Tutorial on Probabilistic Context-Free Grammars

Raphael Hoffmann

590AI, Winter 2009

Outline

- PCFGs: Inference and Learning
- Parsing English
- Discriminative CFGs
- Grammar Induction

Image Search for "pcfg"

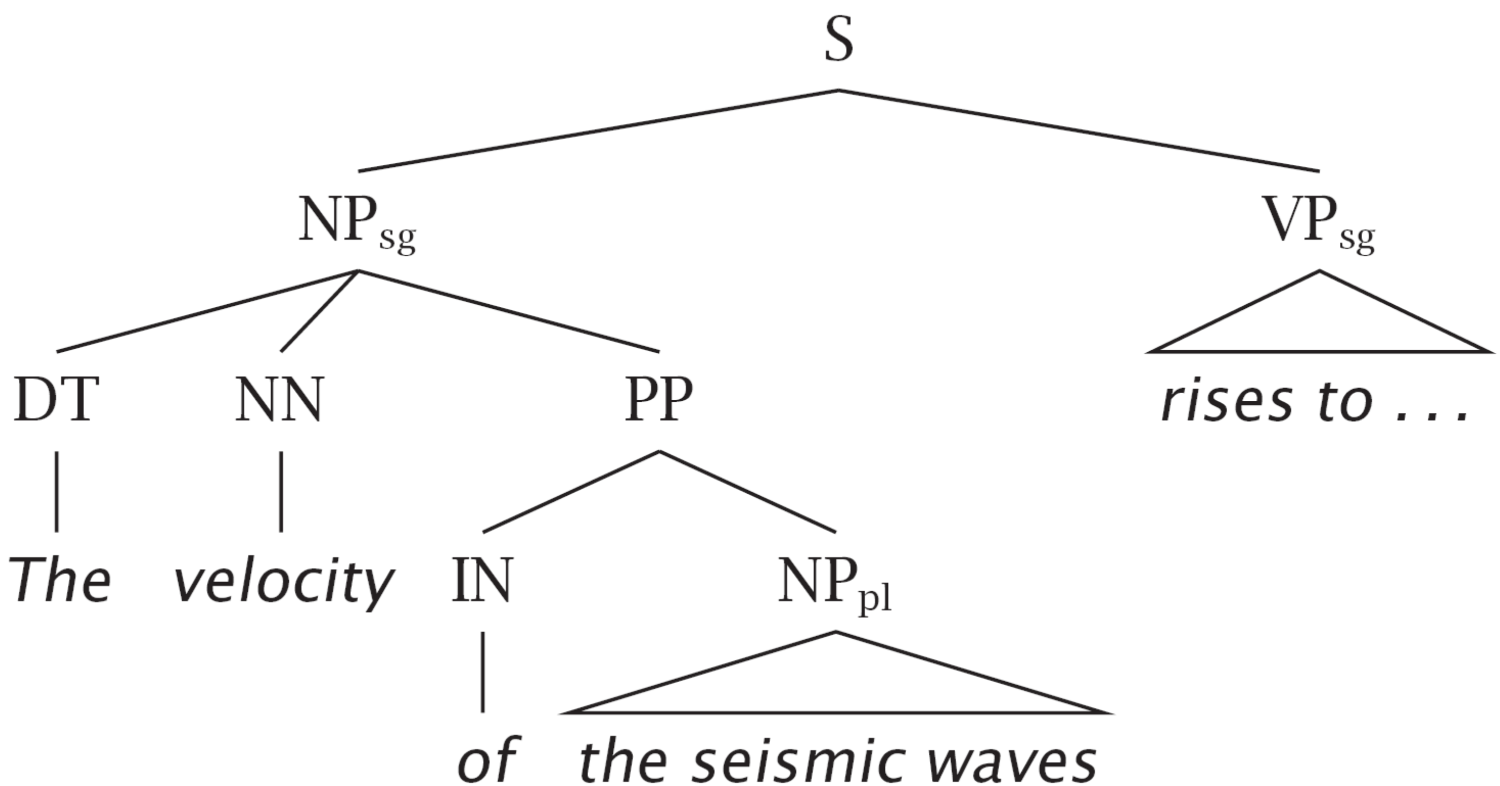
Google

Live Search

Outline

- PCFGs: Inference and Learning
- Parsing English
- Discriminative CFGs
- Grammar Induction

The velocity of the seismic waves rises to …



Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

A CFG consists of

- Terminals $w^1, w^2, \ldots, w^V$
- Nonterminals $N^1, N^2, \ldots, N^n$
- Start symbol N^1
- Rules $N^i \longrightarrow \zeta^j$

where ζ^j is a sequence of terminals and nonterminals

A (generative) PCFG consists of

- Terminals $w^1, w^2, \ldots, w^V$
- Nonterminals $N^1, N^2, \ldots, N^n$
- Start symbol N^1
- Rules $N^i \longrightarrow \zeta^j$

  where ζ^j is a sequence of terminals and nonterminals

- Rule probabilities such that

$$\forall_i \sum_j P(N^i \longrightarrow \zeta^j) = 1$$

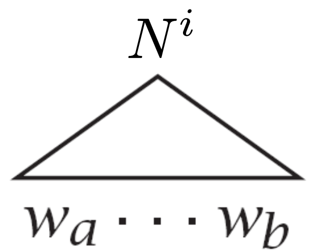
Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

Notation

sentence: sequence of words $w_1 w_2 \ldots w_m$

w_{ab} : the subsequence $w_a \ldots w_b$

N^i_{ab} : nonterminal N^i dominates $w_a \ldots w_b$



$N^i \Longrightarrow^* \zeta$: repeated derivation from N^i gives ζ

Slide based on “Foundations of Statistical Natural Language Processing” by Christopher Manning and Hinrich Schütze

Probability of a Sentence

$$P(w_{1n}) = \sum_t P(w_{1n}, t)$$

where t a parse tree of w_{1n}

Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

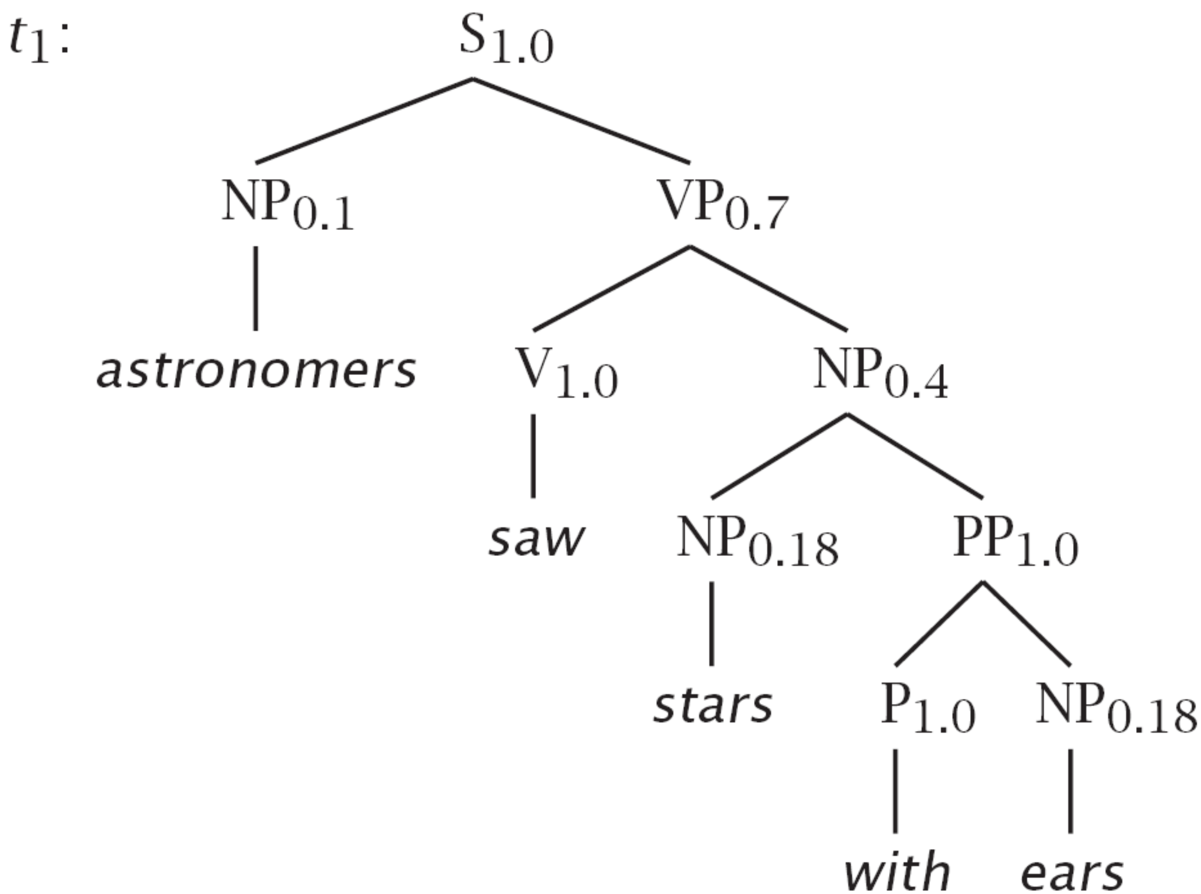
Example

S → NP VP	1.0		NP → NP PP	0.4
PP → P NP	1.0		NP → *astronomers*	0.1
VP → V NP	0.7		NP → *ears*	0.18
VP → VP PP	0.3		NP → *saw*	0.04
P → *with*	1.0		NP → *stars*	0.18
V → *saw*	1.0		NP → *telescopes*	0.1

- Terminals with, saw, astronomers, ears, stars, telescopes
- Nonterminals S, PP, P, NP, VP, V
- Start symbol S

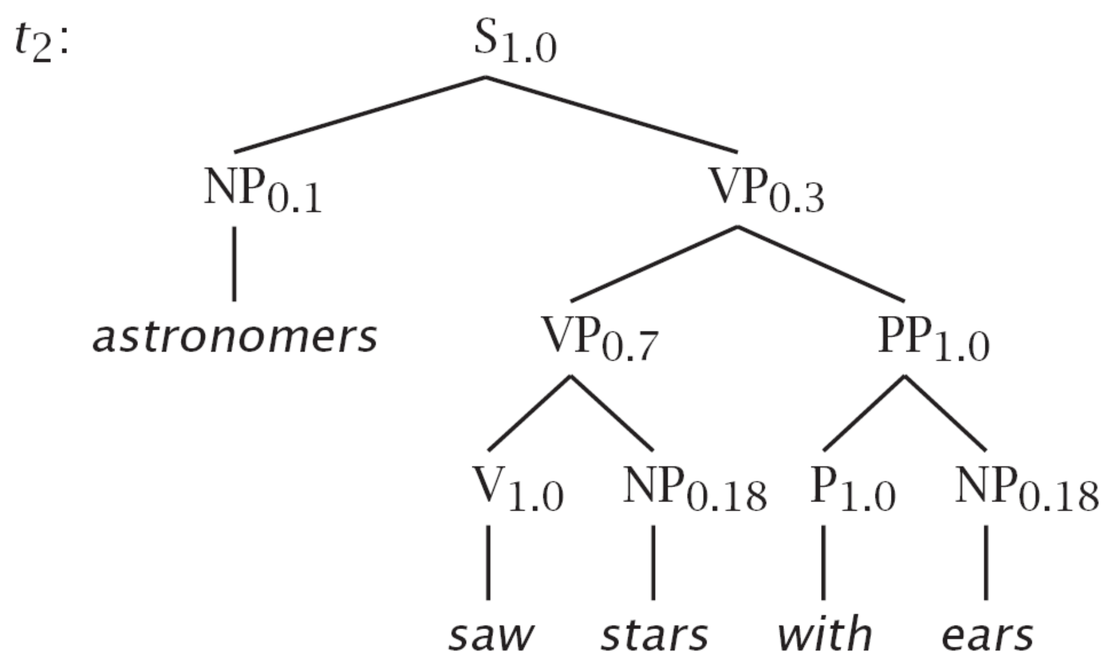
Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

astronomers saw stars with ears



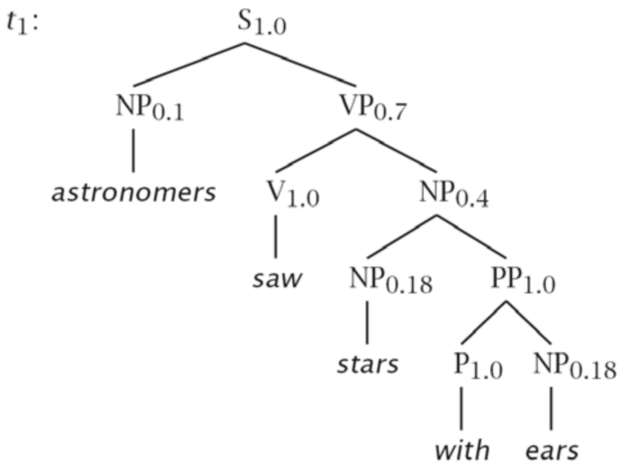
Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

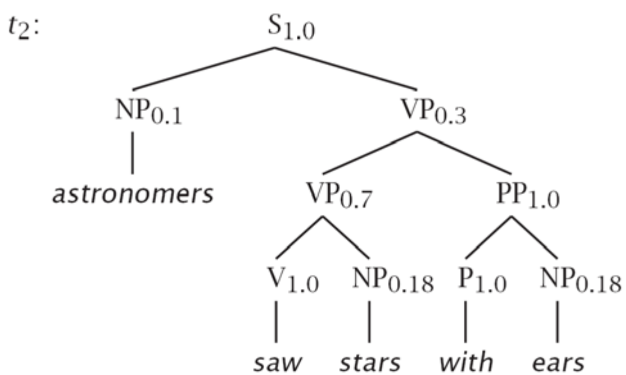
astronomers saw stars with ears



Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

Probabilities





$$
\begin{aligned}
P(t_1) &= 1.0 \times 0.1 \times 0.7 \times 1.0 \times 0.4 \\
&\quad \times 0.18 \times 1.0 \times 1.0 \times 0.18 \\
&= 0.0009072 \\
P(t_2) &= 1.0 \times 0.1 \times 0.3 \times 0.7 \times 1.0 \\
&\quad \times 0.18 \times 1.0 \times 1.0 \times 0.18 \\
&= 0.0006804 \\
P(w_{15}) &= P(t_1) + P(t_2) = 0.0015876
\end{aligned}
$$

Slide based on “Foundations of Statistical Natural Language Processing” by Christopher Manning and Hinrich Schütze

Assumptions of PCFGs

1. Place invariance (like time invariance in HMMs)

$$\forall k \quad P(N^j_{k(k+c)} \longrightarrow \zeta) \quad \text{is the same}$$

Slide based on “Foundations of Statistical Natural Language Processing” by Christopher Manning and Hinrich Schütze

Assumptions of PCFGs

1. Place invariance (like time invariance in HMMs)

$$\forall k \quad P(N^j_{k(k+c)} \longrightarrow \zeta) \quad \text{is the same}$$

2. Context free

$$P(N^j_{kl} \longrightarrow \zeta | \text{ words outside } w_k \ldots w_l) = P(N^j_{kl} \longrightarrow \zeta)$$

Assumptions of PCFGs

1. Place invariance (like time invariance in HMMs)

$$\forall k \quad P(N^j_{k(k+c)} \longrightarrow \zeta) \quad \text{is the same}$$

2. Context free

$$P(N^j_{kl} \longrightarrow \zeta | \text{ words outside } w_k \ldots w_l) = P(N^j_{kl} \longrightarrow \zeta)$$

3. Ancestor free

$$P(N^j_{kl} \longrightarrow \zeta | \text{ ancestor nodes of } N^j_{kl}) = P(N^j_{kl} \longrightarrow \zeta)$$

Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

Some Features of PCFGs

- Partial solution for grammar ambiguity
- Can be learned from positive data alone (but grammar induction difficult)
- Robustness (admit everything with low probability)
- Gives a probabilistic language model
- Predictive power better than that for a HMM

Some Features of PCFGs

- Not lexicalized (probabilities do not factor in lexical co-occurrence)
- PCFG is a worse language model for English than n-gram models
- Certain biases: smaller trees more probable (average WSJ sentence 23 words)

Inconsistent Distributions

- S → *rhubarb* $P = \frac{1}{3}$

  S → S S $P = \frac{2}{3}$
- *rhubarb* $\frac{1}{3}$

  rhubarb rhubarb $\frac{2}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{2}{27}$

  rhubarb rhubarb rhubarb $\left(\frac{2}{3}\right)^2 \times \left(\frac{1}{3}\right)^3 \times 2 = \frac{8}{243}$

  ...
- $P(\mathcal{L}) = \frac{1}{3} + \frac{2}{27} + \frac{8}{243} + \ldots = \frac{1}{2}$
- Improper/inconsistent distribution
- Not a problem if you estimate from parsed treebank: Chi and Geman 1998).

Slide based on “Foundations of Statistical Natural Language Processing” by Christopher Manning and Hinrich Schütze

Questions

Let w_{1m} be a sentence, G a grammar, t a parse tree.

- What is the probability of a sentence?

$$P(w_{1m}|G)$$

Questions

Let w_{1m} be a sentence, G a grammar, t a parse tree.

- What is the probability of a sentence?

$$P(w_{1m}|G)$$

- What is the most likely parse of sentence?

$$\arg\max_t P(t|w_{1m}, G)$$

Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

Questions

Let w_{1m} be a sentence, G a grammar, t a parse tree.

- What is the probability of a sentence?

$$P(w_{1m}|G)$$

- What is the most likely parse of sentence?

$$\arg\max_t P(t|w_{1m}, G)$$

- What rule probs. maximize probs. of sentences?

Find G that maximizes $P(w_{1m}|G)$

Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

Chomsky Normal Form

- Any CFG grammar can be represented in CNF where all rules take the form

$$N^i \longrightarrow N^j N^k$$

$$N^i \longrightarrow w^j$$

Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

HMMs and PCFGs

- HMMs: distribution over strings of certain length

$$\forall n \sum_{w_{1n}} P(w_{1n}) = 1$$

- PCFGs: distribution over strings of language *L*

$$\sum_{w \in L} P(w) = 1$$

HMMs and PCFGs

- HMMs: distribution over strings of certain length

$$\forall n \sum_{w_{1n}} P(w_{1n}) = 1$$

- PCFGs: distribution over strings of language *L*

$$\sum_{w \in L} P(w) = 1$$

- Consider

$$P(\text{John decided to bake a})$$

high probability in HMM, low probability in PCFG

HMMs and PCFGs

- HMMs: distribution over strings of certain length

$$\forall n \sum_{w_{1n}} P(w_{1n}) = 1$$

- PCFGs: distribution over strings of language *L*

$$\sum_{w \in L} P(w) = 1$$

- Consider

$$P(\text{John decided to bake a})$$

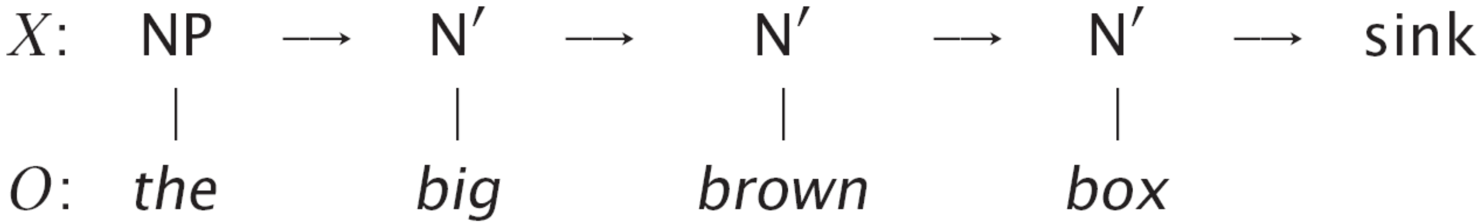
  high probability in HMM, low probability in PCFG

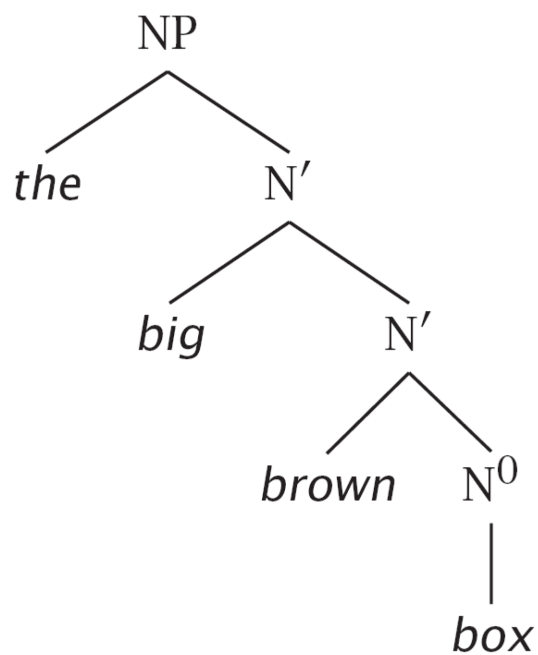
- Probabilistic Regular Grammar

$$N^i \longrightarrow w^j N^k$$
$$N^i \longrightarrow w^j$$

Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

HMMs and PCFGs





Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

Inside and Outside Probabilities

- For HMMs we have

$$
\begin{aligned}
\text{Forwards} \quad \alpha_i(t) &= P(w_{1(t-1)}, X_t = i) \\
\text{Backwards} \quad \beta_i(t) &= P(w_{tT} | X_t = i)
\end{aligned}
$$

Inside and Outside Probabilities

- For HMMs we have

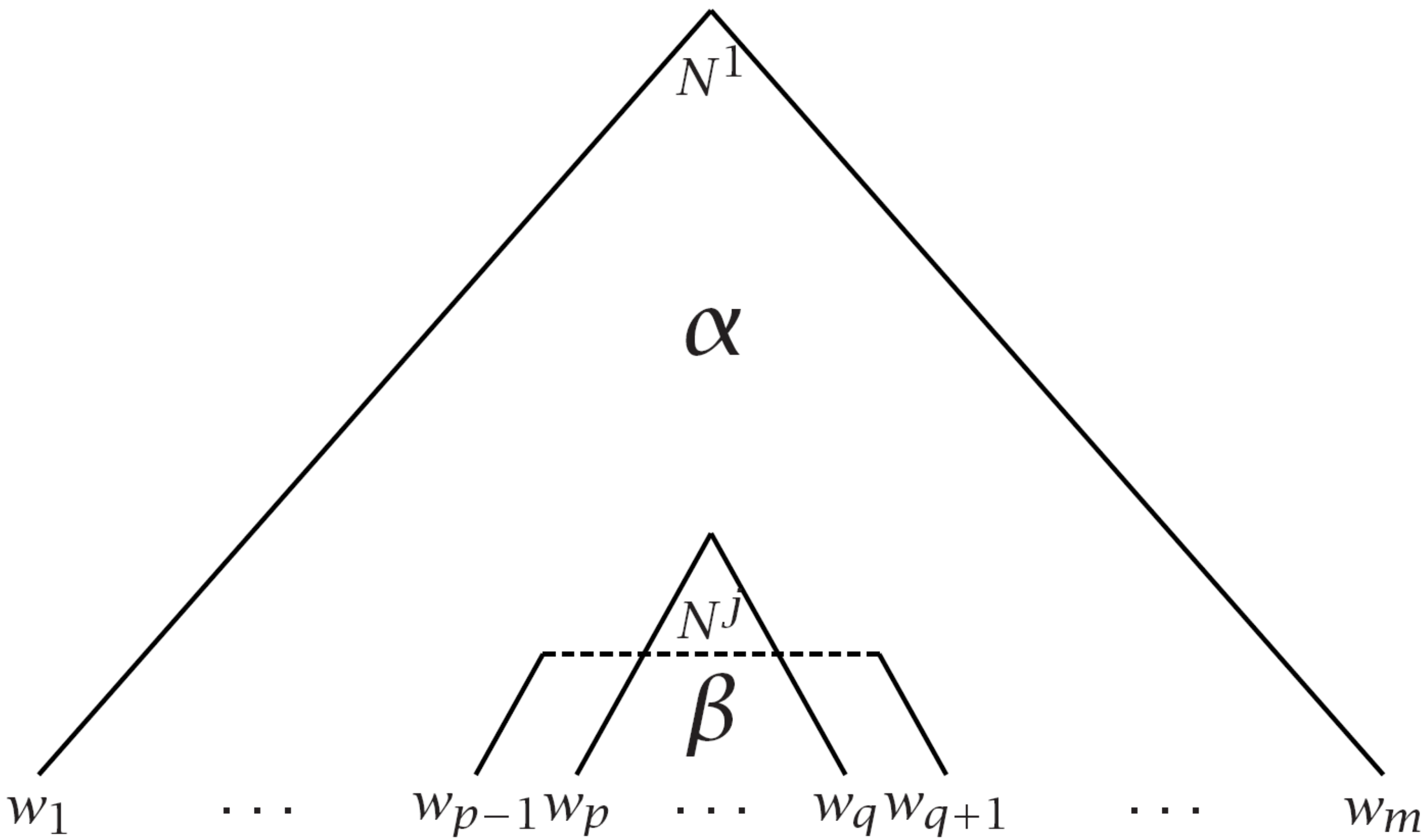
$$
\begin{aligned}
\text{Forwards} \quad \alpha_i(t) &= P(w_{1(t-1)}, X_t = i) \\
\text{Backwards} \quad \beta_i(t) &= P(w_{tT} | X_t = i)
\end{aligned}
$$

- For PCFGs we have

$$
\begin{aligned}
\text{Outside} \quad \alpha_j(p,q) &= P(w_{1(p-1)}, N^j_{pq}, w_{(q+1)m} | G) \\
\text{Inside} \quad \beta_j(p,q) &= P(w_{pq} | N^j_{pq}, G)
\end{aligned}
$$

Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

Inside and Outside Probabilities



Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

Probability of a sentence

Outside $$\alpha_j(p,q) \quad = \quad P(w_{1(p-1)}, N^j_{pq}, w_{(q+1)m}|G)$$

Inside $$\beta_j(p,q) \quad = \quad P(w_{pq}|N^j_{pq}, G)$$

$$P(w_{1m}|G) \quad = \quad \beta_1(1,m)$$

$$P(w_{1m}|G) \quad = \quad \sum_j \alpha_j(k,k) P(N^j \longrightarrow w_k)$$

Inside Probabilities

$$\beta_j(p,q) = P(w_{pq}|N^j_{pq}, G)$$

- Base case

$$\begin{aligned}\beta_j(k,k) &= P(w_{kk}|N^j_{kk}, G) \\ &= P(N^j \longrightarrow w_k|G)\end{aligned}$$

Inside Probabilities

$$\beta_j(p,q) \quad = \quad P(w_{pq}|N^j_{pq}, G)$$

- Base case

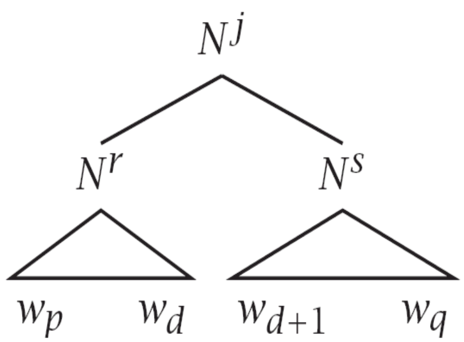
$$\begin{aligned} \beta_j(k,k) &= P(w_{kk}|N^j_{kk}, G) \\ &= P(N^j \longrightarrow w_k|G) \end{aligned}$$

- Induction

Want to find $\beta_j(p,q)$ for $p < q$

Since we assume Chomsky Normal Form,
the first rule must be of the form $N^j \longrightarrow N^r N^s$

So we can divide the sentence in two in
various places and sum the result

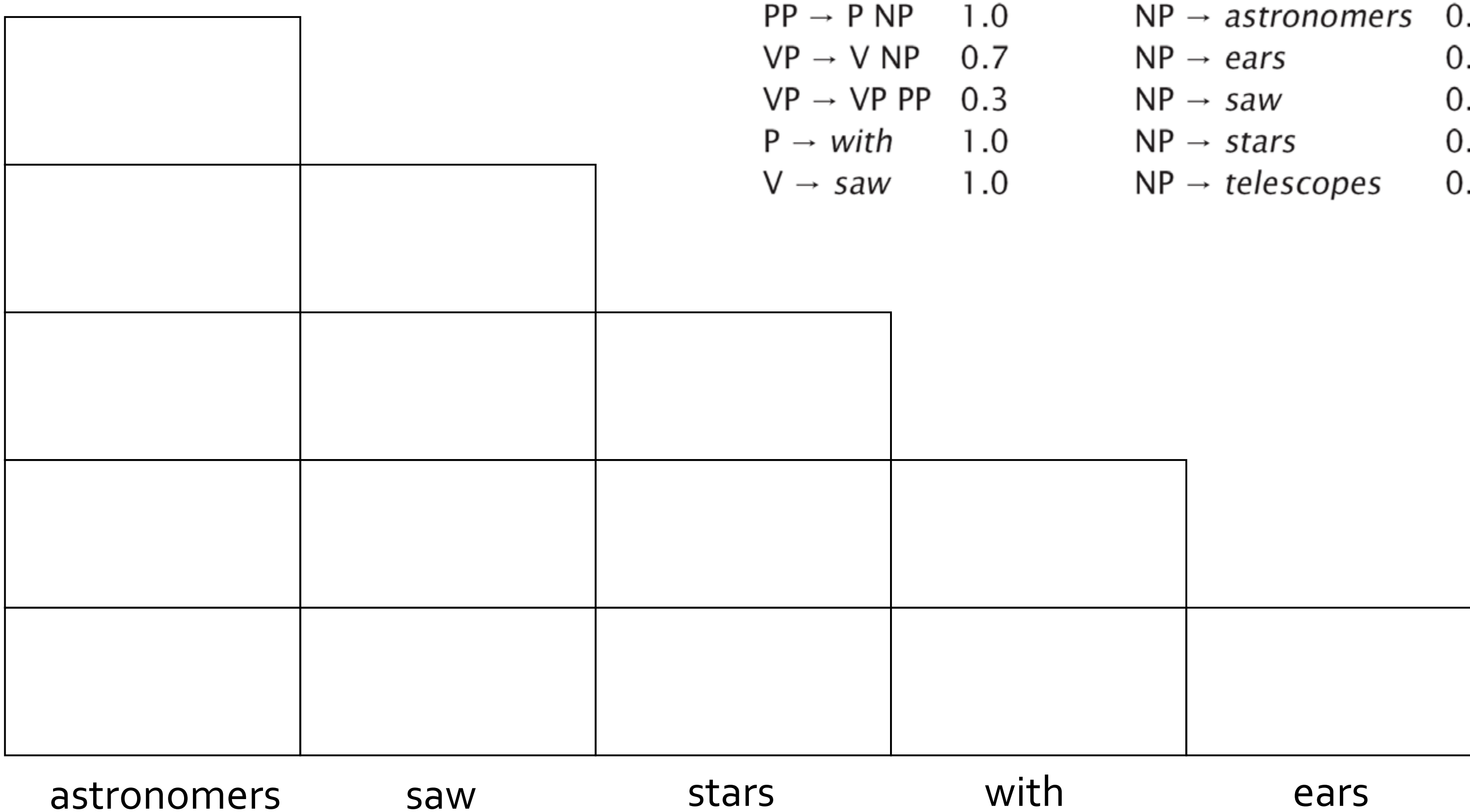


$$\beta_j(p,q) = \sum_{r,s} \sum_{d=p}^{q-1} P(N^j \longrightarrow N^r N^s) \beta_r(p,d) \beta_s(d+1,q)$$

Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

CYK Algorithm

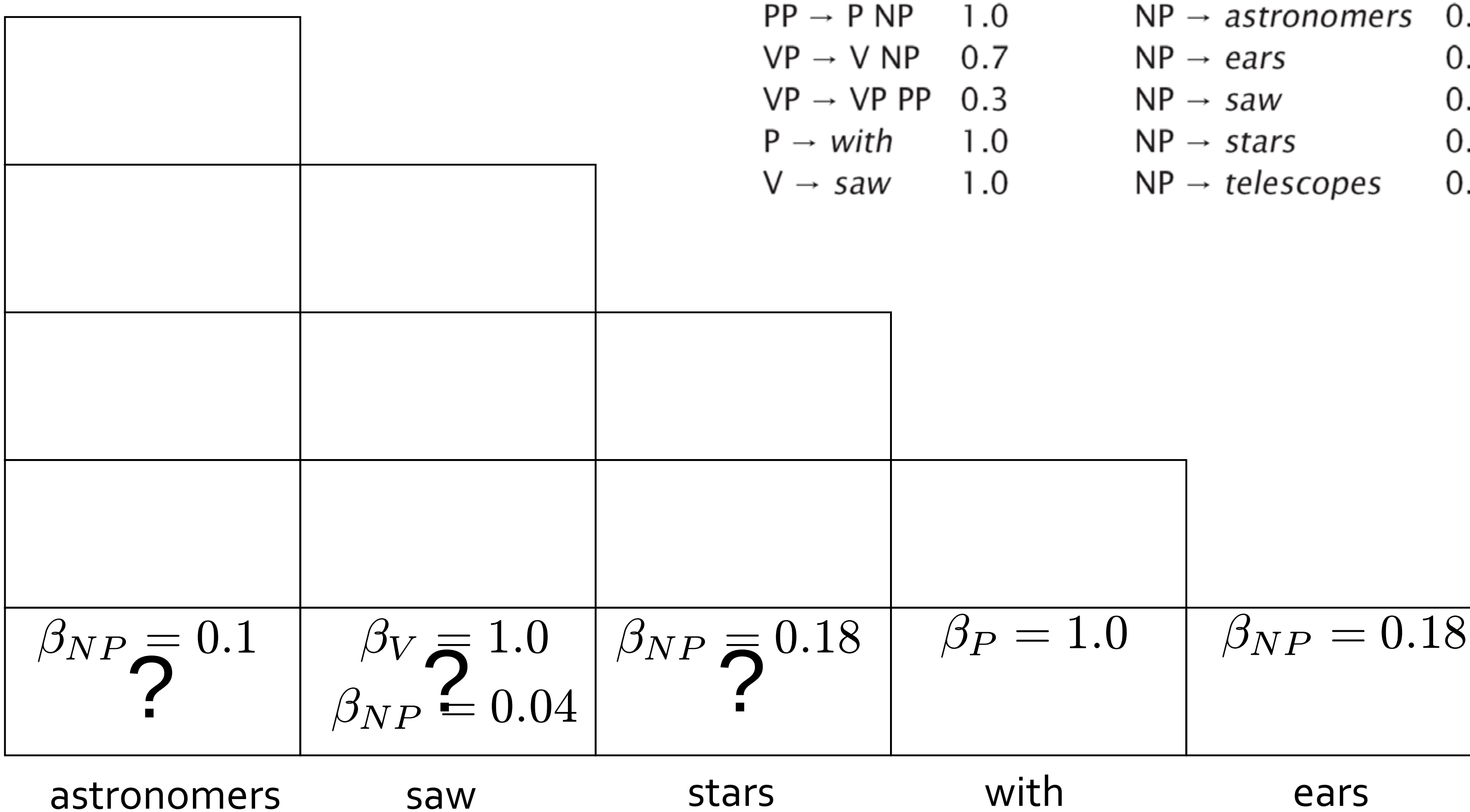
S → NP VP	1.0		NP → NP PP	0.4	
PP → P NP	1.0		NP → *astronomers*	0.1	
VP → V NP	0.7		NP → *ears*	0.18	
VP → VP PP	0.3		NP → *saw*	0.04	
P → *with*	1.0		NP → *stars*	0.18	
V → *saw*	1.0		NP → *telescopes*	0.1	



Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

CYK Algorithm

S → NP VP	1.0	NP → NP PP	0.4
PP → P NP	1.0	NP → *astronomers*	0.1
VP → V NP	0.7	NP → *ears*	0.18
VP → VP PP	0.3	NP → *saw*	0.04
P → *with*	1.0	NP → *stars*	0.18
V → *saw*	1.0	NP → *telescopes*	0.1



CYK Algorithm

S → NP VP	1.0	NP → NP PP	0.4
PP → P NP	1.0	NP → *astronomers*	0.1
VP → V NP	0.7	NP → *ears*	0.18
VP → VP PP	0.3	NP → *saw*	0.04
P → *with*	1.0	NP → *stars*	0.18
V → *saw*	1.0	NP → *telescopes*	0.1

?	$\beta_{VP} = 0.126$?	?	$\beta_{PP} = 0.18$?	
$\beta_{NP} = 0.1$	$\beta_{V} = 1.0$ $\beta_{NP} = 0.04$	$\beta_{NP} = 0.18$	$\beta_{P} = 1.0$	$\beta_{NP} = 0.18$
astronomers	saw	stars	with	ears

Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

CYK Algorithm

S → NP VP	1.0		NP → NP PP	0.4
PP → P NP	1.0		NP → *astronomers*	0.1
VP → V NP	0.7		NP → *ears*	0.18
VP → VP PP	0.3		NP → *saw*	0.04
P → *with*	1.0		NP → *stars*	0.18
V → *saw*	1.0		NP → *telescopes*	0.1

$\beta_S = 0.015876$				
	$\beta_{VP} = 0.015876$			
$\beta_S = 0.0126$?		$\beta_{NP} = 0.01296$		
	$\beta_{VP} = 0.126$		$\beta_{PP} = 0.18$	
$\beta_{NP} = 0.1$	$\beta_V = 1.0$ $\beta_{NP} = 0.04$		$\beta_P = 1.0$	$\beta_{NP} = 0.18$
astronomers	saw	stars	with	ears

Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

CYK Algorithm

Worst case: $O(m^3r)$

m = length of sentence

r = number of rules in grammar

n = number of non-terminals

If we consider all possible CNF rules: $O(m^3n^3)$

$\beta_S = 0.015876$				
	$\beta_{VP} = 0.015876$			
$\beta_S = 0.0126$		$\beta_{NP} = 0.01296$		
	$\beta_{VP} = 0.126$		$\beta_{PP} = 0.18$	
$\beta_{NP} = 0.1$	$\beta_V = 1.0$ $\beta_{NP} = 0.04$	$\beta_{NP} = 0.18$	$\beta_P = 1.0$	$\beta_{NP} = 0.18$
astronomers	saw	stars	with	ears

Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

Outside Probabilities

- Compute top-down (after inside probabilities)
- Base case

$$\alpha_1(1, m) = 1$$

$$\alpha_j(1, m) = 0, \text{ for } j \neq 1$$

- Induction

$$\alpha_j(p, q) = \left(\sum_{f,g} \sum_{e=q+1}^{m} \alpha_f(p, e) P(N^f \longrightarrow N^j N^g) \beta_g(q+1, e) \right) + \left(\sum_{f,g} \sum_{e=1}^{p-1} \alpha_f(e, q) P(N^f \longrightarrow N^g N^j) \beta_g(e, p-1) \right)$$

Slide based on “Foundations of Statistical Natural Language Processing” by Christopher Manning and Hinrich Schütze

Probability of a node existing

- As with a HMM, we can form a product of the inside and outside probabilities.

$$\alpha_j(p,q)\beta_j(p,q) = P(w_{1m}, N^j_{pq}|G)$$

Probability of a node existing

- As with a HMM, we can form a product of the inside and outside probabilities.

$$\alpha_j(p,q)\beta_j(p,q) = P(w_{1m}, N^j_{pq}|G)$$

- Therefore,

$$P(w_{1m}, N_{pq}|G) = \sum_j \alpha_j(p,q)\beta_j(p,q)$$

Probability of a node existing

- As with a HMM, we can form a product of the inside and outside probabilities.

$$\alpha_j(p,q)\beta_j(p,q) = P(w_{1m}, N^j_{pq}|G)$$

- Therefore,

$$P(w_{1m}, N_{pq}|G) = \sum_j \alpha_j(p,q)\beta_j(p,q)$$

- Just in the cases of the root node and the preterminals, we know there will be some such constituent.

Training

- If have data → count

$$\hat{P}(N^j \longrightarrow \zeta) = \frac{C(N^j \longrightarrow \zeta)}{\sum_{\gamma} C(N^j \longrightarrow \gamma)}$$

Training

- If have data → count

$$\hat{P}(N^j \longrightarrow \zeta) = \frac{C(N^j \longrightarrow \zeta)}{\sum_\gamma C(N^j \longrightarrow \gamma)}$$

- else use EM (Inside-Outside-Algorithm)

repeat

compute α_j 's and β_j 's

compute $\hat{P}$'s

$$\hat{P}(N^j \longrightarrow N^r N^s) = \ldots$$

$$\hat{P}(N^j \longrightarrow w^k) = \ldots$$

end

two really long formulas with α_j 's and β_j 's

Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

EM Problems

- Slow: $O(m^3n^3)$ for each sentence and each iteration
- Local maxima (Charniak: 300 trials led to 300 different max.)
- In practice, need >3 times more non-terminals than are theoretically needed
- No guarantee that learned non-terminals correspond to NP, VP, …

Bracketing helps

Pereira/Schabes '92:

- Train on sentences:
  37% of predicted brackets correct
- Train on sentences + brackets:
  90% of predicted brackets correct

Grammar Induction

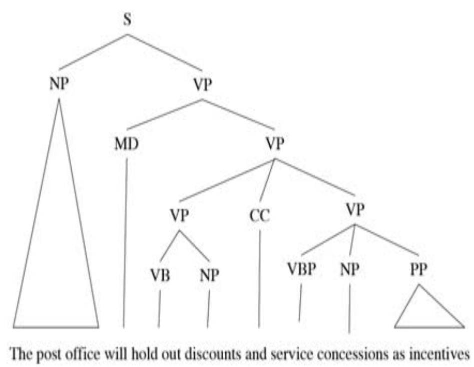
- Rules typically selected by linguist
- Automatic induction is very difficult for context-free languages
- It is easy to find *some* form of structure, but little resemblance to that of linguistics/NLP

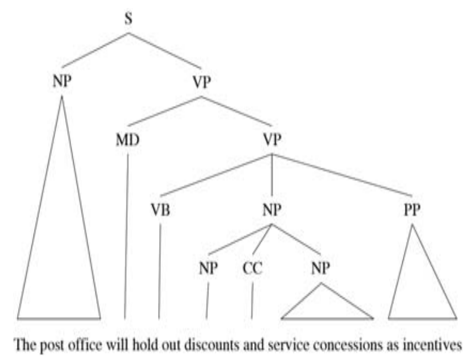
Outline

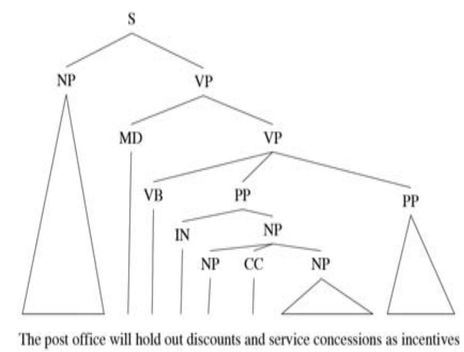
- PCFGs: Inference and Learning
- Parsing English
- Discriminative CFGs
- Grammar Induction

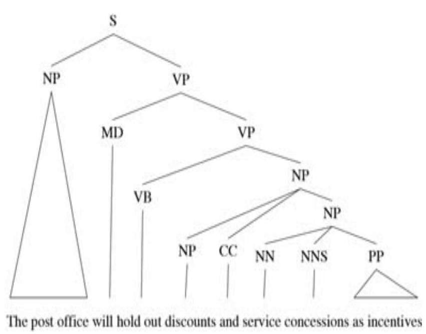
Parsing for Disambiguation

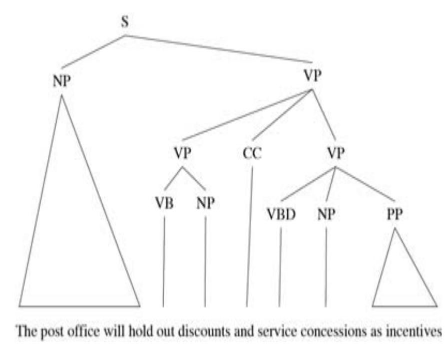
The post office will hold out discounts and service concessions as incentives.











Parsing for Disambiguation

- There are typically many syntactically possible parses
- Want to find the most likely parses

Slide based on “Foundations of Statistical Natural Language Processing” by Christopher Manning and Hinrich Schütze

Treebanks

- If grammar induction does not work, why not count expansions in many parse trees?
- Penn Treebank

```
( (S
    (NP (NBAR (ADJP (ADJ "Battle-tested/JJ")
                    (ADJ "industrial/JJ"))
              (NPL "managers/NNS")))
    (? (ADV "here/RB"))
    (? (ADV "always/RB"))
    (AUX (TNS *))
    (VP (VPRES "buck/VBP")))
    (? (PP (PREP "up/RP")
           (NP (NBAR (ADJ "nervous/JJ")
                     (NPL "newcomers/NNS")))))
    (? (PP (PREP "with/IN")
           (NP (DART "the/DT")
               (NBAR (N "tale/NN"))
                     (PP of/PREP
                         (NP (DART "the/DT")
                             (NBAR (ADJP
                                    (ADJ "first/JJ"))))))))
```

Slide based on "Foundations of Statistical Natural Language Processing" by Christopher Manning and Hinrich Schütze

PCFG weaknesses

- No Context
  - (immediate prior context, speaker, …)
- No Lexicalization
  - “VP NP NP” more likely if verb is “hand” or “tell”
  - fail to capture lexical dependencies (n-grams do)
- No Structural Context
  - How NP expands depends on position

Slide based on “Foundations of Statistical Natural Language Processing” by Christopher Manning and Hinrich Schütze

PCFG weaknesses

Expansion	% as Subj	% as Obj
NP $\longrightarrow$ PRP	13.7%	2.1%
NP $\longrightarrow$ NNP	3.5%	0.9%
NP $\longrightarrow$ DT NN	5.6%	4.6%
NP $\longrightarrow$ NN	1.4%	2.8%
NP $\longrightarrow$ NP SBAR	0.5%	2.6%
NP $\longrightarrow$ NP PP	5.6%	14.1%

Expansion	% as 1st Obj	% as 2nd Obj
NP $\longrightarrow$ NNS	7.5%	0.2%
NP $\longrightarrow$ PRP	13.4%	0.9%
NP $\longrightarrow$ NP PP	12.2%	14.4%
NP $\longrightarrow$ DT NN	10.4%	13.3%
NP $\longrightarrow$ NNP	4.5%	5.9%
NP $\longrightarrow$ NN	3.9%	9.2%
NP $\longrightarrow$ JJ NN	1.1%	10.4%
NP $\longrightarrow$ NP SBAR	0.3%	5.1%

Slide based on “Foundations of Statistical Natural Language Processing” by Christopher Manning and Hinrich Schütze

Other Approaches

- Challenge: use lexical and structural context, without too many parameters, sparse data
- Other Grammars
  - Probabilistic Left-Corner Grammars
  - Phrase Structure Grammars
  - Dependency Grammars
  - Probabilistic Tree Substitution Grammars
  - History-based Grammars

Outline

- PCFGs: Inference and Learning
- Parsing English
- Discriminative CFGs
- Grammar Induction

Generative vs Discriminative

- An HMM (or PCFG) is a *generative* model

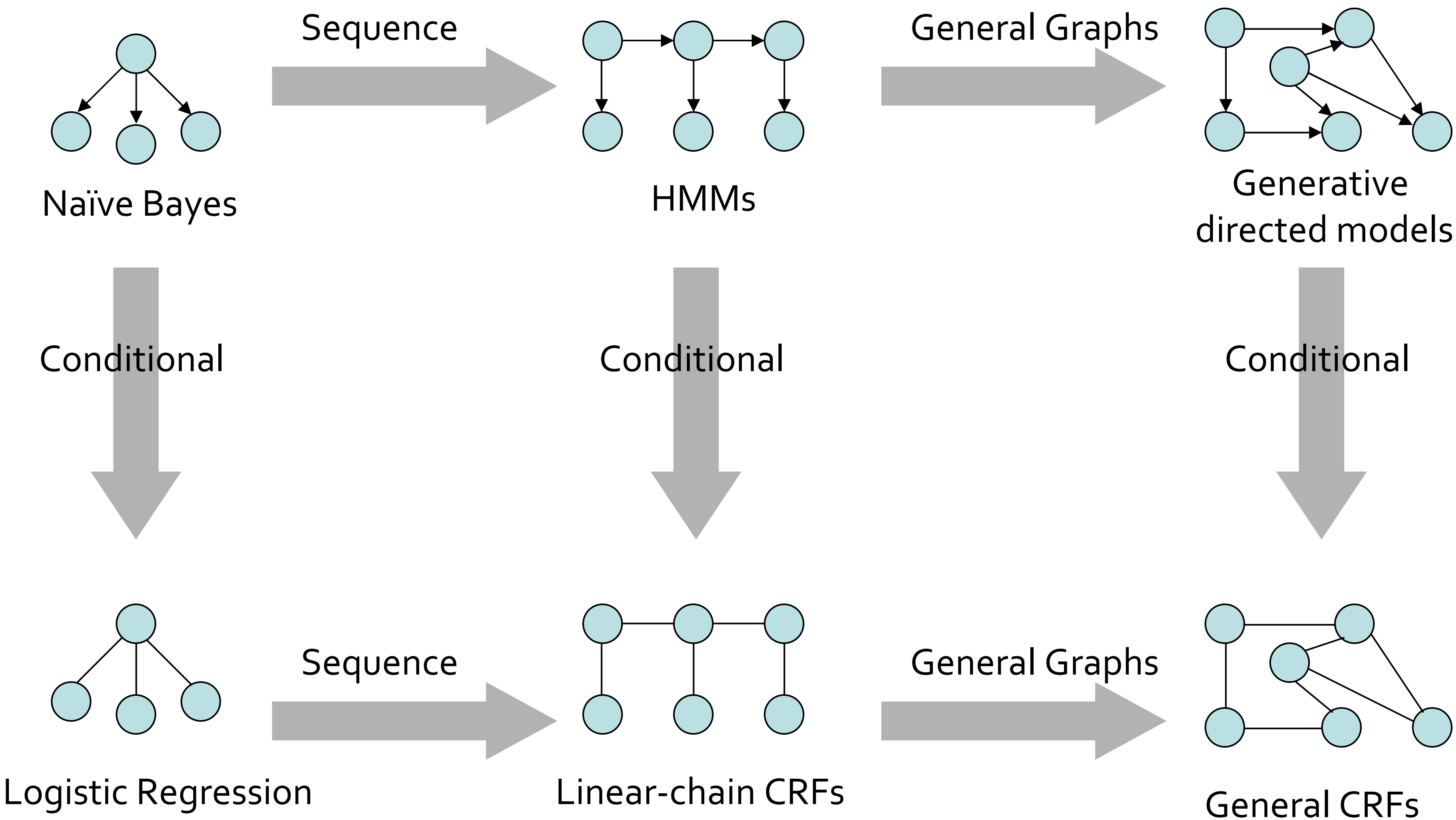
$$P(y, w)$$

- Often sufficient is a *discriminative* model

$$P(y|w)$$

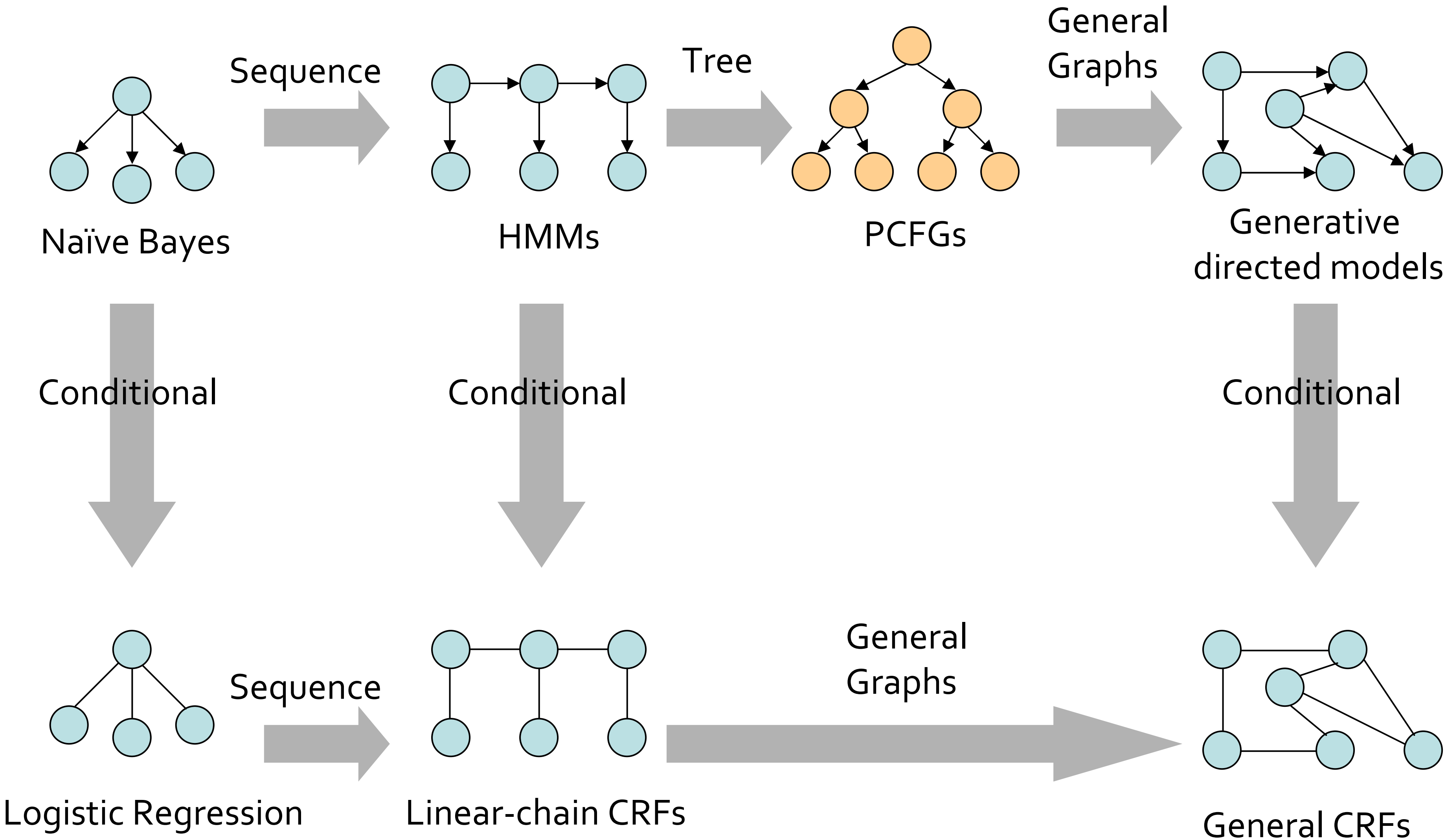
- Easier, because does not contain $P(w)$
- Cannot model dependent features in HMM, so one only picks one feature: word's identity

Generative and Discriminative Models

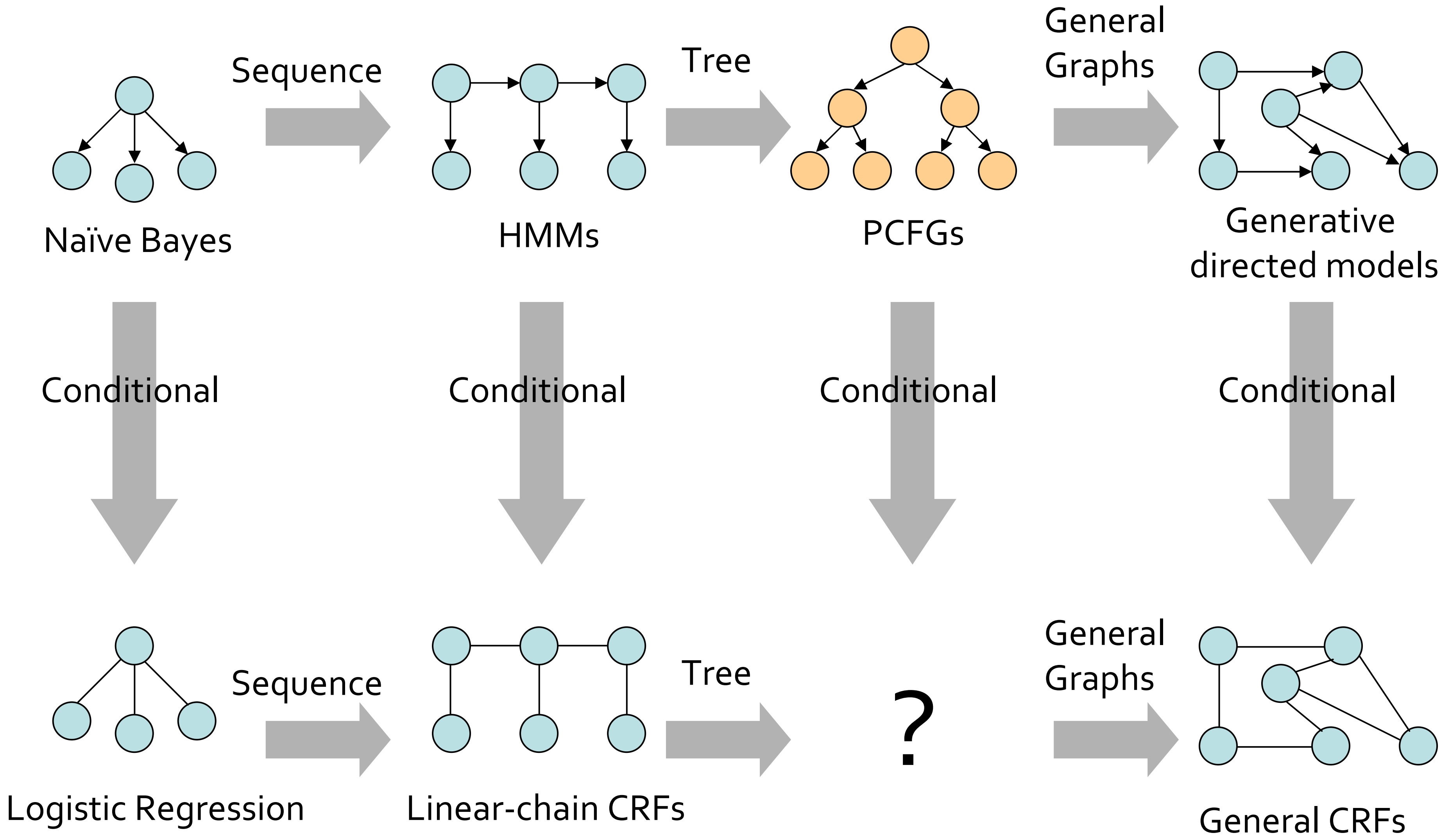


Slide based on “An introduction to Conditional Random Fields for Relational Learning” by Charles Sutton and Andrew McCallum

Generative and Discriminative Models



Generative and Discriminative Models



Discriminative Context-Free Grammars

- Terminals $w^1, w^2, \ldots, w^V$
- Nonterminals $N^1, N^2, \ldots, N^n$
- Start symbol N^1
- Rules $N^i \longrightarrow \zeta^j$

  where ζ^j is a sequence of terminals and nonterminals

- Rule scores

$$S(N^i \longrightarrow \zeta^j, p, q) = \sum_{k=1}^{F} \lambda_k (N^i \longrightarrow \zeta^j) f_k(w_1 w_2 \ldots w_m, p, q, N^i \longrightarrow \zeta^j)$$

Slide based on "Learning to Extract Information from Semi-Structured Text using a Discriminative Context Free Grammar" by Paul Viola and Mukund Narasimhan

Features

$$S(N^i \longrightarrow \zeta^j, p, q) = \sum_{k=1}^{F} \lambda_k(N^i \longrightarrow \zeta^j) f_k(w_1 w_2 \ldots w_m, p, q, N^i \longrightarrow \zeta^j)$$

- Features can depend on all tokens + span
- Consider feature "AllOnTheSameLine"

Mavis Wood
Products

Mavis Wood Products

[compare to linear CRF $f_k(s_t, t_{t-1}, w_1 w_2 \ldots w_m, t)$]

Slide based on "Learning to Extract Information from Semi-Structured Text using a Discriminative Context Free Grammar" by Paul Viola and Mukund Narasimhan

Features

$$S(N^i \longrightarrow \zeta^j, p, q) = \sum_{k=1}^{F} \lambda_k(N^i \longrightarrow \zeta^j) f_k(w_1 w_2 \ldots w_m, p, q, N^i \longrightarrow \zeta^j)$$

- No independence between features necessary
- Can create features based on words, dictionaries, digits, capitalization, …
- Can still do efficient Viterbi inference in $O(m^3 r)$

Example

FirstName	LastName	AddrLine	AddrLine	AddrLine	City	State	Zip
John	Doe	100	Main	Street	Seattle	WA	98195

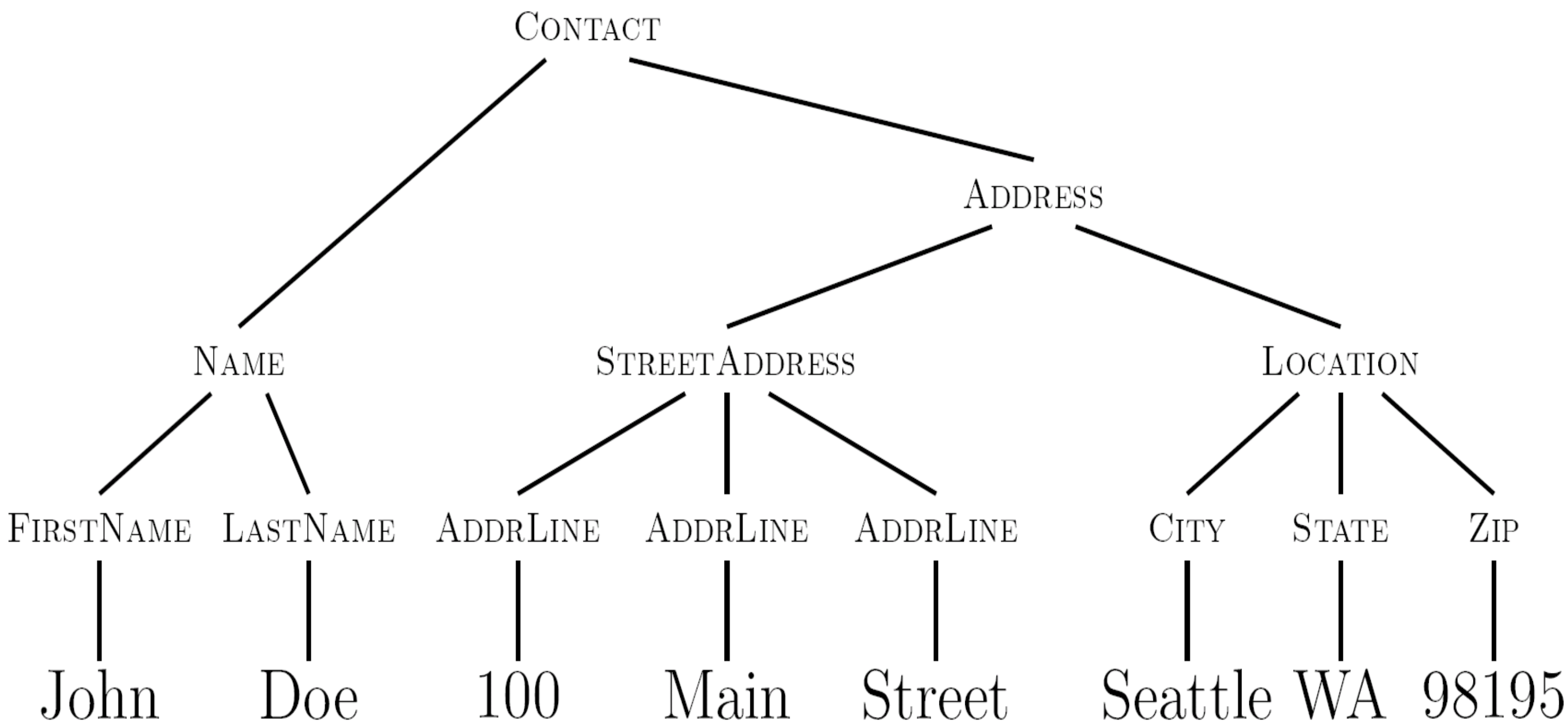
```
Fred Jones
10 Main St.
Cambridge, MA 02146
(425) 994-8021
```

```
Boston College
10 Main St.
Cambridge MA 02146
(425) 994-8021
```

$$\text{BizContact} \longrightarrow \text{BizName Address BizPhone}$$
$$\text{PersonalContact} \longrightarrow \text{BizName Address HomePhone}$$

Slide based on “Learning to Extract Information from Semi-Structured Text using a Discriminative Context Free Grammar” by Paul Viola and Mukund Narasimhan

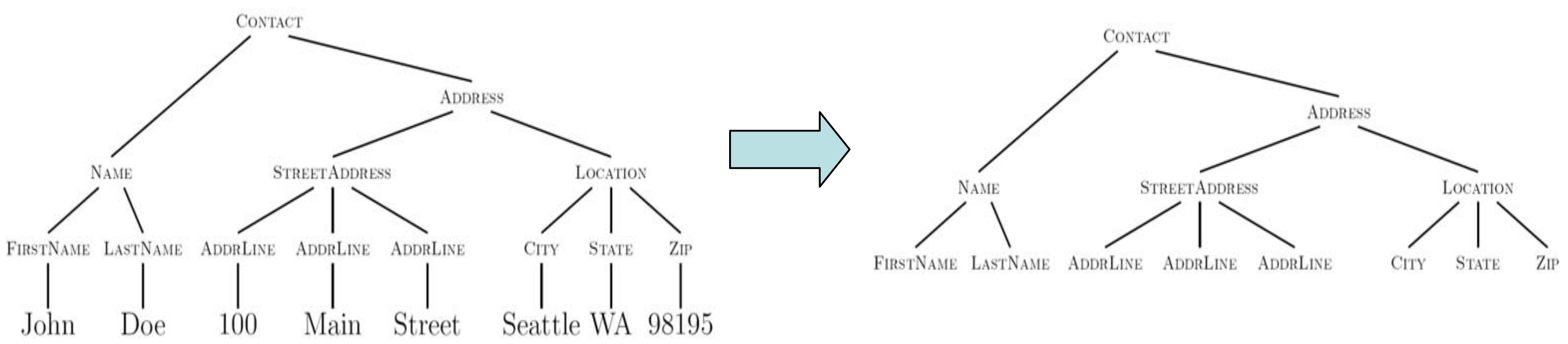
Example



Slide based on “Learning to Extract Information from Semi-Structured Text using a Discriminative Context Free Grammar” by Paul Viola and Mukund Narasimhan

Training

- Train feature weight vector for each rule $\lambda_j(R)$
- Have labels, but not parse trees;
  efficiently create trees by ignoring leaves



Slide based on "Learning to Extract Information from Semi-Structured Text using a Discriminative Context Free Grammar" by Paul Viola and Mukund Narasimhan

Collins' Averaged Perceptron

for $r \leftarrow 1 \ldots numRounds$ **do**
  for $i \leftarrow 1 \ldots m$ **do**
    $T \leftarrow$ optimal parse of w^i with current parameters
    if $T \neq T^i$ **then**
      for each rule R used in T but not in T^i **do**
        if feature f_j is active in w^i **then**
          $\lambda_j(R) \leftarrow \lambda_j(R) - 1;$
        endif
      endfor
      for each rule R used in T^j but not in T **do**
        if feature f_j is active in w^i **then**
          $\lambda_j(R) \leftarrow \lambda_j(R) + 1;$
        endif
      endfor
    endif
  endfor
endfor

Slide based on "Learning to Extract Information from Semi-Structured Text using a Discriminative Context Free Grammar" by Paul Viola and Mukund Narasimhan

Results

	Linear CRF	Discriminative CFG	Improvement
Word Error Rate	11.57%	6.29%	45.63%
Record Error Rate	54.71%	27.13%	50.41%

Slide based on “Learning to Extract Information from Semi-Structured Text using a Discriminative Context Free Grammar” by Paul Viola and Mukund Narasimhan

Outline

- PCFGs: Inference and Learning
- Parsing English
- Discriminative CFGs
- Grammar Induction

Gold’s Theorem ‘67

“Any formal language which has hierarchical structure capable of infinite recursion is unlearnable from positive evidence alone.”

Slide based on Wikipedia

Empirical Problems

- Even finite search spaces can be too big
- Noise
- Insufficient data
- Many local optima

Slide based on "Unsupervised grammar induction with Minimum Description Length" by Roni Katzir

Common Approach

- Minimize total description length
- Simulated Annealing

Initial state

$$
\begin{aligned}
D &:= \texttt{pabikugolatuda...} \\
T &:= T_0 \\
G &:= \begin{cases} \gamma \rightarrow p\,\gamma \\ \gamma \rightarrow a\,\gamma \\ \gamma \rightarrow b\,\gamma \\ \vdots \end{cases}
\end{aligned}
$$

Repeat:

$$
\begin{aligned}
G' &:= \mathit{random\_neighbor}(G) \\
\Delta &:= \mathit{Energy}(G', D) - \mathit{Energy}(G, D) \\
p &:= \begin{cases} 1 & \Delta \leq 0 \\ e^{-\frac{\Delta}{T}} & \Delta > 0 \end{cases} \\
G &:= G' \text{ with probability } p \\
T &:= \alpha T
\end{aligned}
$$

Slide based on “Unsupervised grammar induction with Minimum Description Length” by Roni Katzir

random_neighbor(G)

- Insert:

$$G := \begin{cases} A \to B\ C \\ B \to D\ E \end{cases} \Rightarrow G := \begin{cases} A \to B\ X\ C \\ B \to D\ E \end{cases}$$

- Delete

$$G := \begin{cases} A \to B\ C \\ B \to D\ E \end{cases} \Rightarrow G := \begin{cases} A \to B \\ B \to D\ E \end{cases}$$

- New Rule

$$G := \begin{cases} A \to B\ C \\ B \to D\ E \end{cases} \Rightarrow G := \begin{cases} A \to B\ C \\ B \to D\ E \\ Y \to \end{cases}$$

- Split

$$G := \begin{cases} A \to B\ C \\ B \to D\ E \end{cases} \Rightarrow G := \begin{cases} A \to D\ E\ C \\ B \to D\ E \end{cases}$$

- Substitute

$$G := \begin{cases} A \to B\ C \\ B \to D\ E \end{cases} \Rightarrow G := \begin{cases} A \to Z\ C \\ B \to D\ E \\ Z \to B \end{cases}$$

Slide based on “Unsupervised grammar induction with Minimum Description Length” by Roni Katzir

Energy

$$Energy(G, D) \quad := \quad |G| + |code(D|G)|$$

Define binary representation for G, code(D|G)

$$G \quad := \quad \begin{cases} A \rightarrow & B \; A \\ A \rightarrow & B \\ B \rightarrow & C \; D \\ \vdots & \\ E \rightarrow & F \; G \end{cases}$$

$$G \quad := \quad \texttt{ABA\#AB\#BCD\#...\#EFG\#\#}$$

Slide based on "Unsupervised grammar induction with Minimum Description Length" by Roni Katzir

Experiment 1

- Word segmentation by 8-month old infants
- Vocabulary: `pabiku, golatu, daropi, tibudo`
- Saffran '96: use speech synthesizer, no word breaks, 2 minutes = 180 words
- Infants can distinguish words from non-words
- Now try grammar induction (60 words)

Experiment 1

Step 37: current temp. $= 14.994450998883487$ Grammar:
$\gamma \rightarrow g\ \gamma; \gamma \rightarrow l\ \gamma; \gamma \rightarrow a\ \gamma; \gamma \rightarrow p\ \gamma; \gamma \rightarrow k\ \gamma; \gamma \rightarrow .\ \gamma; \gamma \rightarrow$
$t\ \gamma; \gamma \rightarrow u\ \gamma; \gamma \rightarrow o\ \gamma; \gamma \rightarrow d\ \gamma; \gamma \rightarrow b\ \gamma; \gamma \rightarrow i\ \gamma; \gamma \rightarrow$
$r\ \gamma; l \rightarrow$
Grammar length: 146 Encoding length: 1442

Step 618: current temp. $= 14.907585393190937$
Grammar:
$k \rightarrow k\ g\ 23; i \rightarrow p; 23 \rightarrow o; \gamma \rightarrow; \gamma \rightarrow g\ \gamma; \gamma \rightarrow p\ a\ \gamma; \gamma \rightarrow$
$a\ \gamma; \gamma \rightarrow t\ \gamma; \gamma \rightarrow u\ \gamma; \gamma \rightarrow o\ \gamma; \gamma \rightarrow k\ u\ \gamma; \gamma \rightarrow o\ l\ \gamma; \gamma \rightarrow$
$o\ p\ \gamma; \gamma \rightarrow d\ \gamma; \gamma \rightarrow b\ \gamma; \gamma \rightarrow i\ \gamma; \gamma \rightarrow r\ \gamma; o \rightarrow; t \rightarrow g$
Grammar length: 200 Encoding length: 1199

Step 5837: current temp. $= 14.149508793558308$
Grammar: $k \rightarrow p; \gamma \rightarrow t\ i\ b\ u\ d\ o\ \gamma; \gamma \rightarrow \gamma; \gamma \rightarrow$
$p\ a\ b\ i\ k\ u\ \gamma; \gamma \rightarrow i\ a\ \gamma\ \gamma\ g; \gamma \rightarrow d\ a\ r\ o\ p\ i\ \gamma; \gamma \rightarrow$
$g\ o\ l\ a\ t\ u\ \gamma; b \rightarrow u\ d\ o\ o\ \gamma; b \rightarrow k\ u\ l; d \rightarrow; l \rightarrow; t \rightarrow \gamma$
Grammar length: 179 Encoding length: 183

Slide based on “Unsupervised grammar induction with Minimum Description Length” by Roni Katzir

Experiment 2

$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Nm \mid D\ N \\
Nm &\rightarrow max \mid sam \mid kim \mid bill \mid mary \\
D &\rightarrow the \mid a \\
N &\rightarrow man \mid dog \mid cat \mid turtle \\
VP &\rightarrow Vin \mid Vtr\ NP \mid Vtl\ NP\ CP \mid Vsy\ CP \\
Vin &\rightarrow walks \mid runs \\
Vtr &\rightarrow kills \mid hits \\
\ldots &
\end{aligned}
$$

- Sample sentences:

```
 aturtleknowsthatsamtellskimthattheturtlewalksands
amkillsmax.kimknowsthatsamtellsmaxthatkimtellsk
imthatkimhitstheman.kimtellsaturtlethatmaxruns
```

Slide based on “Unsupervised grammar induction with Minimum Description Length” by Roni Katzir

Experiment 2

Step : 423000 Temperature : 26.2
$1 \rightarrow aman1; 1 \rightarrow hits1o; m \rightarrow h; 1 \rightarrow bill1m; m \rightarrow au; m \rightarrow \epsilon; 1 \rightarrow or1; 1 \rightarrow knowsthat1u; 1 \rightarrow urtle1; a \rightarrow n; 1 \rightarrow eman; m \rightarrow a; 1 \rightarrow heui; 1 \rightarrow edog; 1 \rightarrow saysthat1; t \rightarrow e; a \rightarrow x; 1 \rightarrow wak1e; a \rightarrow o; 1 \rightarrow and1; 1 \rightarrow \epsilon; 1 \rightarrow tells1; 1 \rightarrow walks1c; 1 \rightarrow raac; 1 \rightarrow runs1; a \rightarrow uu; 1 \rightarrow acat1cx; 1 \rightarrow x; 1 \rightarrow kim1ky; m \rightarrow o; a \rightarrow \epsilon; 1 \rightarrow th11; 1 \rightarrow at1w; m \rightarrow r; 1 \rightarrow ma11; 1 \rightarrow et; 1 \rightarrow r; 1 \rightarrow ecat; 1 \rightarrow adog1ty; 1 \rightarrow inksthat; 1 \rightarrow ry; 1 \rightarrow sam1os; 1 \rightarrow kills1; 1 \rightarrow a$
Grammar length: 778 Encoding length: 10620 Energy: 11398

- Accurate segmentation
- Inaccurate structural learning

Prototype-Driven Grammar Induction

- Semi-supervised approach
- Give only a few dozen prototypical examples (for NP e.g. determiner noun, pronouns, …)
- On English Penn Treebank: $F_1 = 65.1$ (52% reduction over naïve PCFG induction)

Aria Haghighi and Dan Klein.
Prototype-Driven Grammar Induction.
ACL 2006

Dan Klein and Chris Manning.
A Generative Constituent-Context Model for Improved Grammar Induction.
ACL 2001

That’s it!