Sept. 22, 2014

Notes for DST4L training- Free Your Metadata
Bonus: OpenRefine!

Data cleaning, linking and enriching with OpenRefine
Cleaning Data - wrangling it, working with it, making it usable
Goal - to make the tools available more useful to us

Starting with open refine because of the easier learning curve - stepping stone

Presenters:
Seth van Hooland – Historian, library and information science person
Ruben Verborgh – Engineer

What do we currently associate with linked data?
Better look at RDF

IDs, triples, Semantic Web,
~Libraries and the semantic web 2010 - cartoon, check it out:
http://www.youtube.com/watch?v=t0dEqgNKH0Y
Check out Google trends: https://www.google.com/trends/
"Raw data now" - Tim Berners-Lee TEDtalk:
http://www.ted.com/talks/tim_berners_lee_the_year_open_data_went_worldwide?language=en

Dbpedea (http://dbpedia.org/) - user interface isn't very user friendly, hard to make sense of but is a common tool.
Digital humanities projects - often mostly engineers and not so many humanities
Librarians and archivists often get put in the role of content providers instead of collaborators

Humanities people must understand RDF in order to bring more to the table

Linked Data for Libraries, Archives and Museums (First chapter -most imp, largest chpt- is available for free online http://freeyourmetadata.org/publications/
Their paper is available here - preprints available

Find value in your unspecified fields - key words ext.

Usage of RDF discussion:
Entity relationship model

Entity, attributes, relationship

Author	Has written	Book
Name		Title

Different kinds of databases:
Relational databases:

Problem with relationship databases - **don't know the full scope from the needs**

Over time, must add and change things but this impacts other tables
Makes **change** slow and challenging and expensive - problem in current climate

In the context of the web, when are we angry with our database with the invention of the web?
Mapping an original schema to a new database schema
2000-2004 the coolest new technology at that time? **XML**
XML allows structured data to be more flexible then when the relationships are stored entirely in a database management software.
<book>
<author>John Smith</author>
</book>

Need separate XML scheme in order to validate your XML fields. Gets complex.
More technology independent but still has challenges. (All tags are personally made and can be used and kept in schema)

RDF:

Subject, predicate, object
Not just associate values - use URIs
Dublin core (or similar ontology) field for predicate, accessible for humans and machines. Or VIAF (http://viaf.org/).
Use URI (link) for the author's Wikipedia page or personal website as SUBJECT
Use URI (link) for book or other object in Wikipedia or Amazon page

Schema neutral - you still need to know that the schema exists, but we'd all be using the same schema.

VIAF - Virtual Authority File (all the big countries putting their files together)
HTP - you can look up what things mean.
RDF - always the same structure. 3 parts. (Can always add extra triples as needed)
URI - can look up what things mean

When triples are amassed on the web, the semantics can be quite problematic. Must have *clear* predicates because otherwise it can screw other people's work
E.g. "a" is a terrible predicate – since it's technically an article, but people use it anyway

Look for slides about different data models! Very cool visual representation!

Data Quality: The Accuracy Dimension - Best, most practical

Theoretical data work has trouble moving into and being used in reality

Data cleaning tools

- Different communities:
  - Corporate world: Inforamatica (http://informatica.com), Trillium (http://trilliumsoftware.com)

- Academia: Wrangler and Potters Wheel AVD
- Hackers: Datapipes

Open Refine - Project leader: Tom (leader tomorrow)

- Knowledge base called Freebase
- Freebase is the core of the Google knowledge graph
- Simple but powerful work
- Runs locally in browser
- Open source project - same code base as the original
  - Training and developing happening communally

Consider writing a blog story on what we're doing here today!

Recap: the first part we covered the history of linked data; the transitions from local databases to XML to RDF; context for why clean data

~Break~
Begin walkthrough on GoogleRefine

Get metadata used in book: http://data.freeyourmetadata.org/
Character set or encoding problems can be spotted before clicking create project in Google Refine - click the Character encoding box

Scroll through the preview to make sure that it makes sense. If not make sure it splits correctly based on selections at the bottom section.

If the import is wrong, you end up wasting time fixing import issues instead of fixing the data. Check to make sure the correct number of rows. Check for the special characters that should be there.

Name it and click create project (clear names are important/useful)

Check out your data
Record ID - use carrot to access Facet tool to analyze info in the left side by different info visually
The visualization can help you see outliers (you will also see Numeric or Non-numeric which can help you clean quickly)

What if two rows have same info? Use Duplicates facet to see. Group the duplicates together

Sort can be a view or permanent - these are separated clicks so be sure you make it permanent when ready/as needed.

Refine remembers your history. Anything you change in OpenRefine, does not change your original on your computer while you're working. (but it does auto save snapshots every 5 mins or so. Don't worry!)

"Blank down" to find repeats (perhaps repeated record ID)
Check some visually
If registration numbers are unique it is safe to delete (as long as blank down is on and you are sorted to the blank record IDs)

If there are leading zeros and you have the ID # import as numbers the 000s will go away.
Edit cells -> common transforms
    Can transform anything to numeric or to text or various other things
Edit cells-> blank out
    Makes the items in the column blank can then remove blanks
To do an operation that relies on sort - must "Reorder rows permanently" in the sort link at the top of the screen

Blank down only works when duplicates are next to each other
Blank down is useful for the context of this particular collection. Only use blank down if it is appropriate for the collection. Do this for unique identifier

Remember to experiment with OpenRefine because it's easy to go back using the Undo/Redo tab.

If you have fields that need to have multiple words in the same field. Like keywords.
(Side note: RDF is a good way to avoid field overloading.)
    Refine has a way to deal with multivalue cells (split them)
    Rows view vs Records view allows for sorting in different ways. Records view keeps whole record together (this view is based on indentation)

Cluster view available in Facet -> text facet for any column
"Fingerprint" keying function - this is a tool that is very effective and not very aggressive. It spots highly similar phrases (generally based on capitalization) there are two more that are very aggressive search options

Upload or post any ideas you have to go over tomorrow (data or things to do with data)

Recap: Basic data cleaning

Break

Weaving the Web by Berners-Lee, Tim

Data reconciliation

In our dataset:
- The Categories are based loosely on an in house thesaurus (partly free text): challenge to link to other vocabularies
- How will they bring these categories into a controlled vocabulary of concepts

Links are uni-directional
Can link to LCSH but the LCSH website doesn't link back
- Can't link to everything.
- Forms could be useful for linked data
- But links would have to become your identifier

Dbpedia
- The purely data aspect of Wikipedia
- It's RDF triples

- When used in a link - unidirectional

Can search pieces of RDF that use a resource as identifiers (basically to search for things that are linked to something else - Want to find all objects related to Belgium can search for that if an identifier instead of a simple link without a triple)

See http://linkeddatafragments.org/ for a way to search

When compatible with outside regulated vocabularies can search related (Broader or Narrower) terms if use a triple to connect to them.

Give RDF extension information on what site to use for reconciling

http://Freeyourmetadata.org/reconciliation

This has an example endpoint (different results depending on the end point)

To see if there is something in a resource links to other concepts

Concept - abstract thought

Term - the specific word in whatever language

Reconciling takes several hours

Reconciling turns data from an array of letters into a concept through linked data

You must approve the correct term; GoogleRefine cannot choose between subject headings correctly or which of a list could be correct in this case.

SPARQL endpoint set up for LCSH

There might be SPARQL endpoint findable from elsewhere

If you have an RDF file for a thesaurus, you can use it as a document

Dbpedia and The Getty Research Institute (http://vocab.getty.edu/) both have SPARQL endpoints

Linked data used to generate more linked data

Obtaining the reconciled URLs of the links for the successfully linked fields.

Can create a new column based on an existing column

Can then tell it to have in that column be the links that are beneath the blue highlighted matched items

If reconciliation is not successful with one endpoint try a different end point.

SPARQL

- A query language
  - Like SQL is a query language
- For RDF triples
- Endpoints are the RDF triples
- SPARQL is the way to get to those triples
- Can choose different endpoints just like LCSH or Dbpedia
- Allows you to make sub selections of data that were not originally created together.
  - Freebase ("evil twin of dbpedia" because not free) interesting to check out - https://www.freebase.com/
- SELECT * WHERE ?pp skos:prefLable "Pablo Picasso" ?artist ex:influencedBy ?pp. ?artist exiborn ?place
- Find people who were influenced by Pablo Picasso

Endpoints are less reliable than normal websites.

How to derive extra value out of longer chunks of text:

On December 5th, 2013 (computer can spot date)
We went to see
The White House (computer can spot proper noun -Interesting concept)
In Washington (computer can spot proper noun -Interesting concept)

Identify important items or concepts (underlined above)
How to disambiguate Washington from all the other important Washingtons? Connect it to a URL!
Associate an entity with a URL
Need API keys to make things work effectively
There are APIs

Alchemy	Alchemyapi.com/api/register.html
Zemanta	Developer.zemanta.com/member/register/

Zemanta is better at both finding the important concepts *and* disambiguating them correctly
It has the best algorithm
Research question: how are these services complementary (to what extent?)
How do you assess the quality of the extraction process?
Start by annotating the corpus manually (2 different people individually reviewing - get gold standard)
This API process is called "Extraction"
Theoretically this would be slower, but actually, it is not.
It gives the "Best" URI (some services can choose how many URIs you want to receive).
Matches can look like they match but actually have subtle differences.

Distant reading
Accurate linking to information is challenging because what is accurate information?

APIs

How to expose data in a sustainable way?
Export project from OpenRefine
How to expose finished data to the public?
The Final answer of how to expose data has changed over time.

1997	HTML 3.2
2000	XML
2005	JSON
2014	RDF?
2020	????

REST
With a website- how to make machine readable

Can have the tech people make you an API but is that what you need?
A way to publish data on the Web for humans AND machines.
    Now and into the future!
What is "Sustainable"?
    How to access info - stay the same
    How to store and format info - change with times
Will PHP exist in the future? JSON? How long into the future? 1 yr, 5 yrs, 10 yrs

What is REST?
    Representational State Transfer
    It is an **architectural style** for distributed hypermedia systems
        A set of constraints with benefits
        Uniform interface constraints - Most important
- What an interface should look like:
- Identification of resources
  - Basic building blocks
  - A resource relates an identifier to a concept
  - A URL with a PHP file in it doesn't bookmark/share/move well
  - A URL with an identifier and NOT technology based (e.g. .PHP)
  - Object on site based on identifier doesn't move
- Manipulation through representations:
- Each resource can have multiple representations
  - One URL for all the versions of devices
  - One URI that:
    - Gives HTML
    - Gives JSON
    - Gives RDF
  - (Much more sharable if this works)
  - Support same identifier and added resources
- Servers and clients send **self-descriptive messages**
  - Don't need conversation stream to find resource again
- The interaction should be driven by **hypermedia**
  - Receive information with links you will need
- APIs should work with hypermedia in order to function
- Treat all clients equally: clients want sustainable information (machines are clients too)
- REST:
  - Uses information
  - Technically no longer meaningful
  - Hypermedia API is actual REST API
- Fallacy of the multi-API culture
- Website for humans and website for machines
- More expensive because have to keep updating two sites
- Don't build applications based on APIs because they are too heavy on the system
- Website is your real API - should be providing access equally to machines
- Identify resources, extend representations
- Don't ask for multiple APIs b/c that's a fallacy
  - Machine-readable representations
- Links are identifiers, so we need our resources to have identifiers in there

    - Facebook - can ask for data as HTML, XML, RDF
        - Quite successful
    - Dbpedia.com also good at this
- Handbook
    - Free chapter at book.freeyourmetadata.org
- OpenRefine book
    - Free chapter online
- Follow @freemetadata, @RubenVerborgh, and @sethvanhooland

Wiki OpenRefine recipes page has a few places to check out
Lamont Library tomorrow -

- Passed the café, to your right, down 2 flights, through your doors, just past the photocopier
- room B30
- Bring data if you have it.
- More hands-on
- They'll have datasets as well