Corpus Linguistics, Annotation

Kron

Goals of this lecture

- Focus on annotation:

1. what makes a good annotation scheme;
2. what standards exist;
3. what markup languages exist.

Corpora and annotation

- Unannotated corpora:
  - simple plain text
  - the linguistic information is implicit
  - e.g. no explicit representation of *man* as a noun
- Annotated corpora:
  - no longer just text
  - real repositories of linguistic information
    - the relevant linguistic information is now explicit

Types of corpora

- Corpora are often defined according to what kind of annotation they contain.
  - part-of-speech annotation (tagging)
    - annotation of morphosyntactic categories (BNC)
  - parsed corpora (treebanks)
    - annotation of syntactic structure (Penn Treebank, SMULTRON)
  - anaphora
    - annotation of pronominal coreferents in context (GNOME corpus)

How is it done?

- Depends on the type of annotation being carried out.
- Many kinds of annotation are done manually.
- Some kinds of annotation, especially POS tagging can be done semi-automatically:
  - many available POS taggers
  - start with a manually tagged sample of text
  - train the tagger on the sample
  - tagger is then applied to new data, and tries to "guess" the POS of new words
  - this is not an error-free process! Current state of the art achieves about 96-7% accuracy

BNC example

- <s> — new sentence
- <w NN2>Explosives — plural noun
- <w VVD>found — past tense verb
- <w PRP>on — preposition
- <w NP0>Hampstead — proper noun
- <w NP0>Heath — proper noun
- <PUN>. — punctuation

Explosives found on Hampstead Heath

The Penn Treebank parsed corpus

```
(S (NPSBJ1 Chris)
   (VP wants
      (S (NPSBJ *1)
         (VP to
            (VP throw
               (NP the ball))))))
```

Empty embedded subject linked to NP subject no. 1

- Predicate Argument Structure:
  wants(Chris, throw(Chris, ball))

The GNOME anaphora corpus

<ne cat="pn" per="per3" num="sing" gen="neut"
ani="inanimate" disc="disc-old">
Dermovate Cream
</ne>
is
<**ne** cat="a-np" per="per3" num="sing" gen="neut"
ani="inanimate" disc="disc-new">
a
<**mod** type="preadj">**strong**</**mod**>
and
<**mod** type="preadj">**rapidly effective**</**mod**>
treatment
</**ne**>

Part 1

Annotation principles, standards and guidelines

Annotation Principles (Leech 1993)

1. Recoverability:
   - it should be possible to remove the annotation and extract the raw text
2. Extractability:
   - it should be possible to extract the annotation itself to store it separately
3. Transparency of guidelines:
   - the annotation should be based on explicit guidelines which are available to the end user

Annotation Principles (II)

4. Transparency of method
   - It should be clear who annotated what (often many people are involved in the project)
   - Typically, projects will also report some statistical measure of inter-annotator agreement
   - The extent to which different annotators agree will reflect on:
     - how good the guidelines are
     - how theory-neutral the annotation is

Annotation principles (III)

5. Fallibility
   - The annotation scheme is not infallible; the user should be made aware of this.
   - E.g. the BNC documentation actually reports on errors in the POS tagging
6. Theory-neutrality
   - As far as possible, the annotation should not be based on narrow theoretical principles.
   - E.g. A treebank with syntactic info is usually parsed with a simple, context-free grammar.
   - Using something more specific, like Chomsky's Principles and Parameters Framework, would mean it's useful to a narrower community.

Annotation principles (IV)

7. Standards:
   - no single annotation scheme has the right to be considered an a priori standard
   - e.g. there are many different formats for annotating part of speech info, or syntactic structure
   - However, there are published standards which provide a minimum for format and amount of information to include.

Comments on Leech (1993)

- Rather than standards, these are "desiderata" for annotation schemes.
- They don't really specify the form or content of an annotation scheme.
- However, there have been concerted efforts to define real standards to which corpora should conform.

The concept of a markup language

- A markup language provides a way of specifying meta-data about a document.
- Why "language"?
  - it specifies a basic "vocabulary" of elements;
  - it specifies a syntax for well-formed expressions.

The “SGML” family of markup languages

- SGML (Standard Generalised Markup Language): one of the first truly standardised formalisms

- Basic idea:
  - create a tag which has some “meaning”
    - e.g. <W> means “word”, <P> means “paragraph”
  - wrap portions of a document with start/end tags
    - e.g. <W>chair</W>
    - end tags can often be omitted: <W>chair
  - the “meaning” of the tag must be specified
  - tag can have attributes:
    - e.g. <S n=101>
  - tags can be nested inside eachother

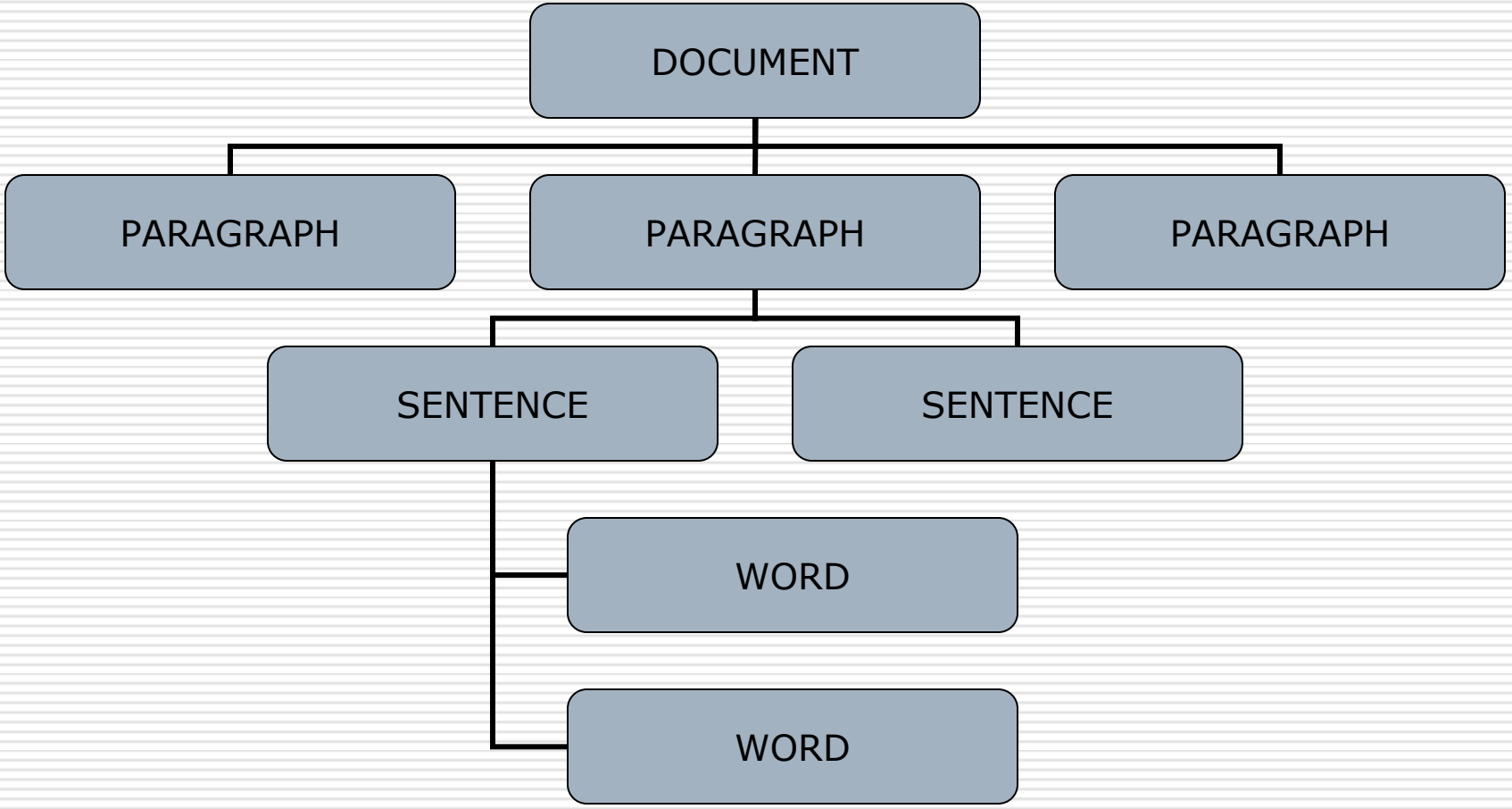
Descendants of SGML: HTML

- HTML: “Hypertext Markup Language”
  - developed by the World-Wide Web Consortium (W3C)
  - based on the SGML tagging principle
  - defines a basic representation language for document layout
  - used by web browsers: when you visit a page, your browser “interprets” the html and renders the layout visually.
  - fixed set of tags such as:
    - <P>: paragraph
    - <IMG>: image
    - etc

Descendants of SGML: XML

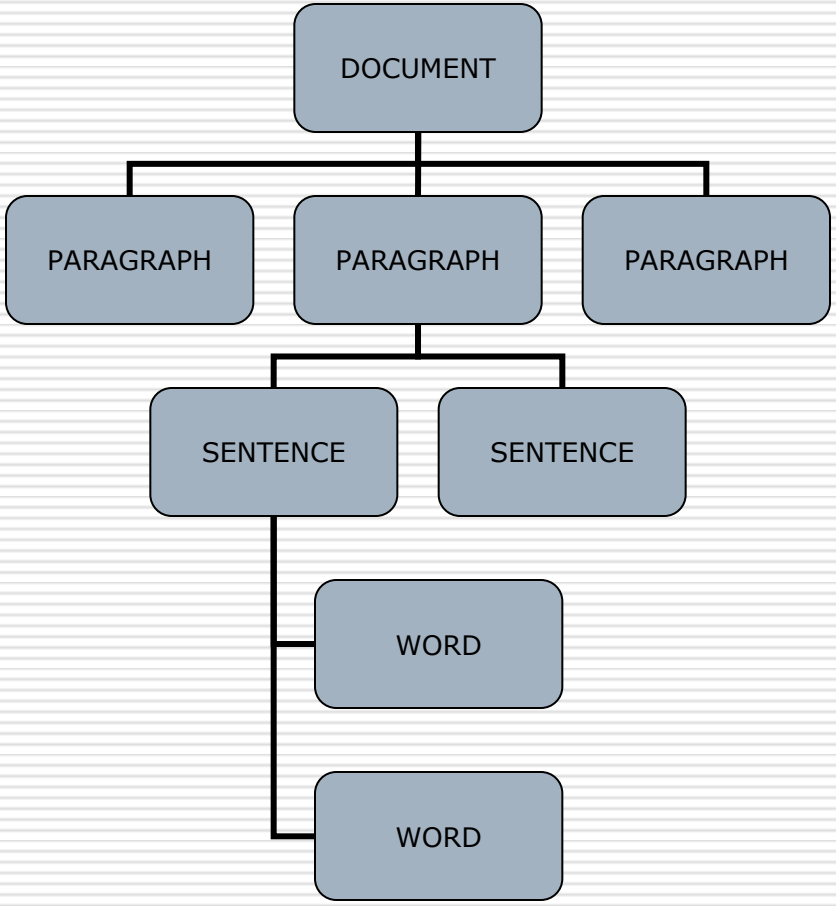
- XML: Extensible Markup Language
  - developed by the World-Wide Web Consortium (W3C)
  - nowadays, this is ubiquitous, and has largely replaced SGML as the markup language of choice
  - stricter syntax than SGML: end-tags can’t be omitted
  - less complex than SGML in other ways
  - unlike HTML, specifies only a syntax; the actual tags can be anything depending on the application.

XML documents are trees



XML Documents are trees

```
<DOCUMENT>
   <PARAGRAPH>
      <SENTENCE>
            <WORD>
            …
            </WORD>
      </SENTENCE>
   …
   </PARAGRAPH>
```



Meta-data in XML

- What properties does a book have?
  - author, ISBN, publisher, number of pages, genre: fiction, etc

```
<BOOK type="fiction">
    <AUTHOR gender="male">John Smith</AUTHOR>
    <PUBLISHER>CUP</PUBLISHER>
    <TITLE>Lost in translation</TITLE>
    ...
</BOOK>
```

- This contains "data" such as *John SMith, CUP, Lost in Translation…*
  - tags have attributes (e.g. *gender* for author, *type* for book)
- It contains meta-data (data about the data) in the form of tags
- Easy for a machine to know which pieces of information are about what.

The Text Encoding Initiative (TEI)

- Sponsored by the main academic bodies with an interest in machine-readable textual markup.

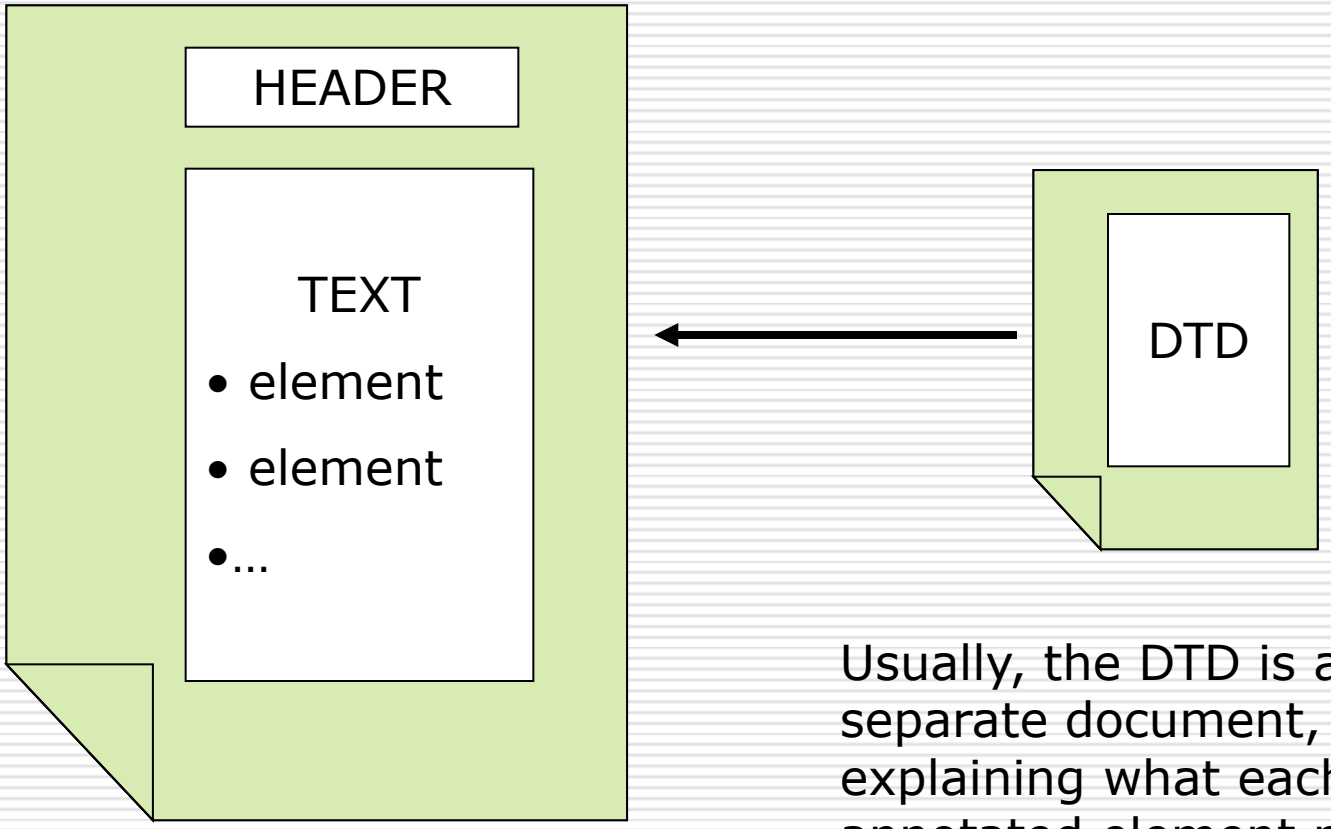
- Aims:
  - provide standardised formats for annotation
  - allow interchange of data: If corpus X is annotated according to TEI standards, then it is easy to:
    - develop tools to "read" the annotation
    - make the annotation comprehensible to others

- NB: The TEI does not specify the content, i.e. what the annotation should contain. It does specify how it should be done, i.e. the form.

The “document” according to TEI

- A document (e.g. a corpus text) consists of:
  - a header
    - information about the text such as *author*, *date*, *source*, etc.
  - the text itself
    - including annotation of textual elements, such as paragraphs, words, etc
      - Encoded using **tags** and **entity references**
  - a Document Type Declaration (DTD)
    - a formal representation which tells a computer program what elements the text contains, and what they mean

In graphics…



Usually, the DTD is a separate document, explaining what each annotated element means

Example: Structure of a BNC document (fragment)

```
<bncdoc>
    <header>
        <fileDesc>
            (description of the file)
        </fileDesc>
        <srcDesc>
            (source of the text, including publisher)
        </srcDesc>
    </header>
    <text>
        (the actual text + annotation)
    </text>
</bncdoc>
```

Markup language

- The TEI uses SGML
- Tags in SGML (and TEI):
  - Always use angle brackets
  - Indicate start and end
    - <tag> text </tag>
    - end-tag often omitted if not required
  - Used for text elements:
    - paragraph, word, sentence…

Markup language (cont/d)

- TEI also specifes a format for entity references:
  - an entity reference is a kind of abbreviation for some detailed formatting or linguistic information
- Format:
  - enclosed using & and ;
- Example:
  - é ➔ represents the letter *e* with an acute accent, i.e. é
  - man&nn1; ➔ represents the information that *man* is a noun in the singular
- Interpretation of entity references:
  - each different entity reference used in the text is defined in detail in the document header

Example: tags and references in a BNC document (fragment)

```
<text>
   <s n=020>
      <w EX0>there
      <w VBB>are
      <w PRP>between
      <w CRD>40&ndash;60,000
      <w NN0>people
   …
</text>
```

Sentence element with number

Word element with Part of Speech

Word element + entity reference
– = a dash

Beyond format: Content guidelines

- EAGLES
  - "Expert Advisory Groups on Language Engineering Standards"
  - EU-sponsored teams of experts who drew up guidelines on many aspects of language engineering, including corpus annotation.
- Aim:
  - "best-practice" recommendations on what to annotate, at all levels (textual, part-of-speech, etc)
  - cover a wide variety of languages
  - guidelines on corpora are TEI-conformant.
- Main document: Corpus Encoding Standard (CES). Assumes SGML as the markup language.
- Later development: XCES: The CES using XML as the markup language.

Part 2

Levels of corpus annotation

Textual/Extra-textual level

- Information about the text, origins etc.
  - cf the earlier example of the BNC header
  - cf. McEnery & Wilson's examples from other corpora
- Extra-textual information can be very detailed, e.g. include gender of author.
- Textual information can include things like questions, abbreviations and their expansions, etc.

Orthographic level

- Problems with different alphabets, accents etc.
  - Maltese: ħ, ġ, ż, ċ; German: umlaut etc; Russian: cyrillic alphabet
- TEI recommends use of entity references:
  - ù ➔ ù
  - ġ ➔ ġ
  - also, recommends sticking to the basic (“English”) ISO-646 character set
- More recently, the UNICODE standard provides for a single, unified representation of all characters in (hopefully) all alphabets and writing systems as they are, without needing any special graphics capabilities.
  - every character is mapped to a unique numeric code
  - all codes are readable by a computer
- TEI also recommends representing changes of typography etc (boldface, italic...) using start/end tags.

The challenges of spoken data

- Speech does not contain “sentences” but “utterances”.

- Transcription of spoken data entails decisions about:
  - whether to assume sentence-based transcription or intonation units
  - what to do about pauses, false starts, coughing...
  - what to do about interruptions and overlapping speech
  - whether to add punctuation

- Example:
  - London-Lund corpus uses intonation units for speech, with no punctuation

Spoken data in the BNC

```
<u who=D00011>
    <s n=00011>
    <event desc="radio on">

        <w PNP><pause dur=34>You
        <w VVD>got
        <w TO0>ta
        <unclear>
        <w NN1>Radio
        <w CRD>Two
        <w PRP>with
        <w DT0>that <c PUN>.
</u>
```

Utterance tag + speaker ID attribute

Sentence tag within utterance

Non-verbal action during speech

Pauses marked with duration

Unclear, non-transcribed speech

- Many other tags to mark non-linguistic phenomena...

Levels of linguistic annotation

- part-of-speech (word-level)
- lemmatisation (word-level)
- parsing (phrase & sentence-level)
- semantics (multi-level)
  - semantic relationships between words and phrases
  - semantic features of words
- discourse features (supra-sentence level)
- phonetic transcription
- prosody

Part of speech tagging

- Purpose:
  - Label every token with information about its part of speech.
- Requirements:
  - A tagset which lists all the relevant labels.

Part of speech tagsets

- Tagging schemes can be very granular. Maltese example:
  - VV1SR: verb, main, 1st pers, sing, perf
    - *imxejt* – "I walked"
  - VA1SP: verb, aux, 1st pers, sing, past
    - *kont miexi* – "I was walking"
  - NNSM-PS1S: noun, common, sing, masc + poss. pronoun, sing, 1st pers
    - *missier-i* – "my father"

How POS Taggers tend to work

1. Start with a manually annotated portion of text (usually several thousand words).
   - the/DET man/NN1 walked/VV
2. Extract a lexicon and some probabilities from it.
   - E.g. Probability that a word is NN given that the previous word is DET.
   - Used for tagging new (previously unseen) words.
3. Run the tagger on new data.

Challenges in POS tagging

- Recall that the process is usually semi-automatic.

- Granularity vs. correctness
  - the finer the distinctions, the greater the likelihood of error
  - manual correction is extremely time-consuming

EAGLES recommendations on POS tagging

- Set of obligatory features for all languages
  - Noun, verb, interjection, unique, residual, etc
- Set of recommended features:
  - Noun: number, gender, case, type
- Set of optional features:
  - generic: apply to "all" languages (e.g. noun=count or mass)
  - language-specific: e.g. Danish has a suffixed definite article, so has a "definiteness" feature for Nouns