CLSINFRA COMPUTATIONAL LITERARY STUDIES INFRASTRUCTURE

XML, TEI and TEITOK

CLS Infra Summerschool Prague 2022

TEITOK

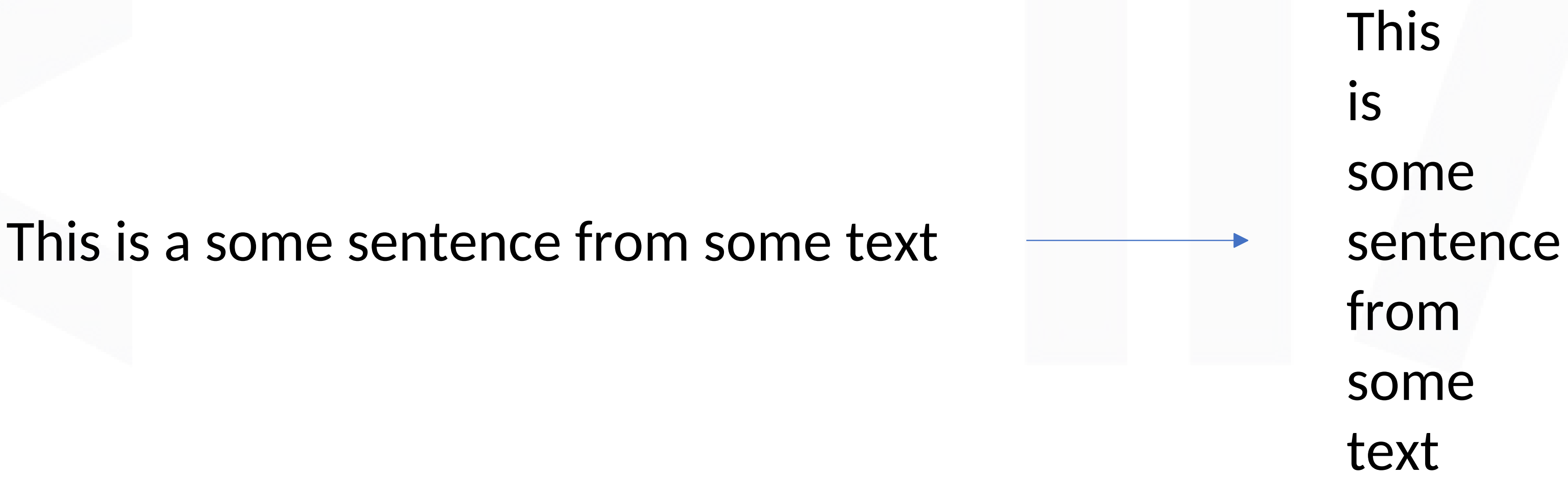
- Online environment for working with annotated tokenized TEI/XML based corpora
  - Create and manage your own corpus
- What is a corpus?
  - What is corpus annotation?
- What is tokenization?
- What is XML?
- What is TEI/XML?

Corpora

- Collections of texts
  - Finding things
  - Counting things
- Representativeness
  - A balanced selections of texts to represent the language
- DracorShake
  - Programmable corpus from within the CLS infra project
  - Texts by Shakespeare
- Corpus for this course
  - Texts you each enter
  - http://www.teitok.org/cls

Tokenization

- Split a text into words
  - Traditionally by putting each word on a line
- Obtaining *tokens* (as opposed to *types*)



This is a some sentence from some text

This
is
some
sentence
from
some
text

Tokenization (2)

- Not a homogeneous notion
  - Various things can count as a “word”
- Graphical tokens
  - Used in OCR: any continuous text block
  - Punctuation part of the token
  - 2 tokens when a word is broken across a line
- Orthographical tokens
  - Anything between two spaces – with punctuation marks split off
- Grammatical tokens
  - *Can’t* consists of two “words”: *can* and *not*
- Not considered

  Fonetic tokens, morphological tokens

Building a Corpus

- Document cleaning
  - Extract text from any document, only words (and paragraphs)
- Throw away any “mark-up”
  - Any non-text (images, graphical elements, page numbers, etc.)
  - Any placement information (titles, tables, margins, columns, etc.)
  - Any font changes (bold, italics, large, small, superscript, etc.)
- TEITOK does NLP without document cleaning
  - All information in the original is kept and shown

Mark-up

- This is a **piece** of text
- The word *piece* is written in bold face
  - Needs to be marked in our document somehow
- Two types of mark-up:
  - Stand-off: characters 11-15 are in bold face
  - In-line: put something around it in the text: This is a *piece* of text
- Famous mark-up languages
  - HTML: HyperText Mark-up Language
  - XML: eXtensible Mark-up Language

XML

- Marking a “word”: putting a *tag* around it
  - Start-tag before, end-tag after
  - <> + name of tag + / for the end tag
  - <bold>piece</bold>
- Adding information to the tags (attributes)
  - Inside the tag: name of the attribute + = + value (between quotes)
  - <typesetting type=“bold”>piece</typesetting>
- Language “flavour” defines tags and meaning
  - HTML:<b> = bold face
  - TEI: <hi rend=“bold”> = highlighted, using bold face

XML (2)

- XML has to be *valid*
  - Syntactically valid – “proper” XML
  - Semantically valid – only using tags defined by the “flavour”
- All tags have to be closed
  - This is a <b>piece of text
- Everything has to be inside a tag
  - <p>This is a <b>piece</b> of text.</p>
- Tags cannot cross
  - <a>Some <b>markup</a> example</b>
- Reserved characters have to be *escaped*
  - No > in an XML text -- you have to use pe. &gt;

XML Display in TEITOK

- XML tags do not have a rendering by themselves
  - Only some XML tags are typographic to start with
- TEITOK lets the browser display the XML
  - XML loaded directly into the HTML page
  - Style sheets (CSS) to define how each tag should be displayed

TEI/XML

- TEI - a standardized framework for digital texts
  - Text Encoding Initiave – XML flavour
- Here, for transcribing source material
  - Faithfully capturing the (relevant) content of the source
  - Mostly initially about standardized philology
- Described at http://tei-c.org

TEI/XML (2)

- General Structure

<TEI>

<teiHeader/> *metadata*

<text/> ***transcription***

<facsimile/><standOff/><sourceDoc/><fsdDecl/>

</TEI>

TEI/XML Tags

- General tags

<p>	Paragraph	
<head>	Any type of header	
<hi>	Highlighted text	@rend – how it was highlighted

Manuscript tags

<add>	Text added later	
<del>	Text deleted by the author	@rend - how it was deleted
<gap/>	Bit missing in the <text>	@reason – why there is a gap
<supplied>	Text added from other source	

TEI/XML Tags (2)

- Spoken tags

<pause>	Pause	@duration – length of the pause
<del>	Retracted speech	@type – repitition, truncation, reformulation
<u>	Utterance	@who – speaker

Other tags

<l>	Verse line	@metric – metric analysis
<lg>	Line group (strofe)	
<foreign>	Bit in another language	@ident – ISO of the language
<stage>	Stage instructions	

Incompatible tags

- Some TEI tags are incompatible with a traditional corpus
  - Mostly those that define multiple texts in a single TEI/XML file
  - They lead to multiple corpora, not a single corpus
  - Not supported in TEITOK or similar tools like TXM

<choice>	Choice between versions	<org> - original / <reg> - regularized version <abbr> - abbreviation / <expan> - expansion
<app>	Apparatus (multiple witnesses)	<rdg> - reading in one witness <lem> - lemma (preferred reading) <rdgGrp> - group of <rdg>

TEITOK Tokenization

- TEITOK works mostly on tokenized TEI/XML documents
  - Start with a non-tokenized TEI/XML file
  - Add inline tokenization
  - Done with a simple click
- TEITOK uses <tok> for token
  - Standard TEI uses <w> for word – and <pc> for punctuation character
  - Tokens are orthographic – they can contain grammatical tokens

<p>A small text.</p> *becomes*

<p><tok>A</tok> <tok>small</tok>
<tok>text</tok><tok>.</tok></p>

Grammatical tokens

- Orthographic tokens can contain multiple grammatical tokens
  - Called <dtok/> - which do not have an inner value
  - <tok>

    ```
     can’t
    <dtok form=“can”/>
    <dtok form=“not”/>
    </tok>
    ```
  - Most tokens have at least one implicit <dtok> below them
    - <tok>can</tok> === <tok>can<dtok form=“can”/></tok>
    - Except (typically) for deleted tokens with no grammatical tokens: <tok><del>can</del></tok>

TEITOK Annotation

- Annotation (primarily) over tokens
- Added a attributes (+value)
  - You have to define your annotations (lemma, pos, deprel, etc.)
- Regularization
  - not halfe so bigge as a round little Worme,
  - <tok>bigge</tok> => <tok reg=“big”>bigge</tok>
- Annotations added/corrected by simply clicking on the word

TEITOK as a GUI interface

- Annotated tokenized TEI/XML files quickly become large
  - Virtually impossible to edit by hand
- TEITOK attempts to help in that
  - Uncluttered display
  - Editing directly using HTML forms

Corpus export

- A linguistic corpus in TEITOK is the sequence of all <tok>
- Search using XML parts - XPath or XQuery.
    - Inefficient both in speed and in expressiveness
    - XML used indirectly in pe. existDB
- Search directly using dedicated search tools
    - There are systems that do that, like BlackLab
- TEITOK exports the TEI/XML documents to a corpus tool
    - Corpus WorkBench – CQL
    - Make a VRT file (one-word-per-line with columns, TSV)
    - Create an indexed corpus