Topics in Natural Language Processing

Shay Cohen

Institute for Language, Cognition and Computation

University of Edinburgh

Lecture 6

Last class

Structure in NLP

B-I-O

sequences

trees

segmentation

chunking

clustering

Today's class

- Grammars (CFGs and TAGs)
- Inference in NLP

Context-free grammars

What is a CFG?

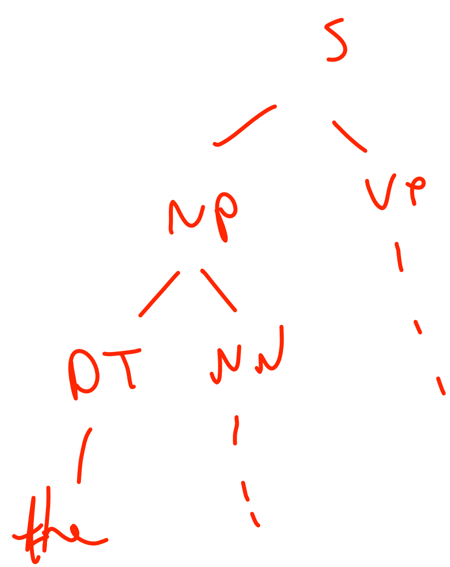
- A set of nonterminals N
- A set of vocabulary terminals V
- A start symbol $S \in N$
- A set of production rules R, $a \to \alpha$ is such that $a \in N$ and $\alpha \in (V \cup N)^*$

$S \to NP \ VP$

$NP \to DT \ NN$

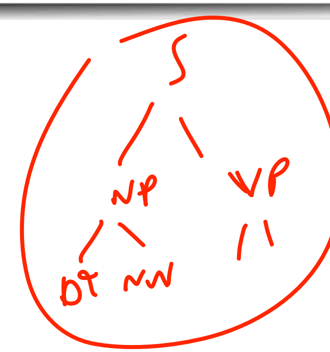
$DT \to the$

S



Are CFGs sufficient for natural language?

Let G be a grammar.



$T(G) =$ set of parse trees allowable by G

$L(G) =$ set of strings allowed by G

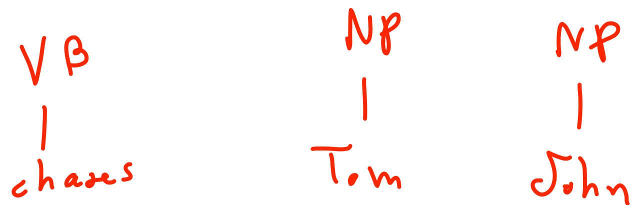
- Dutch - there are structures in Dutch which do not appear in any $T(G)$ for any G context-free grammar
- Swiss-German - there are structures in Swiss-German which do not appear in any $L(G)$ (and hence in any $T(G)$) for any G CFG

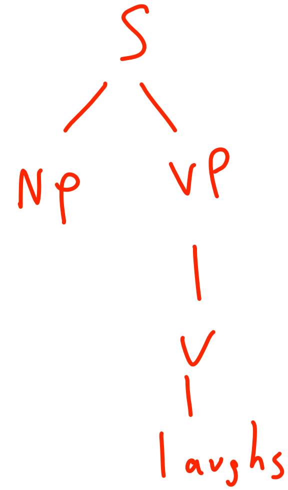
The constructions are similar to demonstrate that. Swiss-German uses case markers.

Tree adjoining grammars

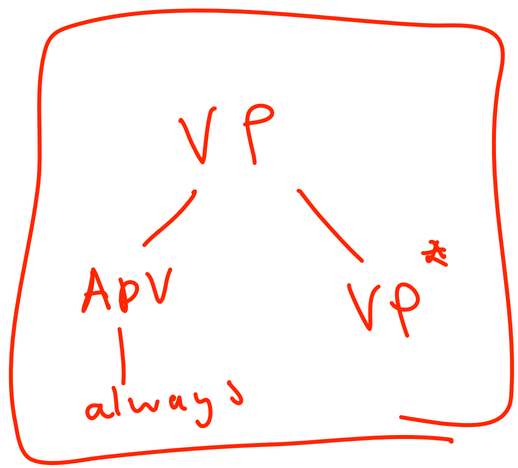
Joshi

Initial trees:

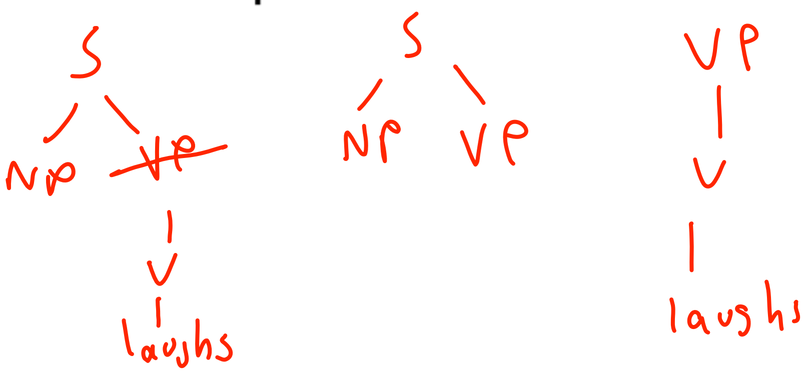


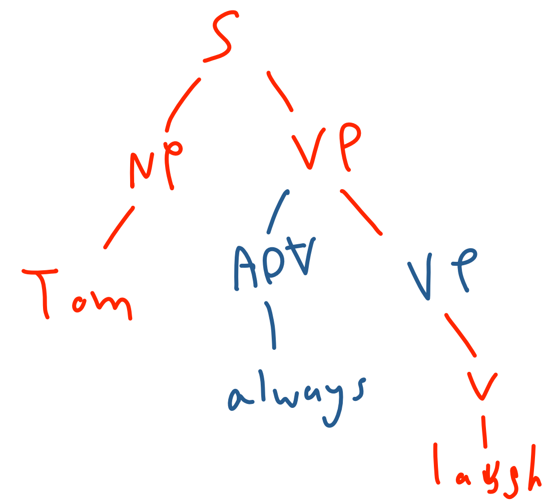


Auxiliary trees:



Derivational process:





Tree adjoining grammars

Quick question: is $\{ww \mid w \in \Sigma^*\}$ a context-free language?

"copy"

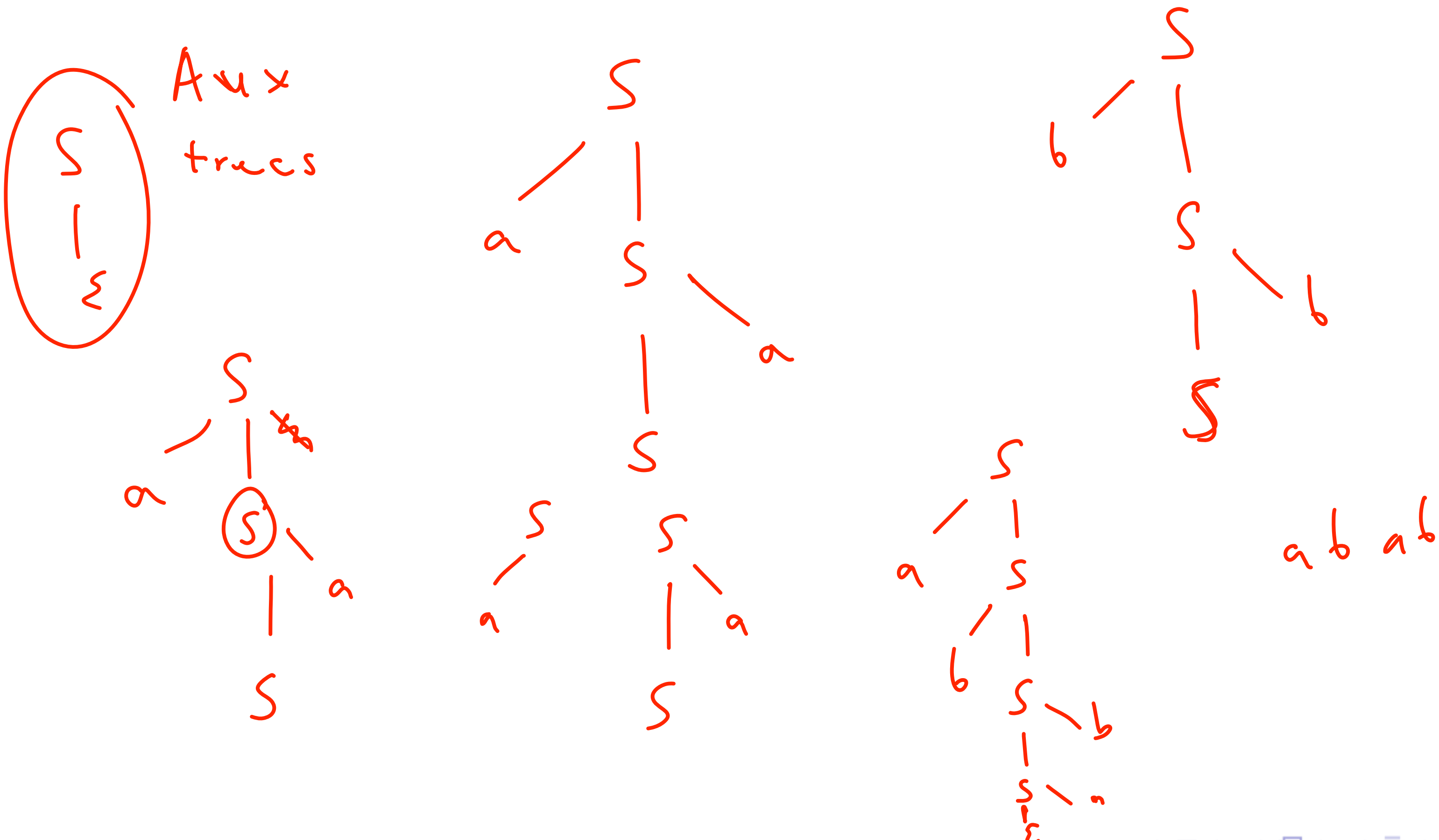
$a^n b^n$

Tree adjoining grammars

Quick question: is $\{ww \mid w \in \Sigma^*\}$ a context-free language?

$\Sigma = \{a, b\}$

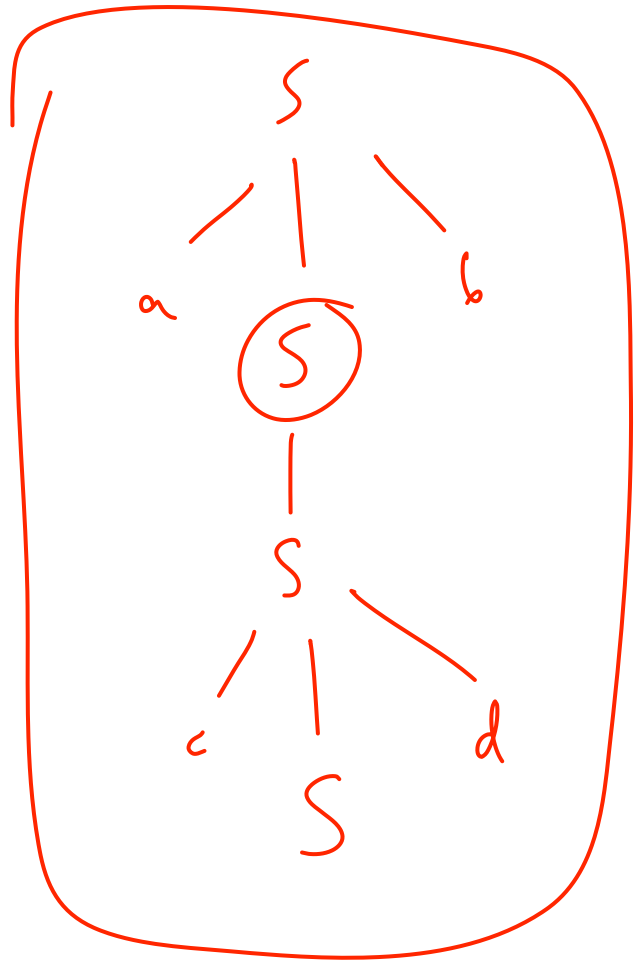
Is it a tree adjoining language?



Tree adjoining grammars

Another quick question: is $\{a^n b^n c^n d^n \mid n \geq 1\}$ a context-free language?

Aux tree



Init. tree

Tree adjoining grammars

Another quick question: is $\{a^n b^n c^n d^n \mid n \geq 1\}$ a context-free language?

Is it a tree adjoining language?

Tree adjoining grammars

They add the “minimum needed” in order to capture phenomena such as cross-serial dependencies

They are part of a family of grammar formalisms called “mildly context sensitive”

Other examples which are weakly equivalent: combinatory categorial grammars, head grammars, linear indexed grammars

Canonical forms of grammars

Canonical form: (1) a specific form for writing a grammar; (2) every general CFG can be converted to an “equivalent” canonical form.

Important example: Chomsky normal form (or binarised form)

$A \to B \quad C$

$A \to w$

A, B, C nonterminals

w word

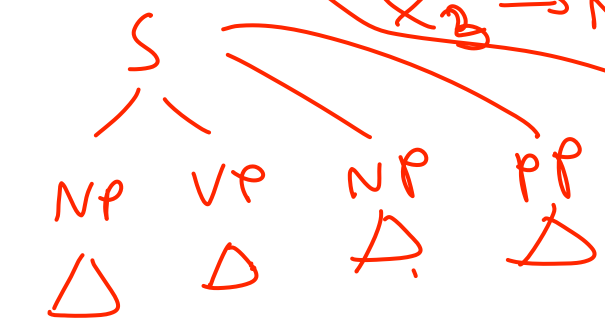
Why is CNF a normal form?

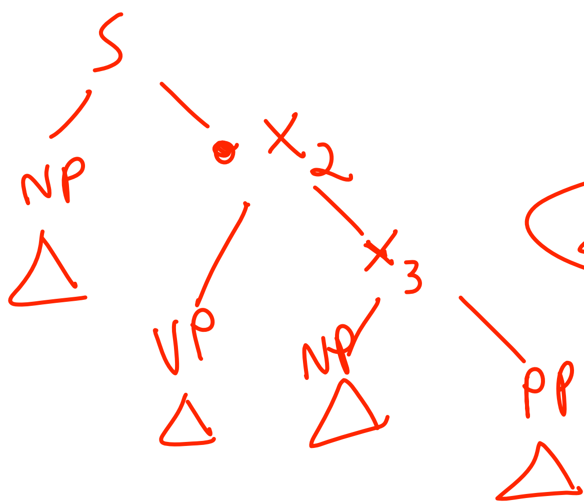
$S \rightarrow NP \;\; VP \;\; NP \;\; PP$

$S \rightarrow NP \;\; X_2$

$X_2 \rightarrow VP \;\; X_3$

$X_3 \rightarrow NP \;\; PP$





G in arbitrary form

$\Downarrow$

G' in binary form

can map any

$t \in T(G)$

to

$t' \in T(G')$

$S \rightarrow Y_1 \;\; PP$

$Y_1 \rightarrow Y_2 \;\; NP$

$Y_2 \rightarrow NP \;\; VP$

Probabilistic grammars

$p(A \to \beta)$ for every rule

$p(A \to \beta) \geq 0$

$$\sum_{A \to \beta \in R(A)} p(A \to \beta) = 1 \quad \forall A \in N$$

rules of A

"Usually", in NLP

$$\sum_{tree \in T(G)} p(tree) = 1$$

$S \to S\ S \quad 0.7$

$$p(tree) = \prod_{A \to \beta \in tree} p(A \to \beta \mid A)$$

$S \to a \quad 0.3$

$\sum p(tree) < 1$

Weighted grammars

Arbitrary positive weights to the rules

$$p(\text{tree}) \propto \prod_{\text{rule}} w(\text{rule})$$

$$p(\text{tree}) = \frac{\prod w(\text{rule})}{Z(G)}$$

$$\boxed{Z(G) = \sum_{\text{tree} \in T(G)} \prod_{\text{rule} \in \text{tree}} w(\text{rule})}$$

Basic inference with grammars

The probability of a derivation:

p(tree) product of rule probabilities

We estimate a PCFG. How do we parse a sentence?

Estimation

We learned how to do estimation:

- Maximum likelihood estimate
- Bayesian posterior summarisation
- ... There are many other ways

What's next?

Inference

What's inference?

Given a statistical model, find probable structure, classification, etc. for the input

Inference

Our Ω was usually a cross-product of inputs and outputs

Now, given an input, we need to find the correct output

$$\text{output}^* = \arg\max_{\text{output}} p(\text{output} \mid \text{input}) = \arg\max_{\text{output}} \frac{p(\text{output}, \text{input})}{p(\text{input})} = \arg\max_{\text{output}} p(\text{input}, \text{output})$$

doesn't depend on output

Inference

Our Ω was usually a cross-product of inputs and outputs

Now, given an input, we need to find the correct output

$$\arg\max_{\text{output}} p(\text{output}|\text{input})$$

Linear Score Function

$p(r_i)$

Consider a model which is a PCFG.

Probability of a tree: $t = (r_1, \ldots, r_n)$

$\mathrm{freq}(r, t)$

$$p(t) = \prod_{i=1}^{n} p(r_i) = \prod_{r \in t} p(r)$$

"Best" tree y given sentence x:

$$t^*(x) = \arg\max_{\mathrm{yield}(t)=x} p(t)$$

sentence

Linear Score Function

$\log(a \cdot b) = \log(a) + \log(b)$

apply $\Rightarrow$ $\log(a^y) = y \log a$

"Best" tree given sentence x:

take log

$$y^* = \arg \max_{y:\ \text{yield}(y)=x} \prod_{r \in y} p(r)^{\text{freq}(y,r)} = \arg\max_{\text{yield}(y)=x} \sum \left((\log p(r)) \times \text{freq}(r, y) \right)$$

$$= \arg\max_{y,\ \text{yield}(y)=x} \sum_{r \in R} w(r) \times \underline{\text{freq}(y, r)}$$

parameters "structure"

$$\arg\max_{y} \theta^T f(y, x)$$