Special Topics in Natural Language Processing

CS698O

Ashutosh Modi
CSE Department, IIT Kanpur

Lecture 17: Parsing 3
Feb 12, 2020

PROBABILISTIC CFGs (PCFGs)

A PCFG consists of:

1. A context free grammar $G = (N, \Sigma, R, S)$
2. A parameter $q(\alpha \rightarrow \beta)$ for each rule $\alpha \rightarrow \beta \in R$

q can be interpreted as the conditional probability of choosing rule $\alpha \rightarrow \beta$ in a left most derivation, given that the non-terminal being expanded is α

$$\sum_{\substack{\alpha \rightarrow \beta \in R \\ \alpha = X \in N}} q(\alpha \rightarrow \beta) = 1$$

$$q(\alpha \rightarrow \beta) \geq 0 \; \forall \; \alpha \rightarrow \beta \in R$$

PROBABILISTIC CFGs (PCFGs)

A PCFG consists of:

1. A context free grammar $G = (N, \Sigma, R, S)$
2. A parameter $q(\alpha \to \beta)$ for each rule $\alpha \to \beta \in R$

Given a parse tree $t \in \mathcal{T}_G$ containing rules $\alpha_1 \to \beta_1, \alpha_2 \to \beta_2, \ldots, \alpha_n \to \beta_n$

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

PCFG

$$G = (N, \Sigma, R, S, q)$$

$$\sum_{\substack{\alpha \to \beta \in R \\ \alpha = X \in N}} q(\alpha \to \beta) = 1 \qquad q(\alpha \to \beta) \geq 0 \ \forall \ \alpha \to \beta \in R$$

Two Questions:

1. ***Learning Problem*** : How do we learn the parameters (probabilities)?
2. ***Decoding Problem***: Given a sentence *s* how do we find the most likely tree?

$$\underset{t \in \mathcal{T}_G(s)}{\operatorname{argmax}} \ p(t)$$

LEARNING IN PCFGs

$$G = (N, \Sigma, R, S, q)$$

$$\{s_i, t_i\}_{i=1}^{m}$$

$$s_i = \mathbf{Yield}(t_i)$$

$$\sum_{\substack{\alpha \to \beta \in R \\ \alpha = X \in N}} q(\alpha \to \beta) = 1 \qquad q(\alpha \to \beta) \geq 0 \ \forall \ \alpha \to \beta \in R$$

$$q(\alpha \to \beta)_{ML} = \frac{C(\alpha \to \beta)}{C(\alpha)}$$

CYK DECODING ALGORITHM

Input: sentence $s = x_1 \dots x_n$ and $PCFG\ G = (N, \Sigma, R, S, q)$ in CNF form

Initialization: $\forall\ 1 \leq i \leq n$ and $X \in N$

$$\pi(i, i, X) = \begin{cases} q(X \to x_i) & \text{If } X \to x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Algorithm:

For $k = 1, \dots, (n-1)$

For $i = 1, \dots, (n-k)$

$j = i + k$

$\forall X \in N$, calculate

$$\pi(i, j, X) = \max_{\substack{X \to YZ \in R, \\ s \in (i \dots (j-1))}} (q(X \to YZ) \times \pi(i, s, Y) \times \pi(s+1, j, Z))$$

$$bp(i, j, X) = \operatorname*{argmax}_{\substack{X \to YZ \in R, \\ s \in (i \dots (j-1))}} (q(X \to YZ) \times \pi(i, s, Y) \times \pi(s+1, j, Z))$$

end for

end for

Output: Return $\pi(1, n, S) = \max_{t \in \mathcal{T}_G(s)} p(t)$ and $bp(1, n, S) = \operatorname*{argmax}_{t \in \mathcal{T}_G(s)} p(t)$

PROBABILITY OF A SENTENCE

- Given a PCFG what is the probability of a sentence *s* under the PCFG?

PROBABILITY OF A SENTENCE

- Given a PCFG what is the probability of a sentence *s* under the PCFG?

$$\text{Given } PCFG\ G = (N, \Sigma, R, S, q) \text{ in } CNF \text{ form}$$

$$\text{A sentence } s = x_1 \ldots x_n$$

PROBABILITY OF A SENTENCE

- Given a PCFG what is the probability of a sentence *s* under the PCFG?

$$\text{Given } PCFG\ G = (N, \Sigma, R, S, q) \text{ in } CNF \text{ form}$$

$$\text{A sentence } s = x_1 \ldots x_n$$

$$p(s) = ?$$

PROBABILITY OF A SENTENCE

- Given a PCFG what is the probability of a sentence *s* under the PCFG?

$$\text{Given } PCFG\ G = (N, \Sigma, R, S, q) \text{ in } CNF \text{ form}$$

$$\text{A sentence } s = x_1 \ldots x_n$$

$$p(s) = \sum_{t \in \mathcal{T}_G(s)} p(t)$$

INSIDE ALGORITHM

Input: sentence $s = x_1 \dots x_n$ and $PCFG\ G = (N, \Sigma, R, S, q)$ in CNF form

Initialization: $\forall\ 1 \leq i \leq n$ and $X \in N$

$$\pi(i, i, X) = \begin{cases} q(X \to x_i) & \text{If } X \to x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Algorithm:

For $k = 1, \dots, (n-1)$

  For $i = 1, \dots, (n-k)$

    $j = i + k$

    $\forall X \in N$, calculate

$$\pi(i, j, X) = \sum_{\substack{X \to YZ \in R, \\ s \in (i \dots (j-1))}} (q(X \to YZ) \times \pi(i, s, Y) \times \pi(s+1, j, Z))$$

  end for

end for

Output: Return $\pi(1, n, S) = \sum_{t \in \mathcal{T}(s)} p(t)$

PCFGs Limitations

- PCFGs alone are poor model for statistical parsing

PCFGs Limitations

- PCFGs alone are poor model for statistical parsing
- Two problems:
  - Lack of sensitivity to lexical information
  - Lack of sensitivity to structural preferences

PCFGs

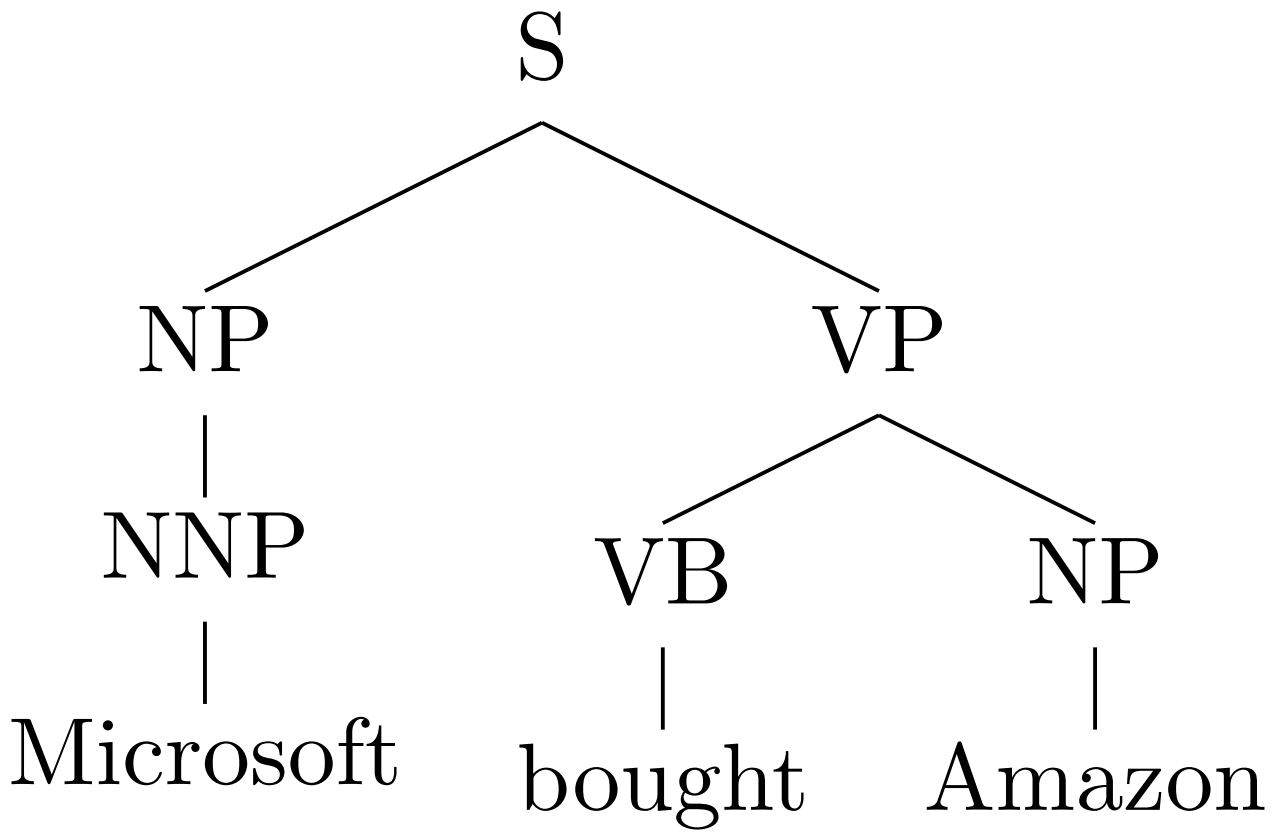
- PCFGs alone are poor model for statistical parsing

- Two problems:

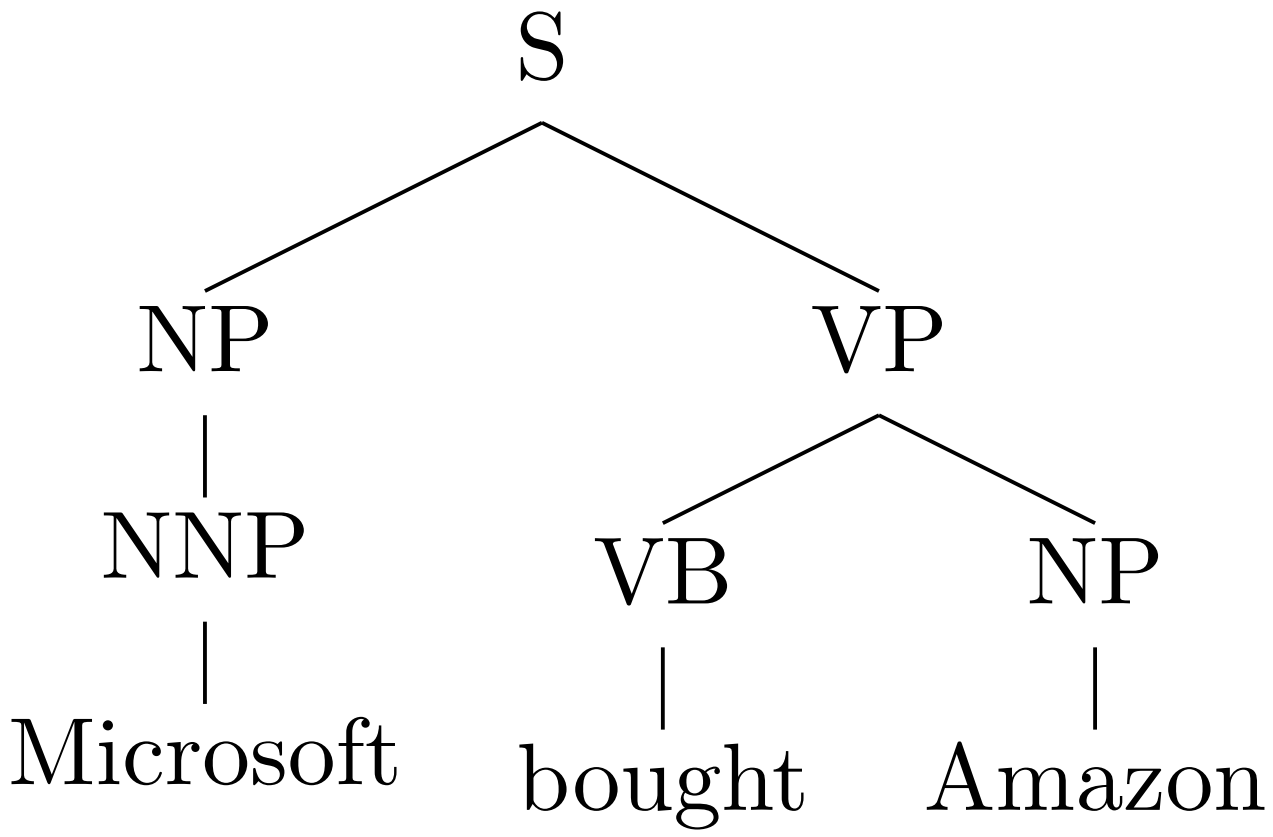
  - Lack of sensitivity to lexical information

  - Lack of sensitivity to structural preferences

PCFGs: Lack of Sensitivity to Lexical Information

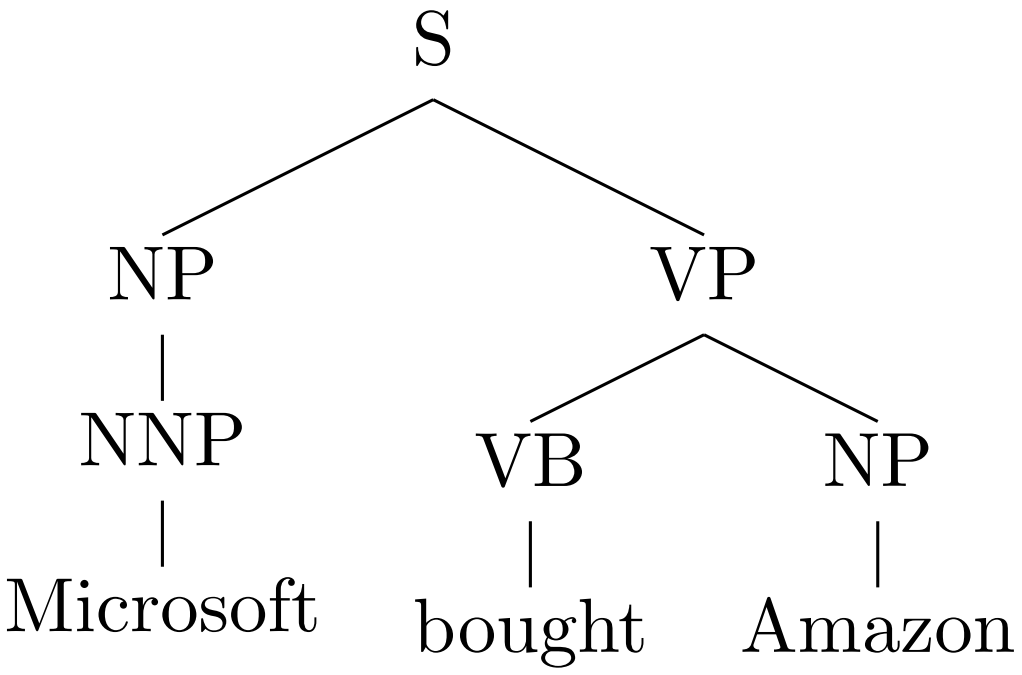


PCFGs: Lack of Sensitivity to Lexical Information



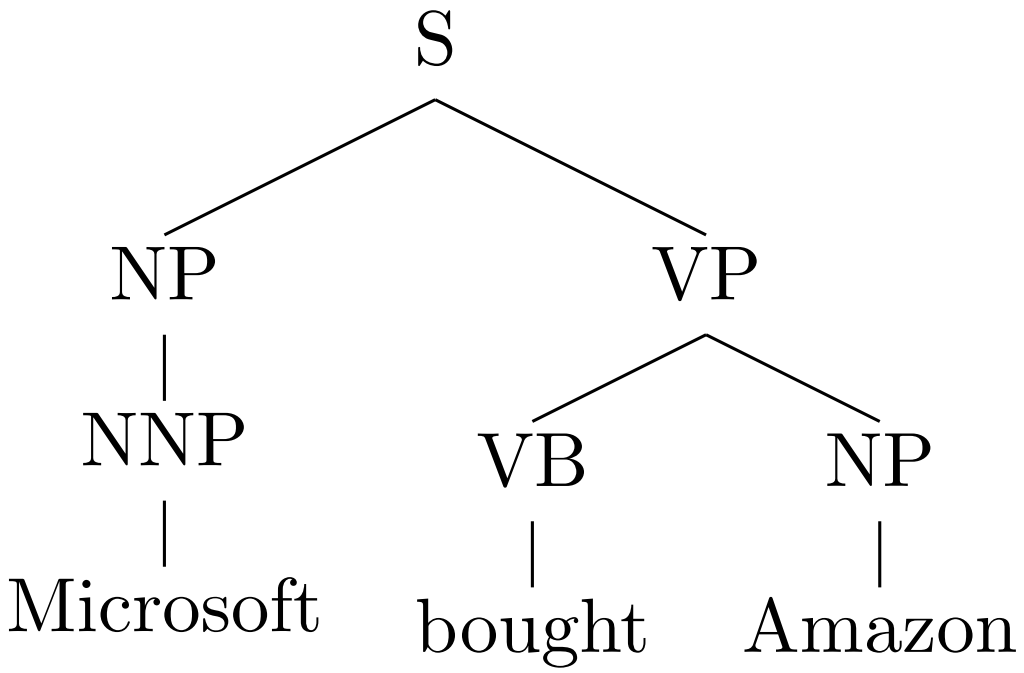
$$q(S \rightarrow NP\ VP) \times q(VP \rightarrow VB\ NP) \times q(NNP \rightarrow Microsoft)$$
$$\times\ q(VB \rightarrow bought) \times q(NP \rightarrow Amazon)$$

PCFGs: Lack of Sensitivity to Lexical Information



- PCFG makes very strong independence assumption
- Each lexical item depends only on the POS above that lexical item in the tree
- It does not depend on any other information in the tree

PCFGs: Lack of Sensitivity to Lexical Information



- PCFG makes very strong independence assumption
- Each lexical item depends only on the POS above that lexical item in the tree
- It does not depend on any other information in the tree

$$p_t(x_i \mid POS \rightarrow x_i, \alpha_1 \rightarrow \beta_1, \ldots, \alpha_k \rightarrow \beta_k) = p_t(x_i \mid POS \rightarrow x_i)$$

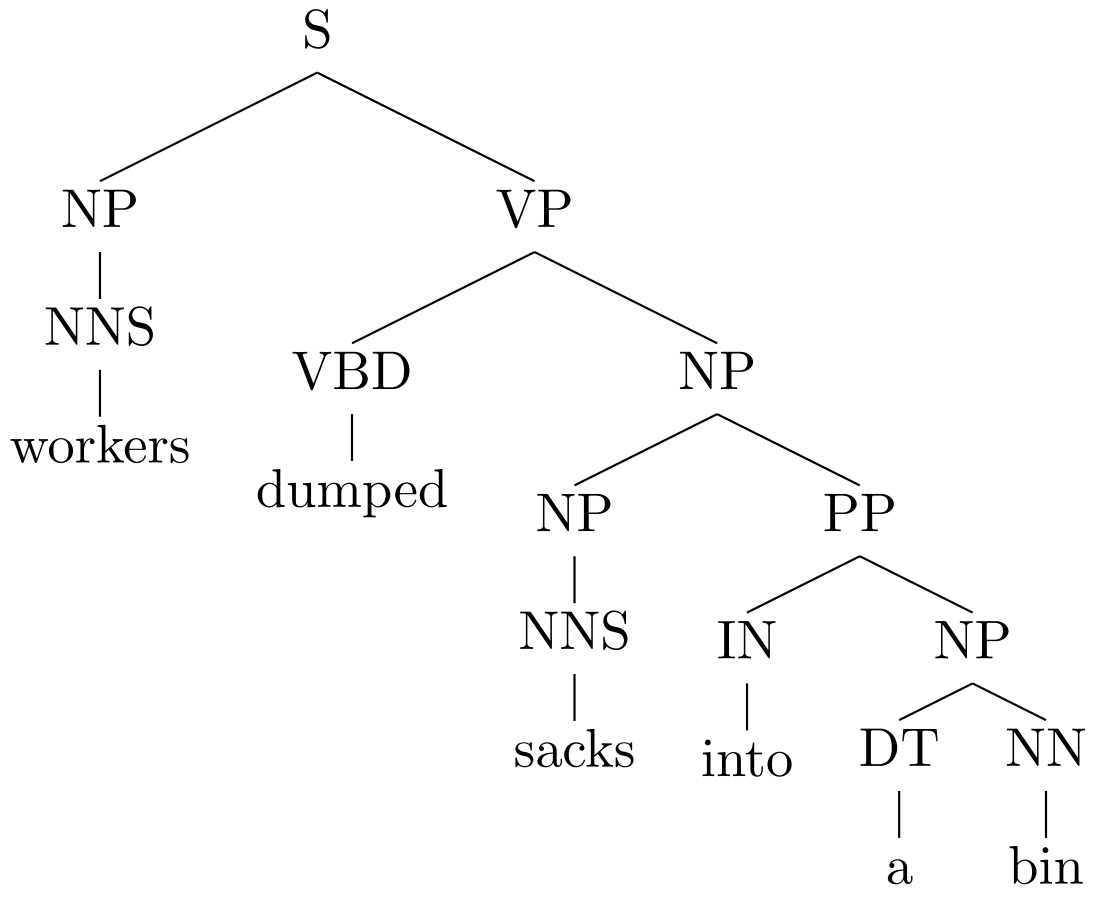
PCFGs: Lack of Sensitivity to Lexical Information

workers dumped sacks into a bin

PCFGs: Lack of Sensitivity to Lexical Information

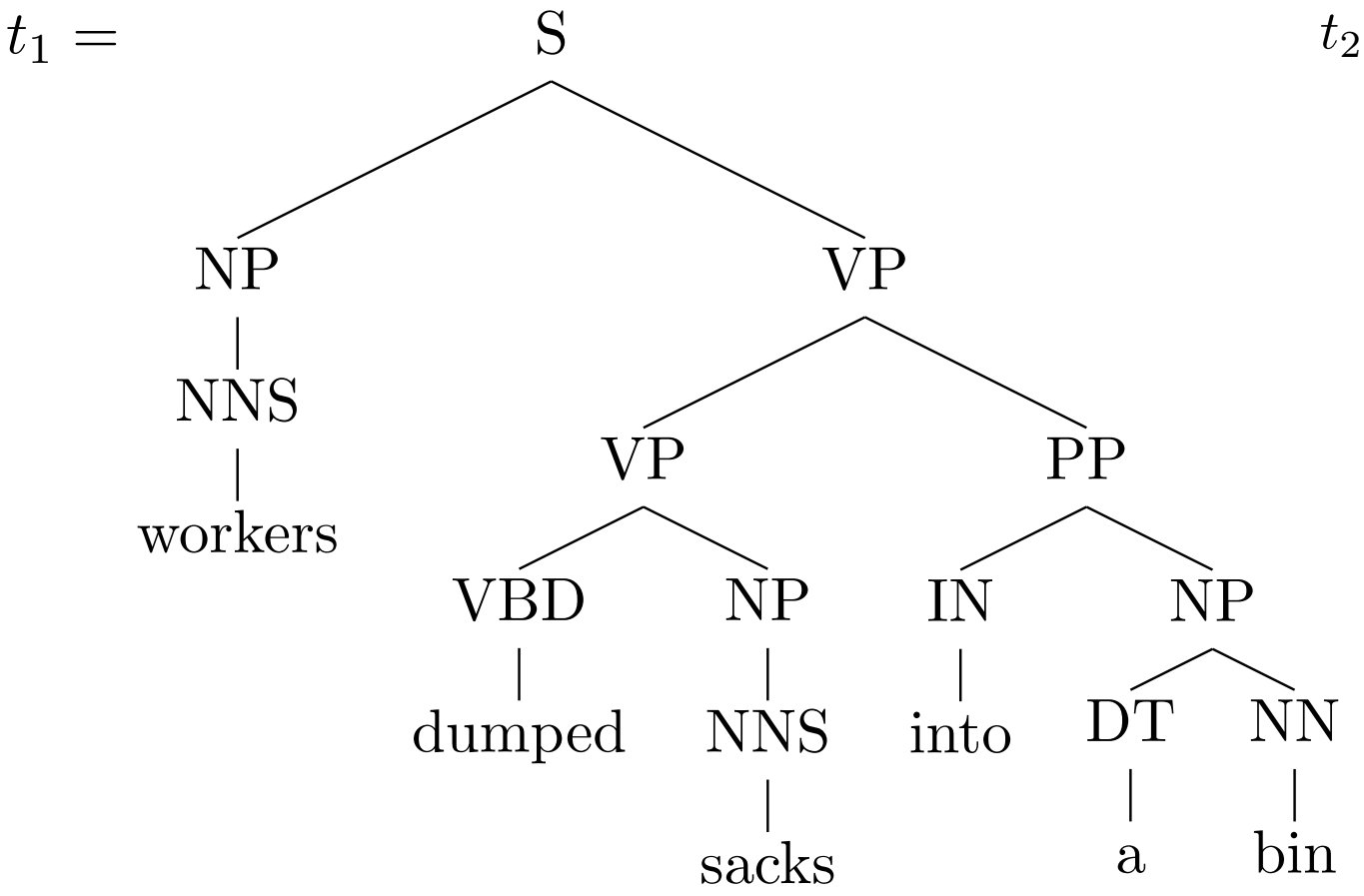
workers dumped sacks into a bin

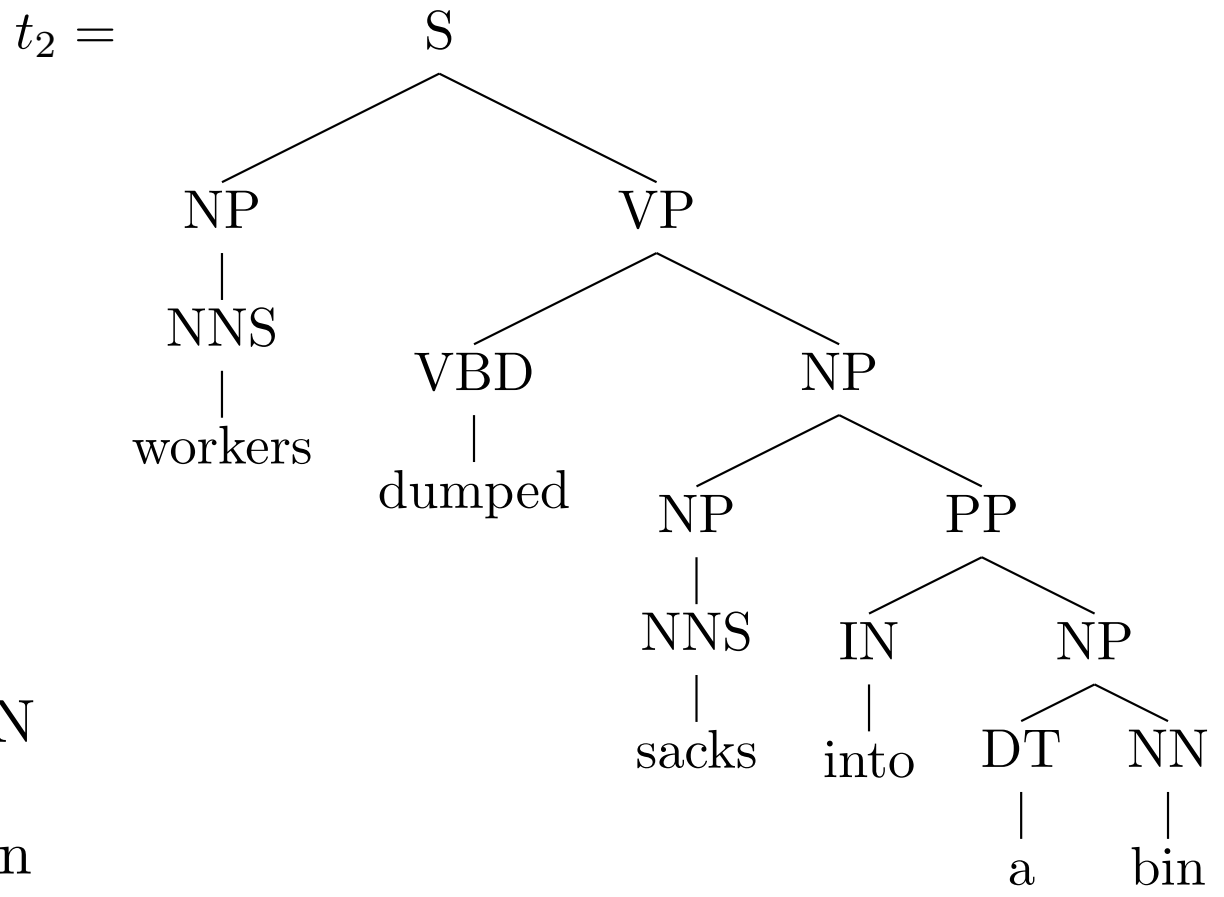
$t_2 =$



PCFGs: Lack of Sensitivity to Lexical Information

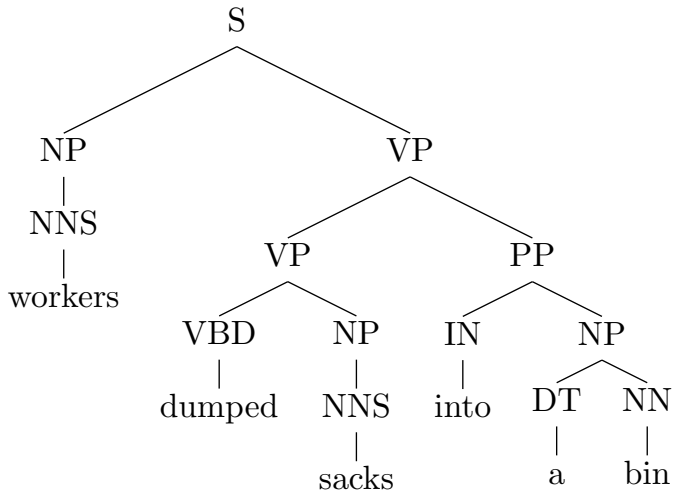
workers dumped sacks into a bin





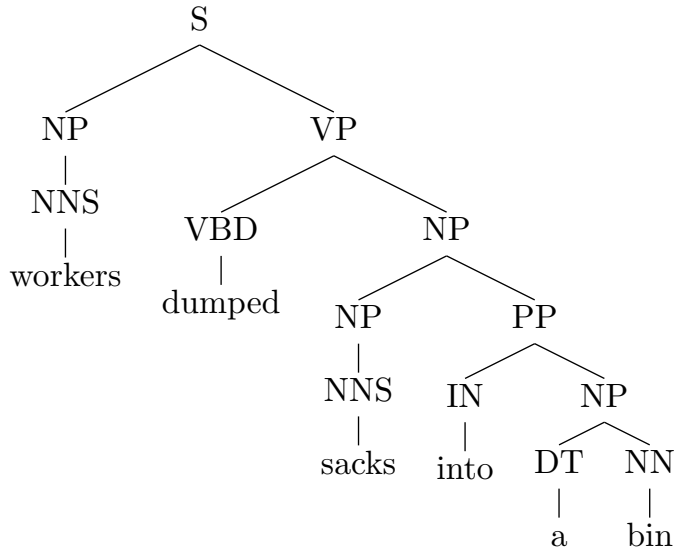
PCFGs: Lack of Sensitivity to Lexical Information

$t_1 =$



workers dumped sacks into a bin

$t_2 =$

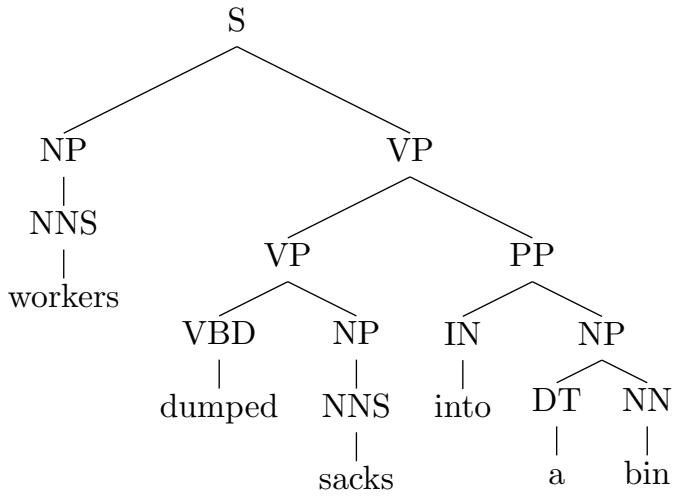


$p(t_1) = ?$

$p(t_2) = ?$

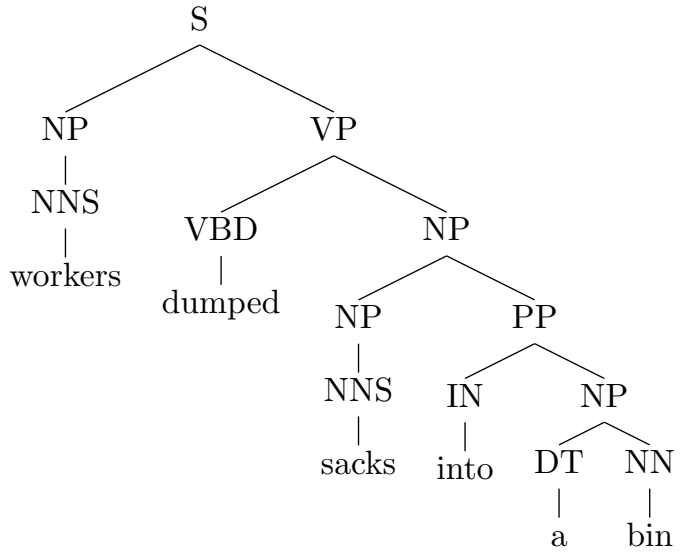
PCFGs: Lack of Sensitivity to Lexical Information

$t_1 =$



workers dumped sacks into a bin

$t_2 =$

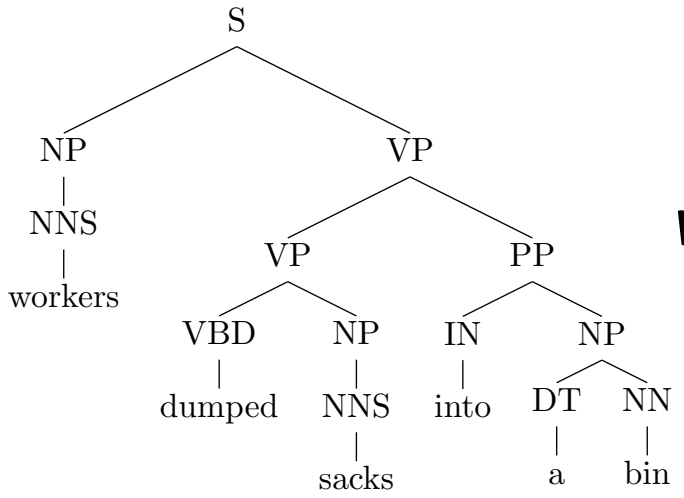


$$p(t_1) = q(S \rightarrow NP\ VP) \times q(NP \rightarrow NNS) \times q(NNS \rightarrow workers) \times q(VP \rightarrow VP\ PP) \times q(VP \rightarrow VBD\ NP) \times q(VBD \rightarrow dumped) \times q(NP \rightarrow NNS) \times q(NNS \rightarrow sacks) \times q(PP \rightarrow IN) \times q(IN \rightarrow into) \times q(NP \rightarrow DT\ NN) \times q(DT \rightarrow a) \times q(NN \rightarrow bin)$$

$$p(t_2) = q(S \rightarrow NP\ VP) \times q(NP \rightarrow NNS) \times q(NNS \rightarrow workers) \times q(VP \rightarrow VBD\ NP) \times q(VBD \rightarrow dumped) \times q(NP \rightarrow NP\ PP) \times q(NP \rightarrow NNS) \times q(NNS \rightarrow sacks) \times q(PP \rightarrow IN) \times q(IN \rightarrow into) \times q(NP \rightarrow DT\ NN) \times q(DT \rightarrow a) \times q(NN \rightarrow bin)$$

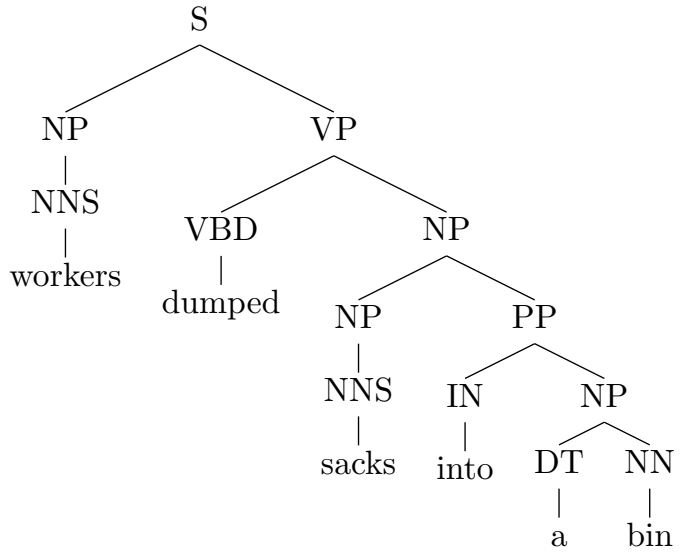
PCFGs: Lack of Sensitivity to Lexical Information

$t_1 =$



workers dumped sacks into a bin

$t_2 =$

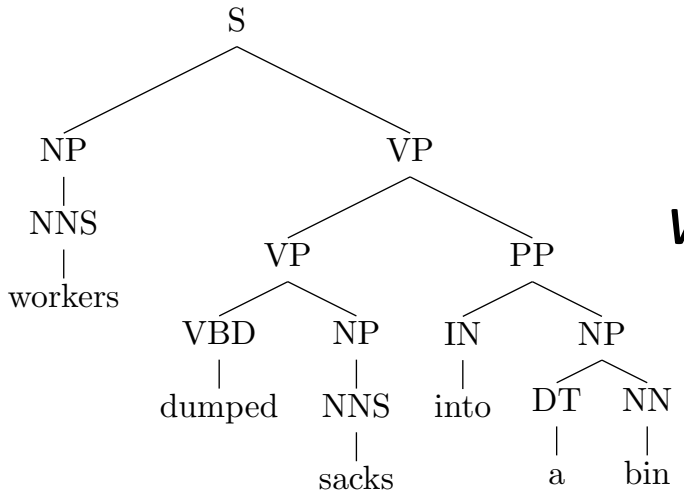


$$p(t_1) = q(S \rightarrow NP\ VP) \times q(NP \rightarrow NNS) \times q(NNS \rightarrow workers) \times q(VP \rightarrow VP\ PP) \times q(VP \rightarrow VBD\ NP) \times q(VBD \rightarrow dumped) \times q(NP \rightarrow NNS) \times q(NNS \rightarrow sacks) \times q(PP \rightarrow IN) \times q(IN \rightarrow into) \times q(NP \rightarrow DT\ NN) \times q(DT \rightarrow a) \times q(NN \rightarrow bin)$$

$$p(t_2) = q(S \rightarrow NP\ VP) \times q(NP \rightarrow NNS) \times q(NNS \rightarrow workers) \times q(VP \rightarrow VBD\ NP) \times q(VBD \rightarrow dumped) \times q(NP \rightarrow NP\ PP) \times q(NP \rightarrow NNS) \times q(NNS \rightarrow sacks) \times q(PP \rightarrow IN) \times q(IN \rightarrow into) \times q(NP \rightarrow DT\ NN) \times q(DT \rightarrow a) \times q(NN \rightarrow bin)$$

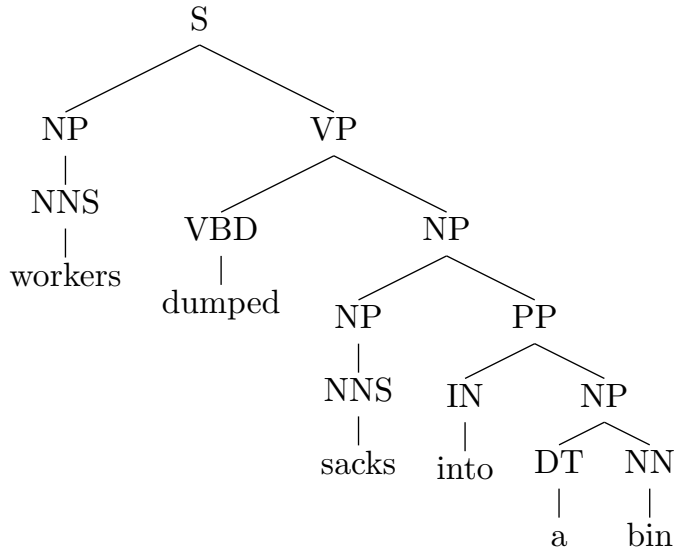
PCFGs: Lack of Sensitivity to Lexical Information

$t_1 =$



workers dumped sacks into a bin

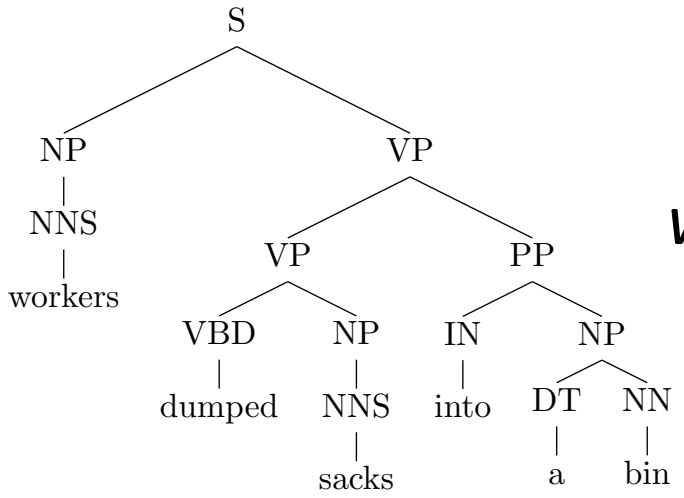
$t_2 =$



$$\underset{t \in \mathcal{T}_G(s)}{\operatorname{argmax}}\, p(t) = \begin{cases} t_1 & \text{if } q(VP \rightarrow VP\ PP) > q(NP \rightarrow NP\ PP) \\ t_2 & \text{if } q(VP \rightarrow VP\ PP) < q(NP \rightarrow NP\ PP) \end{cases}$$

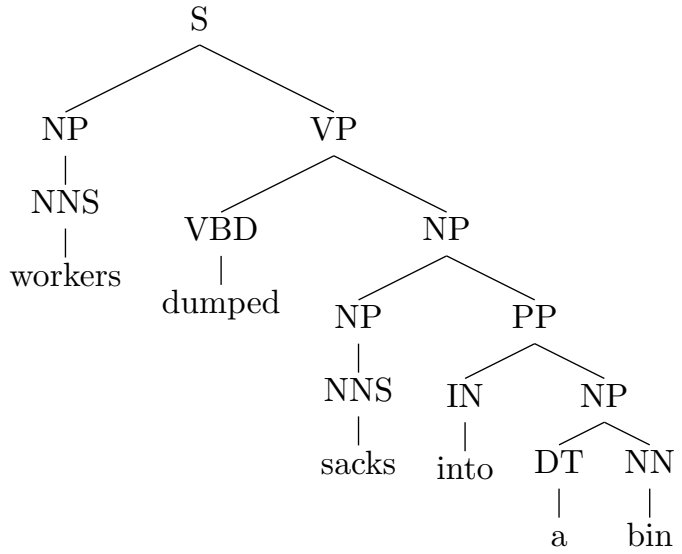
PCFGs: Lack of Sensitivity to Lexical Information

$t_1 =$



workers dumped sacks into a bin

$t_2 =$



$$\underset{t \in \mathcal{T}_G(s)}{\operatorname{argmax}}\, p(t) = \begin{cases} t_1 & \text{if } q(VP \rightarrow VP\ PP) > q(NP \rightarrow NP\ PP) \\ t_2 & \text{if } q(VP \rightarrow VP\ PP) < q(NP \rightarrow NP\ PP) \end{cases}$$

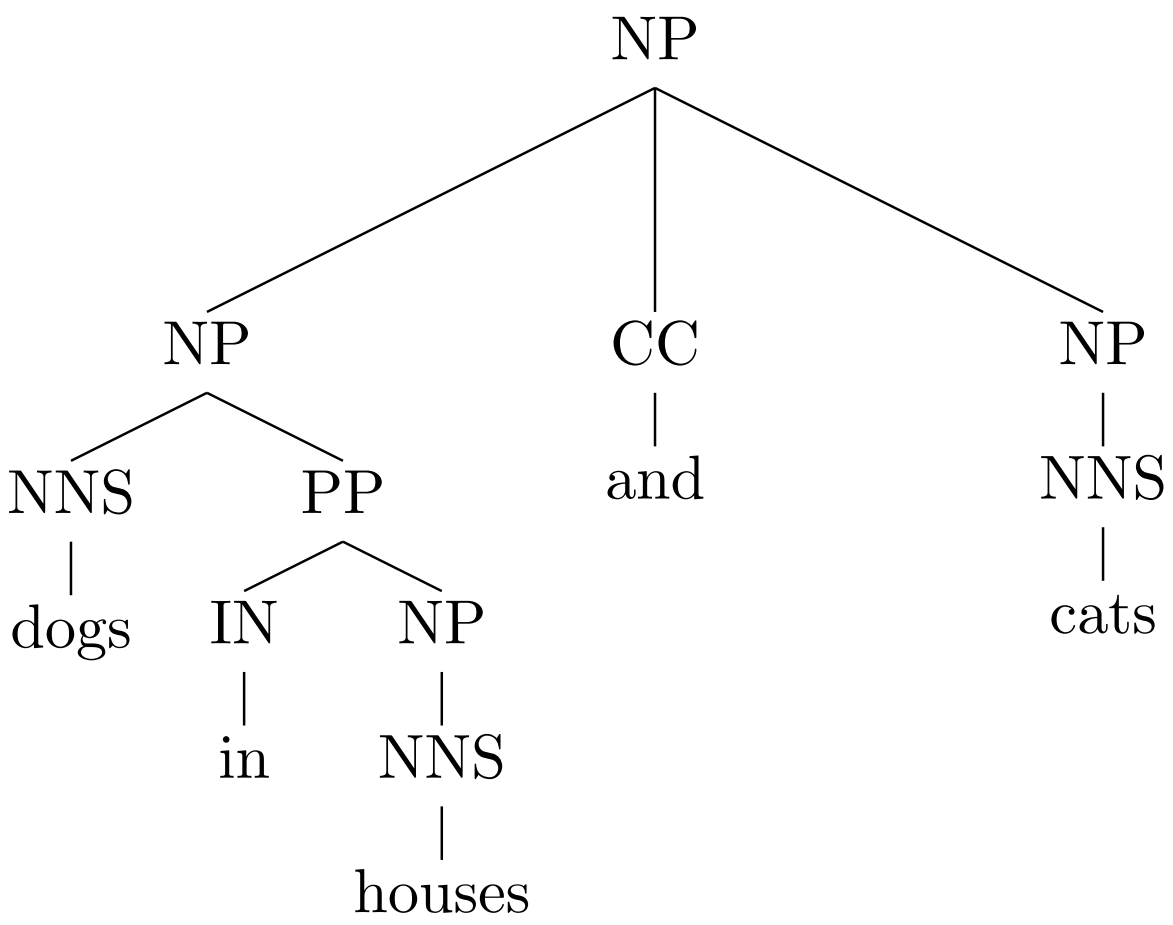
Decision is entirely independent of the lexical information in the sentence

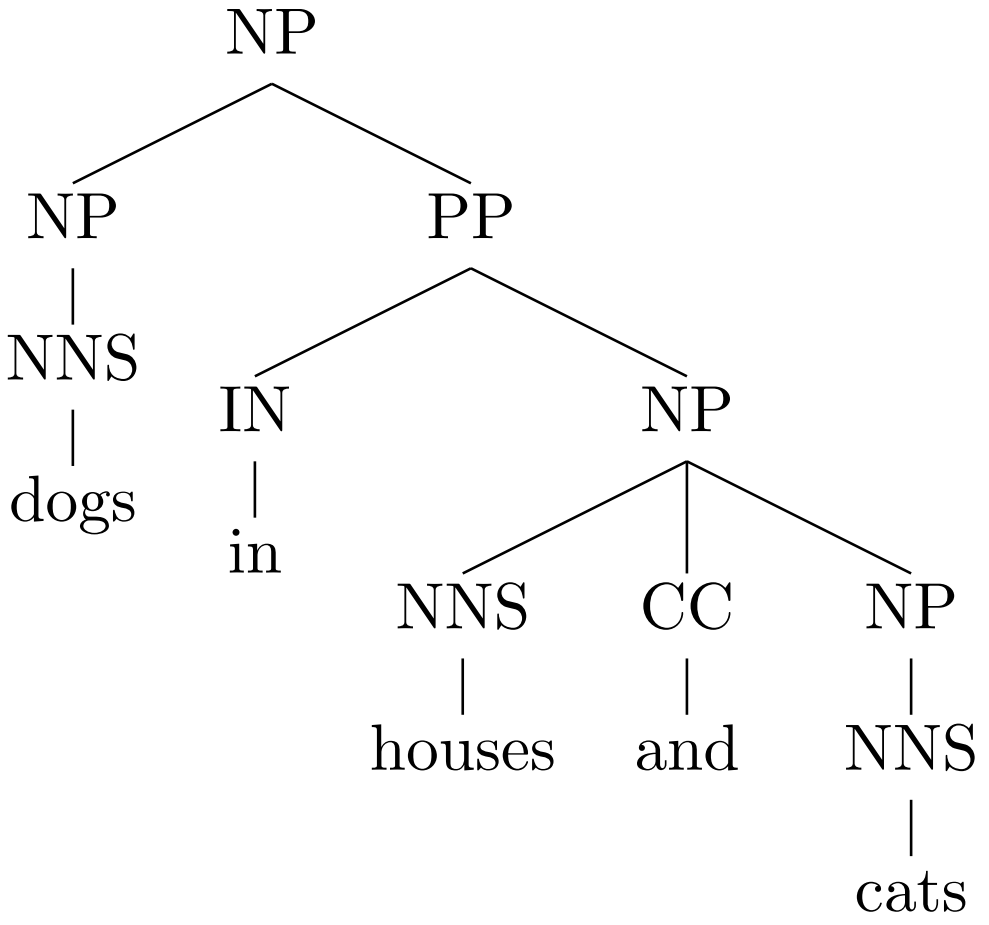
PCFGs: Lack of Sensitivity to Lexical Information

dogs in house and cats

PCFGs: Lack of Sensitivity to Lexical Information

dogs in house and cats





PCFGs

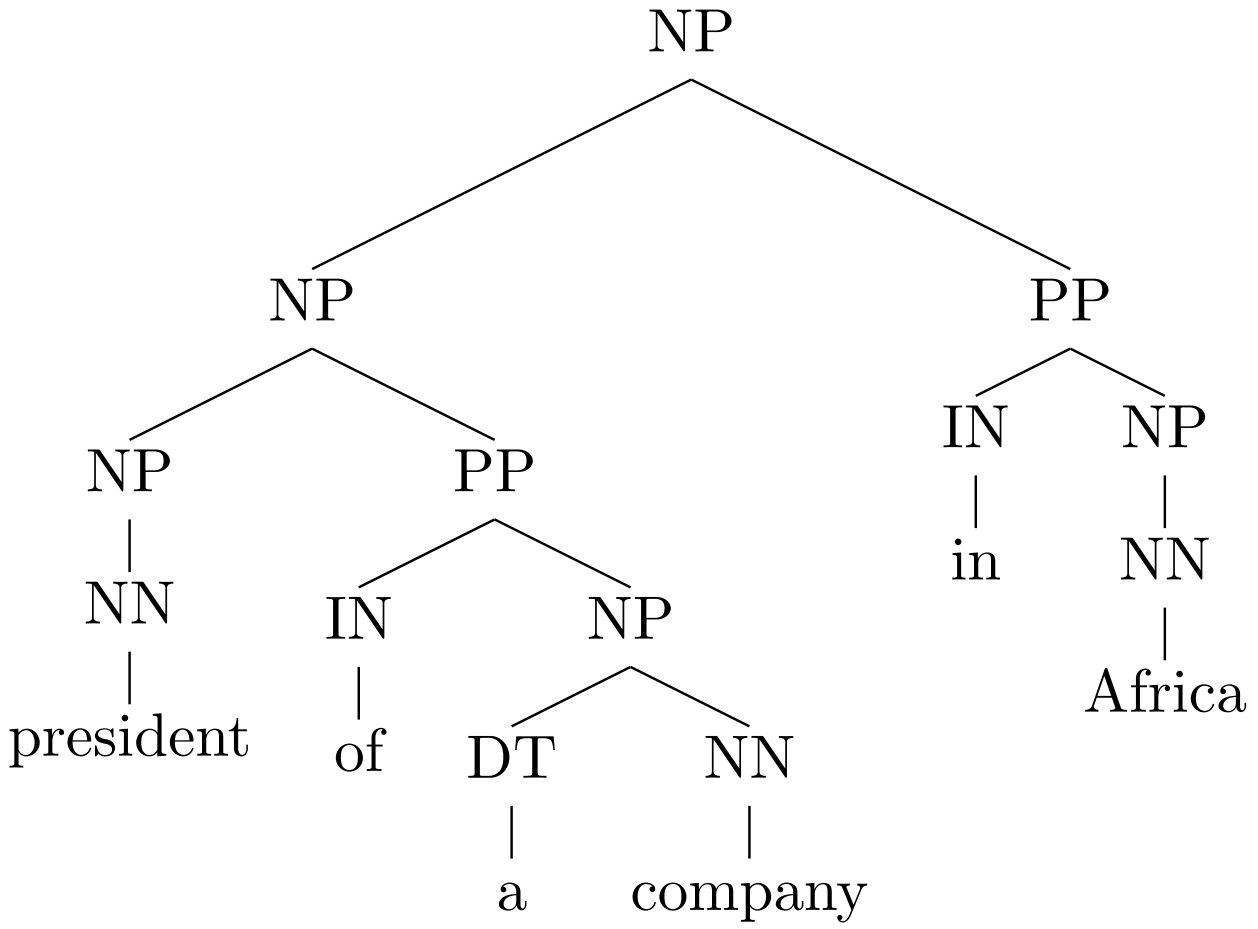
- PCFGs alone are poor model for statistical parsing
- Two problems:
  - Lack of sensitivity to lexical information
  - Lack of sensitivity to structural preferences

PCFGs: Lack of Sensitivity to Structural Preferences

president of a company in Africa

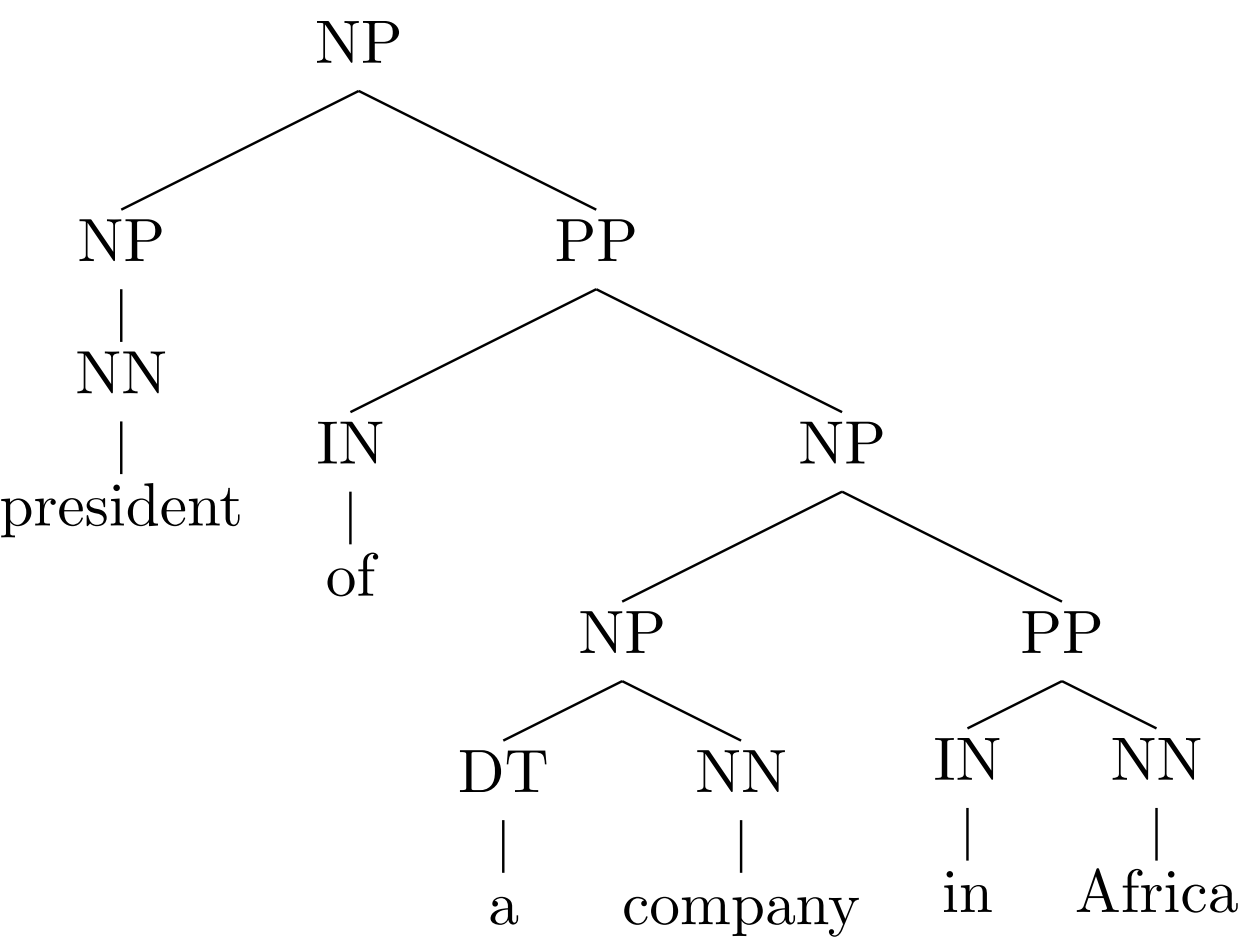
PCFGs: Lack of Sensitivity to Structural Preferences

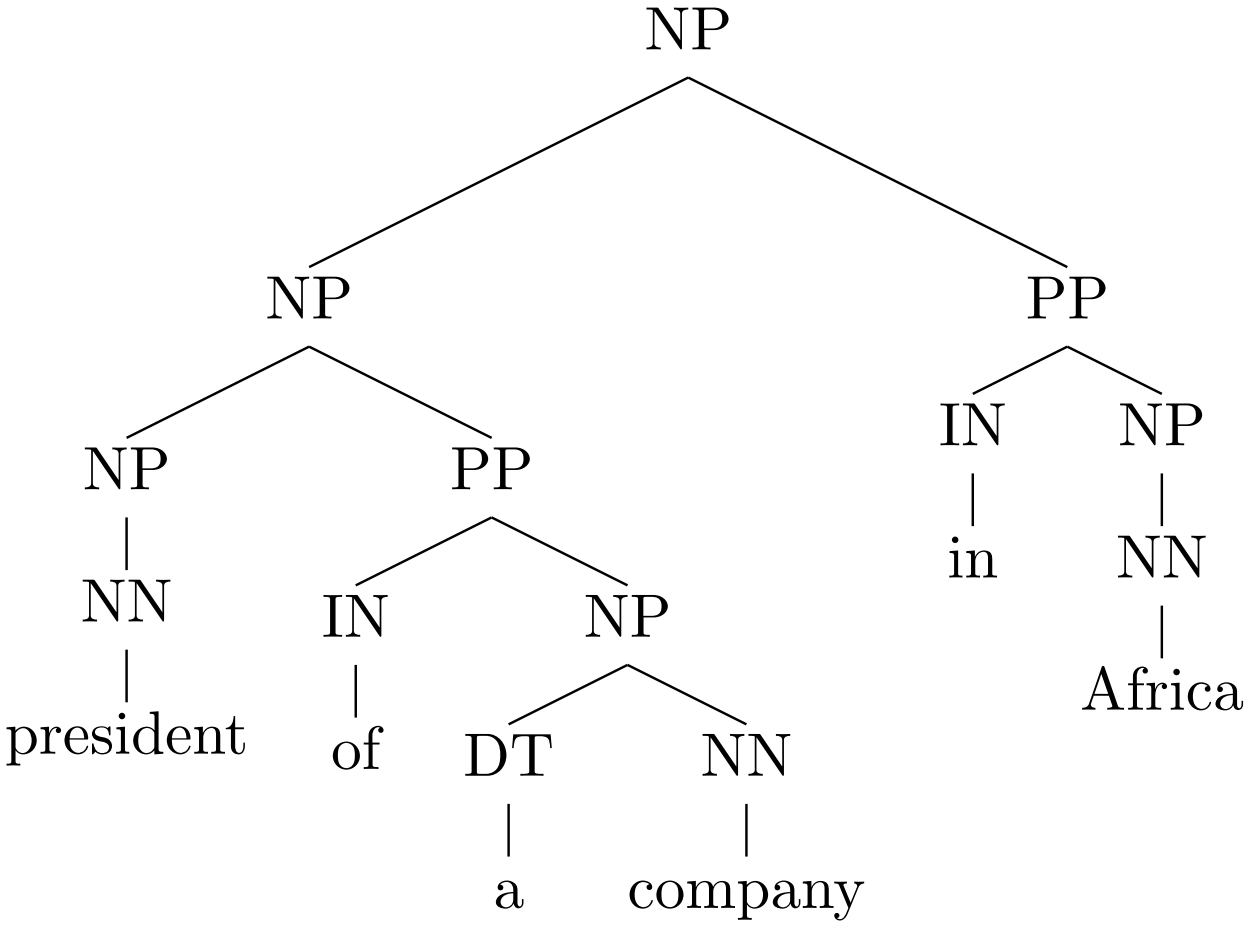
president of a company in Africa



PCFGs: Lack of Sensitivity to Structural Preferences

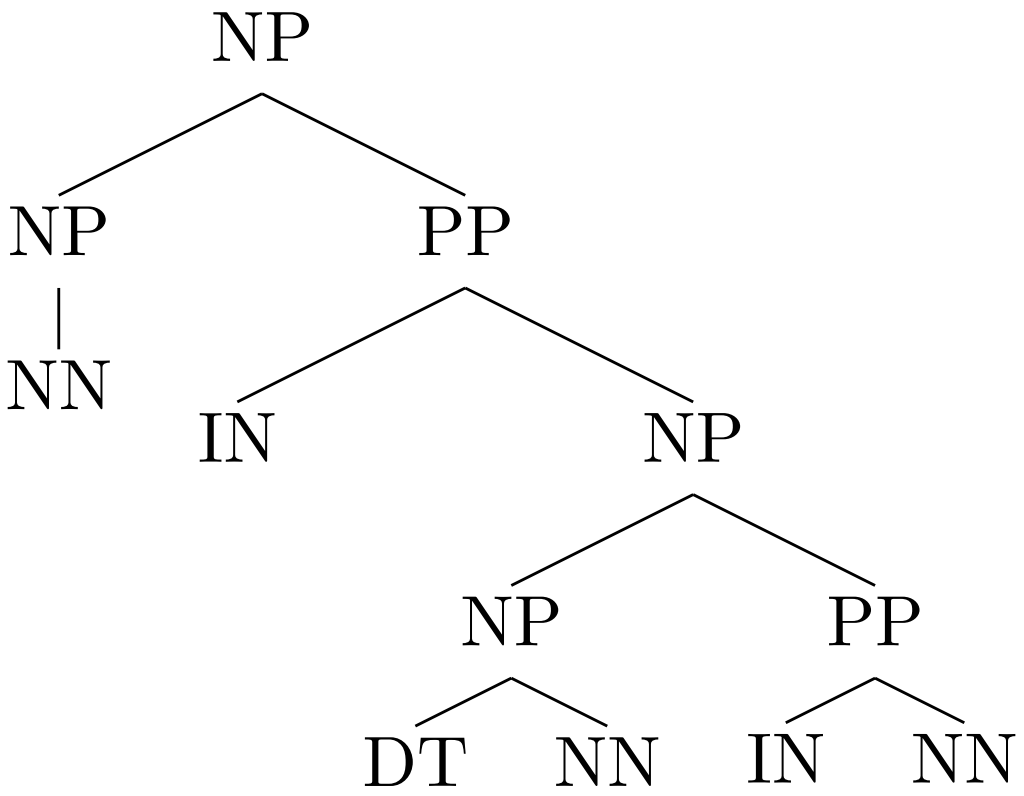
president of a company in Africa

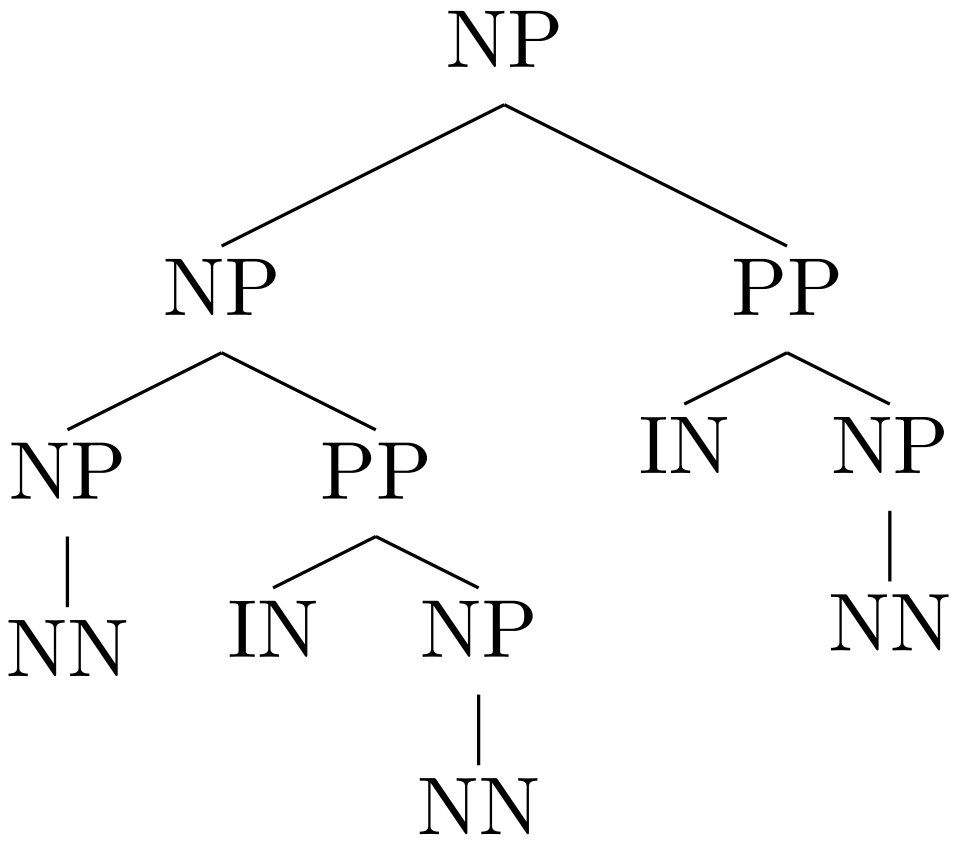




PCFGs: Lack of Sensitivity to Structural Preferences

president of a company in Africa



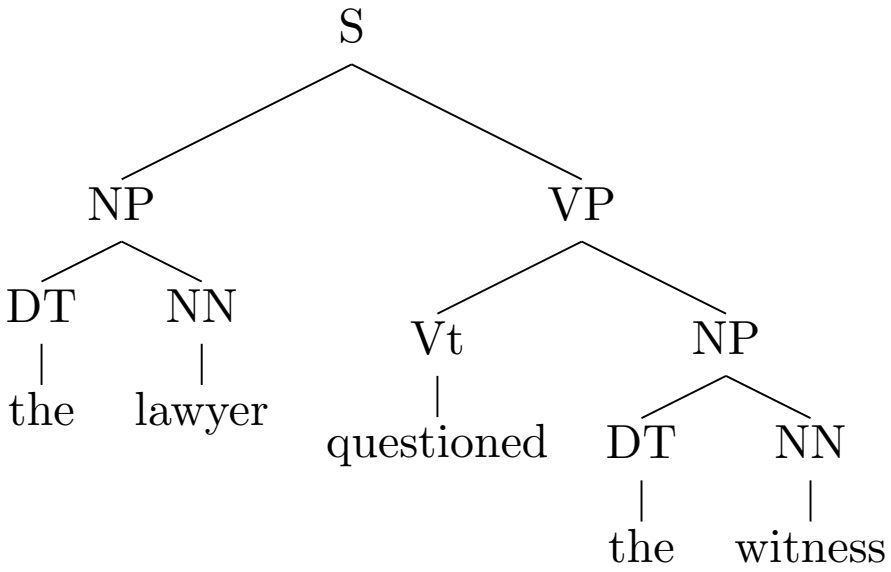


Close Attachment

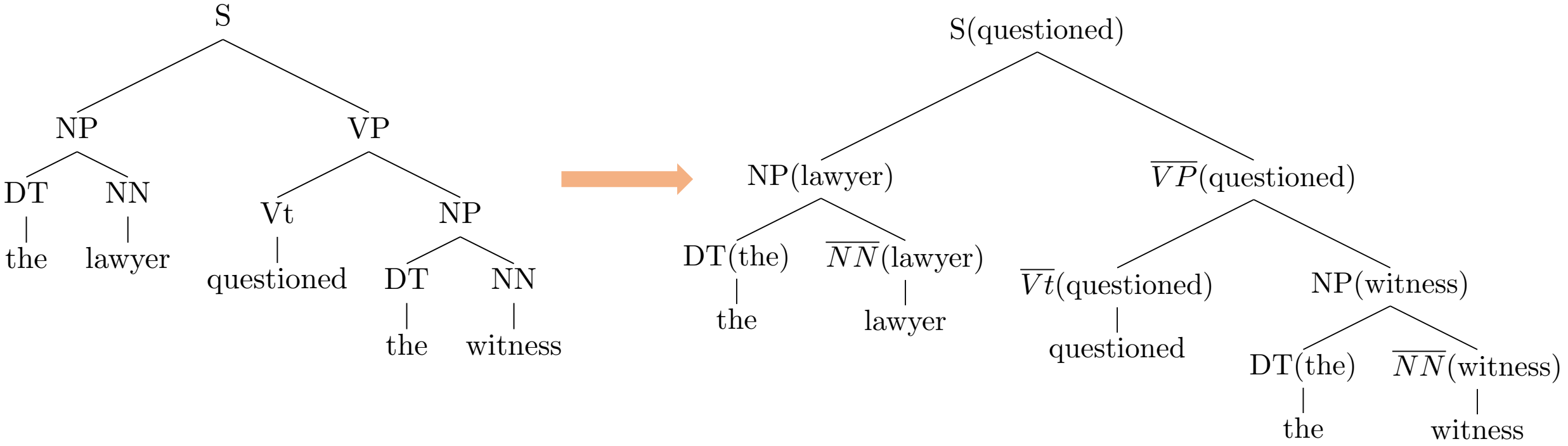
PCFGs: Lack of Sensitivity to Structural Preferences

John was believed to have been shot by Bill

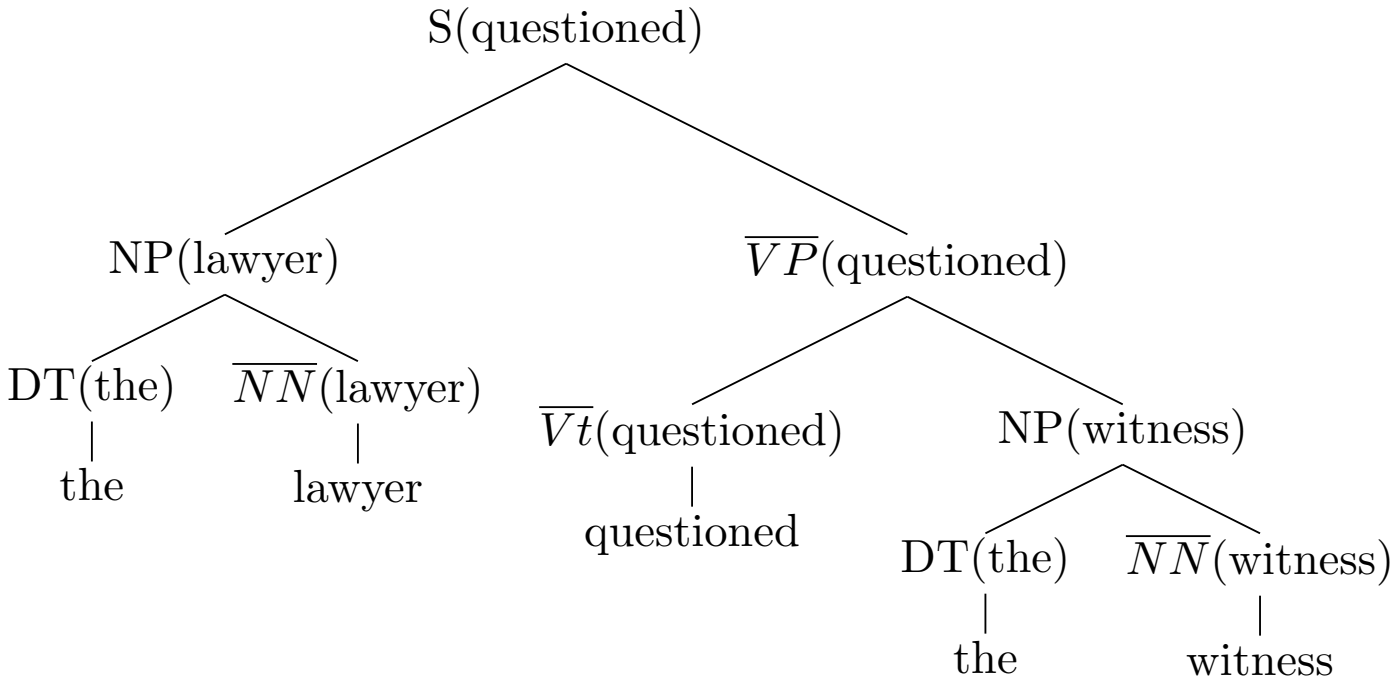
Lexicalization of a Treebank



Lexicalization of a Treebank



Lexicalization of a Treebank



For each context free rule of the form: $X \rightarrow Y_1\ Y_2\ \ldots\ Y_n$

The idea is to identify an index $h \in \{1, 2, \ldots, n\}$ that specifies **head** of the rule

Lexicalization of a Treebank

For each context free rule of the form: $X \rightarrow Y_1\ Y_2\ \ldots\ Y_n$

The idea is to identify an index $h \in \{1, 2, \ldots, n\}$ that specifies **head** of the rule

$$S \rightarrow NP\ VP$$

Lexicalization of a Treebank

For each context free rule of the form: $X \rightarrow Y_1\ Y_2\ \ldots\ Y_n$

The idea is to identify an index $h \in \{1, 2, \ldots, n\}$ that specifies **head** of the rule

$$S \rightarrow NP\ VP \qquad\qquad h = 2$$

Lexicalization of a Treebank

For each context free rule of the form: $X \rightarrow Y_1\ Y_2\ \ldots\ Y_n$

The idea is to identify an index $h \in \{1, 2, \ldots, n\}$ that specifies **head** of the rule

$$S \rightarrow NP\ VP \qquad\qquad h = 2$$

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

Lexicalization of a Treebank

For each context free rule of the form: $X \rightarrow Y_1\ Y_2\ \ldots\ Y_n$

The idea is to identify an index $h \in \{1, 2, \ldots, n\}$ that specifies **head** of the rule

$$S \rightarrow NP\ VP \qquad h = 2$$

$$NP \rightarrow NP\ PP\ PP\ PP \qquad h = 1$$

Lexicalization of a Treebank

For each context free rule of the form: $X \rightarrow Y_1\ Y_2\ \ldots\ Y_n$

The idea is to identify an index $h \in \{1, 2, \ldots, n\}$ that specifies **head** of the rule

$$S \rightarrow NP\ VP \qquad h = 2$$

$$NP \rightarrow NP\ PP\ PP\ PP \qquad h = 1$$

$$PP \rightarrow IN\ NP$$

Lexicalization of a Treebank

For each context free rule of the form: $X \rightarrow Y_1\ Y_2\ \ldots\ Y_n$

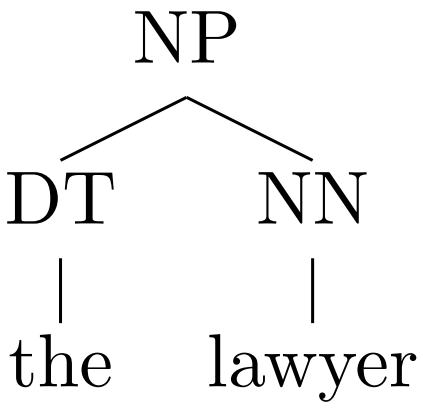
The idea is to identify an index $h \in \{1, 2, \ldots, n\}$ that specifies **head** of the rule

$$S \rightarrow NP\ VP \qquad h = 2$$

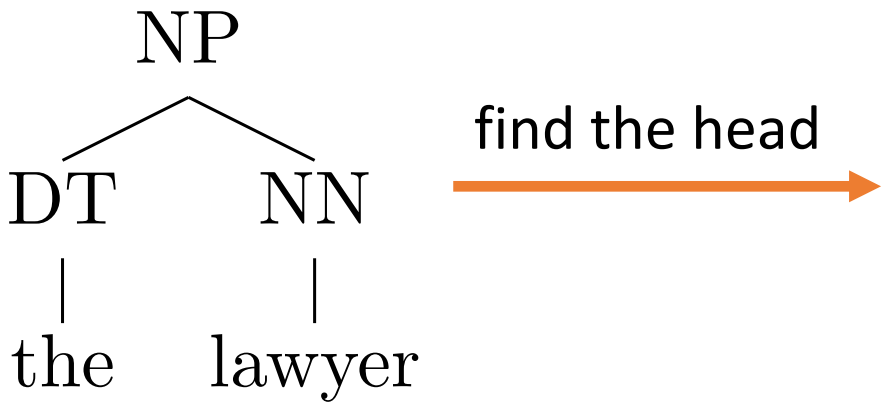
$$NP \rightarrow NP\ PP\ PP\ PP \qquad h = 1$$

$$PP \rightarrow IN\ NP \qquad h = 1$$

Lexicalization of a Treebank

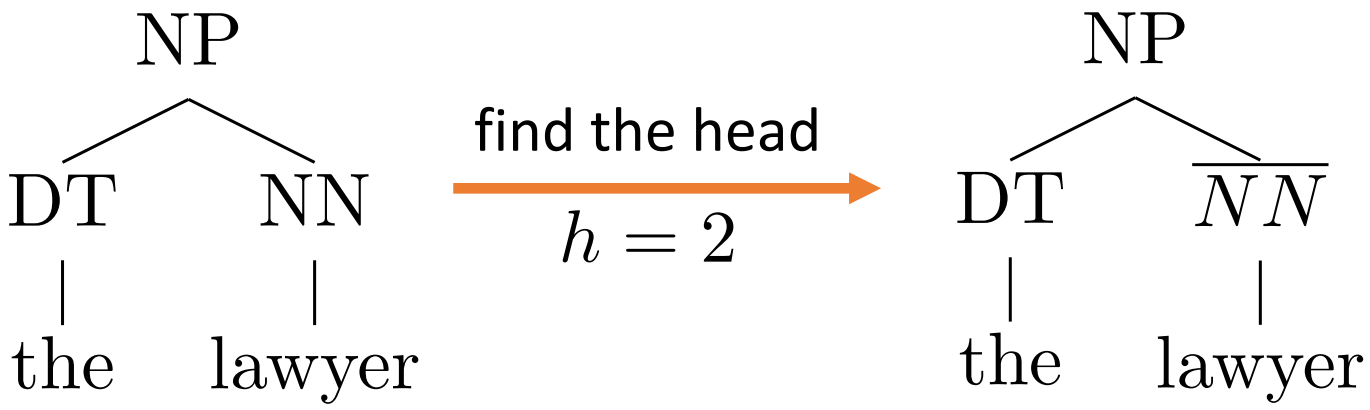


Lexicalization of a Treebank

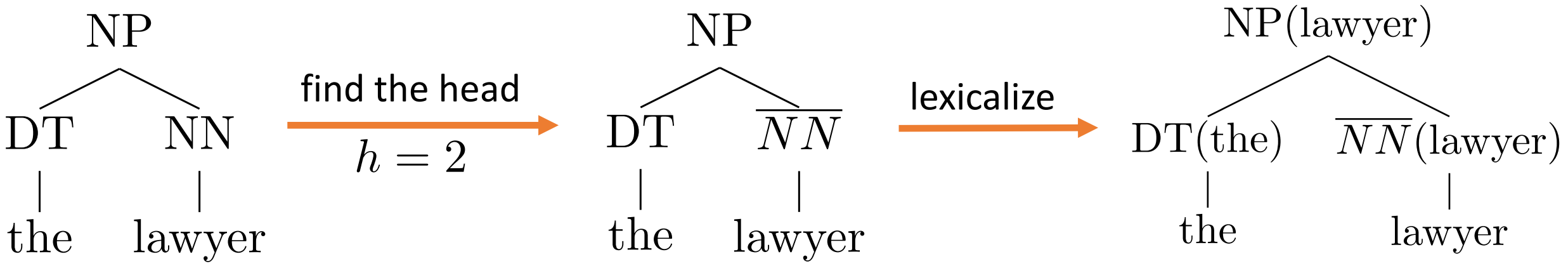


find the head

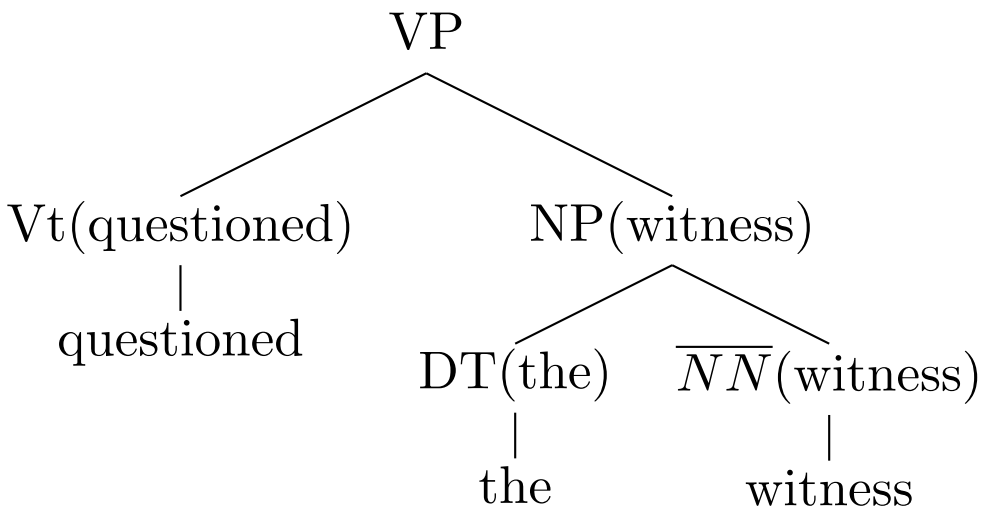
Lexicalization of a Treebank



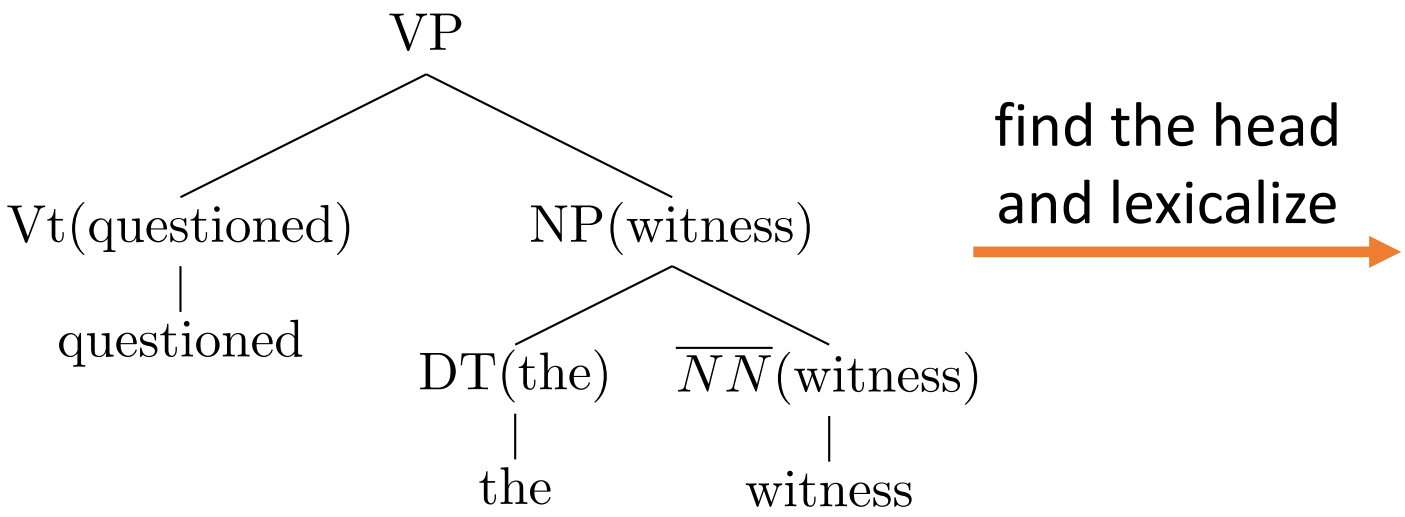
Lexicalization of a Treebank



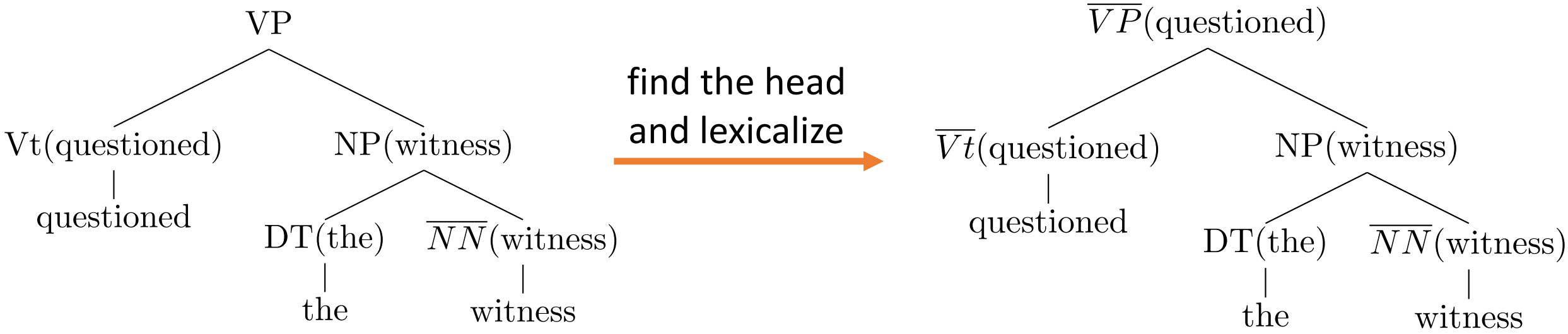
Lexicalization of a Treebank



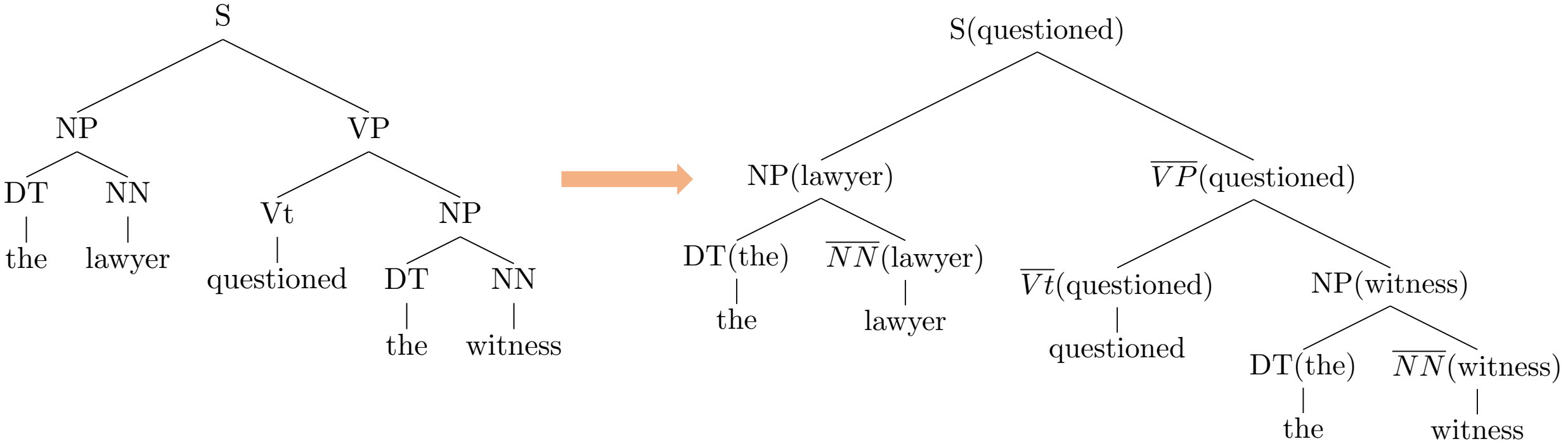
Lexicalization of a Treebank



Lexicalization of a Treebank



Lexicalization of a Treebank



Lexicalization of a Treebank

How to identify the head?

Lexicalization of a Treebank

How to identify the head?

Use Rules/Heuristics (with linguistic knowledge)

Lexicalization of a Treebank

How to identify the head?

Use Rules/Heuristics (with linguistic knowledge)

If the rule contains NN, NNS, or NNP:
  Choose the rightmost NN, NNS, or NNP
Else If the rule contains an NP: Choose the leftmost NP
Else If the rule contains an JJ: Choose the rightmost JJ
Else If the rule contains a CD: Choose the rightmost CD
Else Choose the rightmost child

Lexicalization of a Treebank

How to identify the head?

Use Rules/Heuristics (with linguistic knowledge)

If the rule contains Vi or Vt: Choose the leftmost Vi or Vt
Else If the rule contains an VP: Choose the leftmost VP
Else Choose the leftmost child

Lexicalized PCFGs

- Each rule in PCFG is lexicalized

$$S(examined) \rightarrow NP(lawyer)\ VP(examined)$$

- From formal point of view Lexicalized PCFG is same as regular PCFG

- Number of non-terminals in the grammar expands from a fairly small number to much larger number

- Number of associated parameters also increases

Lexicalized PCFGs

$$S(examined) \rightarrow NP(lawyer)\ VP(examined)$$

$$S(examined) \rightarrow_2 NP(lawyer)\ VP(examined)$$

Lexicalized PCFGs

$$S(examined) \rightarrow NP(lawyer)\ VP(examined)$$

$$S(examined) \rightarrow_2 NP(lawyer)\ VP(examined)$$

$$PP(in) \rightarrow_1 PP(in)\ PP(in)$$

$$PP(in) \rightarrow_2 PP(in)\ PP(in)$$

Lexicalized PCFG Definition

A lexicalized PCFG in CNF is a 6-Tuple $G = (N, \Sigma, R, S, q, \gamma)$ where:

Lexicalized PCFG Definition

A lexicalized PCFG in CNF is a 6-Tuple $G = (N, \Sigma, R, S, q, \gamma)$ where:

N is a finite set of non-terminals in the grammar

Σ is a finite set of lexical items in the grammar

Lexicalized PCFG Definition

A lexicalized PCFG in CNF is a 6-Tuple $G = (N, \Sigma, R, S, q, \gamma)$ where:

N is a finite set of non-terminals in the grammar

Σ is a finite set of lexical items in the grammar

R is a set of rules. Each rule takes one of the following three forms:

$$X(h) \rightarrow_1 Y_1(h)\ Y_2(m)\ where\ X, Y_1, Y_2 \in N, h, m \in \Sigma$$

$$X(m) \rightarrow_2 Y_1(h)\ Y_2(m)\ where\ X, Y_1, Y_2 \in N, h, m \in \Sigma$$

$$X(h) \rightarrow h\ where\ X \in N, h \in \Sigma$$

Lexicalized PCFG Definition

A lexicalized PCFG in CNF is a 6-Tuple $G = (N, \Sigma, R, S, q, \gamma)$ where:

N is a finite set of non-terminals in the grammar

Σ is a finite set of lexical items in the grammar

R is a set of rules. Each rule takes one of the following three forms:

$$X(h) \rightarrow_1 Y_1(h)\ Y_2(m) \ where\ X, Y_1, Y_2 \in N, h, m \in \Sigma$$

$$X(m) \rightarrow_2 Y_1(h)\ Y_2(m) \ where\ X, Y_1, Y_2 \in N, h, m \in \Sigma$$

$$X(h) \rightarrow h \ where\ X \in N, h \in \Sigma$$

For each rule $r \in R$ there is an associated parameter: $q(r),\ where,$

$$q(r) \geq 0 \quad and \quad \sum_{r \in R : LHS(r) = X(h)} q(r) = 1 \quad \forall X \in N, h \in \Sigma$$

Lexicalized PCFG Definition

A lexicalized PCFG in CNF is a 6-Tuple $G = (N, \Sigma, R, S, q, \gamma)$ where:

N is a finite set of non-terminals in the grammar

Σ is a finite set of lexical items in the grammar

R is a set of rules. Each rule takes one of the following three forms:

$$X(h) \rightarrow_1 Y_1(h)\ Y_2(m)\ where\ X, Y_1, Y_2 \in N, h, m \in \Sigma$$

$$X(m) \rightarrow_2 Y_1(h)\ Y_2(m)\ where\ X, Y_1, Y_2 \in N, h, m \in \Sigma$$

$$X(h) \rightarrow h\ where\ X \in N, h \in \Sigma$$

For each rule $r \in R$ there is an associated parameter: $q(r),\ where,$

$$q(r) \geq 0 \quad and \quad \sum_{r \in R : LHS(r) = X(h)} q(r) = 1 \quad \forall X \in N, h \in \Sigma$$

For each $X \in N, h \in \Sigma$, there is a parameter $\gamma(X, h)$

$$\gamma(X, h) \geq 0 \ \ and \ \sum_{X \in N, h \in \Sigma} \gamma(X, h) = 1$$

Lexicalized PCFG Definition

L-PCFG: $G = (N, \Sigma, R, S, q, \gamma)$

$$q(r) \geq 0 \quad and \quad \sum_{r \in R : LHS(r) = X(h)} q(r) = 1 \quad \forall X \in N, h \in \Sigma$$

$$\gamma(X, h) \geq 0 \quad and \quad \sum_{X \in N, h \in \Sigma} \gamma(X, h) = 1$$

Probability of a left-most derivation $r_1, r_2, \ldots, r_N$ is given by:

Lexicalized PCFG Definition

L-PCFG: $G = (N, \Sigma, R, S, q, \gamma)$

$$q(r) \geq 0 \quad and \quad \sum_{r \in R : LHS(r) = X(h)} q(r) = 1 \quad \forall X \in N, h \in \Sigma$$

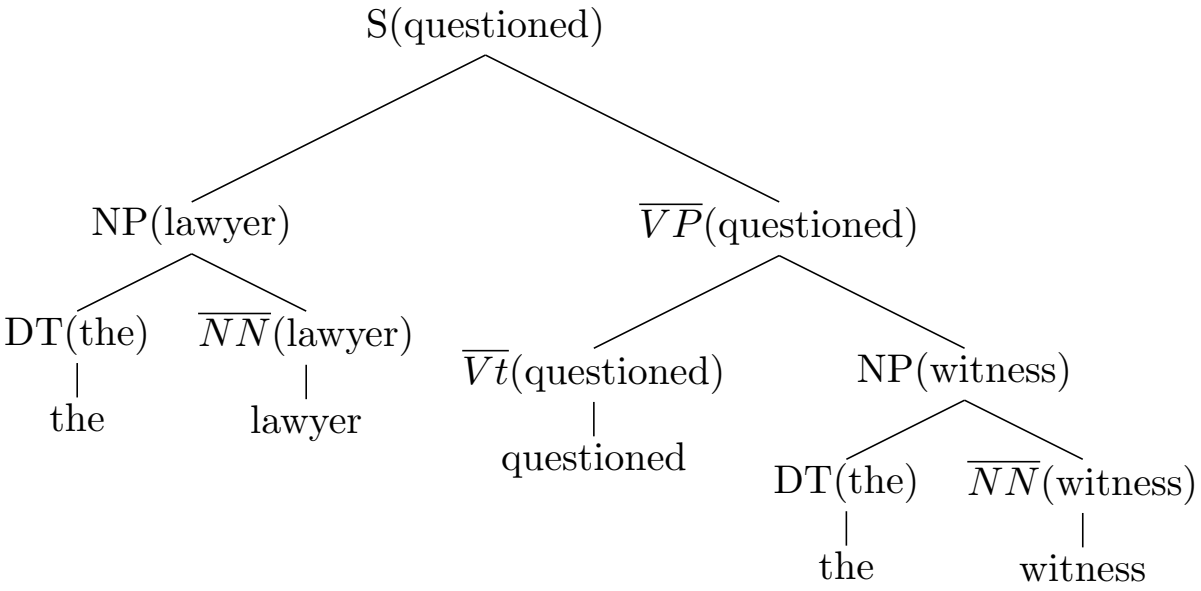
$$\gamma(X, h) \geq 0 \quad and \quad \sum_{X \in N, h \in \Sigma} \gamma(X, h) = 1$$

Probability of a left-most derivation $r_1, r_2, \ldots, r_N$ is given by:

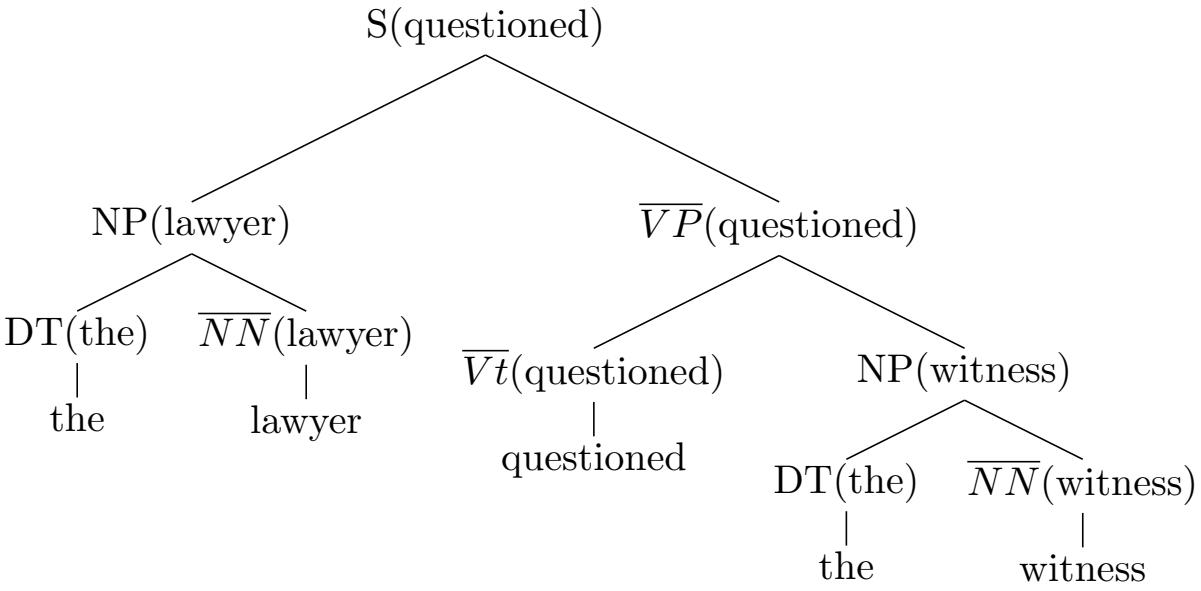
$$\gamma(LHS(r_1)) \times \prod_{i=1}^{N} q(r_i)$$

Lexicalized PCFG Definition

$$p(t) = ?$$



Lexicalized PCFG Definition



$$
\begin{aligned}
p(t) = \gamma(S, questioned) \times \\
q(S(questioned) \rightarrow_2 NP(lawyer)\ VP(questioned)) \times \\
q(NP(lawyer) \rightarrow_2 DT(the)\ NN(lawyer)) \times \\
q(DT(the) \rightarrow_2 the) \times \\
q(NN(lawyer) \rightarrow_2 lawyer) \times \\
q(VP(questioned) \rightarrow_2 Vt(questioned)\ NP(witness)) \times \\
q(NP(witness) \rightarrow_2 DT(the)\ NN(witness)) \times \\
q(DT(the) \rightarrow_2 the) \times \\
q(NN(witness) \rightarrow_2 witness)
\end{aligned}
$$

LEXICALIZED PCFG

$$\text{L-PCFG: } G = (N, \Sigma, R, S, q, \gamma)$$

$$q(r) \geq 0 \quad and \quad \sum_{r \in R: LHS(r) = X(h)} q(r) = 1 \quad \forall X \in N, h \in \Sigma$$

$$\gamma(X, h) \geq 0 \quad and \quad \sum_{X \in N, h \in \Sigma} \gamma(X, h) = 1$$

Two Questions:

1. ***Learning Problem*** : How do we learn the parameters (probabilities)?
2. ***Decoding Problem***: Given a sentence *s* how do we find the most likely tree?

$$\underset{t \in \mathcal{T}_G(s)}{\text{argmax}} \; p(t)$$

LEXICALIZED PCFG

$$\text{L-PCFG: } G = (N, \Sigma, R, S, q, \gamma)$$

$$q(r) \geq 0 \quad and \quad \sum_{r \in R: LHS(r) = X(h)} q(r) = 1 \quad \forall X \in N, h \in \Sigma$$

$$\gamma(X, h) \geq 0 \quad and \quad \sum_{X \in N, h \in \Sigma} \gamma(X, h) = 1$$

Two Questions:

1. ***Learning Problem*** : How do we learn the parameters (probabilities)?
2. ***Decoding Problem***: Given a sentence *s* how do we find the most likely tree?

$$\underset{t \in \mathcal{T}_G(s)}{\text{argmax}} \; p(t)$$

L-PCFG PARAMETER ESTIMATION

- The number of parameters is very large
- This can lead to sparsity problems

L-PCFG PARAMETER ESTIMATION

- The number of parameters is very large
- This can lead to sparsity problems

$$S(examined) \rightarrow_2 NP(lawyer)\ VP(examined)$$

L-PCFG PARAMETER ESTIMATION

- The number of parameters is very large
- This can lead to sparsity problems

$$S(examined) \rightarrow_2 NP(lawyer)\ VP(examined)$$

$$X = S$$

$$H = examined$$

$$R = S \rightarrow_2 NP\ VP$$

$$M = lawyer$$

L-PCFG PARAMETER ESTIMATION

$$S(examined) \rightarrow_2 NP(lawyer)\ VP(examined)$$

$$q(S(examined) \rightarrow_2 NP(lawyer)\ VP(examined))$$
$$= P(R = S \rightarrow_2 NP\ VP, M = lawyer \mid X = S, H = examined)$$

L-PCFG PARAMETER ESTIMATION

$$S(examined) \rightarrow_2 NP(lawyer)\ VP(examined)$$

$$\begin{aligned}
&q(S(examined) \rightarrow_2 NP(lawyer)\ VP(examined)) \\
&= P(R = S \rightarrow_2 NP\ VP, M = lawyer \mid X = S, H = examined) \\
&= P(M = lawyer \mid R = S \rightarrow_2 NP\ VP, X = S, H = examined) \\
&\quad \times P(R = S \rightarrow_2 NP\ VP \mid X = S, H = examined)
\end{aligned}$$

L-PCFG PARAMETER ESTIMATION

$$P(R = S \rightarrow_2 NP\ VP \mid X = S, H = examined)$$

L-PCFG PARAMETER ESTIMATION

$$P(R = S \rightarrow_2 NP\ VP \mid X = S, H = examined)$$

$$q_{ML}(S \rightarrow_2 NP\ VP \mid S, examined) = \frac{Count(R = S \rightarrow_2 NP\ VP, X = S, H = examined)}{Count(X = S, H = examined)}$$

L-PCFG PARAMETER ESTIMATION

$$P(R = S \rightarrow_2 NP\ VP \mid X = S, H = examined)$$

$$q_{ML}(S \rightarrow_2 NP\ VP \mid S, examined) = \frac{Count(R = S \rightarrow_2 NP\ VP, X = S, H = examined)}{Count(X = S, H = examined)}$$

$$q_{ML}(S \rightarrow_2 NP\ VP \mid S) = \frac{Count(R = S \rightarrow_2 NP\ VP, X = S)}{Count(X = S)}$$

L-PCFG PARAMETER ESTIMATION

$$P(R = S \rightarrow_2 NP\ VP \mid X = S, H = examined)$$

$$q_{ML}(S \rightarrow_2 NP\ VP \mid S, examined) = \frac{Count(R = S \rightarrow_2 NP\ VP, X = S, H = examined)}{Count(X = S, H = examined)}$$

$$q_{ML}(S \rightarrow_2 NP\ VP \mid S) = \frac{Count(R = S \rightarrow_2 NP\ VP, X = S)}{Count(X = S)}$$

$$P(R = S \rightarrow_2 NP\ VP \mid X = S, H = examined) =$$
$$\lambda_1 \times q_{ML}(S \rightarrow_2 NP\ VP \mid S, examined) + (1 - \lambda_1) \times q_{ML}(S \rightarrow_2 NP\ VP \mid S)$$

L-PCFG PARAMETER ESTIMATION

$$P(M = lawyer \mid R = S \rightarrow_2 NP\ VP, X = S, H = examined)$$

L-PCFG PARAMETER ESTIMATION

$$P(M = lawyer \mid R = S \rightarrow_2 NP\ VP, X = S, H = examined)$$

$$q_{ML}(lawyer \mid S \rightarrow_2 NP\ VP, examined) = \frac{Count(M = lawyer, R = S \rightarrow_2 NP\ VP, H = examined)}{Count(R = S \rightarrow_2 NP\ VP, H = examined)}$$

L-PCFG PARAMETER ESTIMATION

$$P(M = lawyer \mid R = S \rightarrow_2 NP\ VP, X = S, H = examined)$$

$$q_{ML}(lawyer \mid S \rightarrow_2 NP\ VP, examined) = \frac{Count(M = lawyer, R = S \rightarrow_2 NP\ VP, H = examined)}{Count(R = S \rightarrow_2 NP\ VP, H = examined)}$$

$$q_{ML}(lawyer \mid S \rightarrow_2 NP\ VP) = \frac{Count(M = lawyer, R = S \rightarrow_2 NP\ VP)}{Count(R = S \rightarrow_2 NP\ VP)}$$

L-PCFG PARAMETER ESTIMATION

$$P(M = lawyer \mid R = S \rightarrow_2 NP\ VP, X = S, H = examined)$$

$$q_{ML}(lawyer \mid S \rightarrow_2 NP\ VP, examined) = \frac{Count(M = lawyer, R = S \rightarrow_2 NP\ VP, H = examined)}{Count(R = S \rightarrow_2 NP\ VP, H = examined)}$$

$$q_{ML}(lawyer \mid S \rightarrow_2 NP\ VP) = \frac{Count(M = lawyer, R = S \rightarrow_2 NP\ VP)}{Count(R = S \rightarrow_2 NP\ VP)}$$

$$P(M = lawyer \mid R = S \rightarrow_2 NP\ VP, X = S, H = examined) =$$
$$\lambda_2 \times q_{ML}(lawyer \mid S \rightarrow_2 NP\ VP, examined) + (1 - \lambda_2) \times q_{ML}(lawyer \mid S \rightarrow_2 NP\ VP)$$

L-PCFG PARAMETER ESTIMATION

$$q(S(examined) \rightarrow_2 NP(lawyer)\ VP(examined)) =$$
$$\lambda_1 \times q_{ML}(S \rightarrow_2 NP\ VP \mid S, examined) + (1 - \lambda_1) \times q_{ML}(S \rightarrow_2 NP\ VP \mid S)$$
$$\lambda_2 \times q_{ML}(lawyer \mid S \rightarrow_2 NP\ VP, examined) + (1 - \lambda_2) \times q_{ML}(lawyer \mid S \rightarrow_2 NP\ VP)$$

LEXICALIZED PCFG

$$\text{L-PCFG: } G = (N, \Sigma, R, S, q, \gamma)$$

$$q(r) \geq 0 \quad and \quad \sum_{r \in R: LHS(r) = X(h)} q(r) = 1 \quad \forall X \in N, h \in \Sigma$$

$$\gamma(X, h) \geq 0 \quad and \quad \sum_{X \in N, h \in \Sigma} \gamma(X, h) = 1$$

Two Questions:

1. ***Learning Problem*** : How do we learn the parameters (probabilities)?
2. ***Decoding Problem***: Given a sentence *s* how do we find the most likely tree?

$$\underset{t \in \mathcal{T}_G(s)}{\text{argmax}} \; p(t)$$

L-PCFG DECODING

L-PCFG DECODING

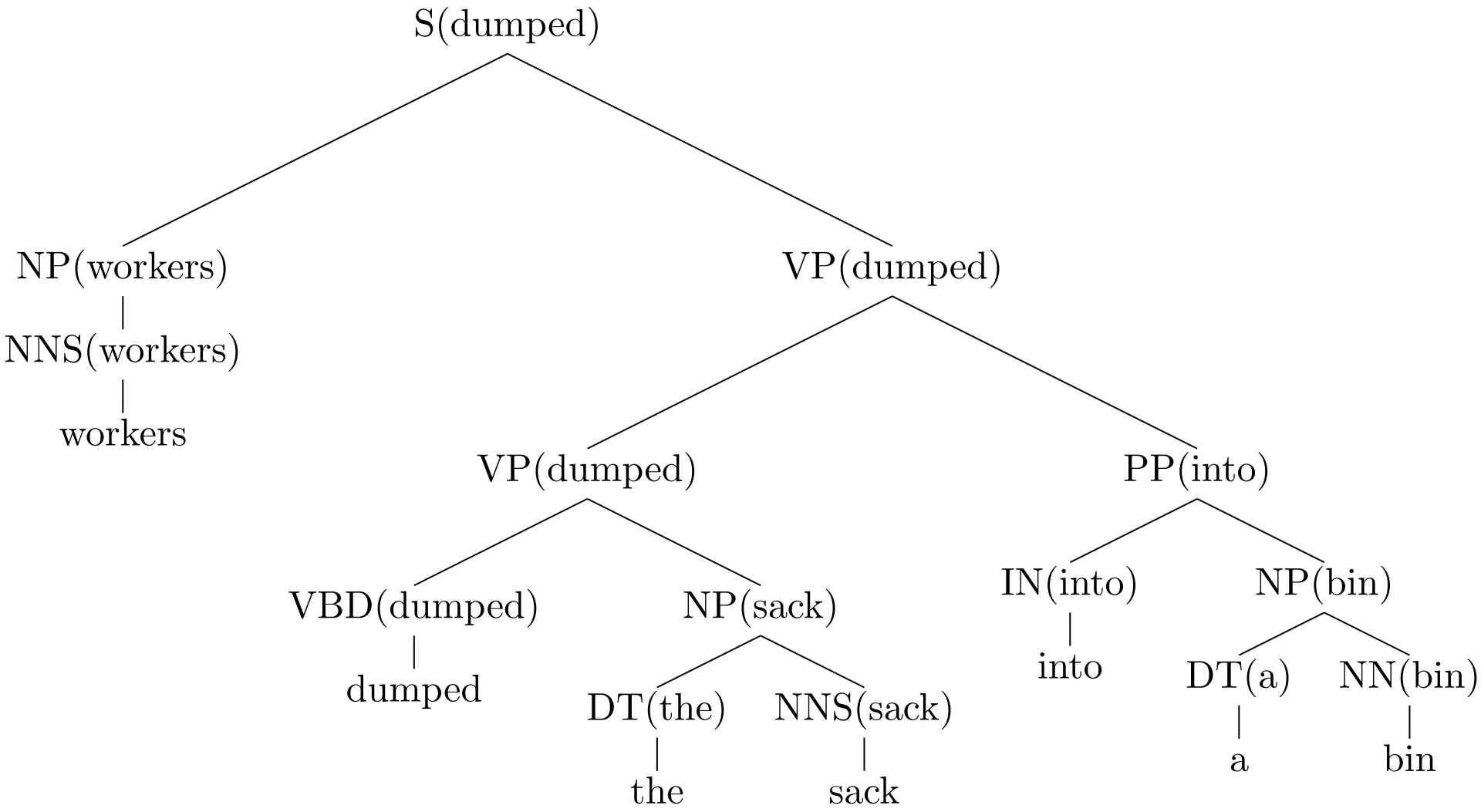
Dynamic Programming similar to the one for PCFG

L-PCFG DECODING

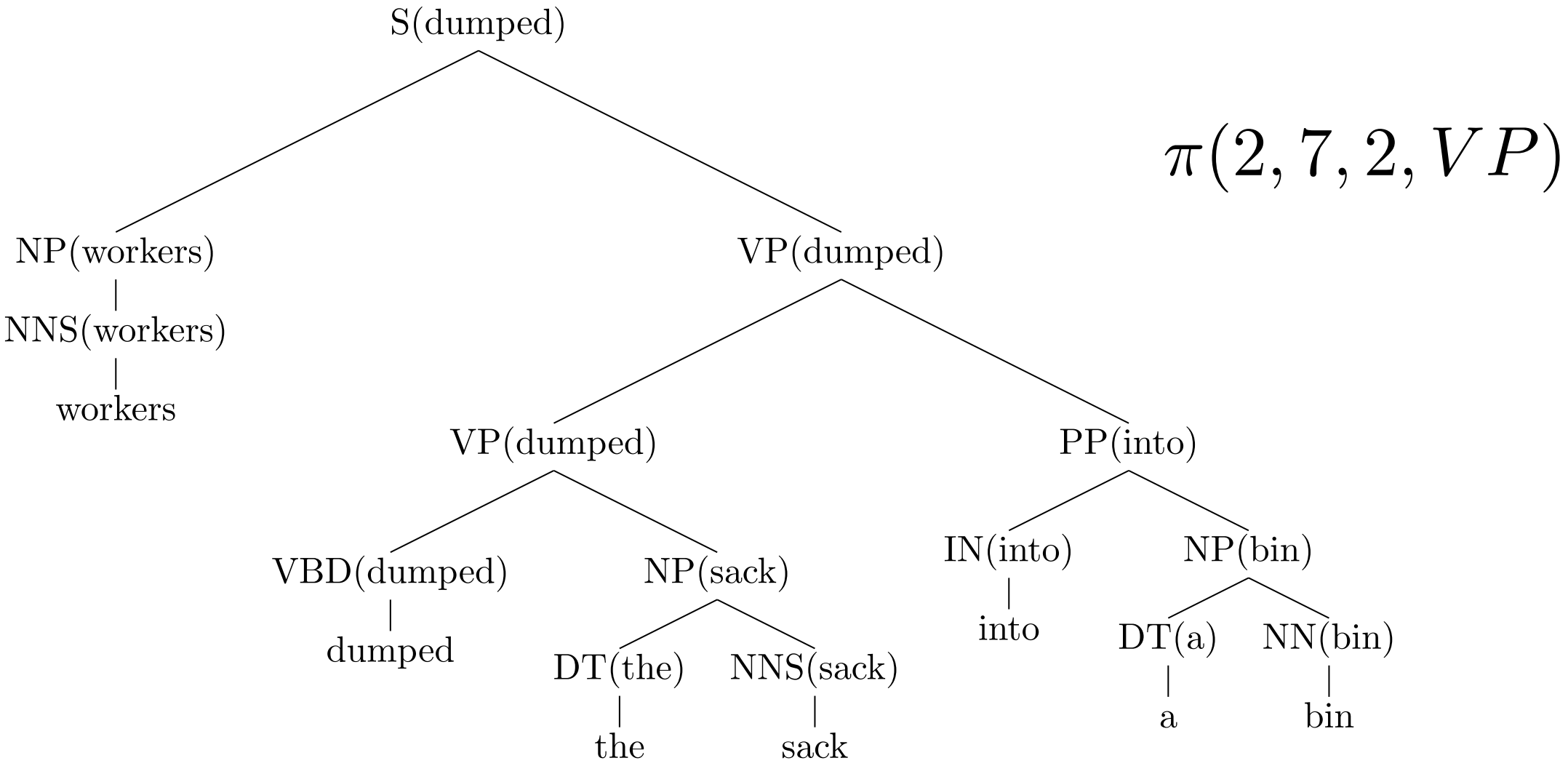
Dynamic Programming similar to the one for PCFG

$$\pi(i, j, h, X) = \text{Highest probability for a parse tree with non-terminal } X \text{ and lexical item } h \text{ at its root, spanning words } i \ldots j \text{ in the input}$$

L-PCFG DECODING



L-PCFG DECODING



$$\pi(2, 7, 2, VP)$$

CYK ALGORITHM FOR L-PCFG

Input: sentence $s = x_1 \dots x_n$ and $L - PCFG\ G = (N, \Sigma, R, S, q, \gamma)$

Initialization: $\forall\ 1 \leq i \leq n$ and $X \in N$

$$\pi(i, i, X) = \begin{cases} q(X \rightarrow x_i) & \text{If } X \rightarrow x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

Algorithm:

For $k = 1, \dots, (n-1)$

For $i = 1, \dots, (n-k)$

$j = i + k$

$\forall X \in N, h \in \{i \dots j\},$ Calculate $\pi(i, j, h, X)$ and $bp(i, j, h, X)$

end for

end for

Output: $(X^*, h^*) = \underset{S \in N, h \in \{1 \dots n\}}{\text{argmax}} \gamma(X, h) \times \pi(1, n, h, X)$

Obtain the highest probability parse tree using $bp(1, n, h^*, X^*)$

L-PCFG DECODING

Calculate: $\pi(i, j, h, X)$

1. $\pi(i, j, h, X) = 0$
2. For $s = h \ldots (j-1)$, for $m = (s+1) \ldots j$, for $X\,(x_h) \rightarrow_1 Y\,(x_h)\,Z\,(x_m) \in R$
   (a) $p = q\,(X\,(x_h) \rightarrow_1 Y\,(x_h)\,Z\,(x_m)) \times \pi(i, s, h, Y) \times \pi(s+1, j, m, Z)$
   (b) If $p > \pi(i, j, h, X)$

$$\pi(i, j, h, X) = p$$
$$bp(i, j, h, X) = \langle s, m, Y, Z \rangle$$

3. For $s = i \ldots (h-1)$, for $m = i \ldots s$, for $X\,(x_h) \rightarrow_2 Y\,(x_m)\,Z\,(x_h) \in R$
   (a) $p = q\,(X\,(x_h) \rightarrow_2 Y\,(x_m)\,Z\,(x_h)) \times \pi(i, s, m, Y) \times \pi(s+1, j, h, Z)$
   (b) If $p > \pi(i, j, h, X)$

$$\pi(i, j, h, X) = p$$
$$bp(i, j, h, X) = \langle s, m, Y, Z \rangle$$

Summary

- PCFG suffer from lack of sensitivity to lexical information and lack of sensitivity to structural preferences
- Solution: L-PCFG $G = (N, \Sigma, R, S, q, \gamma)$
- L-PCFG similar to PCFGs except lexical information included with every non-terminal
- Parameters can be obtained by lexicalizing the treebank and using smoothing techniques
- Most probable tree can be found using CKY algorithm

References

1. Michael Collin's NLP Lecture Notes: http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/lexpcfgs.pdf

2. Chapter 13, Speech and Language Processing, Dan Jurafsky and James Martin