Natural Language Processing

Syntactic Parsing

Based on slides from Yoav Goldberg, Jason Eisner, Michael Collins, Shuli Wintner

So far

- Word vectors
- Language modeling
    - n-gram models
    - neural models
- Tagging
    - HMMs
    - locally-normalized linear models
    - globally-normalized linear models
    - Deep learning models

Future

- Parsing: trees over sentence

- Generation: generate text/structure conditioned on textual input

Plan for today - syntax

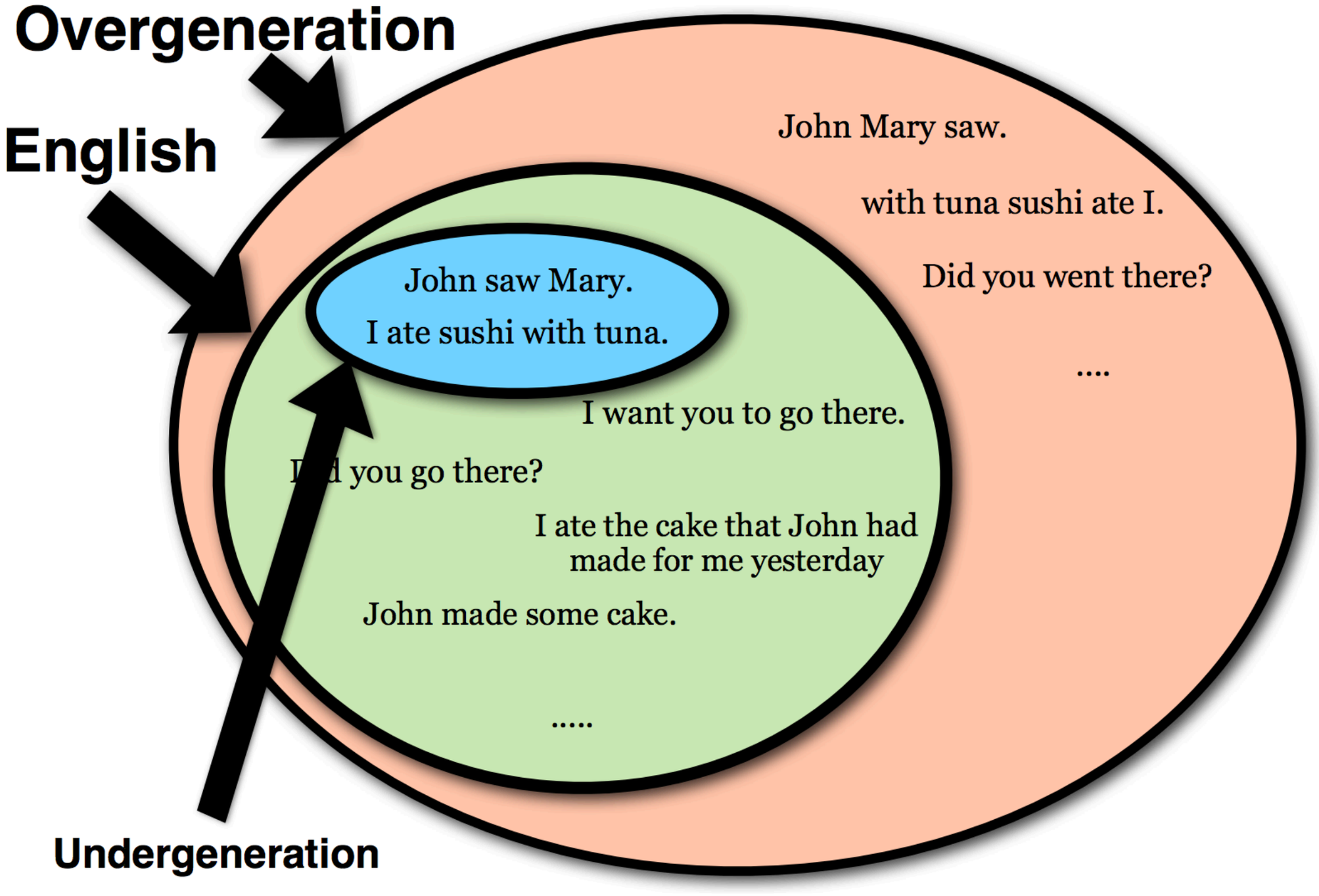
- Grammars
- Parsing
- Context-free grammars
- The syntax of English

Grammars

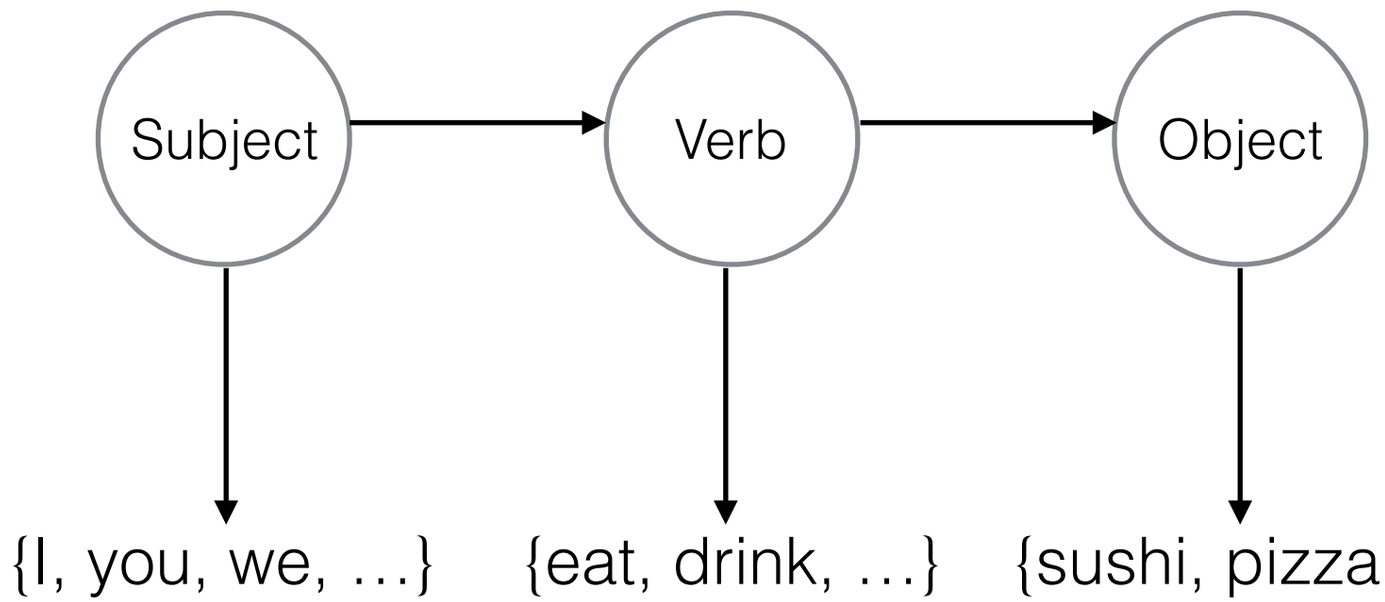
What are grammars?

set of structural rules governing the composition of clauses, phrases, and words in any given natural language...

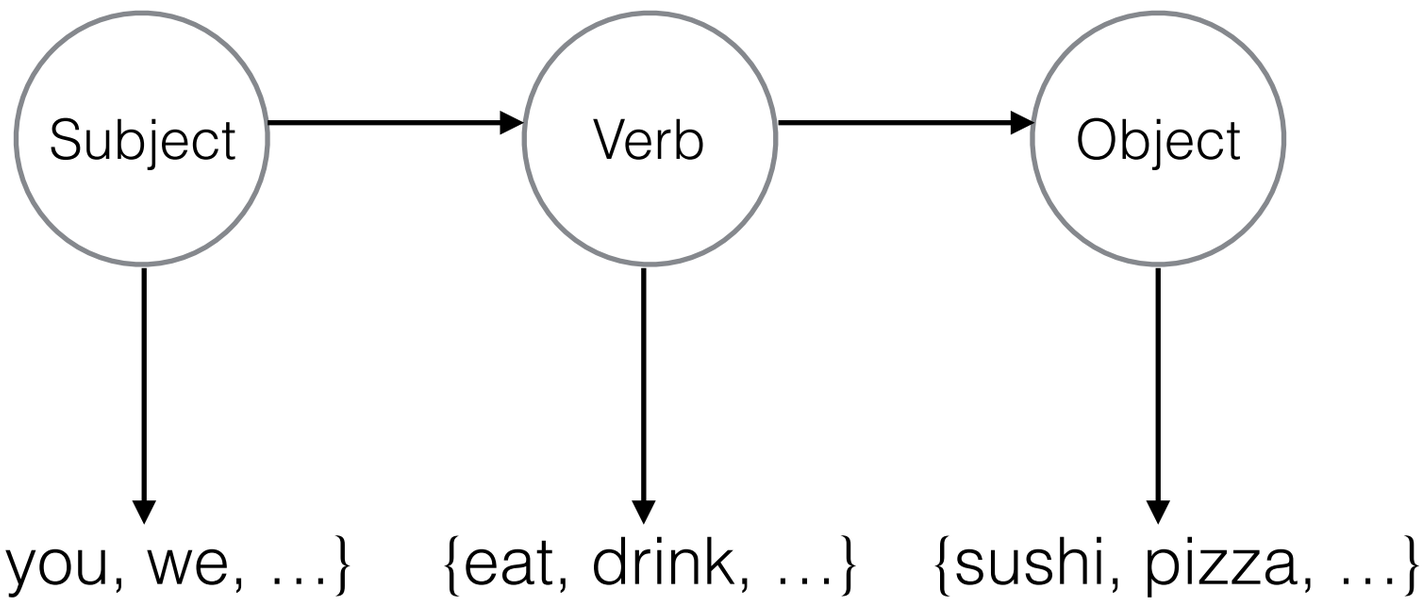
- Formalism
  - A method for describing the structure of language (CFG, TAG, HPSG, LFG, …)
- Instance
  - An implementation of a formalism in a particular language
  - Defines the infinite set of grammatical sentences


Overgeneration
English
John Mary saw.
with tuna sushi ate I.
Did you went there?
....
John saw Mary.
I ate sushi with tuna.
I want you to go there.
d you go there?
I ate the cake that John had made for me yesterday
John made some cake.
.....
Undergeneration

Sequence model?

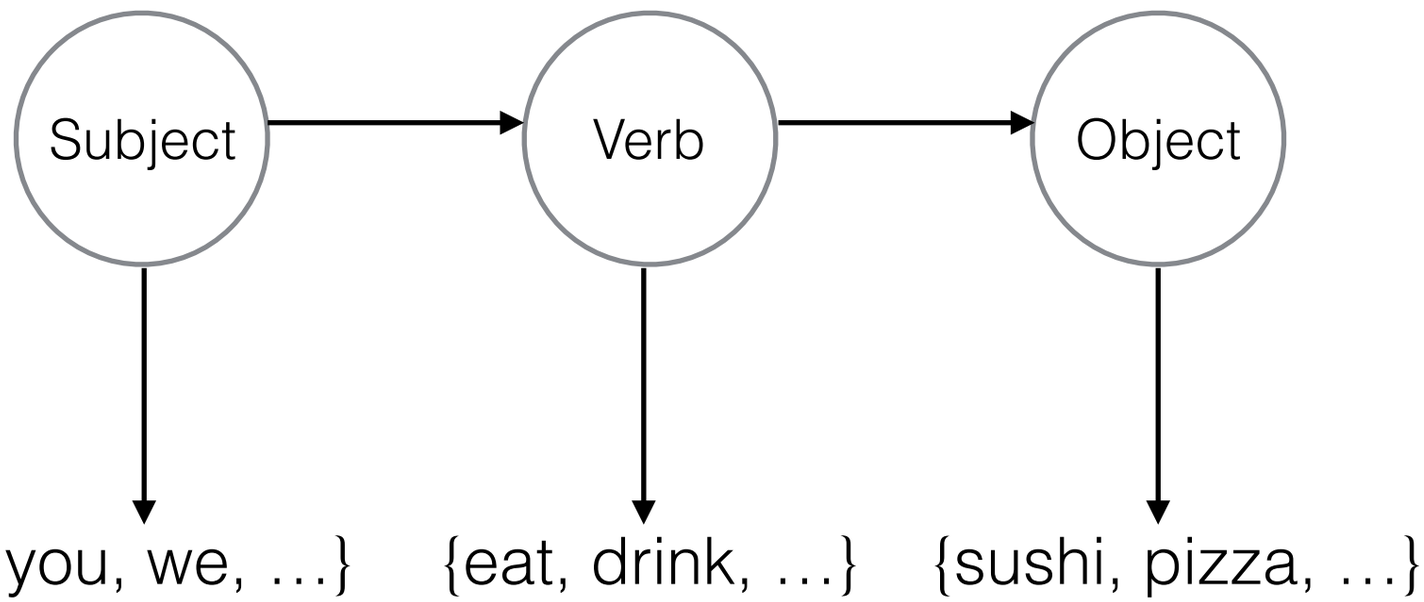


Sequence model?



Undergeneration: *I sleep*

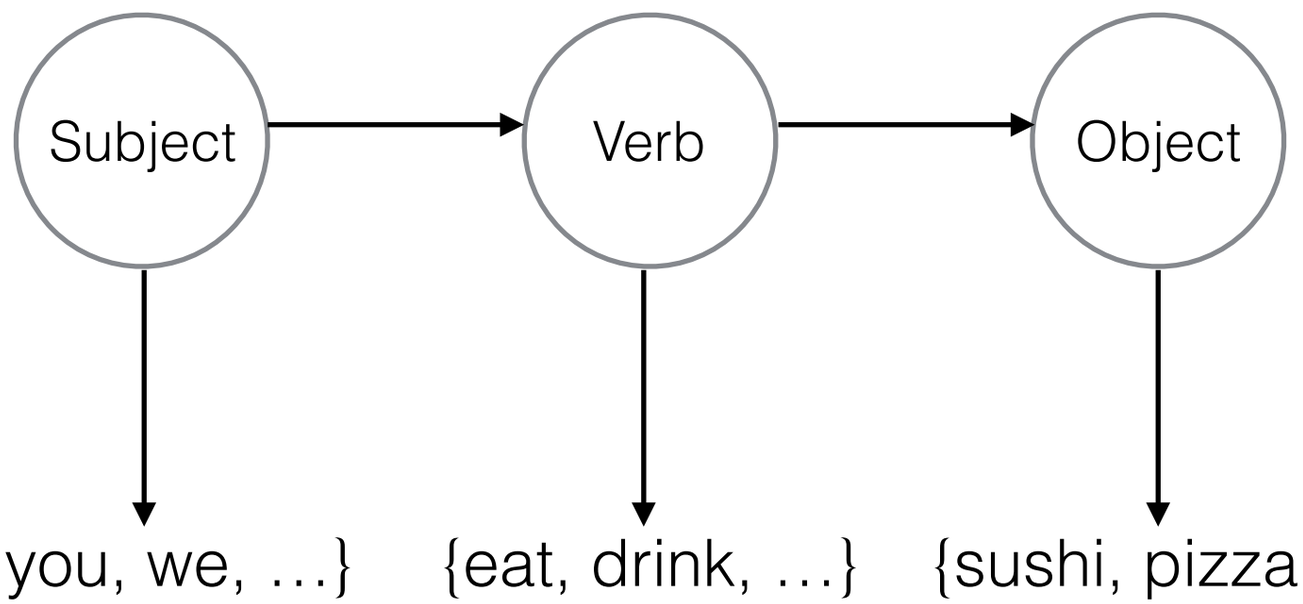
Sequence model?



Undergeneration: *I sleep*

Overgeneration: *I sleep sushi, I drink pizza*

Sequence model?

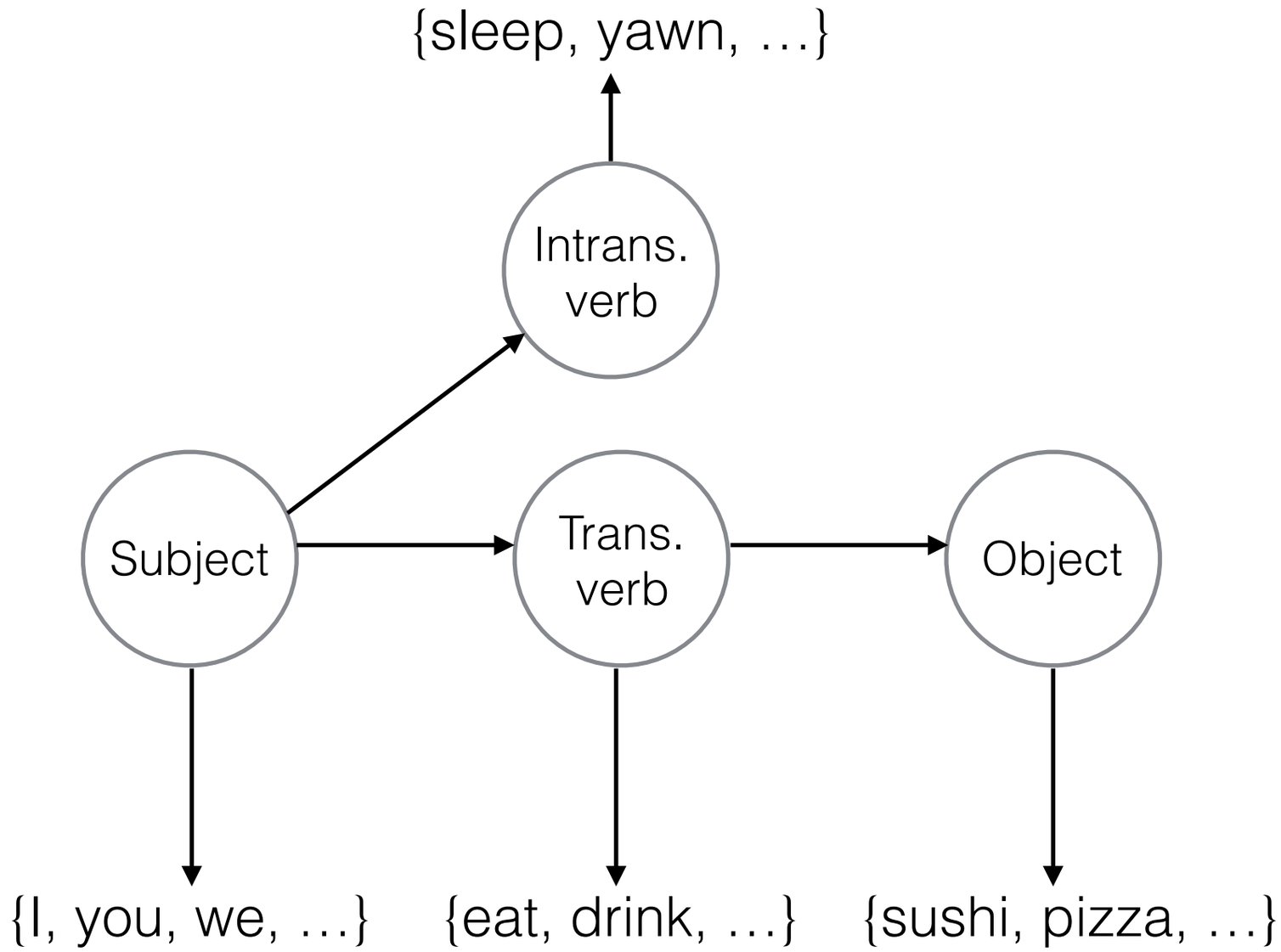


Undergeneration: *I sleep*

Overgeneration: *I sleep sushi, I drink pizza*

- Subcategorization: different verbs take a different number of arguments
- Selectional perference: verbs take certain types for semantic arguments

Sequence model?



Language is recursive

the ball

the big ball

the big red ball

the big red heavy ball

Nouns can take an infinite number of adjectives

Language is recursive

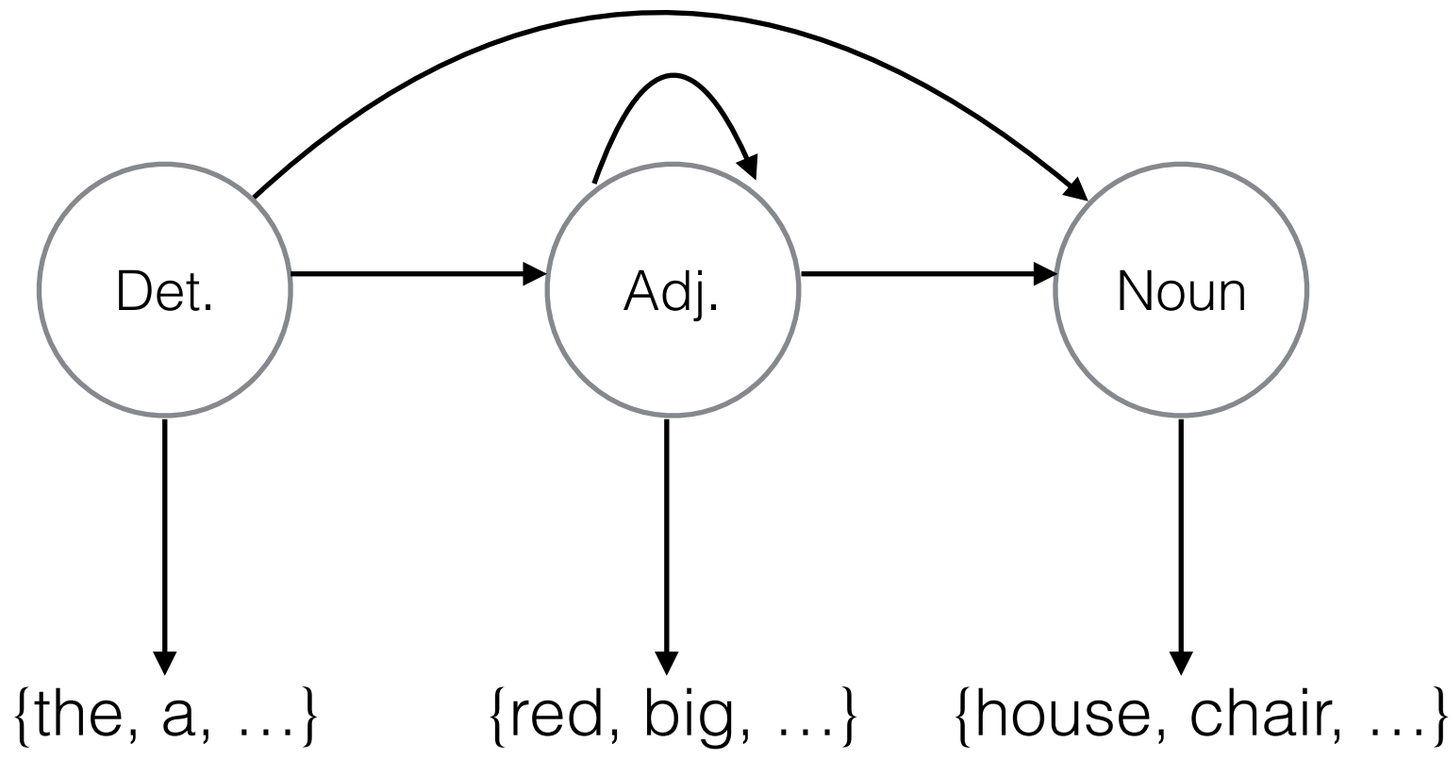
the ball
the big ball
the big red ball
the big red heavy ball

Nouns can take an infinite number of adjectives

*English is weird about adjective order (opinion, size, shape, age, color, nationality material)

- the big round old red house
- the red old round big house

Recursive sequence model



Hierarchical recursive structure

the cat likes tuna
the cat the dog chased likes tuna
the cat the dog the rat bit chased likes tuna
the cat the dog the rat the elephant admired bit chased likes tuna

$a^n b^n$ construction (not a regular langauge)

Competence vs. performance (Chomsky):

- Competence: idealized capacity
- Performance: what we actually utter

Example from Shuli Wintner

Context-free?

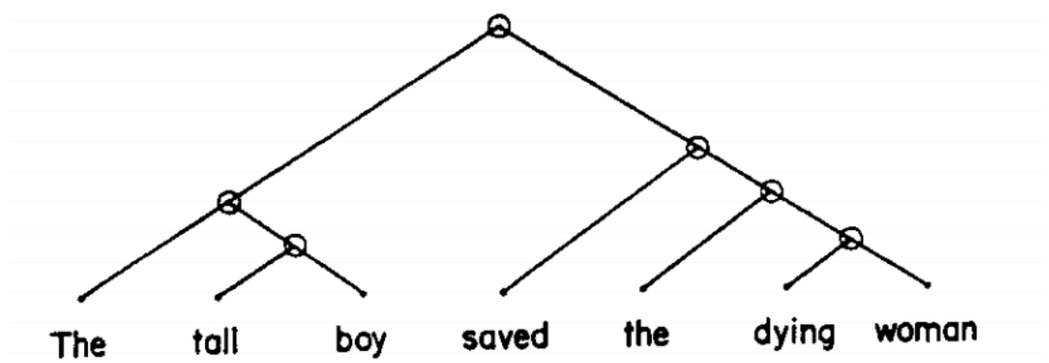
Chomsky (1957):
“English is not a regular language. As for context-free languages, I do not know whether or not English is itself literally outside the range of such analyses”

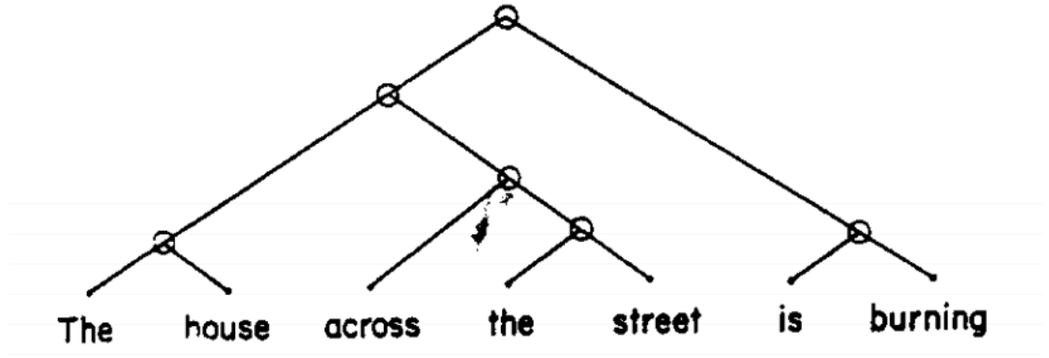
In swiss-german there are constructions like:

$$(\textit{acc-noun})^n\ (\textit{dative-noun})^m\ (\textit{acc-verb})^n\ (\textit{dative-verb})^m$$

Johnson, 1965

Language has structure





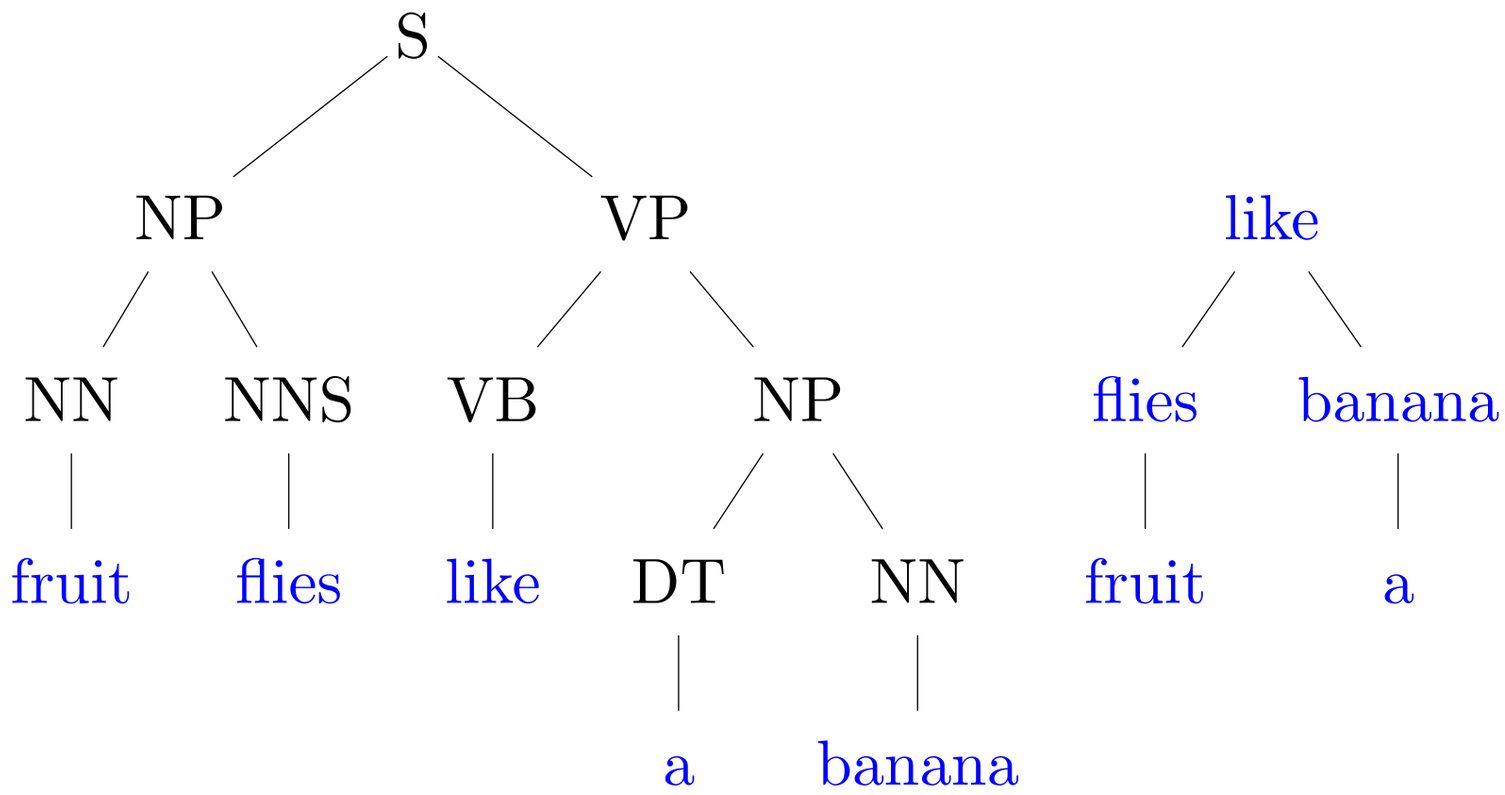
- Subjects asked to memorize sentences
- Probability of error related to phrase structure

Conclusion: our representation will be hierarchical

Strong vs. weak capacity

- Formal language theory:
  - Language is a set of strings
    - We care about generating the right set
- Formal syntax:
  - Language is a set of strings with structure
  - We care about strings having the right structure

Constituency vs. dependency



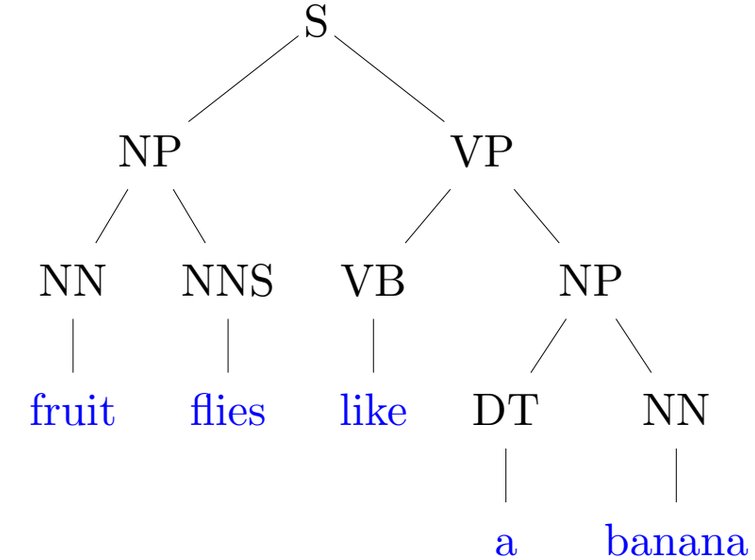
Constituency: words are leaves with part-of-speech tags as parents. Other nodes are syntactic categories

Dependency: All nodes are words. Each word is a **modifier** to a single **head**

Parsing

Goal

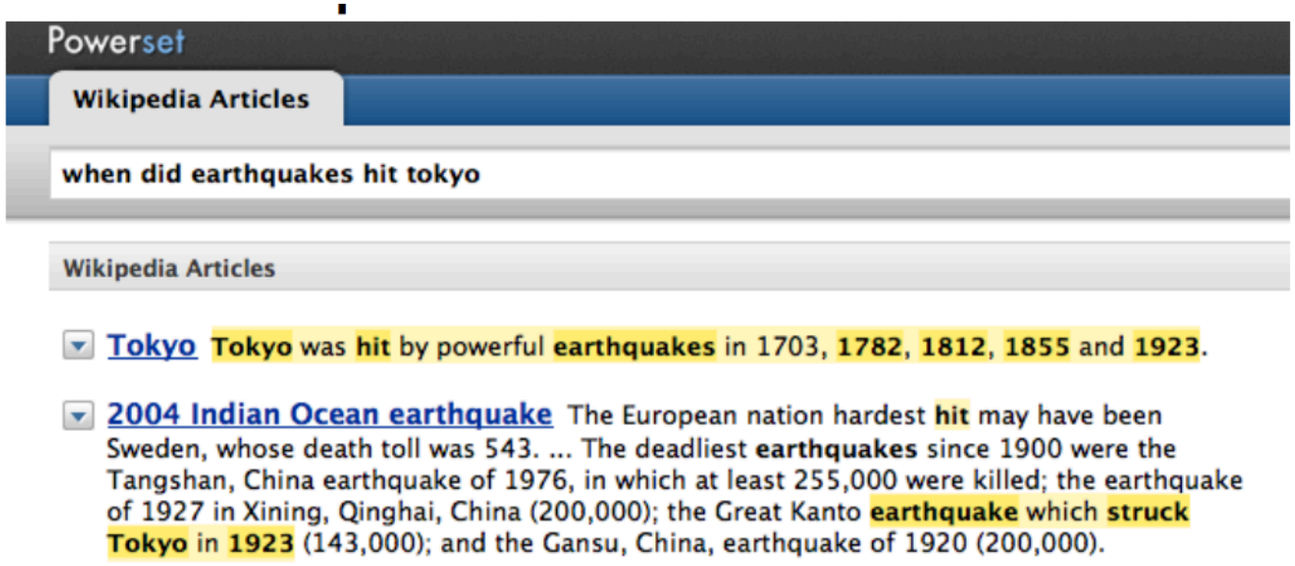
- Input: sentence
  - *Fruit flies like a banana*
- Output: constituency parse tree
- Method: supervised learning
  - Given (x,y) pairs of sentences and parse trees, learn a mapping from sentences to parse trees



What is it good for?

Information extraction

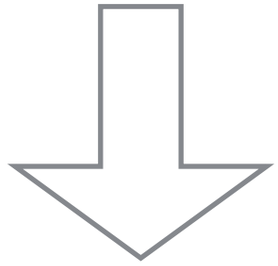
Question answering



What is it good for?

Summarization/simplification

The first new product, ATF Protype, is a line of digital postscript typefaces that will be sold in packages of up to six fonts.



ATF Protype is a line of digital postscript typefaces that will be sold in packages of up to six fonts .

What is it good for?

Machine translation: re-ordering of parse trees for English-Japanese translation

[SUBJECT] + TIME + PLACE/IMPLEMENT + INDIRECT OBJECT + OBJECT + ACTION VERB

Sources said that IBM bought Lotus yesterday
Sources yesterday IBM Lotus bought that said

Why is it hard?

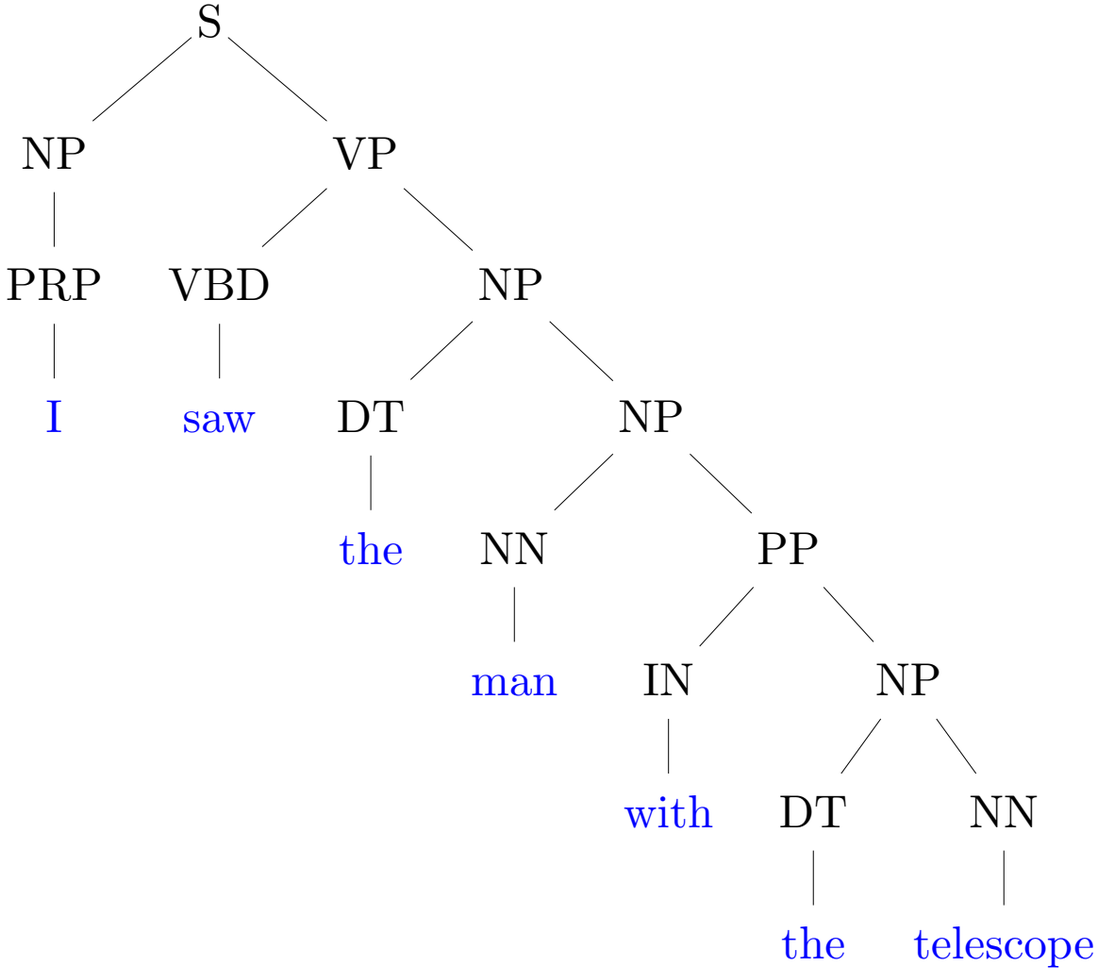
- Real sentences are long:

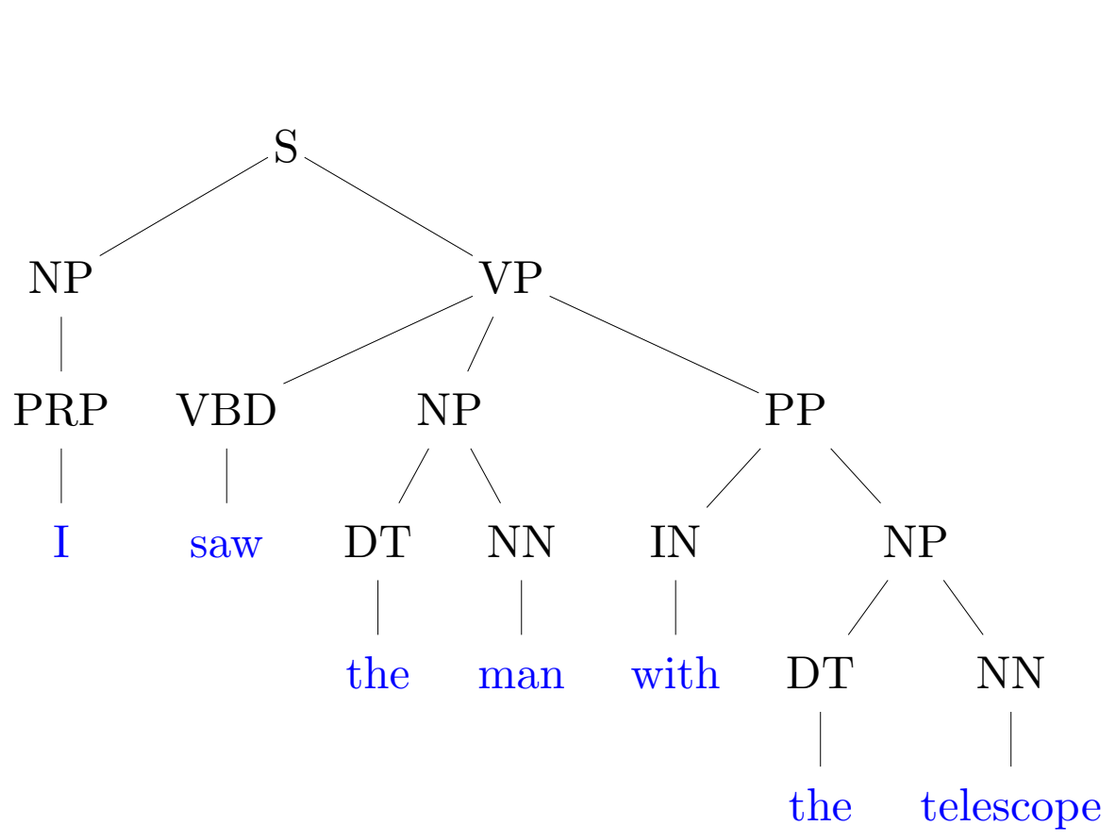
“Former Beatle Paul McCartney today was ordered to pay nearly $50M to his estranged wife as their bitter divorce battle came to an end.”

“Welcome to our Columbus hotels guide, where you’ll find honest, concise hotel reviews, all discounts, a lowest rate guarantee, and no booking fees.”

Why is it hard?

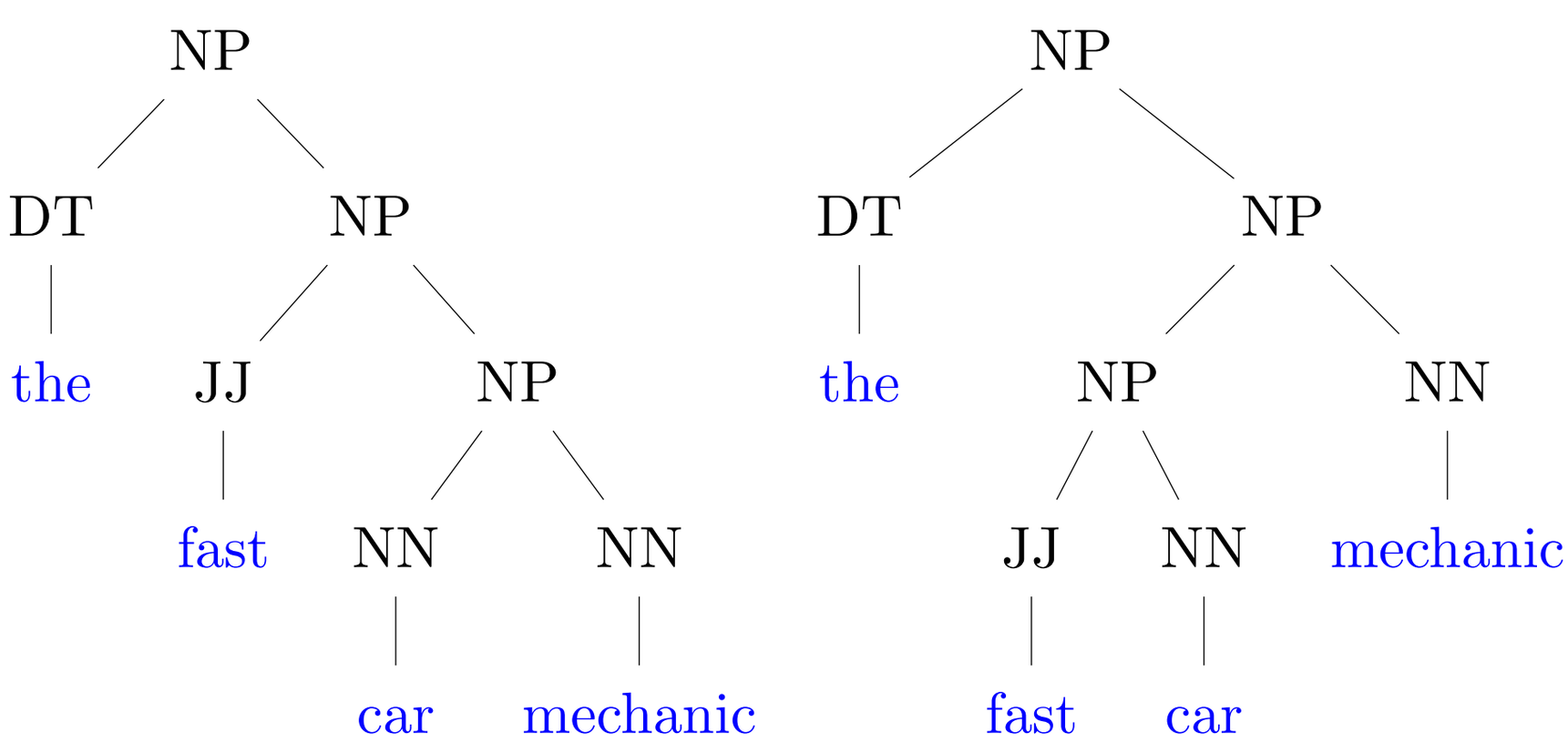
- Ambiguities: prepositional attachment





Why is it hard?

- Ambiguities:



Why is it hard?

- Ambiguities:
  - *She announced a program to promote safety in trucks and vans*

Context-free grammars

Context-free grammars

A context-free grammar (CFG) is a 4-tuple $G = (N, \Sigma, R, S)$:

N is a set of non-terminal symbols

Σ is a set of terminal symbols

R is a set of rules $X \to Y_1 Y_2 \ldots Y_n, n \geq 0, X \in N, Y_i \in N \cup \Sigma$

$S \in N$ is a special start symbol

Example

N = {S, NP, VP, PP, DT, Vi, Vt, NN, IN}

Σ = {sleeps, saw, man, woman, telescope, the, with, in}

R:

S	—>	NP VP	Vi	—>	sleeps
VP	—>	Vi	Vt	—>	saw
VP	—>	Vt NP	NN	—>	man
VP	—>	VP PP	NN	—>	woman
NP	—>	DT NN	NN	—>	telescope
NP	—>	NP PP	DT	—>	the
PP	—>	IN NP	IN	—>	with
			IN	—>	in

Left-most derivations

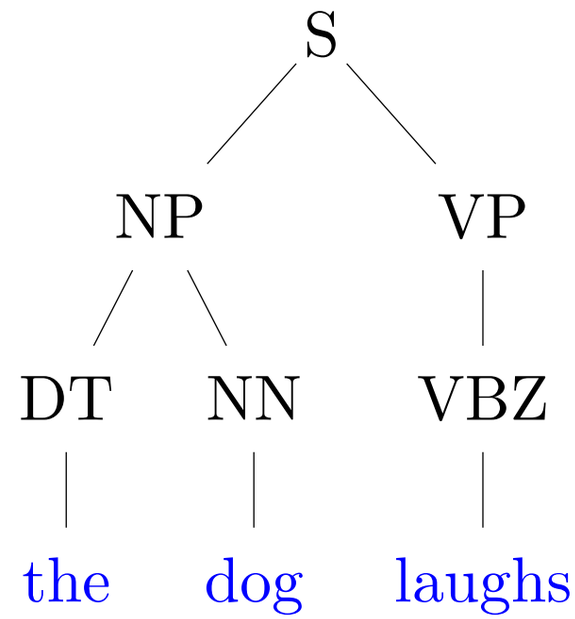
- Sequence of strings $s_1 \ldots s_n$, where
  - $s_1 = S$
  - s_n is a string in Σ^*
  - Each s_i for i=2…n is derived from s_{i-1} by picking the left-most non-terminal X in s_{i-1} and replacing it by some β where X—> β is a rule in R

Example

	Derivation	**Rule**
s_1	S	$S \rightarrow NP\ VP$
s_2	$NP\ VP$	$NP \rightarrow DT\ NN$
s_3	$DT\ NN\ VP$	$DT \rightarrow$ the
s_4	the $NN\ VP$	$NN \rightarrow$ dog
s_5	the dog VP	$VP \rightarrow VB$
s_6	the dog VBZ	$VBZ \rightarrow$ laughs
s_7	the dog laughs	

Example

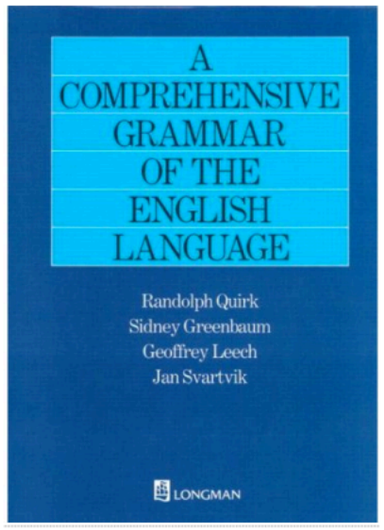
	Derivation	**Rule**
s_1	S	$S \rightarrow NP\ VP$
s_2	$NP\ VP$	$NP \rightarrow DT\ NN$
s_3	$DT\ NN\ VP$	$DT \rightarrow$ the
s_4	the $NN\ VP$	$NN \rightarrow$ dog
s_5	the dog VP	$VP \rightarrow VB$
s_6	the dog VBZ	$VBZ \rightarrow$ laughs
s_7	the dog laughs	



Properties of CFGs

- A CFG defines a set of derivations
- A string is in the language if there is a derivation that yields it
- Ambiguity is when the same string can be derived in multiple ways with left-most derivations

The syntax of English



Product Details (from Amazon)
Hardcover: 1779 pages
Publisher: Longman; 2nd Revised edition
Language: English
ISBN-10: 0582517346
ISBN-13: 978-0582517349
Product Dimensions: 8.4 x 2.4 x 10 inches
Shipping Weight: 4.6 pounds

English syntax

- Parts-of-speech (saw that already)

Noun phrase grammar

Ñ	—>	NN	NN	—>	box
Ñ	—>	NN Ñ	NN	—>	car
Ñ	—>	JJ Ñ	NN	—>	mechanic
Ñ	—>	Ñ Ñ	NN	—>	pigeon
NP	—>	DT Ñ	DT	—>	the
			DT	—>	a
			JJ	—>	fast
			JJ	—>	metal
			JJ	—>	idealistic
			JJ	—>	clay

We can generate:

- *the car, the fast car, the fast metal car*
- *the car mechanic, the fast car mechanic*

Prepositions

Ñ	—>	NN	NN	—>	box
Ñ	—>	NN Ñ	NN	—>	car
Ñ	—>	JJ Ñ	NN	—>	mechanic
Ñ	—>	Ñ Ñ	NN	—>	pigeon
NP	—>	DT Ñ	DT	—>	the
PP	—>	IN NP	DT	—>	a
Ñ	**—>**	**Ñ PP**	JJ	—>	fast
			JJ	—>	metal
			JJ	—>	idealistic
			JJ	—>	clay
			IN	**—>**	**in I under I…**

We can generate:

- *the fast car mechanic under the pigeon in the box*

Verbs, verb phrases and sentences

- Verb types
  - Vi: intransitive verbs (*sleeps, walks, yawns*)
  - Vt: transitive verbs (*see, like, hug, kiss*)
  - Vd: ditransitive verbs (*give, send*)
- VP rule
  - VP —> Vi
  - VP —> Vt NP
  - VP —> Vd NP NP
- Sentence rule
  - S —> NP VP

The dog gave the mechanic the fast car

PPs modifying verb phrases

- VP —>VP PP
    - *sleeps in the car, walks like the mechanic, gave the mechanic the fast car on Tuesday*

Complementizer and SBARs

- COMP—> that | which | …
- SBAR —> COMP S
  - *that the man sleeps, that the mechanic saw the dog*

More verb types

- V[5]—>said | reported
- V[6]—>told | informed
- V[7]—>bet

- VP —> V[5] SBAR
- VP —> V[6] NP SBAR
- VP —> V[7] NP NP SBAR
  - *said that the man sleeps*
  - *told the dog that the mechanic likes the pigeon*
  - *bet the pigeon 50$ that the mechanic owns a fast car*

Coordination

- CC —> and | or | but | …
- NP —> NP CC NP
- Ñ —> Ñ and Ñ
- VP —> VP CC VP
- S —> S CC S
- SBAR —> SBAR CC SBAR

There's more…

- Agreement
  - *the dog laughs* vs. *the dogs laugh*
- Wh-movement
  - *The dog that the cat liked ____*
- Active vs. passive
  - *the dog saw the cat* vs. *the cat was seen by the dog*