



NIEDERSÄCHSISCHE STAATS- UND
UNIVERSITÄTSBIBLIOTHEK GÖTTINGEN



Working with CERL data

Andreas Walker
CERL Annual General Meeting
2020-10-06

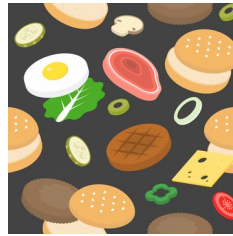


GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

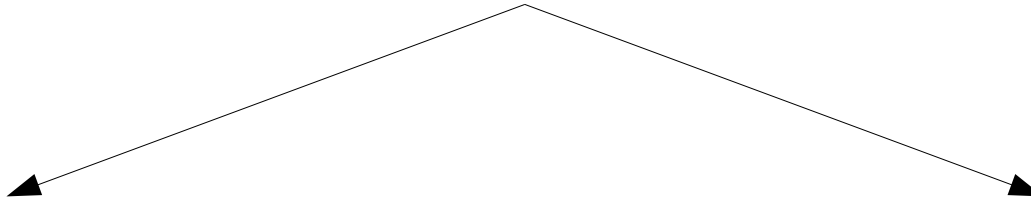
Preliminary notes

- The presentation is about **half an hour**, the rest of the time is for questions and discussion
- We will be **recording** the presentation and discussion session
- **Slides are online** here: gwdg.de/~walker5/docs/20201006_slides.pdf (useful if you want to click on the links)

How to get a burger



- **Raw ingredients**



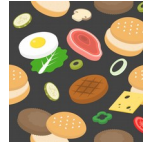
- **Eat ready-made**, low effort but somewhat dependent on the cook



- **Make your own** from ingredients, packed for your convenience

Image credits: [Burger Ingredients Vectors by Vecteezy](#) | [Cheese Burger by Jun Seita](#) | [Grocery Bag by Dawn Hudson](#)

Two views on our data



Internal format
(JSON)



Web interface

- **Aims at:**
 - Usability for human users
 - Generalistic tool
- **Restricted by:**
 - Available technologies
 - CERL development resources
 - Earlier design decisions



Serializations

- **Aims at:**
 - Machine-readable data
 - Building / using your own tools
- **Restricted by:**
 - Available technologies
 - Your own resources

Capabilities of the web interface – what it does really well

- **Search** for records
 - Both full-text search and search in specific data fields
 - Fairly sophisticated search syntax (ElasticSearch)
 - See e.g. [MEI Searching Guidelines](#) or [ElasticSearch documentation](#)
 - Filter search results further by pre-defined facets
- **Display and edit** records in a document-like view
 - Shows *all* the information available in a record
 - Aims at human reader who intelligently extracts needed information
 - Assumes you are working with a small number of records and can still do the work of intelligent extraction for each one

Capabilities of the web interface – what it does really well

Multilingual labels

CERL

SuchenIndexVerlaufLesezeichenMehr ...

Oxford, Bodleian Library (GB): Douce 66.

ISTC Nr.
Verfasser
Titel
Erscheinungsvermerk
Format
Sprache
GW Nr.

ib01085100
Brant, Sebastian
Das Narrenschiff [German]
Basel : Johann Bergmann, de Olpe, 12 Feb. 1499
4°
Deutsch
GW 5047

zum vollständigen Datensatz

Themenbereich
Stichwörter
Zeitraum

Literature
jest; poetry
humanist

Beschreibung des Exemplars

Exemplar Nr.
Besitzende Bibliothek
Signatur
Anmerkung
Weitere Identifier

00205433
Oxford, Bodleian Library (GB)
Douce 66.
BodInc-Id: B-504(1)
ib01085100 (TextInc)

Provenienz 1700

Zeitraum
Provenienzzname

Anmerkung

- 1700
Schedel, Sebastian (d.1628), - 1628 [per] great-grandson of Hartmann Schedel; see Nicolas Barker, *Hortus Cystettensis: The Bishop's Garden and Besler's Magnificent Book* (London, 1994), 32; Wagner 82.
Provenance: Sebastian Schedel (†1628); on front pastedown printed and coloured armorial book-plate: a moor's head, with motto 'Ich lass passiern', inscribed 'Sebastian Schedel', see Siebmacher, *Bürgerliche Wappen* V.1 pl. 76 for an identical coat of arms, but with an uncoloured head, from an Album amicorum dated Vienna, 7 Feb. 1594.

Provenienz 1701 - 1900

Ort
Region
Zeitraum
Provenienzzname

London (Geonames Id: 2643743)
England
1701 - 1900
Douce, Francis (1757-1834), 1757 - 1834 [per; Vorbesitzer] (Männlich, Gelehrter/Wissenschaftler, Keine Charakterisierung / Laie) Keeper of Manuscripts at the British Museum, bequeathed his collection to the

Alle Exemplare

Oxford, Bodleian Library (GB): Douce 66.
Wien, Österreichische Nationalbibliothek (AT): Ink 12.H.16

Information pulled from other databases for display (record only contains ID)

Related information grouped visually

GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

AGM 2020 – Working with CERL data – Slide 6

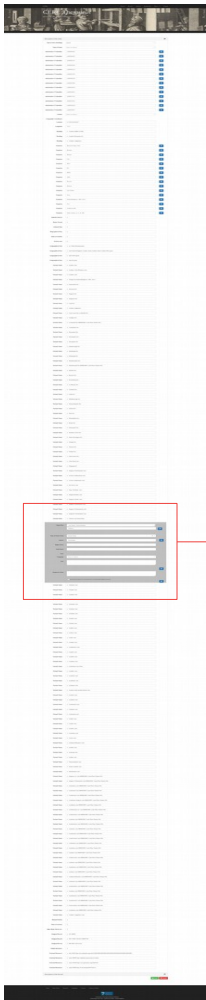
NIEDERSÄCHSISCHE STAATS- UND
UNIVERSITÄTSBIBLIOTHEK GÖTTINGEN

SUB

Limitations of the web interface – where it struggles

- Filter records based on more **complex criteria**
 - Search syntax is based on presence of information in particular data fields, but cannot usually describe more complex data structures
 - **Example:** Search in MEI can show you documents that have (a) a 16th century provenance and (b) information on purchase prices, but it cannot be used to know whether the price information is *in* the 16th century provenance block
- Display and edit records based on **particular tasks**
 - Because displaying a record is a non-trivial task that requires decisions about mapping the data model to a human-readable display, we can only provide generic solutions
 - Users and editors end up doing the cognitive labor of ignoring data they do not need for the task at hand, or adding data from external sources

Limitations of the web interface – where it struggles



Name Part: Last Name / Entry Element
Chelsea

Type of Name Form: Variant Name

Source: Deschamps

Begin (Year):

End (Year):

Note:

Language: Select an Option

Text:

Temporary Data:

☐ protected (cannot be overwritten by an automated update process)

It takes **a lot** of scrolling to edit this piece of information

Going beyond the web interface – cooking your own

- For some tasks, it can be useful to switch from the web interface to our second option: using and/or building **your own tools** for working with CERL data
- This can range from **integrating the data with your existing workflows** (e.g. working with CERL data in your favorite spreadsheet software) to **building specialized software** (e.g. a Python script that produces statistics on a set of CERL data), with different requirements in terms of technical expertise
- Going beyond the web interface will require input from both **domain experts** and **technology experts**, depending on the task and the available tools (and of course the two roles are **not mutually exclusive**)

Going beyond the web interface – what do you need?

- The web interface does not (and cannot) perfectly present the **data model** – it is always a translation. But it often informs our own **mental models** of the data
- A first step in working with the “raw” data: developing a more precise mental model of how the data is structured, and **mapping** it to your own mental model of the domain
- **Example:** You have a fairly complex mental representation of a book purchase, and as a human being you can easily manipulate it to ask questions centered on particular elements of that representation, like prices within a date range. How does this mental representation correspond to the data model?

Going beyond the web interface – reading the model

- You need to
 - find the *corresponding elements* in the data model,
 - diagnose how they are *related by structure*,
 - understand whether the combined information of data elements and structure can *answer your question* and
 - **decide whether you need to *enrich or simplify* the data for the task at hand**

Going beyond the web interface – reading the model

The screenshot shows a JSON viewer interface with a tree view on the left and a detailed view on the right. The tree view shows a JSON object with a 'timeperiod' field and an 'agent' field. The 'timeperiod' field is expanded, showing 'start' and 'end' values. The 'agent' field is also expanded, showing 'role', 'gender', 'type', 'professionOrType', 'ownerId', 'dates', 'name', 'characterisation', 'priceCurrency', 'acquisitionMethod', 'certainty', 'place', 'note', 'type', 'source', 'area', 'areaCode', and 'priceAmount'. The 'priceCurrency' and 'priceAmount' fields are highlighted with red boxes. The detailed view on the right shows the values for these fields: 'timeperiod' with 'start: 1550' and 'end: 1600', 'priceCurrency' with 'Lire: Soldi', and 'priceAmount' with '1:16'.

```
JSON Rohdaten Kopfzeilen
Speichern Kopieren Alle einklappen Alle ausklappen JSON durchsuchen

{
  "timeperiod": {
    "start": 1550,
    "end": 1600
  },
  "agent": {
    "0": {
      "role": "R390",
      "gender": 1002,
      "type": "person",
      "professionOrType": {
        "0": "unk",
        "ownerId": "00010051",
        "dates": "1550 - 1600",
        "name": "Georgiis, Georgius de"
      },
      "characterisation": {
        "0": "ari",
        "priceCurrency": "Lire: Soldi",
        "acquisitionMethod": "a",
        "certainty": "a"
      },
      "place": {
        "0": {
          "name": "Feltre",
          "geonamesId": 3177120
        }
      },
      "note": "A c. alr nota di possesso successivamente depennata, anche se ancora parzialmente leggibile: '[?]ri Georgij Geor. et suorum [?], mi costò L. 1 S. 16'. Sull'angolo superiore esterno della carta, la stessa mano annota 'N° 181'.",
      "type": "a",
      "source": "a",
      "area": {
        "0": {
          "areaCode": "e-it",
          "priceAmount": "1:16"
        }
      },
      "stampsNote": ""
    }
  }
}
```

▼ timeperiod:
start: 1550
end: 1600

priceCurrency: "Lire: Soldi"

priceAmount: "1:16"

All elements **grouped** in a single provenance block

Going beyond the web interface – simplification

- A standard case of **simplification** is representing the data in a format that is cognitively more accessible to humans but does not capture the full complexity of the data model's structure:
- **Example:** You want to edit records in your favorite spreadsheet application, so you need to simplify the tree-like structure of a record into a tabular structure of columns and rows
- **Challenges:**
 - There is **not a single mapping** from a more complex to a simpler data structure, so you have to decide which one to use (which is exactly what happens if you use the CSV download functionality in AMPLE)
 - Simplification always means a **loss of information**. This is particularly important if you later want to manipulate the data and put it back in the database

Going beyond the web interface – simplification

CERL

Material Evidence in Incunabula

Every item of data recorded (a/c provenance, therefore it can be Europe and through the centuries

1 Edition in 3 Copies

Bartolus de Saxoferrato : Super secunda parte Codicis

Milan : Leonardus Pachel and Uldericus Sch

Oxford, Bodleian Library (GB). – Auct. 3Q

Bartolus de Saxoferrato : Super secunda parte Codicis

Choose a format

Excel (csv)

Exports the data into a csv file, which can be opened easily with Excel or Calc (or similar). The separator is a semi-colon. Please select the fields you want to export and choose if you prefer one line per copy (MEI id) or per provenance.

Export the following fields

ISTC No. x Author x Title x data.provenance.bindingType x

A new table row for each

MEI Identifier

Submit

JSON

Machine-readable format, for re-using the MEI data in your own scripts

JSON

JSON (one record per line)

Similar to JSON with one JSON object per record

JSON (one record per line)

YAML (plain text)

Text based format. Contains the complete record.

YAML (plain text)

Limit your search

by Author

Bartolus de Saxoferrato (3)

by Century

15 (2)

16 (2)

17 (1)

18 (1)

19 (1)

by Holding Institution

Choosing the appropriate **simplification** for your task

Going beyond the web interface – enrichment

- A standard case of **enrichment** is adding data from an external source, e.g. another database or even your own research
- **Example:** A spreadsheet of information about book purchases contains ISTC numbers; bibliographic information about the titles is then pulled from the database and put into the spreadsheet automatically
- **Challenges:**
 - You need to **match up the two data sources** with one another, e.g. by using unique identifiers
 - The two data sources may have **incompatible structures**, making it necessary to simplify both to a shared common core

Going beyond the web interface – we at DCG

- We want to make life easier for both domain and technology experts interested in working with CERL data (that includes ourselves)
- We provide **various serialization formats** for our data that help communicate to technology experts what can be done computationally with the data
- We **choose helpful ontologies** for our RDF representation that connect our data model to well-known standards in the GLAM world
 - However, standards develop over time, so this needs to be revisited (see also [my blog post](#) on the topic)
 - I would suggest forming a small CERL working group for people interested in the question of how to best present our data using existing ontologies (as an on-going task rather than a one-time decision)

Going beyond the web interface – ontologies

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rel: <http://purl.org/vocab/relationship/>.
@prefix gn: <http://www.geonames.org/ontology#>.
@prefix rdaGr3: <http://rdvocab.info/ElementsGr3/>.
@prefix skos: <http://www.w3.org/2004/02/skos/core#>.
@prefix gnd: <http://d-nb.info/gnd/>.
@prefix ct: <http://www.cerl.org/namespaces/thesaurus/>.
@prefix rdaGr2: <http://rdvocab.info/ElementsGr2/>.
@prefix rdaRelGr2: <http://metadataregistry.org/>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix wgs84_pos: <http://www.w3.org/2003/01/geopos/>.
@prefix edm: <http://europeana.eu/terms/>.
```

Status: **Deprecated**

The screenshot shows the 'open metadata registry' web interface. The main content area displays details for the 'RDA Group 3 Elements (deprecated)' element set. The status is clearly marked as 'Deprecated'. The interface includes a search bar at the top right and a 'Browse...' section on the right side. The main content area is divided into sections: Detail, Elements, History, Maintainers, Export, and Import. The 'Detail' section shows the following information:

- Owner:** Metadata Management Associates
- Label:** RDA Group 3 Elements (deprecated)
- Available Languages:** English
- Default Language:** English
- Status:** Deprecated
- Namespace:** http://rdvocab.info/ElementsGr3
- URI:** http://rdvocab.info/ElementsGr3
- URI:** http://rdvocab.info/ElementsGr3
- Publishing:**
 - URI: http://rdvocab.info/ElementsGr3
 - URI: http://rdvocab.info/ElementsGr3
- Documentation:**
 - URI: http://rdvocab.info/ElementsGr3
 - URI: http://rdvocab.info/ElementsGr3
- Users:**

Name	Administrator	Maintainer	Registrar
Gordon Dunsmuir	✓	✓	✓
Danesh	✓	✓	✓
Jon Phipps	✓	✓	✓

Who is using our data, and how?

- See the DCG paper on Linked Open Data from May 2020 for a small bibliography of **external projects** making use of our data to, e.g.:
 - Normalize their own data based on the Thesaurus
 - Answer questions about large-scale bibliographic trends
- But we also have use cases **inside CERL**:
 - Providing researchers with tabular representations of our data sets
 - Enriching externally created spreadsheets with data from our databases
 - Answering questions that cannot be answered by search
- **Coming up:**
 - We are providing MEI data for *Coding da Vinci*, a regional Hackathon in Hannover this year ([What is a hackathon?](#))

Going beyond the web interface: the DH community

- The DH community offers a lot of **ready-made tools** for secondary steps like visualization and analysis of data (still usually needed: data pre-processing, transformation, cleanup)
- For some possibilities of employing these tools for working with CERL data, see Alex Jahnke's talk at *Printing Revolution & Society 1450-1500. Venice Conference, Palazzo Ducale, 19-21 Sept. 2018*: [Watch on YouTube](#)
- Also, don't forget about the new [DH working group](#) at CERL (and join us?)
- But so far, we have seen **little uptake of our data** in that community
 - Is it there, but we don't hear about it? (Please tell us!)
 - Are there **barriers** to using our data?
 - Necessary domain expertise may not be available easily
 - It may simply not be well-known enough

Towards a „GLAM workbench“ for CERL

- There is still **a lot of untapped potential** in our data
- Tim Sherratt presents the idea of the [GLAM workbench](#), a **collection of tools and building blocks** for working with data from GLAM institutions
- We envision something similar for CERL:
 - Examples for making use of our data outside the interface
 - Explain our data from both perspectives (domain and technology)
 - Position as a “translation” between CERL community and DH community
- Thoughts on concrete implementation:
 - Collection of [Jupyter Notebooks](#) (heavily annotated Python code)
 - See also the National Library of Scotland’s [newly launched notebooks](#)
 - **DCG to provide some initial contents** (see next slides)
 - Aim to get more contributions from wider CERL community (and beyond)



Image credits: [Project Jupyter Logo by Project Jupyter Contributors](#)

The anatomy of a Jupyter Notebook

- Mixes Python code (or other languages) with detailed descriptions, making it possible to explain every step conceptually and technically
- Can be loaded in various environments as an interactive tool, exported as a script, or exported to a static reading version (e.g. a website or PDF)

What's a good name for a printer?

Plotting categorical data from the CERL Thesaurus on a bar chart

In this notebook, we will demonstrate how data from the CERL Thesaurus (or any other AMPLE database) can be visualized using the AMPLE API, Python and the visualization library `bokeh`. To keep the notebook free from clutter, we will be hiding some of the functionality in a Python library called `amplelib` which is currently under development and can be shared on request.

Setup

The library `amplelib` provides an `AmpleDB` class which can query an AMPLE instance, return search results and download entire records as Python dictionaries (equivalent to JSON). The library also provides a `ThesaurusRecord` class that provides functionality around such a Python dictionary, e.g. access by dot notation (as in the web interface search function).

```
In [26]: # Set up a connection to the CERL Thesaurus
from amplelib.adapters.ample import AmpleDB
thesaurus = AmpleDB('data.cerl.org', 'thesaurus')

# Wrapper around the JSON record
from amplelib.domain.record import ThesaurusRecord

# Set up the bokeh library
from bokeh.io import push_notebook, show, output_notebook
from bokeh.plotting import figure
output_notebook(hide_banner=True)
```

Detailed comments, which include formatting, links, images etc.

Block of code, with colored syntax highlighting for better readability

Querying AMPLE

The `.query()` method takes a search string with the same syntax as in the web interface. It returns a `QueryResult` object that provides access to the number of search results, the JSON representation of the search results and their IDs.

```
In [27]: # Search for printers related to Göttingen
result = thesaurus.query("related_to:cn100029316 AND type:cn1")
print(f"Found {result.hits} printers related to Göttingen")

Found 57 printers related to Göttingen
```

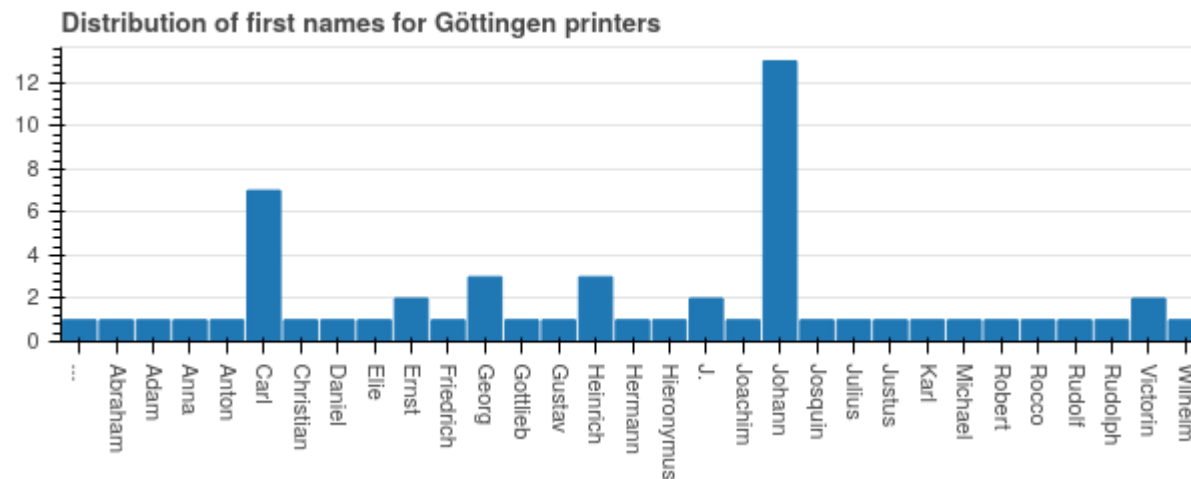
Block of code, with output printed below

Loading full records

As the search result JSON only includes abbreviated records, full records must be loaded from the IDs in the search result. The full records can be used to instantiate a `ThesaurusRecord` object.

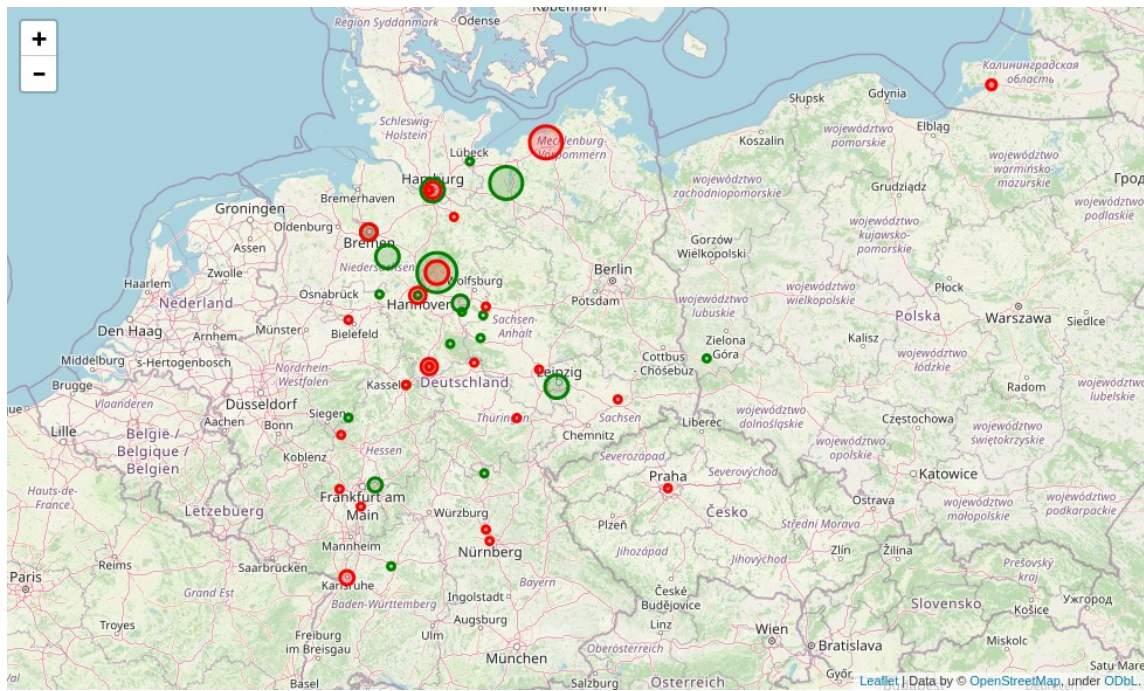
Some examples

- “What’s a good name for a printer from Göttingen?”
 - displaying **bar charts for categorical data** in the CERL Thesaurus
- [Preview of the notebook](#)



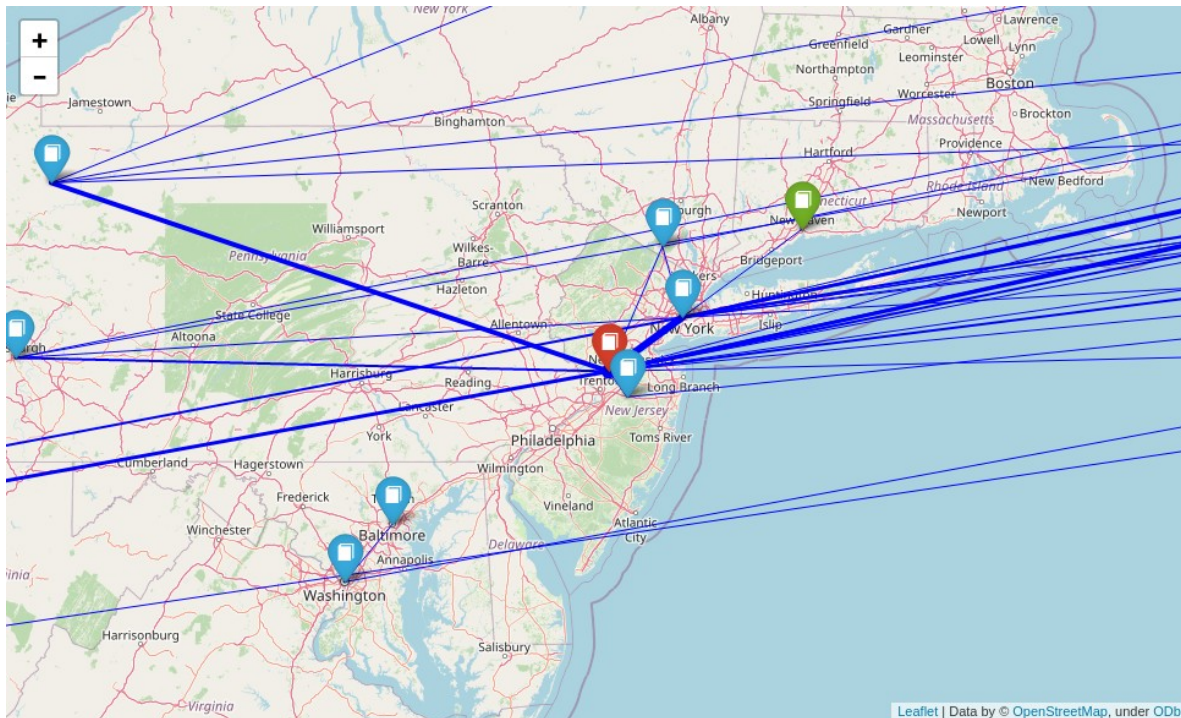
Some examples

- “Who came to Göttingen, and where did they go?”
 - displaying a **map view for geocoordinates** from the CERL Thesaurus
- [Preview of the notebook](#)



Some examples

- “How did these books come to Princeton?”
 - displaying a **map view for geographic networks** from MEI
- [Preview of the notebook](#)





NIEDERSÄCHSISCHE STAATS- UND
UNIVERSITÄTSBIBLIOTHEK GÖTTINGEN



Thank you

Contact: Andreas Walker (walker@sub.uni-goettingen.de)



GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Some more notebooks

- Biblioteca Virtual Miguel de Cervantes:
<http://data.cervantesvirtual.com/blog/notebooks/>
- National Library of Scotland:
<https://data.nls.uk/tools/jupyter-notebooks/>
- Tim Sherratt's [presentation](#) (LIBER Webinar)
- List of Jupyter notebooks beyond cultural data:
<https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks>