Cleaning & Publishing Metadata

A Hands on Tutorial with OpenRefine

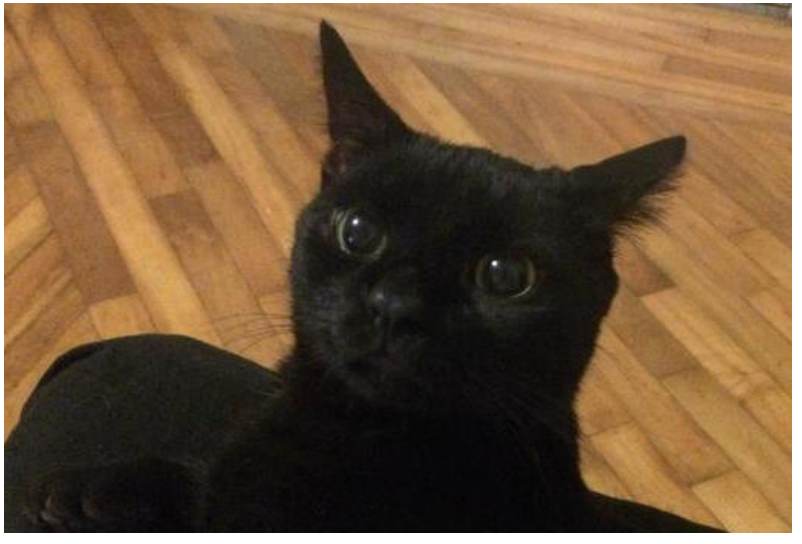
Christina Harlow, @cm_harlow

LODLAM Workshop, DLF Forum 2015

Slides, Examples, + Install

https://github.com/cmh2166/DLF15LODLAM

Didn't do Installation HW?



Ernest is not happy with you.

Quick Installation Backup

Go to Installation instructions, follow RefinePro options, & hope you get instance on server where DERI extension works.

Agenda/Method

1. Introduction (5)
2. Importing Data (15)
3. Data Munging (20)
4. Reconciliation (30)
5. Mapping, Exporting (30)
6. Wrap-up (5)

Quick Introduction

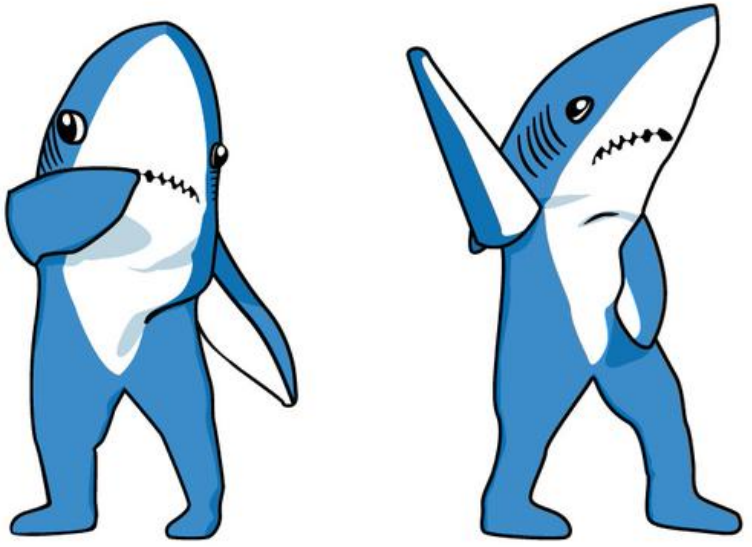
1. Introduction <==
2. Importing Data
3. Data Munging
4. Reconciliation
5. Mapping, Exporting
6. Wrap-up

Learning LOD by Working with LOD

Goal: Learn Linked Open Data by working with it in context of Libraries, Archives, & Museums metadata

Help: Raise Hand, Ask Friend, Check Instructions, Check online

Let's All Left-shark It



But Also...

"Hacker School Rules"

- No feigning surprise
- No well-actually's
- No back-seat driving
- No subtle -isms

https://www.recurse.com/manual;

Quick Intro to OpenRefine

- OpenRefine = power data tool
- Since 2012, community-sourced
- OpenRefine.org
- github.com/OpenRefine/Openrefine
- Java (& Jetty) tool that runs locally
- GUI runs in your chosen browser (NOT INTERNET EXPLORER)

OpenRefine & RDF

- Native support for importing RDF/XML, RDF Ntriples
- Original Freebase Extension
- DERI RDF Extension, LODRefine
  - RDF Document Reconciliation
  - RDF Skeleton, Mapping
  - RDF Export: RDF/XML, RDF Turtle

DERI RDF Extension & LODRefine

No longer actively supported

```
Each annoyance re:slowness, bugs = 1

 If reaches 30, we promise to learn
 Java + become LODRefine committers
```

Why so much Recon?
Where's the RDF?

Using RDF data & tools like
OpenRefine = better entity matching

Possible Influences:

- VIVO Recon Service
- Nomenklatura
- Ecco!

Importing Data

1. Introduction
2. Importing Data <==
3. Data Munging
4. Reconciliation
5. Mapping, Exporting
6. Wrap-up

Importing Data

3 'tracks' for this workshop:

1. Import sample CSV data to make into local SKOS/RDF authority
2. Import sample DC/XML metadata to make into DC/RDF
3. Import your own data & DIY it

Starter data: Cleaning Metadata > Data

Import data into OpenRefine

1. Start up OpenRefine or LODRefine
2. Click on Create Project Tab
3. Click on Web Addresses (URLs)
4. Enter the URL for GitHub Raw Object of Starter Dataset you want to use

(Or download/save your metadata to working environment & use 'This Computer')

Import data into OpenRefine

1. Preview your data as project
2. Change settings as needed
    - XML, Json: need to choose 'record' object
    - CSV, Excel: review for header rows
    - RDF: Preview options for loading
3. Once ready, give name, Create Project

Viewing OpenRefine Project

- Saved Automatically with Undo / Redo Panel
- Rows/Records divide = VERY IMPORTANT
- Extensions, Export Options in Top Right Corner
- Facet, Filter panel on left
- If something freezes, refresh the browser (gahhhh)

Import Your Data

Go ahead and import the data for the track you'd like to pursue. Explore the possible options.

Bonus: Once your main project is created, export one of the sample RDF documents to see how it looks as an OpenRefine project. This differs from what the DERI extension expects.

Data Munging

1. Introduction
2. Importing Data
3. Data Munging <==
4. Reconciliation
5. Mapping, Exporting
6. Wrap-up

Metadata Munging in OpenRefine

Ways to Normalize, Remediate Data:

- Join, Split Rows
- Splitting, Renaming Columns
- Faceting, Clustering, Filtering
- Google Refine Expression Language (GREL)

https://github.com/OpenRefine/OpenRefine/wiki;

Prepare Your Data

- Get columns renamed as needed
- Get cells joined
- Facet, review
- Facet, cluster, normalize
- Filter to target, map values to new fields

Reconciliation

1. Introduction
2. Importing Data
3. Data Munging
4. Reconciliation <==
5. Mapping, Exporting
6. Wrap-up

OpenRefine Reconciliation

Reconciliation broadly: Compare values in my dataset with values in an external dataset, if deemed a match, link and pull in external datapoint information

Cleaning Metadata > Instructions > Reconciliation

Add column by fetching URL...

- HTTP requests to external data API in UI
- takes far longer to pull data
- requires parsing returned data with GREL

Standard Recon Service API

- RESTful API between OpenRefine and external data
- handles JSON reconciliation objects btwn datasource API + Openrefine

Cleaning Metadata > Instructions > Reconciliation

DERI RDF Extension

- no longer actively supported
- Standard Recon Service API to work with RDF, SPARQL endpoints
- RDF docs held in memory
- SPARQL recon dependent on SPARQL server details

Reconciliation Demos

- LCSH via SPARQL
- Languages via RDF Doc
- Geonames via Recon Service
- VIAF hosted service

OpenRefine Recon

1. Run Recon according to your choosing - see options in Recon instructions
2. Pull URIs for a particular field
3. Pull other information helpful for your projects
4. Make sure to pull in URIs, information

Mapping & Exporting RDF

1. Introduction
2. Importing Data
3. Data Munging
4. Reconciliation
5. Mapping, Exporting <==
6. Wrap-up

DERI RDF Creation

RDF Extension button > Edit RDF Skeleton...

- Add Namespaces/Utilize Namespaces
- Can assign types, create blank nodes
- Preview the Output
- Save your skeleton
- Export > RDF...

http://refine.deri.ie/

Map & Export

Map your data to RDF using the RDF skeleton, preview the Turtle, then export when you're ready.

Use the SKOS docs and the DPLA MAP v.4 doc to help guide the creation.

Bonus: Export your doc then use for a test RDF Doc reconcile.

Wrap-Up

1. Introduction
2. Importing Data
3. Data Munging
4. Reconciliation
5. Mapping, Exporting
6. Wrap-up <==

Links + Contact

cmharlow@gmail.com

http://openrefine.org/

http://github.com/openrefine/openrefine

@openrefine, @cm_harlow