Pre-processing and annotation

Raw data from a linguistic source can't be exploited directly. We first have to perform:

- *pre-processing:* identify the basic units in the corpus:
  - tokenization;
  - sentence boundary detection;
- *annotation:* add task-specific information:
  - parts of speech;
  - syntactic structure;
  - dialogue structure, prosody, etc.

Tokenization

Tokenization: divide the raw textual data into tokens (words, numbers, punctuation marks).

Word: a continuous string of alphanumeric characters delineated by whitespace (space, tab, newline).

Example: potentially difficult cases:

- amazon.com, Micro$oft
- John's, isn't, rock'n'roll
- child-as-required-yuppie-possession
  (As in: "The idea of a child-as-required-yuppie-possession must be motivating them.")
- cul de sac

Sentence Boundary Detection

Sentence boundary detection: identify the start and end of sentences.

Sentence: string of words ending in a full stop, question mark or exclamation mark.

This is correct 90% of the time.

Example: potentially difficult cases:

- Dr. Foster went to Gloucester.
- He said "rubbish!".
- He lost cash on lastminute.com.

The detection of word and sentence boundaries is particularly difficult for *spoken data*.

Corpus Annotation

Annotation: adds information that is not explicit in the data itself, increases its usefulness (often application-specific).

Annotation scheme: basis for annotation, consists of a tag set and annotation guidelines.

Tag set: is an inventory of labels for markup.

Annotation guidelines: tell annotators (domain experts) how tag set is to be applied; ensure consistency across different annotators.

Part-of-speech (POS) annotation

Part-of-speech (POS) tagging is the most basic kind of linguistic annotation.

Each linguistic token is assigned a code indicating its *part of speech*, i.e., basic grammatical status.

Examples of POS information:

- singular common noun;
- comparative adjective;
- past participle.

POS tagging forms a basic first step in the disambiguation of homographs.

E.g., it distinguishes between the verb “boot” and the noun “boot”.

But it does not distiguish between “boot” meaning “kick” and “boot” as in “boot a computer”, both of which are transitive verbs.

Example POS tag sets

- CLAWS tag set (used for BNC): 62 tags; (Constituent Likelihood Automatic Word-tagging System)
- Brown tag set (used for Brown corpus): 87 tags:
- Penn tag set (used for the Penn Treebank): 45 tags.

Category	Examples	CLAWS	Brown	Penn
Adjective	happy, bad	AJ0	JJ	JJ
Adverb	often, badly	PNI	CD	CD
Determiner	this, each	DT0	DT	DT
Noun	aircraft, data	NN0	NN	NN
Noun singular	woman, book	NN1	NN	NN
Noun plural	women, books	NN2	NN	NN
Noun proper singular	London, Michael	NP0	NP	NNP
Noun proper plural	Australians, Methodists	NP0	NPS	NNPS

POS Tagging

Idea: Automate POS tagging: look up the POS of a word in a dictionary.

Problem: POS ambiguity: words can have several possible POS's; e.g.:

> Time flies like an arrow. (1)
>
> time: singular noun or a verb;
> flies: plural noun or a verb;
> like: singular noun, verb, preposition.

Combinatorial explosion: (1) can be assigned $\mathbf{2 \times 2 \times 3 = 12}$ different POS sequences.

Need more information to resolve such ambiguities.

It might seem that higher-level meaning (semantics) would be needed, but in fact great improvement is possible using the *probabilities* of different POS.

Probabilistic POS tagging

Observation: words can have more than one POS, but one of them is more frequent than the others.

Idea: assign each word its most frequent POS (get frequencies from manually annotated training data). Accuracy: around 90%.

Improvement: use frequencies of POS sequences, and other context clues. Accuracy: 96–98%.

Example output from a POS tagger (not XML format!):

> Our/PRP$ enemies/NNS are/VBP innovative/JJ and/CC resourceful/JJ ,/, and/CC so/RB are/VB we/PRP ./. They/PRP never/RB stop/VB thinking/VBG about/IN new/JJ ways/NNS to/TO harm/VB our/PRP$ country/NN and/CC our/PRP$ people/NN, and/CC neither/DT do/VB we/PRP ./. (George W. Bush)

Use of markup languages

An important general application of markup languages, such as XML, is to separate *data* from *metadata*.

In a corpus, this serves to keep different types of information apart;

- *Data* is just the raw data.
  In a corpus this is the text itself.
- *Metadata* is data about the data.
  In a corpus this is the various annotations.

Nowadays, XML is the most widely used markup language for corpora.

The example on the next slide is taken from the BNC XML Edition, which was released only in 2007.
(The previous BNC World Edition was formatted in SGML.)

Example from the BNC XML Edition

```
<wtext type="FICTION">
 <div level="1">
 <head> <s n="1">
   <w c5="NN1" hw="chapter" pos="SUBST">CHAPTER </w>
   <w c5="CRD" hw="1" pos="ADJ">1</w>
 </s> </head>
 <p> <s n="2">
   <c c5="PUQ"> </c>
   <w c5="CJC" hw="but" pos="CONJ">But</w>
   <c c5="PUN">,</c> <c c5="PUQ">  </c>
   <w c5="VVD" hw="say" pos="VERB">said </w>
   <w c5="NP0" hw="owen" pos="SUBST">Owen</w>
   <c c5="PUN">,</c> <c c5="PUQ"> </c>
   <w c5="AVQ" hw="where" pos="ADV">where </w>
   <w c5="VBZ" hw="be" pos="VERB">is </w>
   <w c5="AT0" hw="the" pos="ART">the </w>
   <w c5="NN1" hw="body" pos="SUBST">body</w>
   <c c5="PUN">?</c> <c c5="PUQ"> </c>
 </s> </p>
 ....
 </div>
</wtext>
```


Aspects of this example

This example is the opening text of J10, a novel by Michael Pearce.

Some aspects of the tagging:

- The **`wtext`** element stands for *written text*. The attribute **`type`** indicates the genre.
- The **`head`** element tags a portion of header text (in this case a chapter heading).
- The **`s`** element tags sentences. (N.B., a chapter heading counts as a sentence.) Sentences are numbered via the attribute **`n`**.
- The **`w`** element tags words. The attribute **`pos`** is a POS tag, with more detailed POS information given by the **`c5`** attribute, which contains the CLAWS code. The attribute **`hw`** represents the *root form* of the word (e.g., the root form of "said" is "say").
- The **`c`** element tags punctuation.

Syntactic annotation (parsing)

Syntactic annotation: information about the structure of sentences.
Prerequisite for computing meaning.

Linguists use phrase markers to indicates which parts of a sentence belong together:

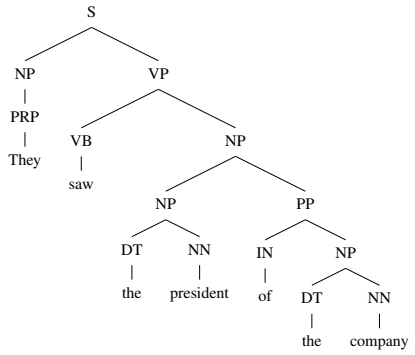
- noun phrase (NP): noun and its adjectives, determiners, etc.
- verb phrase (VP): verb and its objects;
- prepositional phrase (PP): preposition and its NP;
- sentence (S): VP and its subject.

Phrase markers group hierarchically in a *syntax tree*.

Syntactic annotation can be automated. Accuracy: around 90%.

Example syntax tree

Sentence from the Penn Treebank corpus:



The same syntax tree in XML:

```
<s>
 <np><w pos="PRP">They</w></np>
 <vp><w pos="VB">saw</w>
   <np>
     <np><w pos="DT">the</w> <w pos="NN">president</w></np>
     <pp><w pos="NN">of</w>
       <np><w pos="DT">the</w> <w pos="NN">company</w></np>
     </pp>
   </np>
 </vp>
</s>
```

Note the conventions used in the above document: phrase markers are represented as elements; whereas POS tags are given as attribute values.

N.B. The tree on the previous slide is *not* the XML element tree generated by this document.