national registers as well as Data Hub, *the community-run catalogue of useful sets of data on the Internet*[5]. CKAN data is especially interesting for Linked Open Data community, but currently not all CKAN datasets in the Data Hub are provided as RDF (either SPARQL endpoint or RDF dump). A lot of datasets are provided as files with comma separated values (CSV), as Excel files or XML. Our goal is to show how it can be relatively easy transformed into triplets using LODGrefine in a similar way as described in previous example with novels.

[4] http://dbpedia.org/ontology/Person
[5] http://thedatahub.org/about

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .


<http://zemanta.com/example/novel#1> a dbo:WrittenWork ;
        rdfs:label "Atlas Shrugged" .

<http://dbpedia.org/resource/Ayn_Rand> rdfs:label "Ayn Rand" .

<http://zemanta.com/example/novel#1> dbo:writer <http://dbpedia.org/resource/Ayn_Rand> .

<http://zemanta.com/example/novel#2> a dbo:WrittenWork ;
        rdfs:label "The Fountainhead" ;
        dbo:writer <http://dbpedia.org/resource/Ayn_Rand> .
```

Fig. 4. Turtle representation of first few rows.

3.3 Looking for entities in extracted links

In the last example we will demonstrate the full power of LODGrefine by using it to clean and filter links extracted from blog posts to obtain links, that could be considered as descriptions for entities.

Many times bloggers include links into their blog posts to point the reader to Wikipedia or any other web page, that can be considered as information resource (e.g. Google Maps, Crunchbase[6] - free database of technology companies, Amazon). Ideally, links that could be considered as entity candidates, have non-empty anchor text (entity surface form) and href attribute set to external URL, which directs to a page containing a description of concept/person/object (disambiguation page) mentioned in anchor text.

Unfortunately, blog posts also contain even more links that can be considered as noise or even spam, e.g. links with anchor text without semantic value (e.g. here, this, Read more). We used LODGrefine faceted browsing and filtering abilities to quickly identify patterns of occurring anchor texts or target links, which could be considered either as entity candidates or noise. We used GREL[7] expressions to simply extract features from columns containing anchor texts and target URL, e.g. number of words in anchor texts, flag whether first word in anchor text is capitalized or not, domain part of the target URL, path level of the target URL and more.

When applying faceted browsing on a large number of rows it is not always possible to display all unique values. LODGrefine offers the ability to display all

[6] www.crunchbase.com

[7] GREL - Google Refine Expression Language: http://code.google.com/p/google-refine/wiki/GRELFunctions

different values by choice count, which can be further used in mathematical expressions. For example, if only a few anchor texts appear 100x more frequently than the rest of the anchor texts, it is difficult to filter out anchor texts with occurrences between 20 and 35. In this case it is better to use logarithmic scale. It is worth mentioning, that filtering in LODGrefine works really fast even for 100 000 rows, where some other tools might start having problems.

After filtering we reconciled entity candidates against DBpedia and/or Freebase to link them to existing entities and then exported entity candidates in Turtle representation.

4 Conclusions

LOD-enabled version of Google Refine is one of the best open-source tools for cleaning and linking. With the examples we demonstrated its versatility and powerfulness for transforming tabular data to Linked Data for different problem domains.

5 Acknowledgments

This work was supported by a grant from European Union's 7th Framework Programme (2007-2013) provided for the project LOD2 (GA no. 257943).

References

1. S. Auer, J. Lehmann, and A.-C. N. Ngomo. Introduction to linked data and its lifecycle on the web. In *Reasoning Web*, pages 1–75, 2011.
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. R. Cyganiak and A. Jentzsch. Linking open data cloud diagram. http://lod-cloud.net/.
4. G. Inc. Google refine homepage. http://code.google.com/p/google-refine/.
5. F. Maali and R. Cyganiak. RDF Refine homepage. http://refine.deri.ie/.
6. F. Maali, R. Cyganiak, and V. Peristeras. Re-using cool uris:entity reconciliation against lod hubs. In *Linked Data on the Web*, volume 813. CEUR-WS, 2011.
7. T. Schandl and A. Blumauer. Poolparty: Skos thesaurus management utilizing linked data. In L. Aroyo, G. Antoniou, E. Hyvnen, A. ten Teije, H. Stuckenschmidt, L. Cabral, and T. Tudorache, editors, *The Semantic Web: Research and Applications*, volume 6089 of *Lecture Notes in Computer Science*, pages 421–425. Springer, 2010.
8. M. Verlic. LodGrefine homepage. http://code.zemanta.com/sparkica/lodgrefine/.
9. Zemanta. Zemanta developers page. http://developer.zemanta.com/.