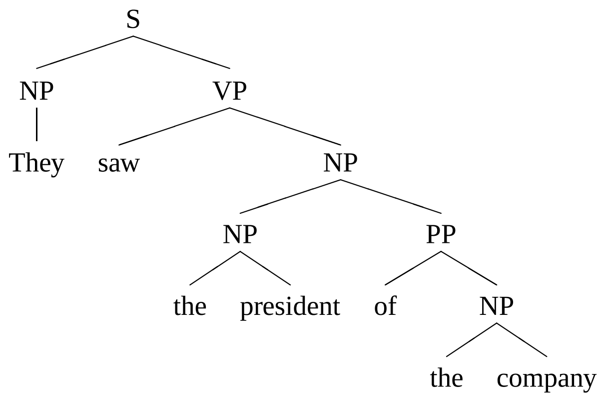
Shallow parsing

Full-blown parsing recovers *all* syntactic relationships:



Shallow parsing

- Requires several hard problems to be solved:
  - Dealing with undergeneration (grammar misses rules).
  - Selecting the best parse.
  - Efficiently recovering parses (despite worse-case exponential behaviour).
- However, don't always need to recover all relationships:
  - Information extraction.
  - Information retrieval.
  - Document classification.
  - Etc.

Shallow parsing

- Shallow parsing: recovery of simple units (chunks):

  (they) saw (the president) of (the company)
- Here we've recovered simple NPs (not PPs, VPs etc).
- Chunks don't overlap (and are not nested).
  - Considerably simplifies parsing.
- Shallow parsing now a lot faster (linear time).

Annotating raw sentences

- For each word, only need to specify:
  - Whether a word begins a chunk.
  - Whether a word is in a chunk.
  - Whether a word is out of a chunk.

  (they/BNP) saw/O (the/BNP president/INP) of/O (the/BNP company/INP)
- Sentences might also have part-of-speech tags.

Modelling shallow parsing

Modelling the full joint probability is hard:

- The only information possible: surrounding words, POS tags and previous chunk labels.
- Using all previous information implies vary sparse statistics.
- Need to use some maximally large context that can be reliably estimated.
- Contexts encoding previous decisions reduces independence assumptions.

Typically, only the current word, a few previous words and their labels are used.

Precision and recall

- *Precision* measures the proportion of selected items that are correct:

$$\text{precision} = \frac{tp}{tp + fp}$$

- *Recall* records the proportion of correct items selected:

$$\text{recall} = \frac{tp}{tp + fn}$$

- Need to report both.

Modelling shallow parsing

- Let W be some sequence of words.
- Let L be some sequence of labels.
- We are now going to specify $P(L,W)$

Shallow parsing as POS tagging

POS taggers are specified using labelled material:

- Words.
- The set of possible POS tags.

Possible to train POS tagger to do shallow parsing:

- Use a training set consisting of words and shallow parsing labels.

Results (F-score)

Context	Results
Nothing	32.5
Words	92.8
Tags	92.7
Words and tags	93.7
Words, tags, next tag and current label	94.8

Error analysis

- Some errors due to mistagging.
- Other errors due to noise in training material:

  Assistant Secretary
  (Assistant Secretary)
- A few errors due to local nature of model (ie labelling word as in a chunk when no previous beginning chunk label).

F-Measure

- *F-measure*:

$$\mathrm{F}score = \frac{2PR}{P+R}$$

- *F score* assumes that precision and recall are equally important.

Error analysis

Errors:

- Co-ordination:

  Guess: (an aerospace), (electronics, automotive and graphics company)
  Answer: (an aerospace, electronics, automotive and graphics company)
- Ditransitive VPs as being transitive VPs:

  Guess: granted (the Shah of Iran asylum in Panama)
  Answer: granted (the Shah of Iran) (asylum in Panama)

Summary

- Shallow parsing = simplified parsing.
- Possible to view as tagging task.
- Best results need carefully selected contextual information.
- Possible to use other classifiers.

Other approaches

- Shallow parsing is a classification task, so possible to use other classifiers:
  - Memory-based learning.
  - Support-vector machines.
  - Maximum entropy.
  - Neural nets.
  - Etc.
- Idea: replace $P(l_i \mid w_i, \ldots)$ with some other approach which assigns the 'best' label to a word (and returns some confidence score).
- Each classifier will have strengths / weaknesses.

Further reading

- Look at CoNLL00 homepage:

  `http://lcg-www.uia.ac.be/conll2000/chunking/`
- In particular, look at:
  - Erik F. Tjong Kim Sang and Sabine Buchholz, Introduction to the CoNLL-2000 Shared Task.
  - Rob Koeling, Chunking with Maximum Entropy Models.
  - Erik F. Tjong Kim Sang, Text Chunking by System Combination.
  - Taku Kudoh and Yuji Matsumoto, Use of Support Vector Learning for Chunk Identification.