ACS Introduction to NLP

Lecture 1: Automatic Linguistic Annotation

UNIVERSITY OF CAMBRIDGE

Stephen Clark

Natural Language and Information Processing (NLIP) Group

sc609@cam.ac.uk

England’s fencers won gold on day 4 in Delhi with a medal-winning performance. This is Dr. Black’s second gold of the Games.

England’s fencers won gold on day 4 in Delhi with a medal-winning performance.

This is Dr. Black’s second gold of the Games.

Tokenisation (What’s a Word?)

```
England ’s fencers won gold on day 4 in Delhi with a
medal -winning performance .

This is Dr. Black ’s second gold of the Games .
```

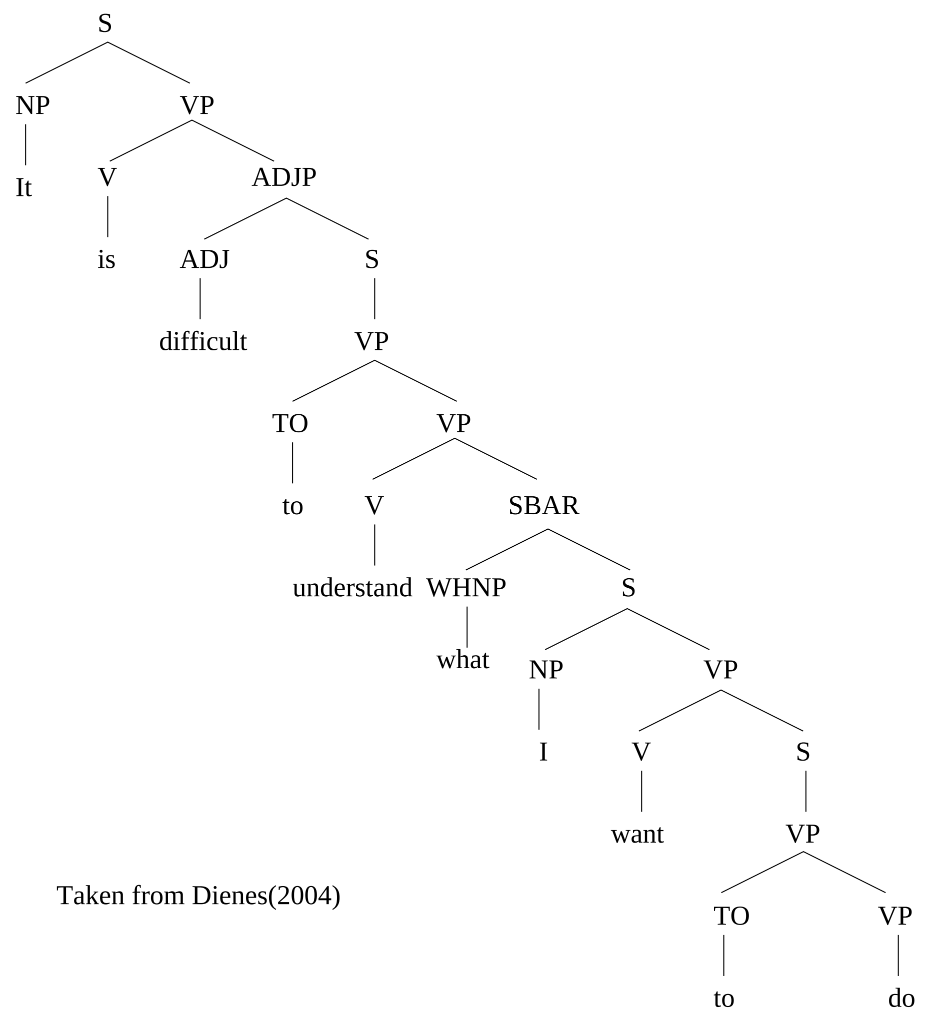
```
England|NNP ’s|POS fencers|NNS won|VBD gold|NN on|IN
day|NN 4|CD in|IN Delhi|NNP with|IN a|DT medal|JJ
-winning|JJ performance|NN .|.

This|DT is|VBZ Dr.|NNP Black|NNP ’s|POS second|JJ
gold|NN of|IN the|DT Games|NNP .|.
```

```
England|I-LOC ’s|O fencers|O won|O gold|O on|O
day|I-TIME 4|I-TIME in|O Delhi|I-LOC with|O a|O medal|O
-winning|O performance|O .|O

This|O is|O Prof.|I-PER Black|I-PER ’s|O second|O
gold|O of|O the|O Games|O .|O
```

Chunk Tagging


```
England|I-NP ’s|I-NP fencers|I-NP won|I-VP gold|I-NP
on|I-PP day|I-NP 4|I-NP in|I-PP Delhi|I-NP with|I-PP
a|I-NP medal|I-NP -winning|I-NP performance|I-NP .|.

This|I-NP is|I-VP Dr.|I-NP Black|I-NP ’s|I-NP
second|I-NP gold|I-NP of|I-PP the|I-NP Games|I-NP .|.
```

Syntactic Parsing - Phrase Structure Trees



Taken from Dienes(2004)

Semantic Parsing - Logical Form

```
From 1953 to 1955 , 9.8 billion Kent cigarettes with the filters were
sold , the company said .

  _____________   _________________________________________________________________
 | x1          | | x2 x3                                                           |
 |-------------| |-----------------------------------------------------------------|
(| company(x1) |A| say(x2)                                                         |)
 | single(x1)  | | agent(x2,x1)                                                    |
 |_____________| | theme(x2,x3)                                                    |
                 | proposition(x3)                                                 |
                 |        __________________    ____________   ________________    |
                 |       | x4               |  | x5         | | x6 x7 x8       |   |
                 |   x3: |------------------|  |------------| |----------------|   |
                 |      (| card(x4)=billion |;(| filter(x5) |A| with(x4,x5)    |)) |
                 |       | 9.8(x4)          |  | plural(x5) | | sell(x6)       |   |
                 |       | kent(x4)         |  |____________| | patient(x6,x4) |   |
                 |       | cigarette(x4)    |                 | 1953(x7)       |   |
                 |       | plural(x4)       |                 | single(x7)     |   |
                 |       |__________________|                 | 1955(x8)       |   |
                 |                                            | single(x8)     |   |
                 |                                            | to(x7,x8)      |   |
                 |                                            | from(x6,x7)    |   |
                 |                                            | event(x6)      |   |
                 |                                            |________________|   |
                 | event(x2)                                                       |
                 |_________________________________________________________________|
```

Why Annotate with Linguistic Structure?

- Allows the computer access to (elements of) the **meaning** of the sentence (or document)
- Allows the computer a (rudimentary) “understanding” of the sentence
- Allows the computer to reason (to some extent) about the sentence

[DEMO: http://svn.ask.it.usyd.edu.au/trac/candc/wiki/Demo]

Part-of-Speech (POS) Tagging

- Task: given a set of POS tags and a sentence, assign a POS tag to each word
  - or a set of tags to each word with a probability distribution
- What are the tags?
- How does the computer decide which tag to assign to each word?
  - what knowledge is required and where does it come from?
- What's the algorithm for assigning the tags?

What Tag Set?

- What are the POS tags used for?
  - to provide basic grammatical information, e.g. noun or verb
  - to provide input to more complex annotation, e.g. parsing
- Example tag sets
  - Penn Treebank set is the most common
  - Others exist, e.g. CLAWS (http://ucrel.lancs.ac.uk/claws6tags.html)
- Choice of tag set may depend to some extent on the algorithm being used to assign the tags

[LOOK AT THE PTB TAG SET]

Why is POS Tagging Non-Trivial?

- AMBIGUITY
- e.g. *can* can be a noun or a (modal) verb

[DEMO]

Formalisation of the Tagging Problem

$$y^* = \arg\max_{y \in Y} \text{score}(y, x)$$

where x is a sentence and Y is the set of possible tag sequences for x

- In machine learning this is known as a *sequence labelling* problem
- There are many possible solutions (HMM, CRF, perceptron, ...)

A Probabilistic Formulation of the Tagging Problem

$$y^* = \arg\max_{y \in Y} P(y|x)$$

where x is a sentence and Y is the set of possible tag sequences for x

- More on the motivation for the probabilistic (statistical) approach in the next lecture
- But for now: we use probabilities because the computer is having to make a guess at the correct tag for a word on the basis of incomplete information
- Probability theory is perhaps the best theory we have for *reasoning under uncertainty*

Probabilistic Formulation

$$y^* = \arg\max_{y \in Y} P(y|x)$$

where $x = (x_1, \ldots, x_n)$ is a sentence and $y = (y_1, \ldots, y_n) \in Y$ is a possible tag sequence for x

- Two problems:
  - where do the probabilities come from? (age-old question in statistical approaches to AI)
  - how do we find the arg max?
- Problem 1 is the problem of *model estimation*
- Problem 2 is the *search problem*

Reading for Today's Lecture

- Penn Treebank POS tag manual (http://www.cis.upenn.edu/ treebank/)
- Jurafsky and Martin, Speech and Language Processing, Chapter on Word Classes and Part of Speech Tagging