Introduction to Human Language Technologies
8. Syntactic parsing: grammars

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

Outline

Outline

Goal and motivation

- Syntax studies the combination of words in a sentence.
- Syntactic parsing provides information of the combination of words in a sentence (the syntactic structure).
- Syntactic information is relevant for many LN applications:
  - Authorship recognition
  - Grammar checking

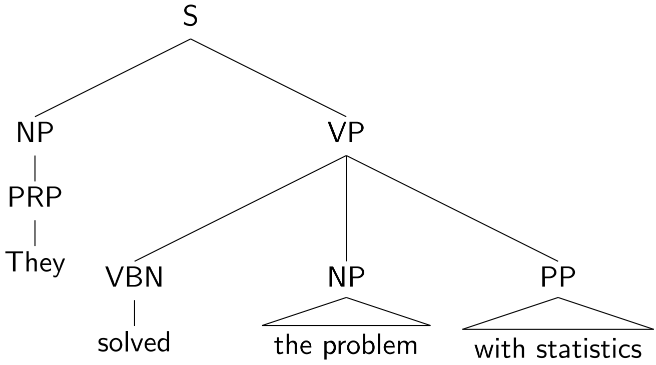
    Ex: 3th-Singular-noun + basic-verb $\Longrightarrow$ error
  - Machine Translation

    Ex: [es] NN+JJ $\Longrightarrow$ [en] JJ+NN
  - Information Extraction

    Ex: $X - [subj] \rightarrow$ visited $\leftarrow [dobj] - Y \Longrightarrow$ visit(X,Y)
  - ...
- **Goal**: find the syntactic structure associated to a sentence.

Outline

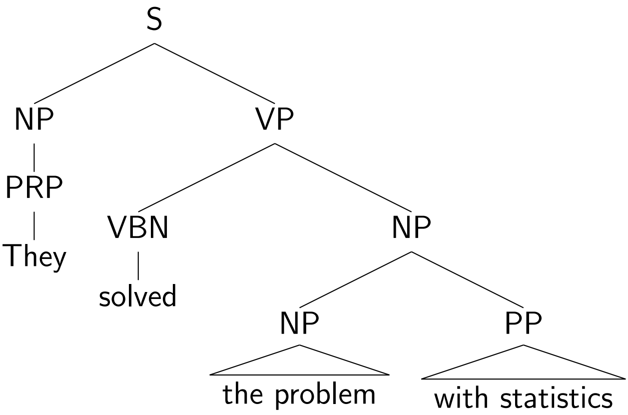
Constituent tree



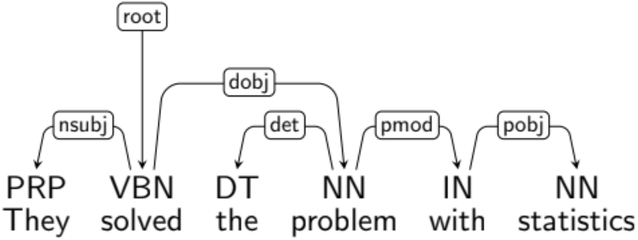
Phrase chunking may be seen as the flattening of this structure

[NP They/PRP][VP solved/VBN] [NP the/DT problem/NN] with/IN [NP statistics/NNS]

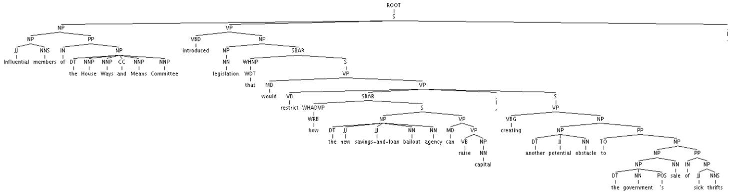
Another constituent Tree



Dependency tree



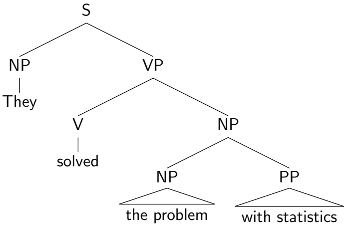
A real sentence



Influential members of the House Ways and Means Committee introduced legislation that would restrict how the new savings-and-loan bailout agency can raise capital, creating another potential obstacle to the government's sale of sick thrifts.

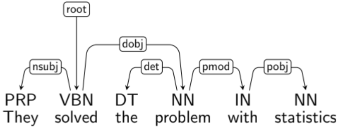
Theories of Syntactic Structure

Constituent Trees



- Main element: constituents (or phrases, or bracketings)
- Constituents = abstract linguistic units
- Results in nested trees

Dependency Trees



- Main element: dependency
- Focus on relations between words
- Handles *free word order* nicely.

Outline

Definition

[Hopcroft and Ullman 1979]
A context free grammar $G = (N, \Sigma, R, S)$ where:

- N is a set of non-terminal symbols
- Σ is a set of terminal symbols
- R is a set of rules of the form $X \rightarrow Y_1 Y_2 \ldots Y_n$ for $n \geq 0, X \in N, Y_i \in (N \cup \Sigma)$
- $S \in N$ is a distinguished start symbol

Context Free Grammars, Example

N = {S, NP, VP, PP, DT, Vi, Vt, NN, IN}
S = S
Σ = {sleeps, saw, man, woman, telescope, the, with, in}

R =

S	$\Rightarrow$	NP	VP
VP	$\Rightarrow$	Vi	
VP	$\Rightarrow$	Vt	NP
VP	$\Rightarrow$	VP	PP
NP	$\Rightarrow$	DT	NN
NP	$\Rightarrow$	NP	PP
PP	$\Rightarrow$	IN	NP

Vi	$\Rightarrow$	sleeps
Vt	$\Rightarrow$	saw
NN	$\Rightarrow$	man
NN	$\Rightarrow$	woman
NN	$\Rightarrow$	telescope
DT	$\Rightarrow$	the
IN	$\Rightarrow$	with
IN	$\Rightarrow$	in

Note: S=sentence, VP=verb phrase, NP=noun phrase, PP=prepositional phrase, DT=determiner, Vi=intransitive verb, Vt=transitive verb, NN=noun, IN=preposition

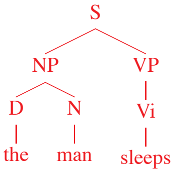
Left-most Derivations in CFGs

A left-most derivation is a sequence of strings $s_1 \ldots s_n$, where

- $s_1 = S$, the start symbol
- $s_n \in \Sigma^*$, i.e. s_n is made up of terminal symbols only
- Each s_i for $i = 2 \ldots n$ is derived from s_{i-1} by picking the left-most non-terminal X in s_{i-1} and replacing it by some β where $X \to \beta$ is a rule in R

For example: [S], [NP VP], [D N VP], [the N VP], [the man VP], [the man Vi], [the man sleeps]

Representation of a derivation as a tree:



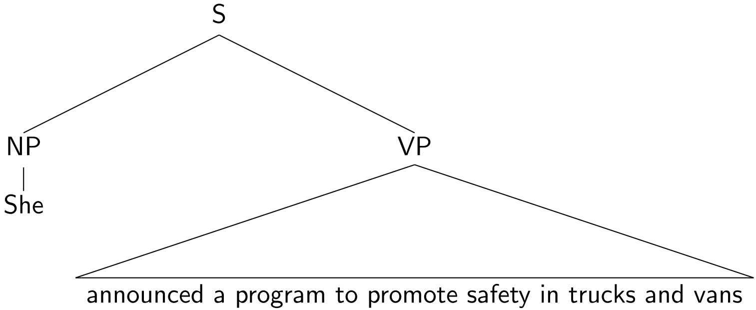
Properties of CFGs

- A CFG defines a set of possible derivations
- A string $s \in \Sigma^*$ is in the *language* defined by the CFG if there is at least one derivation which yields s
- Each string in the language generated by the CFG may have more than one derivation ("ambiguity")

Ambiguities

- I cleaned the dishes from dinner
- I cleaned the dishes with detergent
- I cleaned the dishes in my pajamas
- I cleaned the dishes in the sink

Exercise



- How many parse trees can be read?
- Provide a CFG to get at least one of the possible parse trees

Probabilistic CFGs can be used to know how likely are the parse trees

Outline

Example

S	$\Rightarrow$	NP	VP	1.0
VP	$\Rightarrow$	Vi		0.4
VP	$\Rightarrow$	Vt	NP	0.4
VP	$\Rightarrow$	VP	PP	0.2
NP	$\Rightarrow$	DT	NN	0.3
NP	$\Rightarrow$	NP	PP	0.7
PP	$\Rightarrow$	P	NP	1.0

Vi	$\Rightarrow$	sleeps	1.0
Vt	$\Rightarrow$	saw	1.0
NN	$\Rightarrow$	man	0.7
NN	$\Rightarrow$	woman	0.2
NN	$\Rightarrow$	telescope	0.1
DT	$\Rightarrow$	the	1.0
IN	$\Rightarrow$	with	0.5
IN	$\Rightarrow$	in	0.5

- Probability of a tree t with rules

$$\alpha_1 \rightarrow \beta_1, \alpha_2 \rightarrow \beta_2, \ldots, \alpha_n \rightarrow \beta_n$$

is

$$p(t) = \prod_{i=1}^{n} q(\alpha_i \rightarrow \beta_i)$$

where $q(\alpha \rightarrow \beta)$ is the probability for rule $\alpha \rightarrow \beta$.

Definition

1. *A context-free grammar* $G = (N, \Sigma, S, R)$.
2. *A parameter*
$$q(\alpha \to \beta)$$
for each rule $\alpha \to \beta \in R$. *The parameter* $q(\alpha \to \beta)$ *can be interpreted as the conditional probabilty of choosing rule* $\alpha \to \beta$ *in a left-most derivation, given that the non-terminal being expanded is* α. *For any* $X \in N$, *we have the constraint*
$$\sum_{\alpha \to \beta \in R : \alpha = X} q(\alpha \to \beta) = 1$$
In addition we have $q(\alpha \to \beta) \geq 0$ *for any* $\alpha \to \beta \in R$.

Given a parse-tree $t \in \mathcal{T}_G$ *containing rules* $\alpha_1 \to \beta_1, \alpha_2 \to \beta_2, \ldots, \alpha_n \to \beta_n$, *the probability of* t *under the PCFG is*

$$p(t) = \prod_{i=1}^{n} q(\alpha_i \to \beta_i)$$

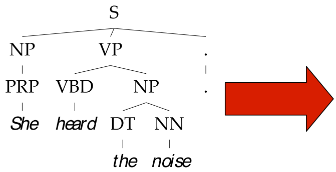
Properties of PCFGs

- Assigns a probability to each *left-most derivation*, or parse-tree, allowed by the underlying CFG
- Say we have a sentence s, set of derivations for that sentence is $\mathcal{T}(s)$. Then a PCFG assigns a probability $p(t)$ to each member of $\mathcal{T}(s)$. i.e., *we now have a ranking in order of probability*.
- The most likely parse tree for a sentence s is

$$\arg\max_{t \in \mathcal{T}(s)} p(t)$$

Learning Treebank Grammars

- Read the grammar rules from a treebank



S $\rightarrow$ NP VP .	1
NP $\rightarrow$ PRP	0.5
NP $\rightarrow$ DT NN	0.5
VP $\rightarrow$ VBD NP	1
PRP $\rightarrow$ She	1
…	

- Set rule weights by maximum likelihood
- Other approaches are out of this course: PCFG with parent annotations, lexicalized PCFG, PCFG with latent variables

Maximum Likelihood Estimates

- Algorithm
  1. Given a treebank, define a CFG **by taking all rules seen in the treebank**
  2. Maximum Likelihood estimates

$$q(\alpha \to \beta) = \frac{\mathrm{Count}(\alpha \to \beta)}{\mathrm{Count}(\alpha)}$$

     where the counts are taken from the examples in the treebank.
- Smoothing issues apply here
- Having the appropriate CFG is critical to success

Exercise

Using the following PCFG

$S \rightarrow NP\ VP$	1.0
$NP \rightarrow NP\ PP$	0.4
$PP \rightarrow P\ NP$	1.0
$VP \rightarrow V\ NP$	0.7
$VP \rightarrow VP\ PP$	0.3
$P \rightarrow with$	1.0
$V \rightarrow saw$	1.0
$NP \rightarrow astronomers$	0.1
$NP \rightarrow ears$	0.18
$NP \rightarrow saw$	0.04
$NP \rightarrow stars$	0.18
$NP \rightarrow telescope$	0.1

Work with the sentence: *'astronomers saw stars with ears'*

a) How many correct parses are there for this sentence?

b) Write them along with their probabilities.