Information Visualization

Auxiliary methods

Lesson 9

Marilena Daquino
Assistant Professor

Department of
Classical Philology
and Italian Studies

marilena.daquino2@unibo.it

Table of contents

01 Data matching query

Reconciliation to Wikidata

02 Web scraping extract

Access and parse HTML documents

03 NER extract

Named entity recognition

04 AJAX query

Query SPARQL endpoints from js

01

Data matching

Tutorial: Reconcile entities with Wikidata API

In a perfect world

01

Reconciliation

URIs are unique

URIs describing entities (e.g. people) are used across data sources, thus interlinking is straightforward.

URIs are linked

In case multiple URIs for the same entity exist, we have a link (e.g. owl:sameAS) between these.

Labels just match...

If we look for an entity by its label, we get exactly what we are looking for.

In the real world

01

Reconciliation

URIs are unique

URIs describing entities (e.g. people) are used across data sources, thus interlinking is straightforward.

We have multiple URIs across sources representing the same thing

URIs are linked

In case multiple URIs for the same entity exist, we have a link (e.g. owl:sameAS) between these.

We don’t have any link between these

Labels just match...

If we look for an entity by its label, we get exactly what we are looking for.

And if we try to match their labels, we end up with wrong links (is Mona Lisa a person, a painting or a brand?).

A holistic approach

Try everything, and never the same

If labels are not enough to reconcile data across datasets, you may need to combine methods and more data, e.g.

- to distinguish people and companies, you may try to match their classes
- to distinguish homonyms, you may compare birth dates
- And so on…

Methods change according to the type of entity you are trying to match and according to the sources and data available.

A holistic approach

01

Reconciliation

Use tools, or try your luck

Some tools for data cleaning exist, and require manual validation, e.g. OpenRefine.

Or you can try the hard way, and implement your methods to reconcile data to some data source.

Ideally, you want to reconcile entities to some **authority file**, that is, a data source that many other sources on the web are likely to link to, e.g. Wikidata, VIAF, Getty vocabularies.

OpenRefine

A free, open source, powerful tool for working with messy data

Why Wikidata?

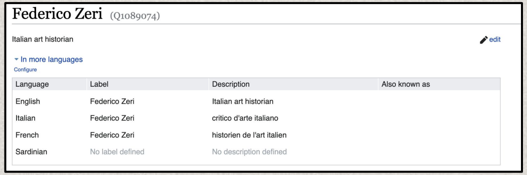
01

Reconciliation

All-in-one

Is a good candidate for the task since:

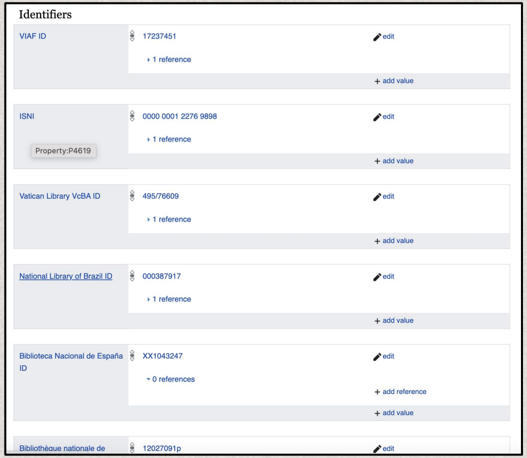
- Many sources link to Wikidata: you can look in third-party datasets for entities that are matched to Wikidata URIs that you matched to
- Wikidata includes **plenty** of links to other external authority files (Getty, VIAF, IMDB, Google Scholar, etc.). If you reconcile your data to wikidata, it works as a gateway to directly access other data sources
- It has very good APIs for automating the process (fast, with a good ranking of results)



Federico Zeri (Q1089074)

Italian art historian

Language	Label	Description	Also known as
English	Federico Zeri	Italian art historian	
Italian	Federico Zeri	critico d'arte italiano	
French	Federico Zeri	historien de l'art italien	
Sardinian	No label defined	No description defined	



Identifiers

VIAF ID	17237451	1 reference
ISNI	0000 0001 2276 9898	1 reference
Vatican Library VcBA ID	495/76609	1 reference
National Library of Brazil ID	000387917	1 reference
Biblioteca Nacional de España ID	XX1043247	0 references
Bibliothèque nationale de	12027091p	

02

Web scraping

Tutorial: Access, parse and traverse tree data (HTML) with BeautifulSoup

So much hidden information

HTML

Is the main source of data on the web.

It is a semi-structured format: there are rules, but the composition of elements can change significantly.

Scraping

A HTML document can be parsed as a tree object. You can query elements that are children, parents or siblings of other elements, and you can interact with their attributes. You can define which **paths** to traverse.

So much wrong data

It’s time consuming

While parsing and querying is made easy by many APIs and libraries, scraping many different websites requires you to define **bespoke rules** for each website.

It’s error-prone

HTML is often manually created. The interesting information you are looking for is often available in **non-homogeneous** ways (elements like to change…) or it is **not identifiable** by any markup element.

03

NER

Tutorial: Recognize entities in natural language text

Named Entity Recognition

03

Extraction

Semantic content

While scraping applies human-defined rules to extract knowledge based on the structure of the document, NER looks for semantic and linguistic structures of sentences to recognize some types of entities.

Pre-trained models

Since it is a well-known task, there are plenty of **pre-trained models** (e.g. Spacy NER) that allow you to extract entities from text without having to create (annotate, test, and validate) your own algorithm.

04

Ajax

Tutorial: Query SPARQL endpoints from js

05

Hands-on

Go to the tutorials: the notebook and the web document

Thanks!

Do you have any questions?

marilena.daquino2@unibo.it

https://github.com/marilenadaquino/information_visualization

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**