CLSINFRA COMPUTATIONAL LITERARY STUDIES INFRASTRUCTURE

Metadata

CLS Infra Summerschool Prague 2022

Corpus Counting

CLSINFRA

- Corpora used for two purposes:
  - Finding things
  - Counting things
- Mostly in context
  - You want to find word by a certain speaker, a certain play, etc.
  - You typically want to compare numbers when counting
  - Only for pe. pure syntax just occurrences

Comparing Numbers

CLSINFRA

- There are 57 occurrences of “yea” in Dracor Shakespeare
  - Is that a lot?
  - There are 53 *per million words*
  - Compared to 307 occurrences of *even*
  - It occurs in 26 of the 37 plays at least once
  - Most frequently in *Cymbeline* (5 times)
- Numbers are always relevant
  - Compare occurences in one type of context with those in another
  - We need to know things *about* the text : metadata

Spreadsheets

CLSINFRA

- Traditionally, metadata were kept in spreadsheet
  - Rows with the title or filename, plus some relevant fields
  - Sometimes no information was kept at all
  - Easy to misplace, misalign, misinterpret
- Only works for text-level metadata
  - Name of the speaker in a play
  - Number of the chapter in a book
- XML based metadata in corpora
  - <text title="Cymbeline">

Metadata in TEI/XML

- Metadata kept together with the transcription

<TEI>
<teiHeader/> ***metadata***
<text/> *transcription*
<facsimile/><standOff/><sourceDoc/><fsdDecl/>
</TEI>

Dublin Core

CLSINFRA

- 15 standard fields to keep about text sources
  - Although moved to linked (open) data
- Creator, Contributor, Publisher, Title, Date, Language, Format, Subject, Description, Identifier, Relation, Source, Type, Coverage, and Rights
  - Basically a spreadsheet
- Not always simple
  - Alternative titles
  - Date of various versions of the text
  - Multiple languages

teiHeader

- Very expressive way to define metadata
  - Multiple layers with (relatively) strict definitions
  - Soft standard: multiple ways to express the same thing

```
<teiHeader>
    <fileDesc/>          Description digital file (bibliographic)
    <encodingDesc/>  Description encoding standard
    <profileDesc/>     Description text (non-bibliographic)
    <revisionDesc/>   Description of changes to the file
    <xenoData/>
</teiHeader>
```

profileDesc

< profileDesc >

<langUsage/> *The language(s) in the text*
<textDesc/> *Description text – channel, purpose, etc.*
<textClass/> *Classification text – genre, taxonomy, etc.*
and more

</ profileDesc>

fileDesc

CLSINFRA

< fileDesc >

<titleStmt/> *Main bibliographic: title, author, date*
<sourceDesc/> *Description source transcribed in the file*
<notesStmt/> *Anything else (often unstructured data)*
<editionStmt/><extent/><publicationStmt/><seriesStmt/>

</ fileDesc>

sourceDesc

< sourceDesc >

<bibl/> *Bibliography source – title, author, date, IDs*

<recordingStmt/><listPerson/><msDesc/>

</ sourceDesc>

Which Metadata

- XML is gracefully degrading
  - You can leave out or add info without breaking anything
- Keep ALL the info you have about a file
  - Even data you have for only one file in your corpus
  - In the teiHeader you can never loose the data
- GDPR
  - Technically speaking, you cannot legally build corpora
  - Copyright rules force you to keep data about the author
  - Privacy rules forbid you to store identifying information

XPath Definitions

- Each TEITOK project defines the relevant fields by XPath
  - Language to specify nodes in XML files
- Define which fields to be editable in the interface
  - Possibly with predefined values (dropdown select)
- Define which fields to export to the searchable corpus
  - With definitions of the type of field to define the search

XPath

- XPath defines the hierarchical position of a node
  - Much like the file path for a file on your computer
  - C:\Documents\Newsletters\CLSInfra.pdf
  - /TEI/teiHeader/fileDesc/sourceDesc/bibl/title
- Results are always list
  - There can be multiple/TEI/teiHeader/fileDesc/sourceDesc/bibl/title
- Double slashes are used for “any number of levels down”
  - //tok refers to any <tok> node in the XML
  - //teiHeader//title refers to any <title> inside a <teiHeader> somewhere
- Node restrictions (attributes) between square brackets
  - //title[@level=“a”] refers to any <title level=“a”>

XPath (2)

CLSINFRA

- Relative positions
  - . refers to the current node, .. to the parent of the current node
  - //p/..//tok – refers to any <tok> inside the parent of a <p> node
- Restrictions can be based on children
  - //p[tok[@lemma=”walk”]]
    - refers to any <p> that has a child <tok lemma=“walk”>
- Axes
  - //tok[@id=“w-10”]/preceding::lb
    - refers to the <lb> preceding <tok id=“w-10>
- Any single column refers to a namespace tei:tok

Below text level

- Parts of texts can have metadata as well
  - In TEITOK – any collection of tokens (but nothing below)
- Attributes over nodes around <tok>
  - <l metric="-+|-+|-+">To be or not to be</l>
- For defined region types – you can define attributes
  - View the text line by line
  - With metadata (attribute-value pairs) below them
- XML Layout editor (TEITOK)

Stand-Off

CLSINFRA

- Various types of text analyses can be self-overlapping
  - *Un gran hombre pequeño*
  - <pair>gran hombre</pair>
  - <pair>hombre pequeño</pair>
- Solved by using stand-off annotations over tokens
  - Identified by IDs
  - <seg corresp=”#w-2 #w-3” position=“pre”/>
- Can be disjoing
  - *The coat of the man in the livingroom was red*
  - <seg corresp=“#w-2 #10” position=“predicative”/>

Corpus Export

CLSINFRA

- (Selected) XML regions exported over sequences of tokens
  - Flat structure with a simple name
    - /TEI/teiHeader/fileDesc/titleStmt/author => <text author="XXX"/>
  - Always calculated explicitly in case of implicit/referenced values
- Empty nodes can be exported as full regions
  - <lb n="1"/> <tok>… => <lb n="1"><tok>….</tok></lb>
- Standoff can be exported as full regions
  - <seg corresp="#w-10" type="1"/>
    - <anno type="1"><tok id="w-10">word</tok></anno>

CLS Infra Corpus

XPath	Display	CQP Field
/TEI/teiHeader/fileDesc/titleStmt/title	Title	title
/TEI/teiHeader/fileDesc/titleStmt/author	Author	author
/TEI/teiHeader/fileDesc/titleStmt/date/@when	Year	year
/TEI/teiHeader/profileDesc/textClass	Text type	type
/TEI/teiHeader/profileDesc/langUsage/language/@ident	Language (ISO)	lang
/TEI/teiHeader/fileDesc/titleStmt/respStmt/resp[@n=“editor”]	Editor	editor