Foundations for Natural Language Processing
Lecture 14
Syntax and Parsing

Ivan Titov

(with slides from Ivan Titov, Alex Lascarides, Sharon Goldwater, Mark Steedman, and Marco Kuhlmann)

School of informatics

Plan

- Finish with tagging: Unsupervised Estimation with HMMs (EM for HMMs)
- Start with Syntax and Parsing

Recap: Hidden Markov Models

- We will consider the part-of-speech (POS) tagging example

John	**carried**	**a**	**tin**	**can**	**.**
NNP	VBD	DT	NN	NN	.

- A "generative" model, i.e.:
  - Model: Introduce a parameterized model of how both words and tags are generated $P(\mathbf{x}, \mathbf{y}|\theta)$
  - Learning: use a labeled training set to estimate the most likely parameters of the model $\hat{\theta}$
  - Decoding: $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} P(\mathbf{x}, \mathbf{y}|\theta)$

Recap: Viterbi

a_{ij}	STOP	NN	VB	JJ	RB
START	0	0.5	0.25	0.25	0
NN	0.25	0.25	0.5	0	0
VB	0.25	0.25	0	0.25	0.25
JJ	0	0.75	0	0.25	0
RB	0.5	0.25	0	0.25	0

b_{ik}	time	flies	fast	…	…	…
NN	0.1	0.01	0.01	…	…	…
VB	0.01	0.1	0.01	…	…	…
JJ	0	0	0.1	…	…	…
RB	0	0	0.1	…	…	…

Initialization: $v_i^1 = a_{START,i} b_{i,x^1}, \quad i = 1, \dots, N;$

$$v_j^t = \left(\max_i v_i^{t-1} a_{ij}\right) b_{j,x^t}, \;\; j = 1, \dots, N, \;\; t = 2, \dots, |x|$$

Final: $v_{STOP}^{|\mathbf{x}|+1} = \max_i v_i^{|\mathbf{x}|} a_{i,STOP}$

	time₁	*flies₂*	*fast₃*	-
NN	0.05	1.25E-4	6.25E-6	-
VB	0.0025	0.0025	6.25E-7	-
JJ	0	0	6.25E-5	-
RB	0	0	6.25E-5 x 0.5	-
STOP	-	-	-	6.25E-5 x 0.5

Recap: computing the likelihood

- From the probability theory we know

$$P(x|\theta) = \sum_y P(x, y|\theta)$$

- But there are an exponential number of sequences *y*
- Again, by computing and storing partial results, we can solve efficiently.

Recap: Forward algorithm (~ a modification of Viterbi)

Initialization: $v_j^1 = a_{START,j} b_{j,x^1}, \quad j = 1, \ldots, N;$

Recomputation: $v_j^t = \left(\sum_i v_i^{t-1} a_{ij} \right) b_{j,x^t}, \; j = 1, \ldots, N, \; t = 2, \ldots, |x|$

Final: $v_{STOP}^{|\mathbf{x}|+1} = \sum_i v_i^{|\mathbf{x}|} a_{i,STOP}$

Recap: EM for Naïve Bayes

		Bayes	your	model	cash	Viagra	class	orderz	spam?
labeled data	lab doc 1	0	1	3	0	0	2	0	-
	lab doc 2	0	2	0	4	0	0	0	+
	lab doc 3	0	2	2	0	0	3	0	-
	lab doc 4	0	3	2	1	3	0	1	+
	lab doc 5	0	1	0	2	0	0	1	+
unlabeled data	unl doc 2	2 x 0.53	2 x0.53	0	0	0	0	0	+ (.53)
		2 x 0.47	2 x 0.47	0	0	0	0	0	- (.47)

$$\hat{P}(\text{your}|+) = (6 + 2 \times 0.53 + \alpha)/(20 + 4 \times 0.53 + \alpha * F)$$

$$\hat{P}(\text{your}|-) = (3 + 2 \times 0.47 + \alpha)/(13 + 3 \times 0.47 + \alpha * F)$$

$$\hat{P}(\text{spam}) = \frac{3 + 0.53}{5 + 1}$$

This is just for one data point

$$\hat{P}(\text{Bayes}|+) = (2 \times 0.53 + \alpha)/(20 + 4 \times 0.53 + \alpha * F)$$

$$\hat{P}(\text{Bayes}|-) = (2 \times 0.47 + \alpha)/(13 + 4 \times 0.47 + \alpha * F)$$

EM for Semi-supervised Learning

1. Train NB on labeled data alone
2. Make soft prediction on on unlabelled data ("E-step")
3. Recompute NB parameters using the soft counts

HMM and Unsupervised Estimation

- N – the number tags, M – vocabulary size
- Parameters (to be estimated from the training set):
  - Transition probabilities $a_{ji} = P(y^t = i | y^{t-1} = j)$, A - [N x N] matrix
  - Emission probabilities $b_{ik} = P(x^t = k | y^t = i)$, B - [N x M] matrix
- Training corpus:
  - $\boldsymbol{x}^{(1)}=$ (In, an, Oct., 19, review, of, ….)
  - $\boldsymbol{x}^{(2)}=$ (Ms., Haag, plays, Elianti,.)
  - …
  - $\boldsymbol{x}^{(L)}=$ (The, company, said,…)

Estimation is trickier: no examples labelled with the right 'answers': all we see are outputs, state sequence is hidden.

Circularity

- If we know the state sequence, we can find the best parameters
  - E.g., use MLE / normalized counts
- If we know parameters, we can find the best state sequence
  - Use Viterbi
- But we do not know them either

- Does not fit the self-training algorithm, but can we make EM work in this setting?

Expectation-maximization (EM)

- As in spelling correction, we can use EM to bootstrap, iteratively updating the parameters and hidden variables.
  - Initialize parameters, $A^{(0)}$ and $B^{(0)}$
  - At each iteration k:
    - E-step: Compute expected counts using $A^{(k-1)}$ and $B^{(k-1)}$
    - M-step: set $A^{(k-1)}$ and $B^{(k-1)}$ using MLE on the expected counts
- Repeat until doesn't converge (a stopping criteria).

Expected counts??

- Counting transitions from ($y^{t-1}= i, y^t=j$):
- Real counts:
  - count 1 each time we see *($y^{t-1}= i, y^t=j$)* in true tag sequence.
- Expected counts:
  - With current A and B, compute probs of all possible tag sequences.
  - If sequence **y** has probability p, count p for each *($y^{t-1}= i, y^t=j$)* in **y**.
  - Add up these fractional counts across all possible sequences

Example

- Notionally, we compute expected counts as follows:

Possible tag sequence			Probability of the sequence
N	N	N	p_1
N	V	N	p_2
N	N	V	p_3
aa	bb	cc	- Sequence of observations (words)

$C_T(N, N) =$

Example

- Notionally, we compute expected counts as follows:

Possible tag sequence	Probability of the sequence
N N N	p_1
N V N	p_2
N N V	p_3
aa bb cc	- Sequence of observations (words)

$$C_T(N, N) = 2\,p_1 + p_3$$

Forward-Backward algorithm

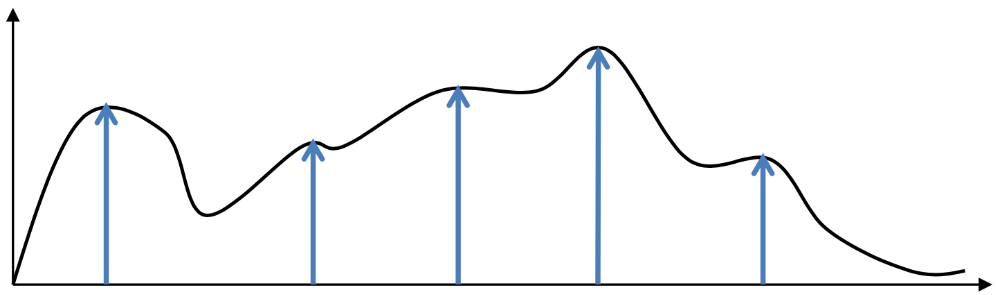
- As usual, avoid enumerating all possible sequences.
- Forward-Backward (Baum-Welch) algorithm computes expected counts (using forward probabilities and backward probabilities)

$$P(y^{t-1} = i, y^t = j | x)$$

- EM idea is much more general: can use for many latent variable models.

EM

- EM is guaranteed to find a local maximum of the likelihood.



- Not guaranteed to find global maximum.
- Practical issues: initialization, random restarts, early stopping. Fact is, it doesn't work well for learning POS taggers!

Summary for HMM / Tagging

- HMM : a generative model of sentences using hidden state sequences
- Dynamic programming algorithms to compute
  - Best tag sequence given words (Viterbi algorithm)
  - Likelihood (forward algorithm)
  - Best parameters from unannotated corpus (forward-backward, an instance of EM)

You can read details of forward-backward in the text-book, not required for the exam (but Viterbi and Forward are)

Syntax

- Basics of Syntax, Context-Free Grammars
- Classes of Syntactic Parsing Algorithms
- Start with the CKY algorithm

Modelling word behaviour

- We've seen various ways to model word behaviour.
  - Bag-of-words models: ignore word order entirely
  - N-gram models: capture a fixed-length history to predict word sequences.
  - HMMs: also capture fixed-length history, using latent variables.
- Useful for various tasks, but a really accurate model of language needs more than a fixed-length history!

Long-range dependencies

- The form of one word often depends on (agrees with) another, even when arbitrarily long material intervenes.

  Sam/Dogs sleeps/sleep soundly
  Sam, who is my cousin, sleeps soundly
  Dogs often stay at my house and sleep soundly
  Sam, the man with red hair who is my cousin, sleeps soundly

- We want models that can capture these dependencies.

Phrasal categories

- We may also want to capture substitutability at the phrasal level.
  - POS categories indicate which words are *substitutable*. For example, substituting **adjectives**:

    I saw a red cat
    I saw a former cat
    I saw a billowy cat

  - Phrasal categories indicate which phrases are substitutable. For example, substituting **noun phrase**:

    Dogs sleep soundly
    My next-door neighbours sleep soundly
    Green ideas sleep soundly

This is one example of "**constituency test**"

Example constituency tests: coordination

- Only constituents (of the same type) can be **coordinated** using conjunction words like *and*, *or*, and *but*
- Pass the test:

Her friends from Peru went to the show.
Mary *and* her friends from Peru went to the show.

Should I go through the tunnel?
Should I go through the tunnel *and* over the bridge?

- Fail the test

We peeled the potatoes.
*We peeled the and washed the potatoes.

Example constituency tests: clefting

- Only a constituent can appear in the frame "_____ *is/are who/what/where/when/why/how* …"
- Pass the test:

They put the boxes in the basement.
In the basement *is where* they put the boxes.

- Fail the test

They put the boxes in the basement.
*Put the boxes is what they did in the basement.

Theories of syntax

- A theory of syntax should explain which sentences are well-formed (grammatical) and which are not.
  - Note that well-formed is distinct from meaningful.
  - Famous example from Chomsky:

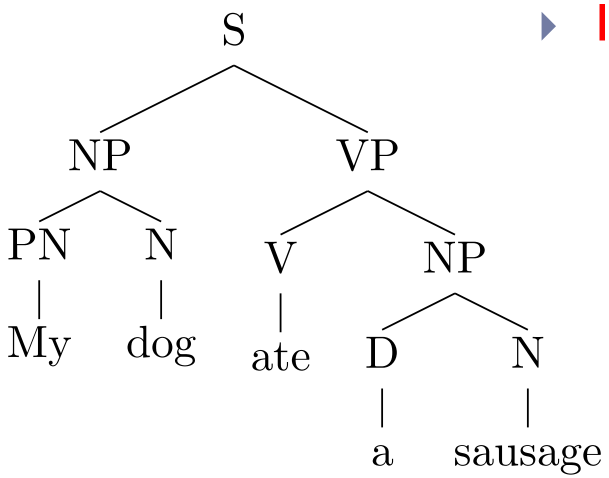
    Colorless green ideas sleep furiously
- However we'll see shortly that the reason we care about syntax is mainly for interpreting meaning.

Theories of syntax

- We'll look at two theories of syntax:
  - Constituency (aka phrase) structures: next two classes
  - Dependency structures: during the last lecture on parsing

- These can be viewed as different models of language behaviour. As with other models, we will look at
  - What each model can capture, and what it cannot.
  - Algorithms that provide syntactic analyses for sentences using these models (i.e., syntactic parser).

Constituent trees



- Internal nodes correspond to phrases

  S – a sentence

  NP (Noun Phrase): My dog, a sandwich, lakes, ..

  VP (Verb Phrase): ate a sausage, barked, …

  PP (Prepositional phrases): with a friend, in a car, …

- Nodes immediately above words are PoS tags

  PN – pronoun
  D – determiner
  V – verb
  N – noun
  P – preposition

Context-Free Grammar

- Context-free grammar is a tuple of 4 elements $G = (V, \Sigma, R, S)$
  - V - the set of non-terminals

    In our case: phrase categories (VP, NP, ..) and PoS tags (N, V, .. – aka preterminals)

  - Σ - the set of terminals

    Words

  - R - the set of rules of the form $X \to Y_1, Y_2, \ldots, Y_n$, where $n \geq 0$, $X \in V,\ Y_i \in V \cup \Sigma$

    $$S \to NP\ \ VP \qquad N \to girl$$
    $$NP \to D\ \ N \qquad N \to telescope$$
    $$NP \to PN \qquad V \to saw$$
    $$NP \to NP\ \ PP \qquad V \to eat$$
    $$PP \to P\ \ NP \qquad \ldots$$
    $$\ldots$$

  - S is a dedicated start symbol

An example grammar

$V = \{S, VP, NP, PP, N, V, PN, P\}$

$\Sigma = \{girl, telescope, sandwich, I, saw, ate, with, in, a, the\}$

$S = \{S\}$

$R:$

Inner rules

$S \rightarrow NP \ \ VP$	(NP A girl) (VP ate a sandwich)
$VP \rightarrow V$	
$VP \rightarrow V \ \ NP$	(V ate) (NP a sandwich)
$VP \rightarrow VP \ \ PP$	(VP saw a girl) (PP with a telescope)
$NP \rightarrow NP \ \ PP$	(NP a girl) (PP with a sandwich)
$NP \rightarrow D \ \ N$	(D a) (N sandwich)
$NP \rightarrow PN$	
$PP \rightarrow P \ \ NP$	(P with) (NP with a sandwich)

Preterminal rules

$N \rightarrow girl$

$N \rightarrow telescope$

$N \rightarrow sandwich$

$PN \rightarrow I$

$V \rightarrow saw$

$V \rightarrow ate$

$P \rightarrow with$

$P \rightarrow in$

$D \rightarrow a$

$D \rightarrow the$

CFGs

S

$$\boxed{S \rightarrow NP \;\; VP}$$

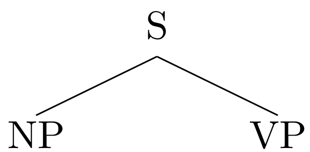
$$VP \rightarrow V$$
$$VP \rightarrow V \;\; NP$$
$$VP \rightarrow VP \;\; PP$$

$$NP \rightarrow NP \;\; PP$$
$$NP \rightarrow D \;\; N$$
$$NP \rightarrow PN$$

$$PP \rightarrow P \;\; NP$$

$$N \rightarrow girl$$
$$N \rightarrow telescope$$
$$N \rightarrow sandwich$$
$$PN \rightarrow I$$
$$V \rightarrow saw$$
$$V \rightarrow ate$$
$$P \rightarrow with$$
$$P \rightarrow in$$
$$D \rightarrow a$$
$$D \rightarrow the$$

CFGs



$$S \rightarrow NP\ \ VP$$

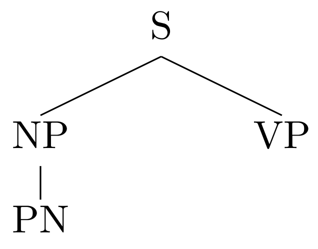
$$VP \rightarrow V$$
$$VP \rightarrow V\ \ NP$$
$$VP \rightarrow VP\ \ PP$$

$$NP \rightarrow NP\ \ PP$$
$$NP \rightarrow D\ \ N$$
$$\boxed{NP \rightarrow PN}$$

$$PP \rightarrow P\ \ NP$$

$$N \rightarrow girl$$
$$N \rightarrow telescope$$
$$N \rightarrow sandwich$$
$$PN \rightarrow I$$
$$V \rightarrow saw$$
$$V \rightarrow ate$$
$$P \rightarrow with$$
$$P \rightarrow in$$
$$D \rightarrow a$$
$$D \rightarrow the$$

CFGs



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

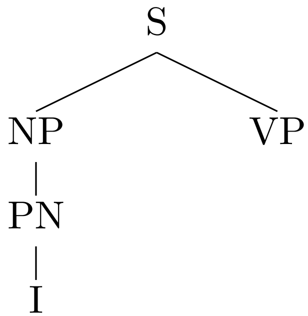
$$VP \rightarrow V$$
$$VP \rightarrow V \;\; NP$$
$$VP \rightarrow VP \;\; PP$$

$$NP \rightarrow NP \;\; PP$$
$$NP \rightarrow D \;\; N$$
$$NP \rightarrow PN$$

$$PP \rightarrow P \;\; NP$$

$$N \rightarrow girl$$
$$N \rightarrow telescope$$
$$N \rightarrow sandwich$$
$$\boxed{PN \rightarrow I}$$
$$V \rightarrow saw$$
$$V \rightarrow ate$$
$$P \rightarrow with$$
$$P \rightarrow in$$
$$D \rightarrow a$$
$$D \rightarrow the$$

CFGs



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

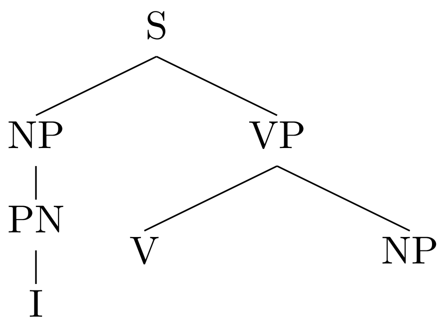
$$VP \rightarrow V$$
$$\boxed{VP \rightarrow V \;\; NP}$$
$$VP \rightarrow VP \;\; PP$$

$$NP \rightarrow NP \;\; PP$$
$$NP \rightarrow D \;\; N$$
$$NP \rightarrow PN$$

$$PP \rightarrow P \;\; NP$$

$$N \rightarrow girl$$
$$N \rightarrow telescope$$
$$N \rightarrow sandwich$$
$$PN \rightarrow I$$
$$V \rightarrow saw$$
$$V \rightarrow ate$$
$$P \rightarrow with$$
$$P \rightarrow in$$
$$D \rightarrow a$$
$$D \rightarrow the$$

CFGs



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

$$VP \rightarrow V$$

$$VP \rightarrow V \;\; NP$$

$$VP \rightarrow VP \;\; PP$$

$$NP \rightarrow NP \;\; PP$$

$$NP \rightarrow D \;\; N$$

$$NP \rightarrow PN$$

$$PP \rightarrow P \;\; NP$$

$$N \rightarrow girl$$

$$N \rightarrow telescope$$

$$N \rightarrow sandwich$$

$$PN \rightarrow I$$

$$\boxed{V \rightarrow saw}$$

$$V \rightarrow ate$$

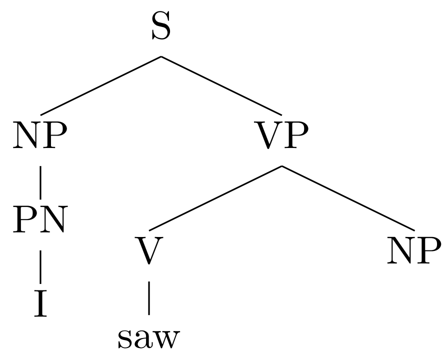
$$P \rightarrow with$$

$$P \rightarrow in$$

$$D \rightarrow a$$

$$D \rightarrow the$$

CFGs



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

$$VP \rightarrow V$$

$$VP \rightarrow V \;\; NP$$

$$VP \rightarrow VP \;\; PP$$

$$\boxed{NP \rightarrow NP \;\; PP}$$

$$NP \rightarrow D \;\; N$$

$$NP \rightarrow PN$$

$$PP \rightarrow P \;\; NP$$

$$N \rightarrow girl$$

$$N \rightarrow telescope$$

$$N \rightarrow sandwich$$

$$PN \rightarrow I$$

$$V \rightarrow saw$$

$$V \rightarrow ate$$

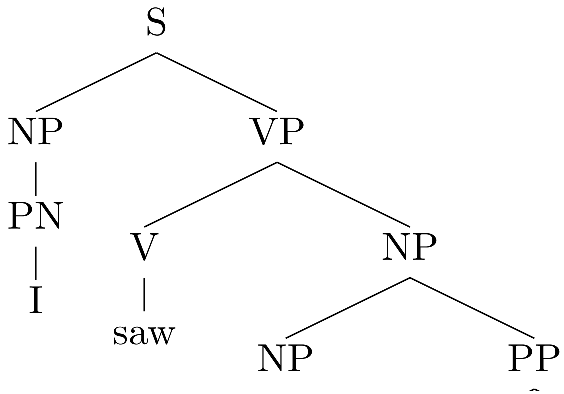
$$P \rightarrow with$$

$$P \rightarrow in$$

$$D \rightarrow a$$

$$D \rightarrow the$$

CFGs



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

$$VP \rightarrow V$$

$$VP \rightarrow V \;\; NP$$

$$VP \rightarrow VP \;\; PP$$

$$NP \rightarrow NP \;\; PP$$

$$\boxed{NP \rightarrow D \;\; N}$$

$$NP \rightarrow PN$$

$$PP \rightarrow P \;\; NP$$

$$N \rightarrow girl$$

$$N \rightarrow telescope$$

$$N \rightarrow sandwich$$

$$PN \rightarrow I$$

$$V \rightarrow saw$$

$$V \rightarrow ate$$

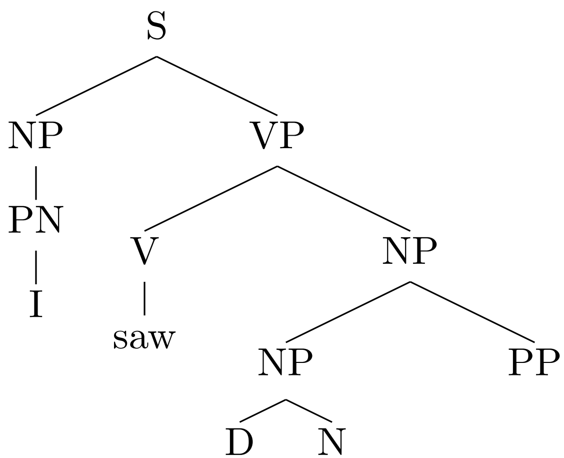
$$P \rightarrow with$$

$$P \rightarrow in$$

$$D \rightarrow a$$

$$D \rightarrow the$$

CFGs



$$S \rightarrow NP \ \ VP$$

$$VP \rightarrow V$$

$$VP \rightarrow V \ \ NP$$

$$VP \rightarrow VP \ \ PP$$

$$NP \rightarrow NP \ \ PP$$

$$NP \rightarrow D \ \ N$$

$$NP \rightarrow PN$$

$$PP \rightarrow P \ \ NP$$

$$N \rightarrow girl$$

$$N \rightarrow telescope$$

$$N \rightarrow sandwich$$

$$PN \rightarrow I$$

$$V \rightarrow saw$$

$$V \rightarrow ate$$

$$P \rightarrow with$$

$$P \rightarrow in$$

$$D \rightarrow a$$

$$D \rightarrow the$$

CFGs

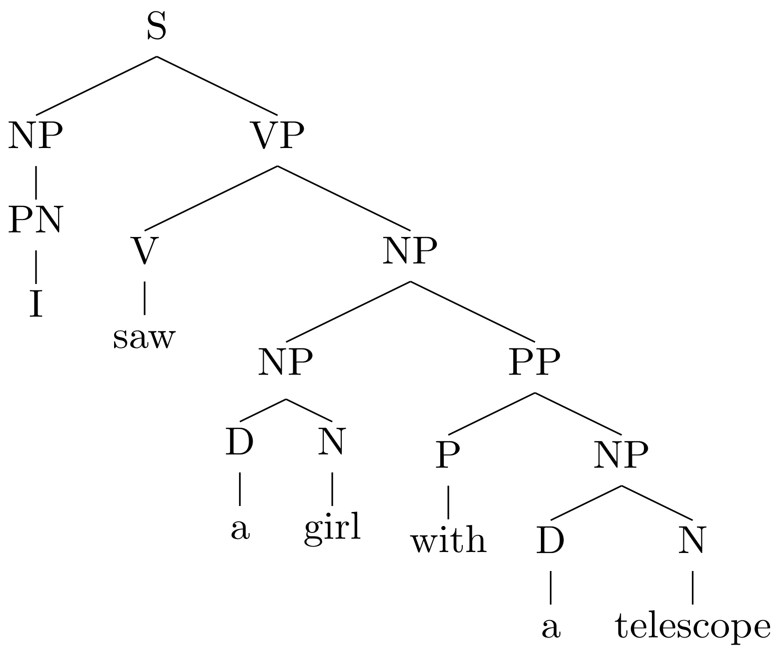
$$S \rightarrow NP\ \ VP$$

$$VP \rightarrow V$$
$$VP \rightarrow V\ \ NP$$
$$VP \rightarrow VP\ \ PP$$

$$NP \rightarrow NP\ \ PP$$
$$NP \rightarrow D\ \ N$$
$$NP \rightarrow PN$$

$$PP \rightarrow P\ \ NP$$

$$N \rightarrow girl$$
$$N \rightarrow telescope$$
$$N \rightarrow sandwich$$
$$PN \rightarrow I$$
$$V \rightarrow saw$$
$$V \rightarrow ate$$
$$P \rightarrow with$$
$$P \rightarrow in$$
$$D \rightarrow a$$
$$D \rightarrow the$$

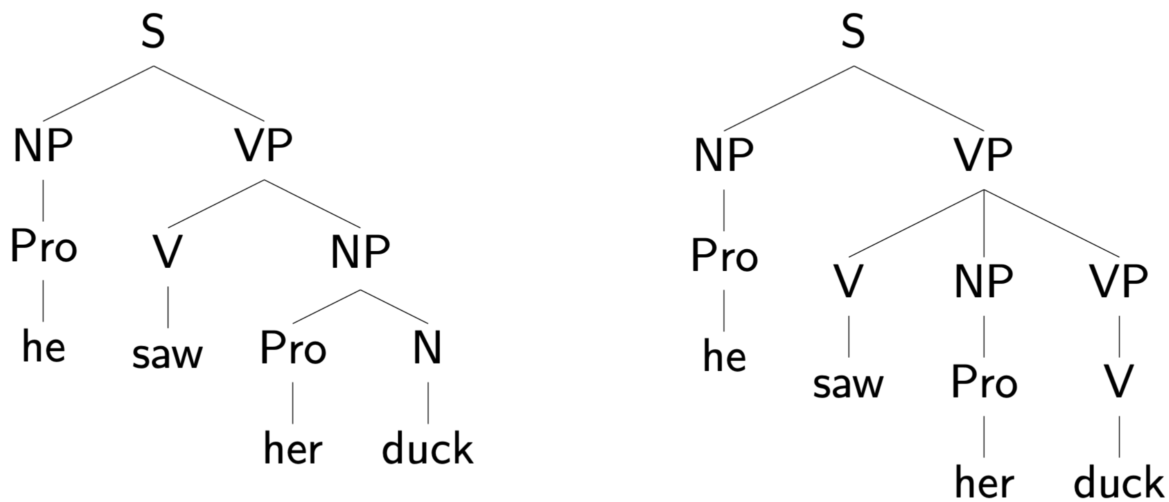


CFG defines both:

- a set of strings (a language)
- structures used to represent sentences (constituent trees)

Structural ambiguity

- Some sentences have more than one parse: structural ambiguity.



- Here, the structural ambiguity is caused by PoS ambiguity in several of the words. (Both are types of syntactic ambiguity.)

Structural ambiguity

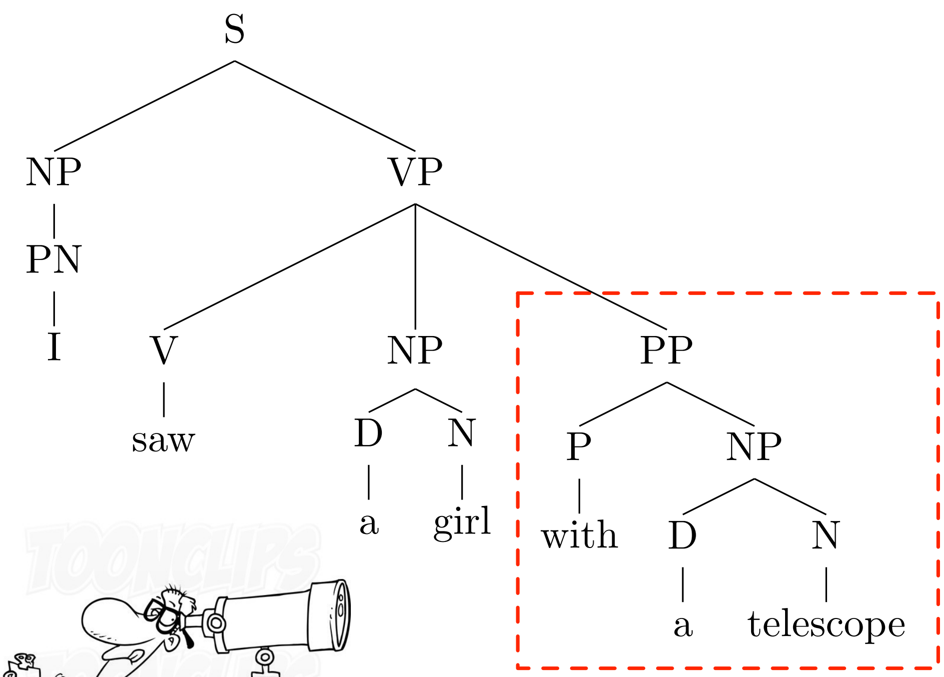
- Some sentences have structural ambiguity even **without** part-of-speech ambiguity. This is called attachment ambiguity.
  - Depends on where different phrases attach in the tree.
  - Different attachments have different meanings:

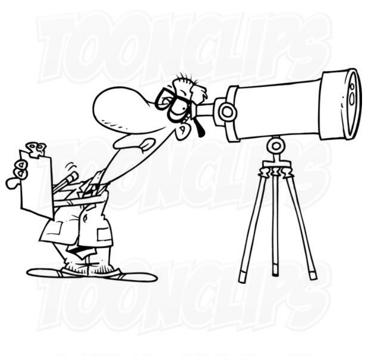
  I saw a girl with a telescope
  She ate the pizza on the floor
  Good boys and girls get presents from Santa

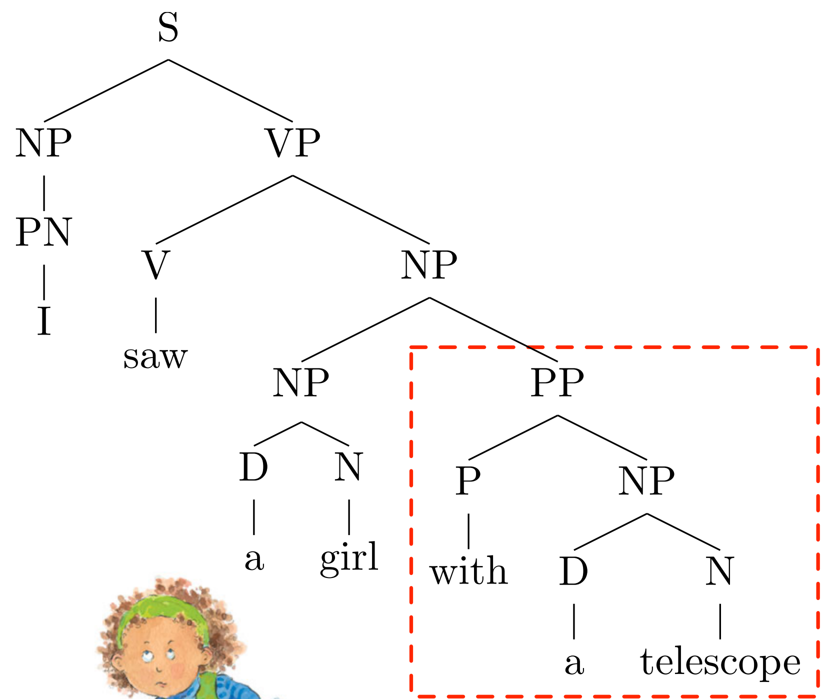
- Next slide shows trees for the first example: prepositional phrase (PP) attachment ambiguity.

Prepositional Phrase (PP-) Attachment Ambiguity

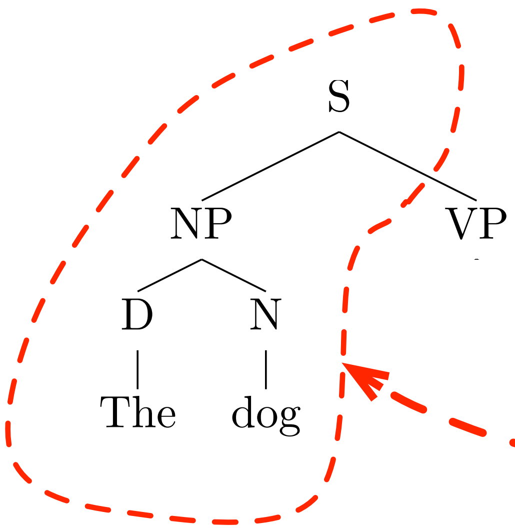
- Prepositional phrase attachment ambiguity





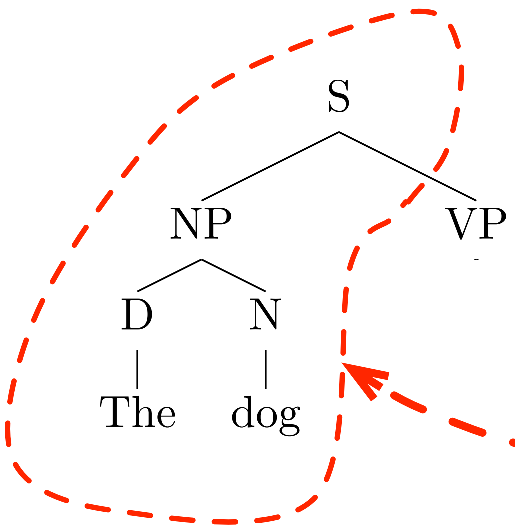


Why context-free?

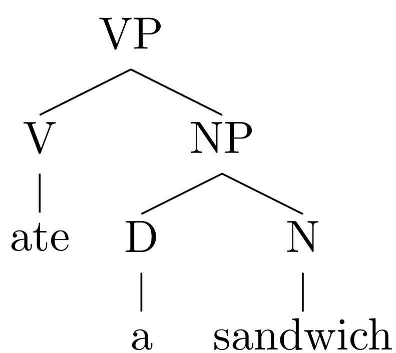


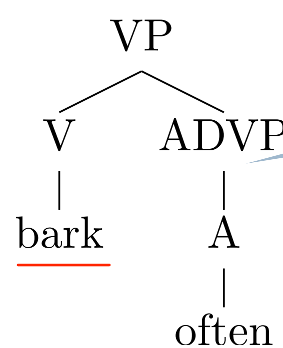
What can be a sub-tree is only affected by what the phrase type is (VP) but not the context

Why context-free?



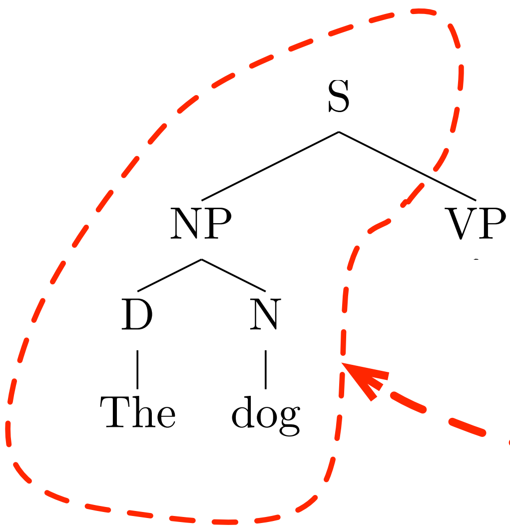
What can be a sub-tree is only affected by what the phrase type is (VP) but not the context



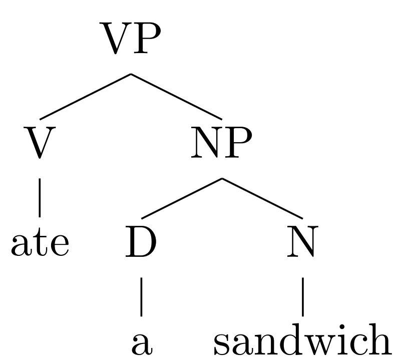


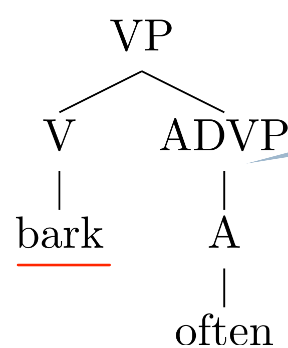
Not grammatical

Why context-free?



What can be a sub-tree is only affected by what the phrase type is (VP) but not the context





Not grammatical

Matters if we want to generate language (e.g., language modeling) but is this relevant to parsing?

Key problems

- Recognition problem: does the sentence belong to the language defined by CFG?
  - That is: is there a derivation which yields the sentence?
- Parsing problem: what is a (most plausible) derivation (tree) corresponding the sentence?

- Parsing problem encompasses the recognition problem

Today

- Basics of Syntax and Context-Free Grammars
- Classes of Syntactic Parsing Algorithms
- Start with the CKY algorithm

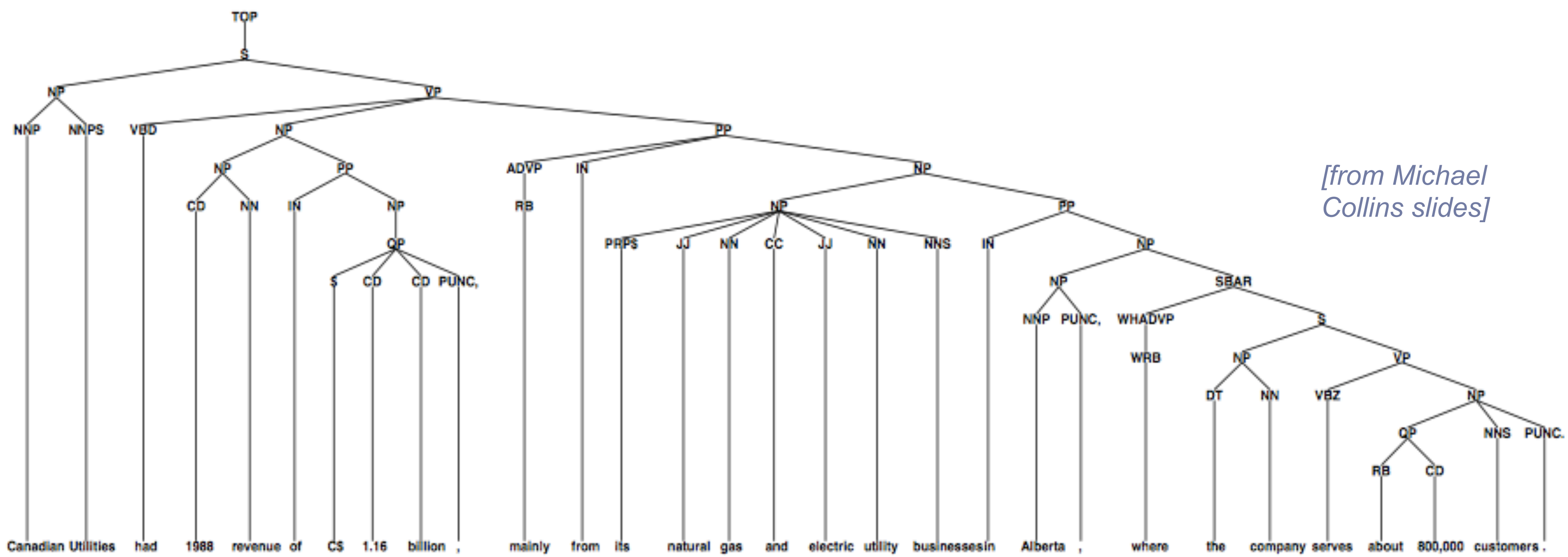
Parsing algorithms

- Goal: compute the structure(s) for an input string given a grammar.
  - (we may want to use the structure to interpret meaning)
  - As usual, ambiguity is a huge problem.

- For correctness: need to find the right structure to get the right meaning.
- For efficiency: searching all possible structures can be very slow
  - want to use parsing for large-scale language tasks

Parsing is hard

- A typical tree from a standard dataset (Penn treebank WSJ)



[from Michael Collins slides]

Canadian Utilities had 1988 revenue of $ 1.16 billion , mainly from its natural gas and electric utility businesses in Alberta , where the company serves about 800,000 customers .

Parser properties

All parsers have two fundamental properties:

- Directionality: the sequence in which the structures are constructed.
  - Top-down: start with root category (S), choose expansions, build down to words.
  - Bottom-up: build subtrees over words, build up to S.
  - Mixed strategies also possible (e.g., left corner parsers)
- Search strategy: the order in which the search space of possible analyses is explored.

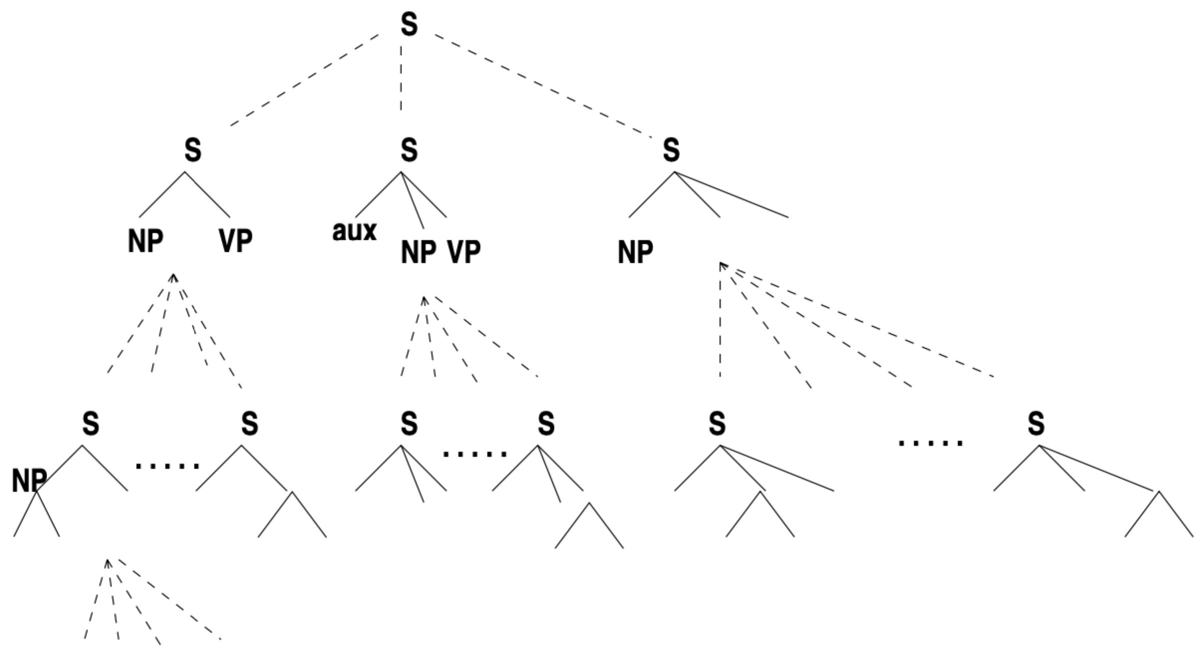
Parser properties

All parsers have two fundamental properties:

- Directionality: the sequence in which the structures are constructed.
  - Top-down: start with root category (S), choose expansions, build down to words.
  - Bottom-up: build subtrees over words, build up to S.
  - Mixed strategies also possible (e.g., left corner parsers)
- Search strategy: the order in which the search space of possible analyses is explored.

Search space for a top-down parser

- Start with S node.
- Choose one of many possible expansions.
- Each of which has children with many possible expansions...
- etc



Search strategies

All parsers have two fundamental properties:

- Depth-first search: explore one branch of the search space at a time, as far as possible. If this branch is a dead-end, parser needs to backtrack.

- Breadth-first search: expand all possible branches in parallel (or simulated parallel). Requires storing many incomplete parses in memory at once.

- Best-first search: score each partial parse and pursue the highest-scoring options first.

We will now consider a bottom-up parser which uses dynamic programming to explore the space

Summary

- Basics of Syntax and Context-Free Grammars
- Classes of Syntactic Parsing Algorithms

Next time:

- CKY algorithm
- Probabilistic parsing with PCFGs
- PCFG parsing beyond treebank grammars