Providing new views on textual data with knowledge graphs

Workshop

Leo Born, Juri Opitz

HCH19
Ruprecht-Karls-Universität Heidelberg

July 19th 2019

A quick introduction

Why do we do what we do?

- Humanities researchers can be confronted with large bodies of text
  - Obtaining a bigger picture can be difficult
  - KGs can help to obtain such bigger picture

The KG motivation

- Transform a large body of text to a graph, storing the 'text's essentials'

What is a graph?

- In many cases, a collection of triples $\{(source_i, edge_i, target_i)\}_{i=1}^{N}$
  - e.g., $(birds, are\_capable\_of, flying)$

A little 'common sense' graph

$(bird, capable\_of, flying)$ $(plane, capable\_of, flying)$
$(mosquito, capable\_of, flying)$ $(bird, eats, mosquito)$
$(mosquito, annoys, human)$ $(mosquito, is\_a, animal)$

A more visually appealing view

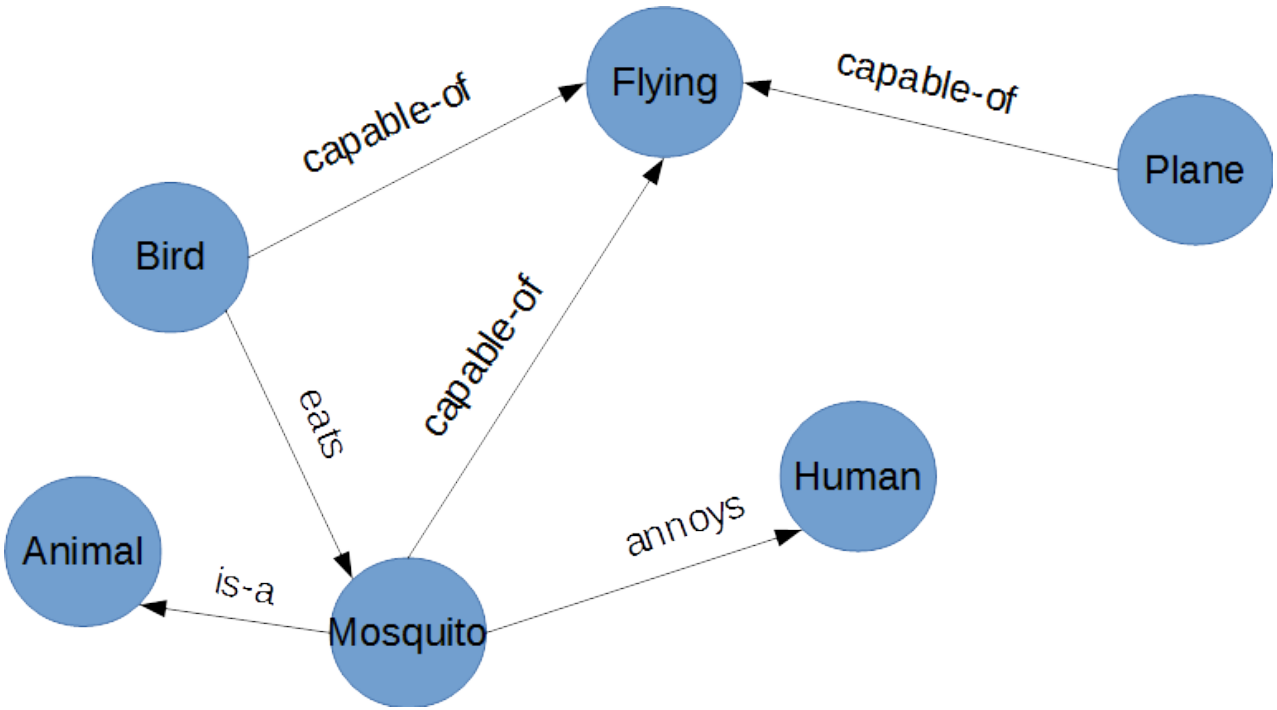


Figure: A more visually appealing view

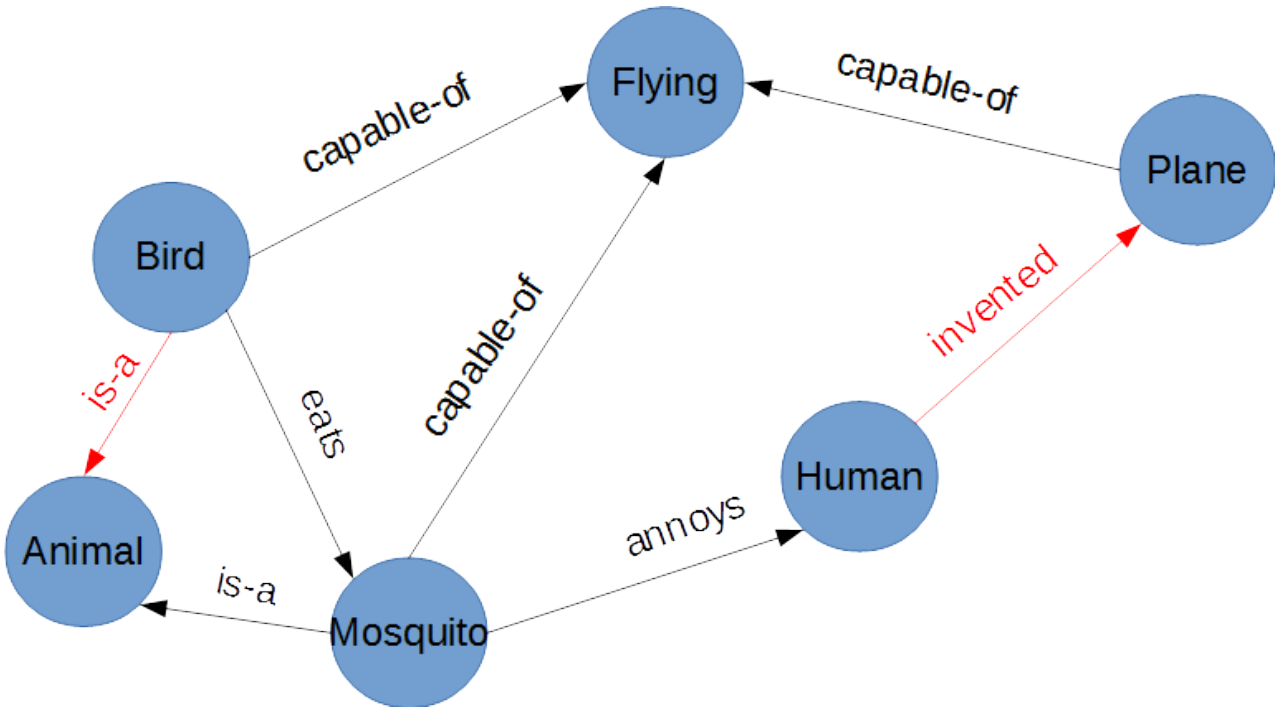


Figure: KGs are missing edges

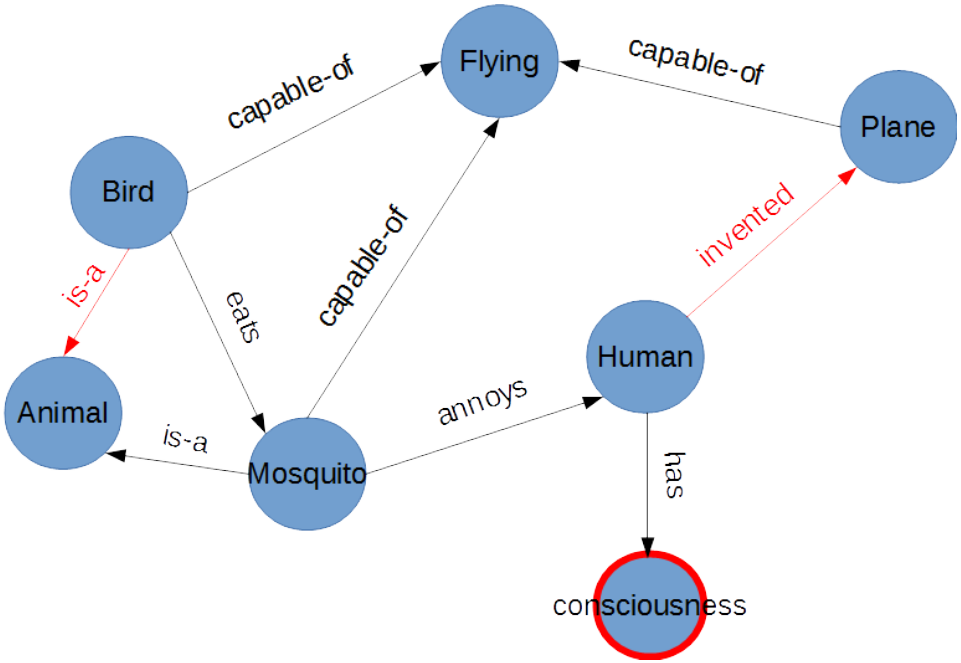


Figure: KGs are missing nodes

Intermediate insight

KGs are incomplete

Transforming historical corpora to KGs

A bunch of historical documents

S1271, f. 11r-v: A.D. 844 (? for 843). (1) Ceolred, bishop (of Leicester), to Berhtwulf, king of Mercia; grant of 14 hides (manentes) at Pangbourne, *s. xiii in.*

S278, ff. 11v-12v: A.D. 835 (Dorchester-on-Thames, Oxon.). Egbert, king of Wessex, to Abingdon Abbey; grant of 50 hides (manentes) at Marcha

S302, f. 12v: A.D. 854 (Wilton, Wilts., 22 April). Æthelwulf, king of Wessex, to the Church; general grant of land and privileges ('Second Decimati

S93, f. 13r: A.D. 726 x 737. Æthelbald, king, to St Mary's Minster, Abingdon; confirmation of lands and grant of 27 hides (cassati) at Watchfield a

S335x, f. 13v: A.D. 862 (Micheldever, Hants.). Æthelred, king of Wessex, to Æthelwulf, princeps; grant of 10 hides (cassati) at (Little) Wittenham,

S1201, f. 14r: A.D. 868. Æthelswith, queen of Mercia, to Cuthwulf, minister.; grant of 15 hides (manentes.) at Lockinge, Berks. *s. xiii in.*

S225, f. 14r-v: A.D. 878 for 915 (Weardburg, 16 Sept.). Æthelflæd, ruler of the Mercians, to Eadric, minister; grant of permission to acquire 10 hi
King Offa to Bynna, Wulflaf's great-great-grandfather (abavus), had been destroyed in a fire. *s. xiii in.*

S355, f. 16r-v: A.D. 892 x 899. Alfred, king of the Anglo-Saxons, to Deormod; grant of 5 hides (mansi) at Appleford, Berks., in exchange for land

S999, ff. 16v-17r: A.D. 1043. King Edward to Ælfstan, his minister; grant of 10 hides (mansae) at Sevington in Leigh Delamere, Wilts. *s. xiii in.*

S369, ff. 17v-18r: A.D. 903 (Southampton). King Edward to Tata, his fasallus; renewal of a charter of King Æthelwulf, king of Wessex, covering 3
by immersion. *s. xiii in.*

S404, ff. 18r-19r: A.D. 930. King Athelstan to Cynath, abbot; grant of 10 hides (mansiunculae) at Dumbleton, Gloucs., with 2 hides at Aston Som
recording King Edgar's confirmation of the land to Osulf, bishop of Ramsbury (A.D. 959 x 970). *s. xiii in.*

S409, f. 19r-v: A.D. 931. King Athelstan to the church of St Mary, Abingdon; grant of 12 hides (cassati) at Shellingford, Berks. *s. xiii in.*

S410, ff. 19v-20r: A.D. 931. King Athelstan to the church of St Mary, Abingdon; grant of 5 hides (cassati) at Swinford, Berks. *s. xiii in.*

S408, f. 20r-v: A.D. 931. King Athelstan to the church of St Mary, Abingdon; grant of 15 hides (cassati) at Sandford, Oxon. *s. xiii in.*

S1208, ff. 20v-21r: c. A.D. 931. Athelstan, senator, to St Mary's, Abingdon; grant of land at Uffington, Berks. *s. xiii in.*

A bunch of historical documents

S1271, f. 11r-v: A.D. 844 (? for 843). (1) Ceolred, bishop (of Leicester), to Berhtwulf, king of Mercia; grant of 14 hides (manentes) at Pangbourne, Berks., in return for the freedom
s. xiii in.

S278, ff. 11v-12v: A.D. 835 (Dorchester-on-Thames, Oxon.). Egbert, king of Wessex, to Abingdon Abbey; grant of 50 hides (manentes) at Marcham, Berks. *s. xiii in.*

S302, f. 12v: A.D. 854 (Wilton, Wilts., 22 April). Æthelwulf, king of Wessex, to the Church; general grant of land and privileges ('Second Decimation'). *s. xiii in.*

S93, f. 13r: A.D. 726 x 737. Æthelbald, king, to St Mary's Minster, Abingdon; confirmation of lands and grant of 2
es (cassati) at Watchfield and 10 by Ginge Brook, Berks., wit

S335x, f. 13v: A.D. 862 (Micheldever, Hants.). Æthelred, king of Wessex, to Æthelwulf, princeps; grant of
i) at (Little) Wittenham, Berks. *s. xiii in.*

S1201, f. 14r: A.D. 868. Æthelswith, queen of Mercia, to Cuthwulf, minister.; grant of 15 hides (
s. s. xiii in.

S225, f. 14r-v: A.D. 878 for 915 (Weardburg, 16 Sept.). Æthelflæd, ruler of the Mercians
to acquire 10 hides (manentes) at Farnborough (W
King Offa to Bynna, Wulflaf's great-great-grandfather (abavus), had been destroyed

S355, f. 16r-v: A.D. 892 x 899. Alfred, king of the Anglo-Saxons, to Deorm
, Berks., in exchange for land at Harandun (Horn Down near Eas

S999, ff. 16v-17r: A.D. 1043. King Edward to Ælfstan, his minister;
n in Leigh Delamere, Wilts. *s. xiii in.*

S369, ff. 17v-18r: A.D. 903 (Southampton). King Edward t
of King Æthelwulf, king of Wessex, covering 3 hides (manentes) at Hardwell in C
by immersion. *s. xiii in.*

S404, ff. 18r-19r: A.D. 930. King Athelstan to
aunculae)
recording King Edgar's confirmation of th
D. 959 x 970).

S409, f. 19r-v: A.D. 931. King At
don; grant of 12 hides

S410, ff. 19v-20r: A.D. 931. Ki
Mary, Abingdon; grant of 5 hide

S408, f. 20r-v: A.D. 931. King Athe
St Mary, Abingdon; grant of 15 hides

S1208, ff. 20v-21r: c. A.D. 931. Athels
tor, to St Mary's, Abingdon; grant of land at Uf

S413, ff. 21r-22r: A.D. 931 (Worthy, Hants., 20 June). King Athelstan to Ælfric, minister; grant

S1604, f. 22r-v: (King Athelstan to ?; grant of land) at Bultheswrthe. *s. xiii in.*

S411, ff. 22v-23v: c. A.D. 935 x 938 (? 937). King Athelstan to Ælfheah, minister; grant of 10 h

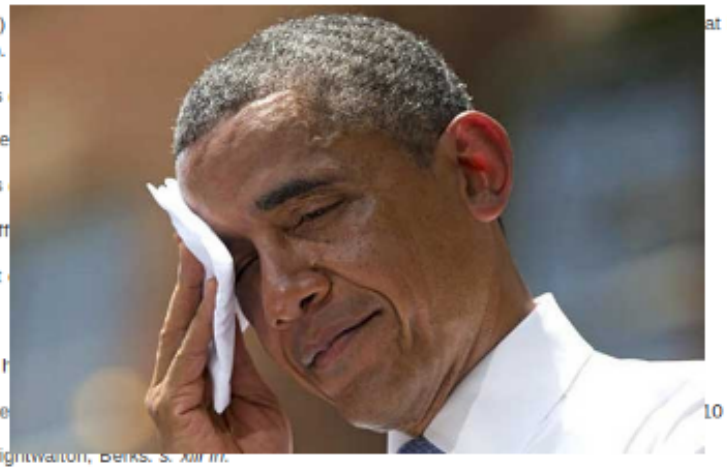
S396, ff. 23v-24r: A.D. 926. King Athelstan to Ealdred, minister; confirmation of 5 hides (mane
10 p

S448, f. 24r-v: A.D. 939. King Athelstan to Eadwulfu, a nun; grant of 15 hides (mansae) at Brightwalton, Berks. *s. xiii in.*

S1567, f. 25v: Bounds of Culham, Oxon. *s. xiii in.*

S471, ff. 25v-26v: A.D. 940 (? for 943). King Edmund to Wulfric, minister; grant of 15 hides (mansae) at Garford, Berks. *s. xiii in.*

There are thousands more!



A pattern

But wait, there appears to be a simple formal pattern

A pattern

S408, f. 20r-v: A.D. 931. King Athelstan to the church of St Mary, Abingdon; grant of 15 hides (cassati) at San

S1208, ff. 20v-21r: c. A.D. 931. Athelstan, senator, to St Mary's, Abingdon; grant of land at Uffington, Berks. s

S413, ff. 21r-22r: A.D. 931 (Worthy, Hants., 20 June). King Athelstan to Ælfric, minister; grant of 20 hides (cas

S1604, f. 22r-v: (King Athelstan to ?; grant of land) at Bultheswrthe. *s. xiii in.*

S411, ff. 22v-23v: c. A.D. 935 x 938 (? 937). King Athelstan to Ælfheah, minister; grant of 10 hides (manentes

Figure: A pattern

A pattern

a SUBJECT (the king) does SOMETHING (e.g., grant) to SOMEONE (e.g. church of St. Mary)

A pattern

X to Y; Z

A pattern

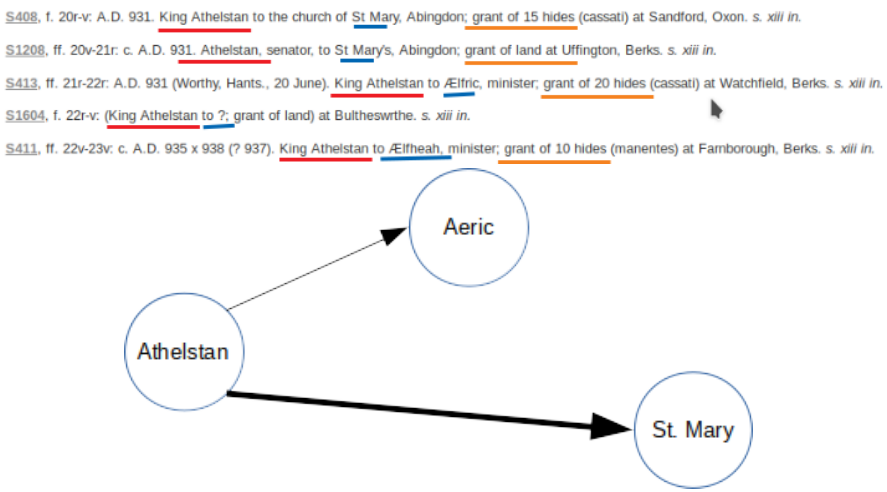


Figure: A high relation 'weight' can indicate a stronger relationship

A pattern

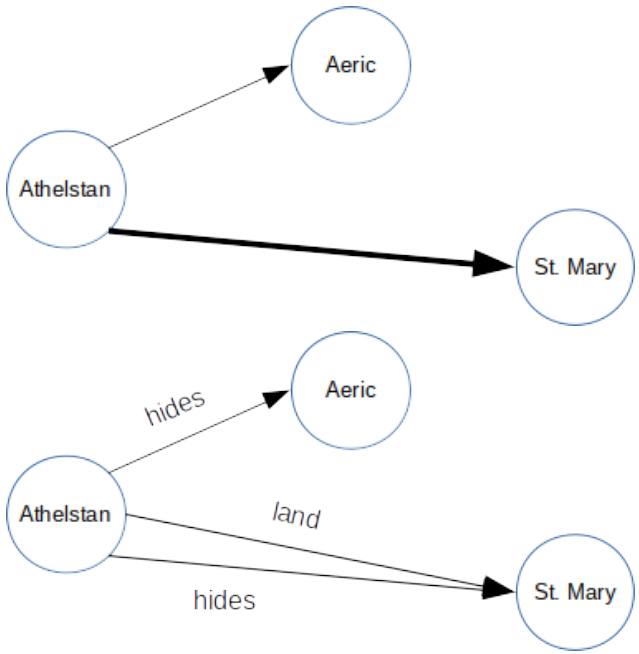


Figure: Design choices

Finally, we have something like this

Let's get our hands dirty

- start a terminal, type 'python'

```
import spacy
nlp = spacy.load('en_core_web_sm')
doc = nlp('Athelstan grants land to St. Mary\'s')
spacy.displacy.serve(doc, style='dep')
```

open the link and discuss. Do you spot an error? Try with other sentences.

Let's get our hands dirty

```
doc = nlp('Athelstan grants land to St. Mary\'s')
for chunk in doc.noun_chunks:
    print(chunk.text)
```

what do you see? What are 'noun chunks'?

Let's get our hands dirty

```
import networkx
G = networkx.DiGraph()
chunks = [chunk.text for chunk in doc.noun_chunks]
G.add_edge(chunks[0], chunks[2], label=chunks[1])
import matplotlib.pyplot as plt
plt.ion()
networkx.draw_networkx(G,with_labels=True)
```

what do you see? What have we done here?

Excercises

1. Add two more triples to the graph
   - one triple where at least one node is already in the graph
   - and one triple with two new nodes
2. make sure that there is one node which is connected to every other node
3. use networkx.info(G) and discuss the results
4. play a little bit around with the graph G (as in 1 or 2) and observe statistical changes with 3

End of part I

Q/A

Part II

Let's get our hands even more dirty!

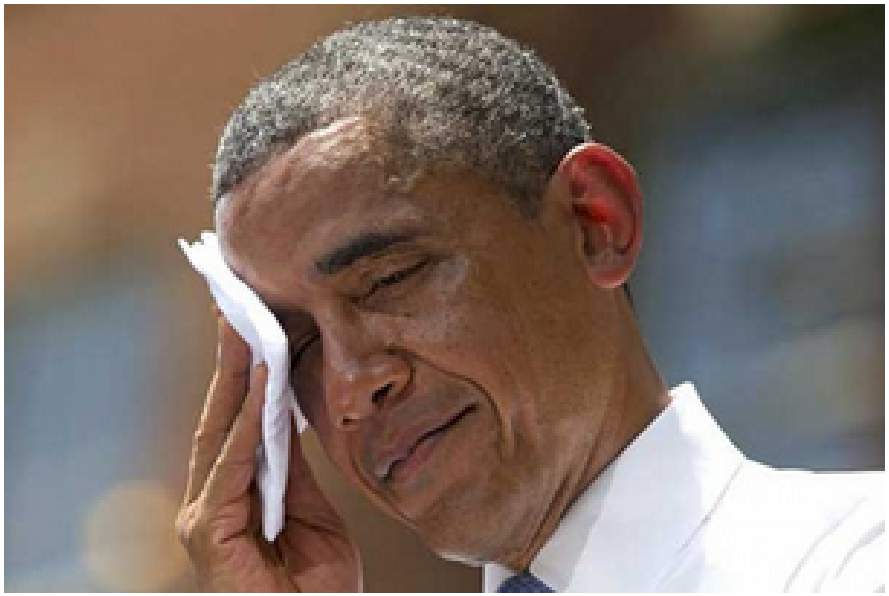
Figure: What do you see?

Old Bailey corpus

- contains transcripts from historical trials in London[1]
  - Structured and annotated
  - 18th, 19th and 20th century
  - 637 'proceedings'

[1] http://fedora.clarin-d.uni-saarland.de/oldbailey/index.html

24.4 mio. spoken words!



Old Baileys 2 KG

- aim 1: test hypotheses (e.g. were males and females differently punished in historical London, did that change over time?)
- aim 2: explore the KG, what are centered nodes? Was there a person sentenced multiple times, etc.?

With a KG, we can engage these and many more questions in a very straightforward way

Plain text vs. in-depth annotations

- What information you can use depends on...
  - ... whether your data is structured (e.g. XML/TEI-annotated) or unstructured (plain text)
  - ... what language it is in and what tools there are
  - ... what you want

Intermediate insight

- In the simplest case, we start from **high-quality and extensive annoations**
- In the hardest case, we start from **plain text**
  - make out and exploit **formalized patterns** (e.g. the charters we have seen)
  - use automatic extraction tools, e.g. extract subject-verb-object triples with **dependency parsing**
    - caveat: not every language follows SVO-patterns...

Old Baileys 2 KG

Luckily for us, the corpus has been extensively annotated by a large research project[2]

[2] http://www1.uni-giessen.de/oldbaileycorpus/

Design choices

We want:

- trial nodes (ids), named entity nodes (e.g., the defandant's name), offence nodes (e.g., theft), description nodes (e.g. what was stolen), punishment nodes (e.g., prison)
- edges to connect trial nodes to defendants, punishments etc.

OBC examples

Some examples...

Extracting the defendant

```
<persName id="t17751206-3-defend341" type="defendantName">
WILLIAM
CLARKE
<interp inst="t17751206-3-defend341"
type="surname" value="CLARKE"/>
<interp inst="t17751206-3-defend341"
type="given" value="WILLIAM"/>
<interp inst="t17751206-3-defend341"
type="gender" value="male"/>
</persName>
```

Extracting the offence

```
<rs id="..." type="offenceDescription">
<interp inst="..." type="offenceCategory" value="theft"/>
<interp inst="..." type="offenceSubcategory"
value="grandLarceny"/>
stealing two gold and three silver watches,
and about 80 l. in money
</rs>
```

Extracting the offence

```
<rs id="..." type="offenceDescription">
<interp inst="..." type="offenceCategory" value="theft"/>
<interp inst="..." type="offenceSubcategory"
value="grandLarceny"/>
stealing two gold and three silver watches,
and about 80 l. in money
</rs>
```

What was stolen? This is more difficult to extract ... it is not annotated

NLP to the rescue

If things are not annotated, but annoation is very desirable, we must automatically 'annotate' them

Let's parse this text

- start a terminal, type 'python'

```
import spacy
nlp = spacy.load('en_core_web_sm')
doc = nlp('stealing two gold and three silver watches')
spacy.displacy.serve(doc, style='dep')
```

open the link and discuss. Do you spot an error? Does it help us to see what exactly was stolen?

Excercise

- insert a few random empty spaces e.g., ‘stealing two gold and three silver watches’.
  - Discuss what happens
- insert: ‘on the 10th of December 1827’ between ‘stealing’ and ‘two’
  - Discuss what happens

Intermediate insight

NLP systems are a bit like 'princesses on peas'

$\Rightarrow$ a super small change in the environment can easily disturb them

Figure: Princess on a pea.

However

... sometimes it's okay if we don't catch everything.

- catching only the word 'gold' or 'watches'
  - would certainly be better than catching nothing
  - and probably also better than using the full text as a stolen-item-node
    - Question: why?!

OBC2KG

So, how do we build a KG from the OBC?

OBC2KG

- start a terminal, type 'git clone https://gitlab.cl.uni-heidelberg.de/born/obc2kg.git'
- type 'cd hch-kg', then 'ls -l'
  - data/ contains small subset of OBC
  - output/ is used to store constructed knowledge graphs
  - src/ contains scripts
  - visualization/ contains visualization suite

OBC2KG

- type 'cd src', then 'ls -l'
- script *graph_builder.py* does all the heavy lifting
  - iterates over data files
  - extracts, for each trial, all nodes and texts
- we will only interact with *main.py*
  - allows for invoking text simplification function from *graph_helpers.py*

OBC2KG

- type 'python main.py -h' to show all available options
- example data contains OBC data for 1720, 1820, and 1913

OBC2KG

- create a graph for the year 1720:
  - type 'python main.py -year 1720 -output_path "../output/example_graph_1720.json"'

OBC2KG: Analysis

- analyze the graph by using *graph_stats.py*
  - type 'python graph_stats.py -general ../output/example_graph_1720.json'

- type 'python graph_stats.py -h' to show all available options
- **Exercise** 1: What are the 10 most central nodes?
- **Exercise** 2: What is the distribution of offences?
- **Exercise** 3: Play around with the other categories

OBC2KG: Analysis

- **Exercise**: do the same for 1820 or 1913
- compare the stats to 1720
  - what differences – if any – do you see?

OBC2KG: Analysis

- **Exercise**: re-run the stats script with '-detail description'
  - what do you notice?

OBC2KG

- recall that text descriptions can be very long
  - e.g. "stealing two gold and three silver watches"
- by simplifying them, we can reduce these to just the most important words/phrases
  - e.g. *gold* or – ideally – *watches*

OBC2KG

- simplification is possible with flag '-text_node_simplification' of *main.py*
  - **Exercise**: create two simplified graphs for one year
  - one using '-text_node_simplification spacy_direct_object'
  - the other using '-text_node_simplification classifier'
- print the stats on descriptions for the new graphs
  - what do you see?

OBC2KG

Terminals are great and stuff, but wasn't there a more interactive and appealing way to look at the KGs?

OBC2KG: Visualizations

- 'cd ../visualization'
- 'python -m http.server'
- open 'http://localhost:8000' in your browser
- open the file browser and go to the output directory

OBC2KG: Visualizations

- Map some node types onto each other, e.g. offence and description
  - What do you see? What kind of descriptions are associated with the offences?
- Do the same with a graph from a different time
  - How do the mappings compare?
  - Did anything change?

OBC2KG: Visualizations

- Visualizations can be a great tool to explore data in a more **intuitive** way
- Looking at diverse transformations or structures of graphs, questions can arise that were not thought about before:
  - Why were verdicts for sexual offences more often *not guilty* in the 18th century and more often *guilty* in the 20th century?
  - Who were the people involved in multiple trials? Do they have any commonalities?
- If, on the other hand, you have specific questions in mind, coding yourself to an answer might give you **more than a visualization tool**

Addendum: plotting gender distributions over the whole corpus

- we have prepared a stats file (*gender_punishment_time.csv*) for plotting
- with it, you can plot the distribution of verdicts and punishments per gender over time

- type 'python plot_gender.py imprison' to generate file *imprison-gender-time.png*
- open it: 'okular imprison-gender-time.png'
  - what do you see?

An example: Tracing the 'invention' of imprisonment as punishment

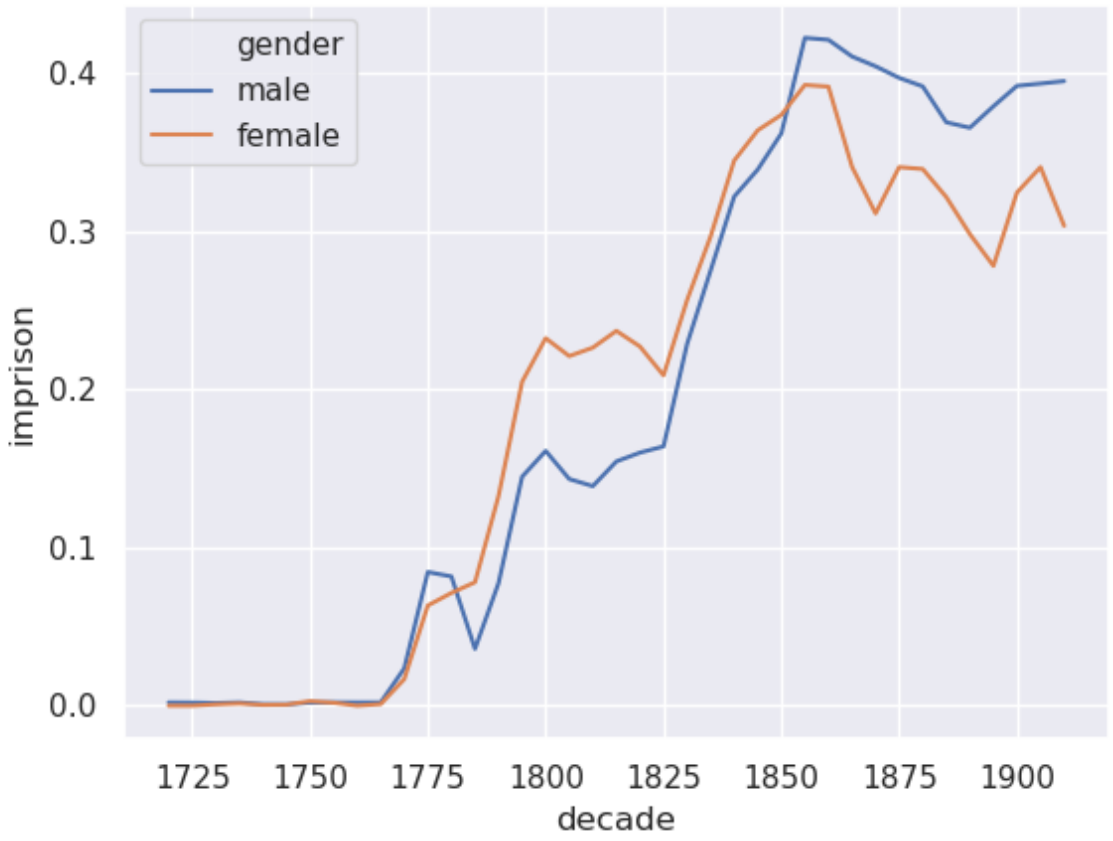


Figure: Ratio of males and females which were sentenced to prison.

OldBailey2KG code repository

`https://gitlab.cl.uni-heidelberg.de/born/obc2kg`

- caveat: code may be not free from bugs and some things may not be modeled ideally
- if you want to build on this work and have questions, don't hesitate to contact us

Contact

lastname@cl.uni-heidelberg.de

Pointers

- visualization: `https://visjs.org/`
- spacy: `https://spacy.io/`
- networkx: `https://networkx.github.io/`
- KG of the Regesta Imperii [OBN18, BON18]
- Holy Roman Emperor itineraries [OBNP19]

References


Leo Born, Juri Opitz, and Vivi Nastase.
A knowledge graph from the regesta imperii: Construction, visualization and macro-level analyses.
In *Inaugural Conference of the European Association for Digital Humanities (EADH)*, Galway, Ireland, 2018.


Juri Opitz, Leo Born, and Vivi Nastase.
Induction of a large-scale knowledge graph from the Regesta Imperii.
In *Proceedings of the Second Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 159–168, Santa Fe, New Mexico, August 2018. Association for Computational Linguistics.


Juri Opitz, Leo Born, Vivi Nastase, and Yannick Pultar.
Automatic Reconstruction of Emperor Itineraries from the Regesta Imperii.
In *Proceedings of the 3rd Conference for Digital Access to Textual Cultural Heritage (DATeCH)*, Brussels, Belgium, 2019.
to appear.