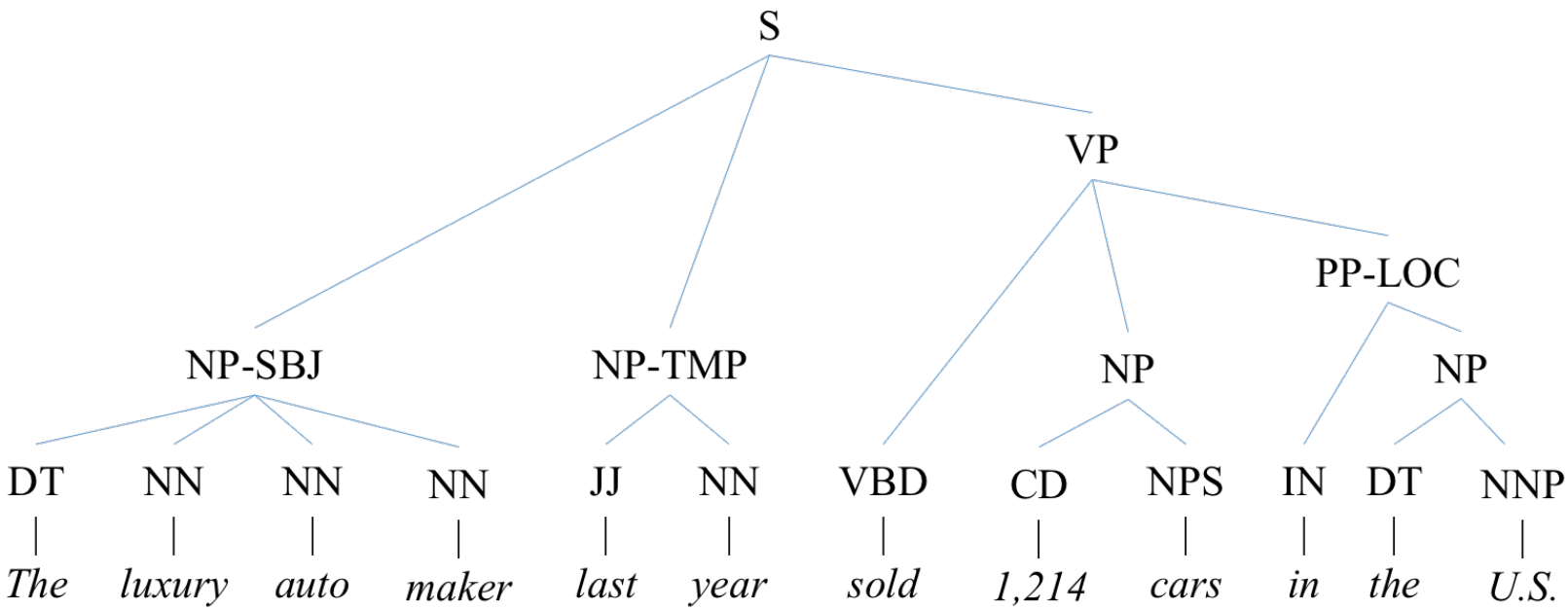
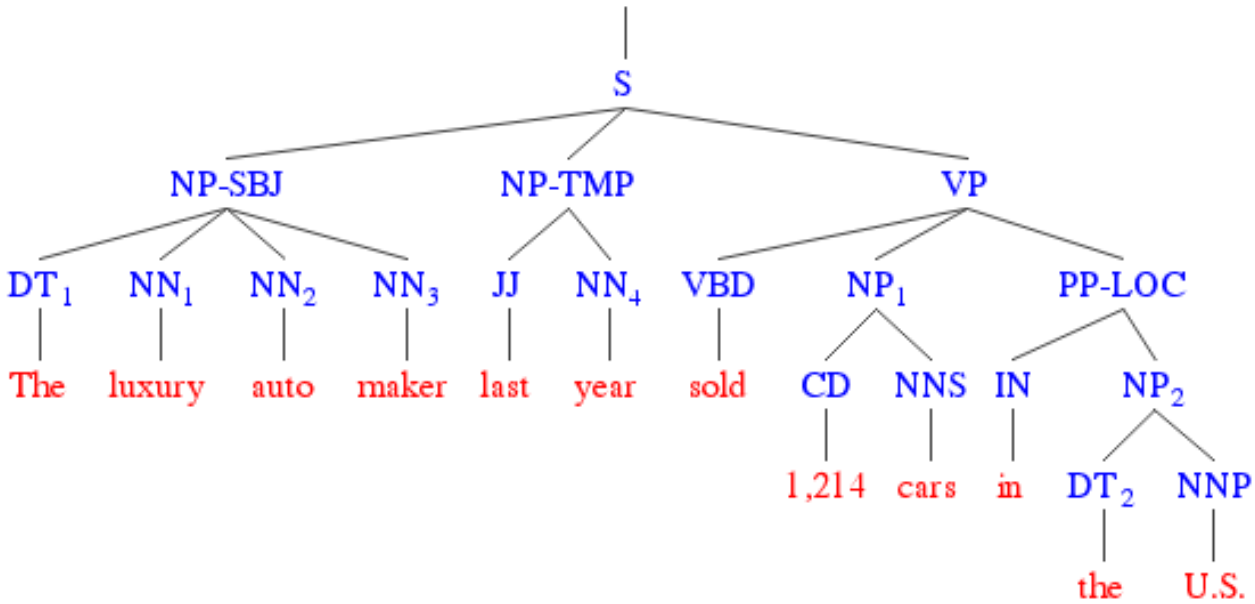
LING439/539 - Statistical NLP
Probabilistic context-free grammars

Monday, September 29 2016

Probabilistic context-free grammars



Tree drawing...




```
[ [S
      [NP-SBJ [DT The] [NN luxury] [NN auto] [NN maker] ]
      [NP-TMP [JJ last] [NN year] ]
      [VP [VBD sold]
            [NP [CD 1,214] [NNS cars] ]
            [PP-LOC [IN in]
                  [NP [DT the] [NNP U.S.] ]]]]
```

http://ironcreek.net/phpsyntaxtree

Treebanks as grammars

$$
\begin{aligned}
\text{S} &\rightarrow \text{NP-SBJ NP-TMP VP} \\
\text{NP-SBJ} &\rightarrow \text{DT NN NN NN} \\
\text{NP-TMP} &\rightarrow \text{JJ NN} \\
\text{VP} &\rightarrow \text{VBD NP PP-LOC} \\
\text{NP} &\rightarrow \text{CD NNS} \\
\text{PP-LOC} &\rightarrow \text{IN NP} \\
\text{NP} &\rightarrow \text{DT NNP} \\
&\ldots
\end{aligned}
$$

Treebanks as grammars

S	$\rightarrow$	NP-SBJ NP-TMP VP
NP-SBJ	$\rightarrow$	DT NN NN NN
NP-TMP	$\rightarrow$	JJ NN
VP	$\rightarrow$	VBD NP PP-LOC
NP	$\rightarrow$	CD NNS
PP-LOC	$\rightarrow$	IN NP
NP	$\rightarrow$	DT NNP
DT	$\rightarrow$	*The*
NN	$\rightarrow$	*luxury*
NN	$\rightarrow$	*auto*
NN	$\rightarrow$	*maker*
...		

Context-free grammar

$$A \quad \rightarrow \quad \gamma$$

where $A \in V$, and $\gamma \in (V \cup \Sigma)^*$

Context-free grammar: $S \to aSa$
$S \to bSb$
$S \to \epsilon$

Context-free language: $S \Rightarrow aSa \Rightarrow aaSaa \Rightarrow aabSbaa \Rightarrow aabbaa$

$L(G) = \{ww^R : w \in \{a, b\}^*\}$

Chomsky normal form (CNF)

A grammar where every production is either of the form

$A \to BC$
or
$A \to c$

where $A, B, C \in V$ and $c \in \Sigma$.

Converting a CFG to the Chomsky normal form

1. START: Eliminate the start symbol from right-hand sides
   - Introduce a new start symbol S_0, and a new rule $S_0 \to S$
   - where S is the previous start symbol. This doesn't change the grammar's produced language, and S_0 won't occur on any rule's right-hand side.

2. TERM: Eliminate rules with nonsolitary terminals
   - To eliminate each rule $A \to X_1 ... \ a \ ... X_n$ with a terminal symbol a being not the only symbol on the right-hand side,
   - introduce a new nonterminal symbol N_a for every such terminal, $A \to X_1 ... N_a ... X_n$
   - and a new rule $N_a \to a$

3. BIN: Eliminate right-hand sides with more than 2 nonterminals
   - Replace each rule $A \to X_1 X_2 ... X_n$ with more than 2 nonterminals $X_1, ..., X_n$ by rules
   - $A \to X_1 A_1$,
   - $A_1 \to X_2 A_2$,
   - ... ,
   - $A_{n-2} \to X_{n-1} X_n$ where A_i are new nonterminal symbols.

4. UNIT: Eliminate unit rules
   - A unit rule is a rule of the form $A \to B$ where A, B are nonterminal symbols, and
   - $B \to X_1 ... X_n$ where $X_1 ... X_n$ is a string of nonterminals and terminals,
   - remove them and add rule $A \to X_1 ... X_n$ unless this is a unit rule which has already been removed.

5. DEL: Eliminate ϵ-rules
   - An ϵ-rule is a rule of the form $A \to \epsilon$, where A is not the grammar's start symbol.
   - To eliminate all rules of this form, first determine the set of all nonterminals that derive ϵ (nonterminals nullable).
     - If a rule $A \to \epsilon$ exists, then A is nullable.
     - If a rule $A \to X_1 ... X_n$ exists, and each X_i is nullable, then A is nullable, too.
   - Obtain an intermediate grammar by replacing each rule $A \to X_1 ... X_n$ by all versions with some nullable X_i omitted. By deleting in this grammar each ϵ-rule, unless its left-hand side is the start symbol, the transformed grammar is obtained.

For example, in the following grammar, with start symbol S_0,

- $S_0 \to AbB|C$
- $B \to AA|AC$
- $C \to b|c$
- $A \to a|\epsilon$

where the nonterminal A, and hence also B, is nullable, while neither C nor S_0 is.

- $S_0 \to AbB|C$
  - $S_0 \to AbB|\not{A}bB|Ab\not{B}|\not{A}b\not{B}|C$
- $B \to AA|AC$
  - $B \to AA|\not{A}A|A\not{A}|\not{A}\not{A}|AC|\not{A}C$
- $C \to b|c$
- $A \to a|\epsilon$
  - $A \to a|\not{\epsilon}$

where the nonterminal A, and hence also B, is nullable, while neither C nor S_0 is.

- $S_0 \to AbB|Ab|bB|b|C$
- $B \to AA|A|AC|C$
- $C \to b|c$
- $A \to a$

Exercise

Consider the CFG:

- $S \to aXbX$
- $X \to aY|bY|\epsilon$
- $Y \to X|c$

Which one is nullable?

DEL:

- $S \to aXbX$
  - $S \to aXbX|a\cancel{X}bX|aXb\cancel{X}|a\cancel{X}b\cancel{X}$
- $X \to aY|bY|\epsilon$
  - $X \to aY|a\cancel{Y}|bY|b\cancel{Y}|\cancel{\epsilon}$
- $Y \to X|c$

Remove Unit $Y \to X$ (UNIT)

- $S \to aXbX|abX|aXb|ab$
- $X \to aY|a|bY|b$
- $Y \to X|c$
  - $Y \to aY|a|bY|b|c$

Add $A \to a$, $B \to b$, $C \to c$ (TERM)

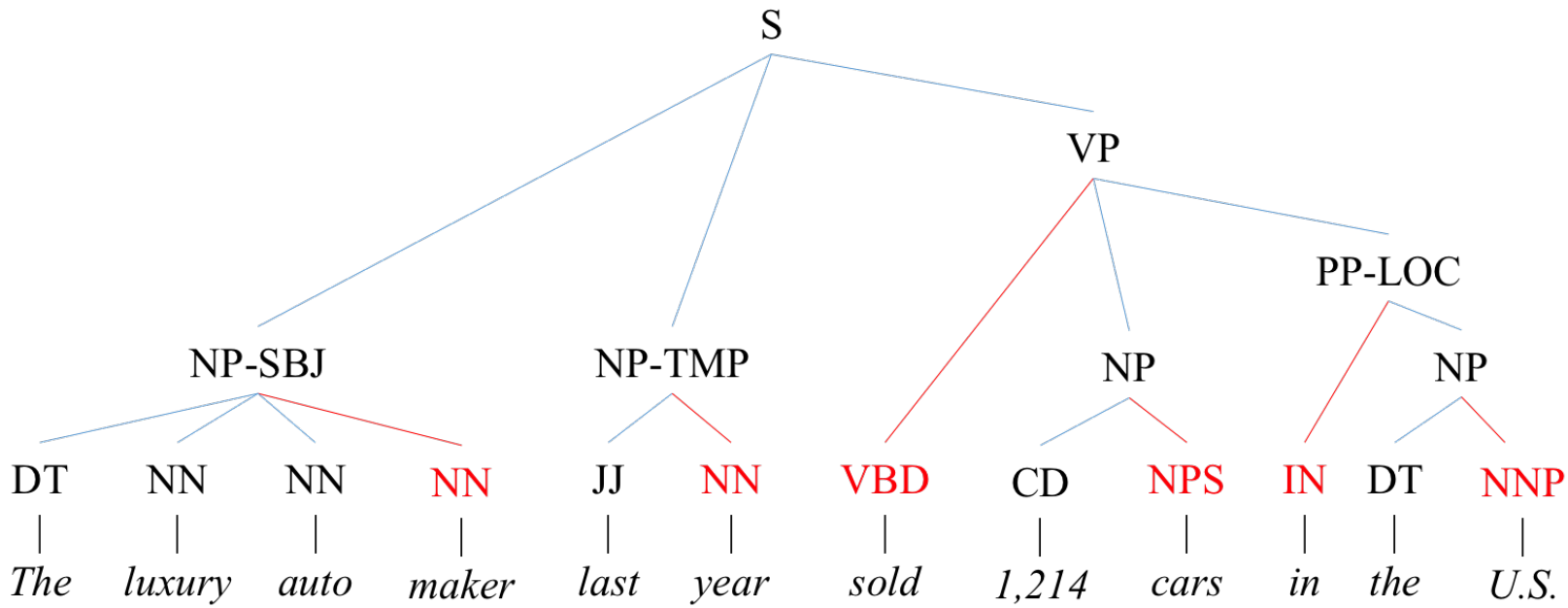
- $S \to aXbX|abX|aXb|ab$
  - $S \to AXBX|ABX|AXB|AB$
- $X \to aY|a|bY|b$
  - $X \to AY|A|BY|B$
- $Y \to aY|a|bY|b|c$
  - $Y \to AY|A|BY|B|C$
- $A \to a$
- $B \to b$
- $C \to c$

BIN:

- $S \to \cancel{AX}MB\cancel{X}N|AB\cancel{X}N|\cancel{AX}MB|AB$
  - $M \to AX$
  - $N \to BX$
- $X \to AY|A|BY|B$
- $Y \to AY|A|BY|B|C$
- $A \to a$
- $B \to b$
- $C \to c$

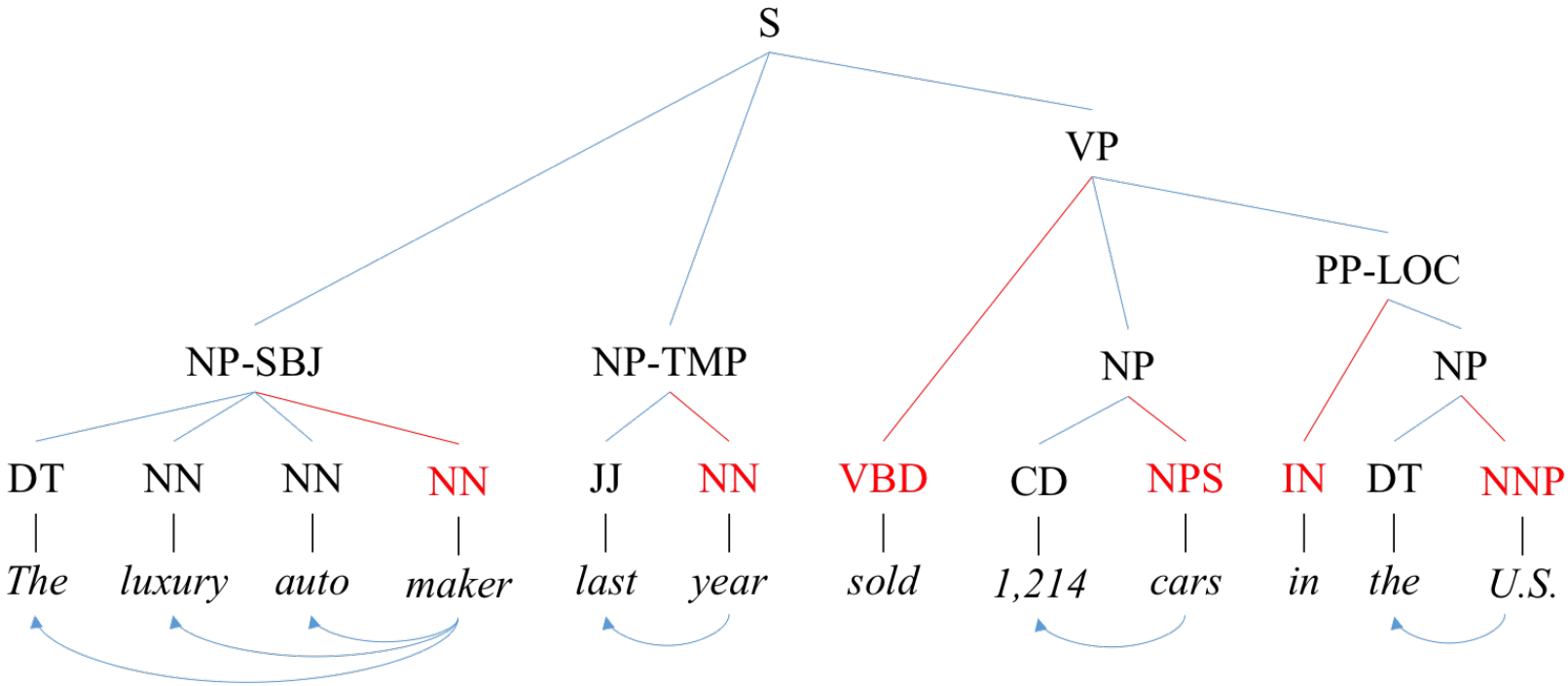
- $S \to MN|AN|MB|AB$
- $M \to AX$
- $N \to BX$
- $X \to AY|A|BY|B$
- $Y \to AY|A|BY|B|C$
- $A \to a$
- $B \to b$
- $C \to c$

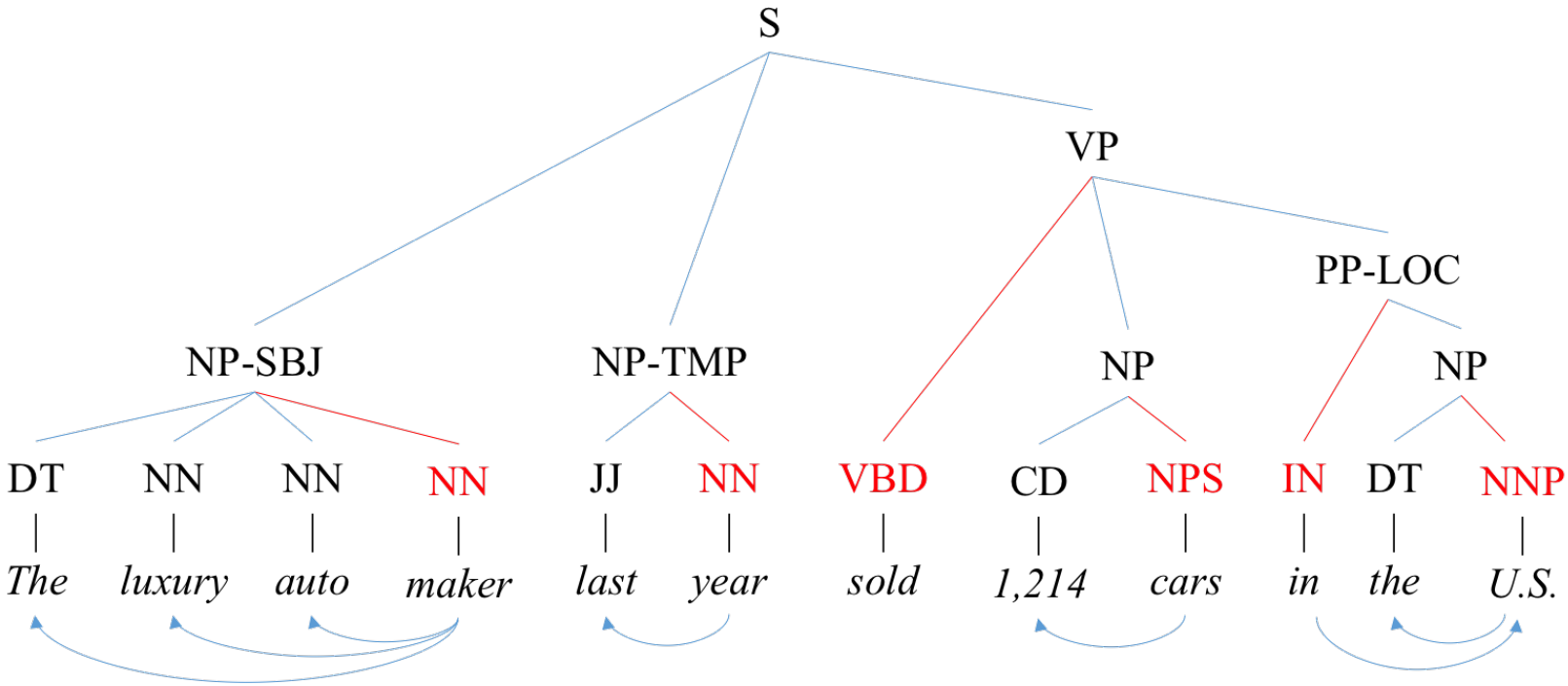
Head and dependencies

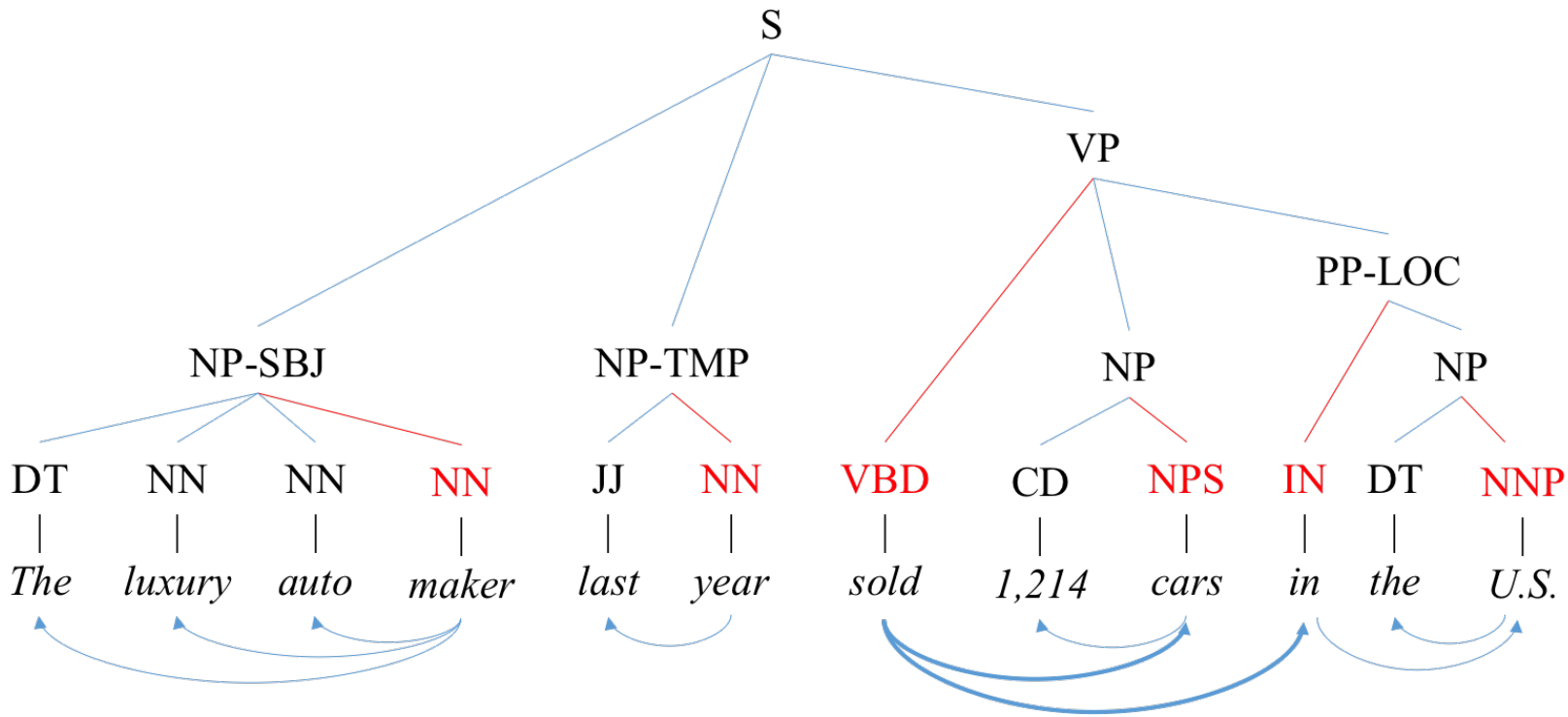


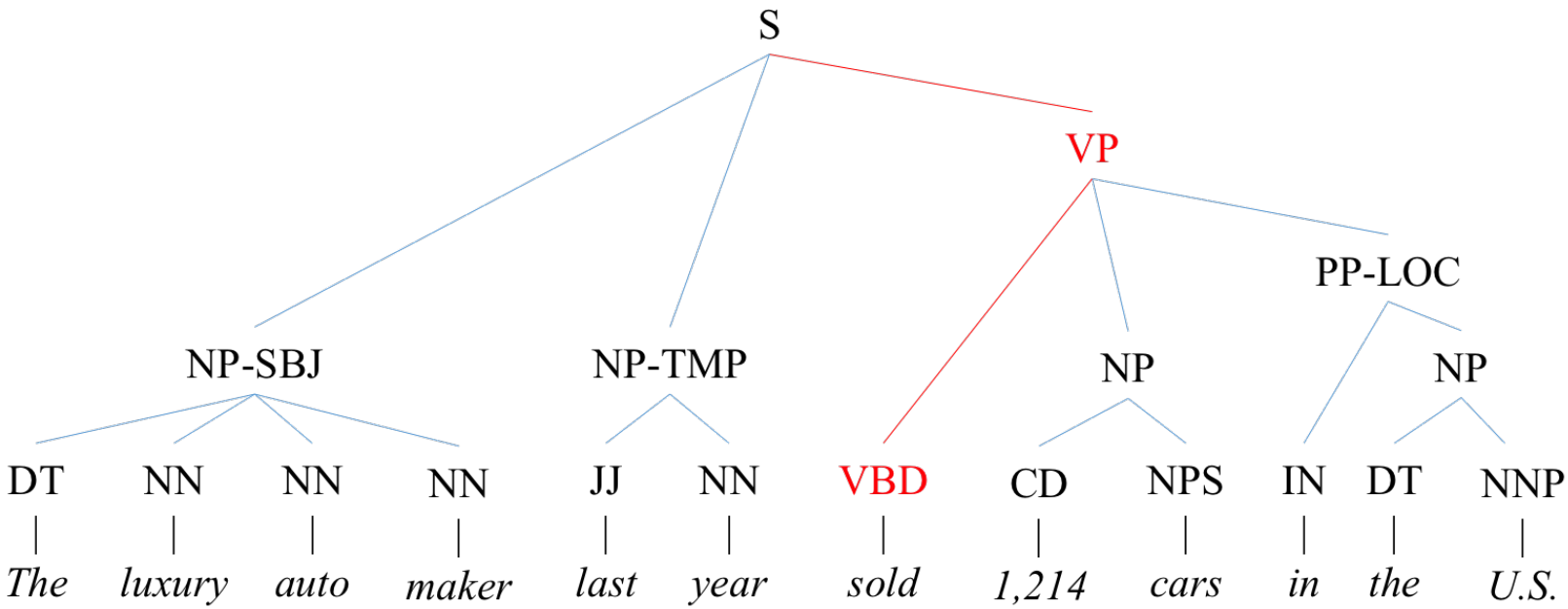
Head percolation

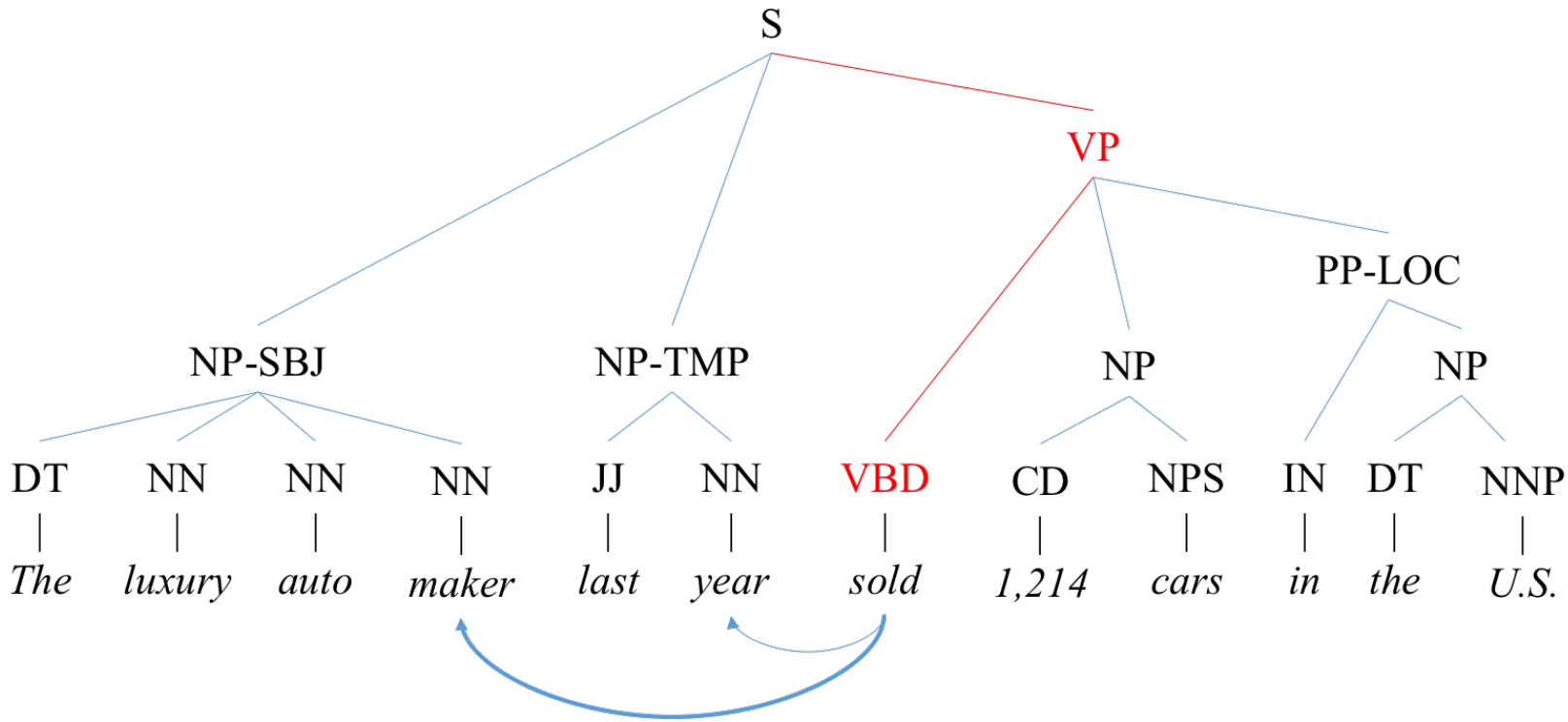
Michael Collins. *Head-Driven Statistical Models for Natural Language Parsing*. PhD Dissertation, University of Pennsylvania, 1999.


S
VP
PP-LOC
NP-SBJ
NP-TMP
NP
NP
DT
NN
NN
NN
JJ
NN
VBD
CD
NPS
IN
DT
NNP
The
luxury
auto
maker
last
year
sold
1,214
cars
in
the
U.S.

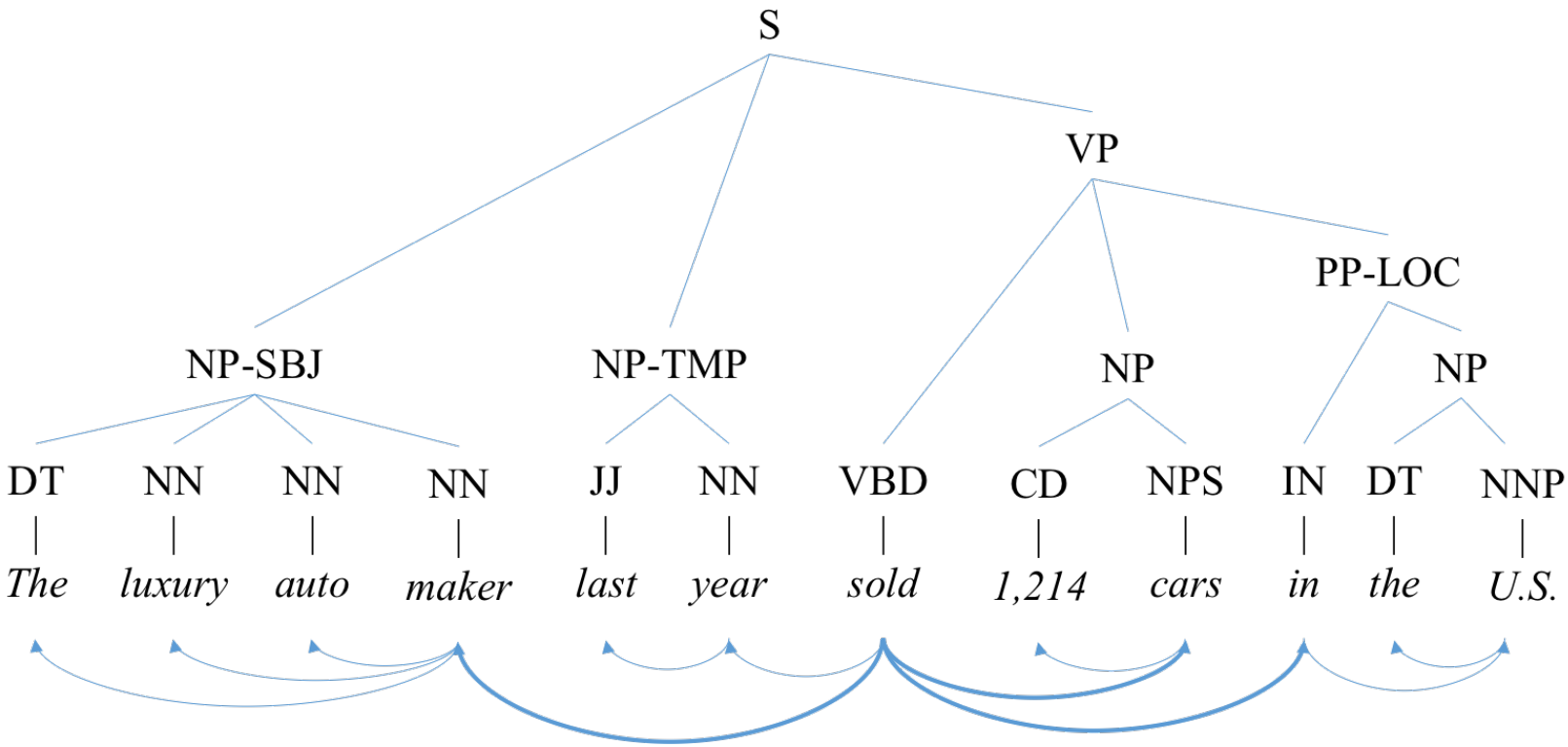

S
VP
PP-LOC
NP-SBJ
NP-TMP
NP
NP
DT
NN
NN
NN
JJ
NN
VBD
CD
NPS
IN
DT
NNP
The
luxury
auto
maker
last
year
sold
1,214
cars
in
the
U.S.


S
VP
PP-LOC
NP-SBJ
NP-TMP
NP
NP
DT
NN
NN
NN
JJ
NN
VBD
CD
NPS
IN
DT
NNP
The
luxury
auto
maker
last
year
sold
1,214
cars
in
the
U.S.

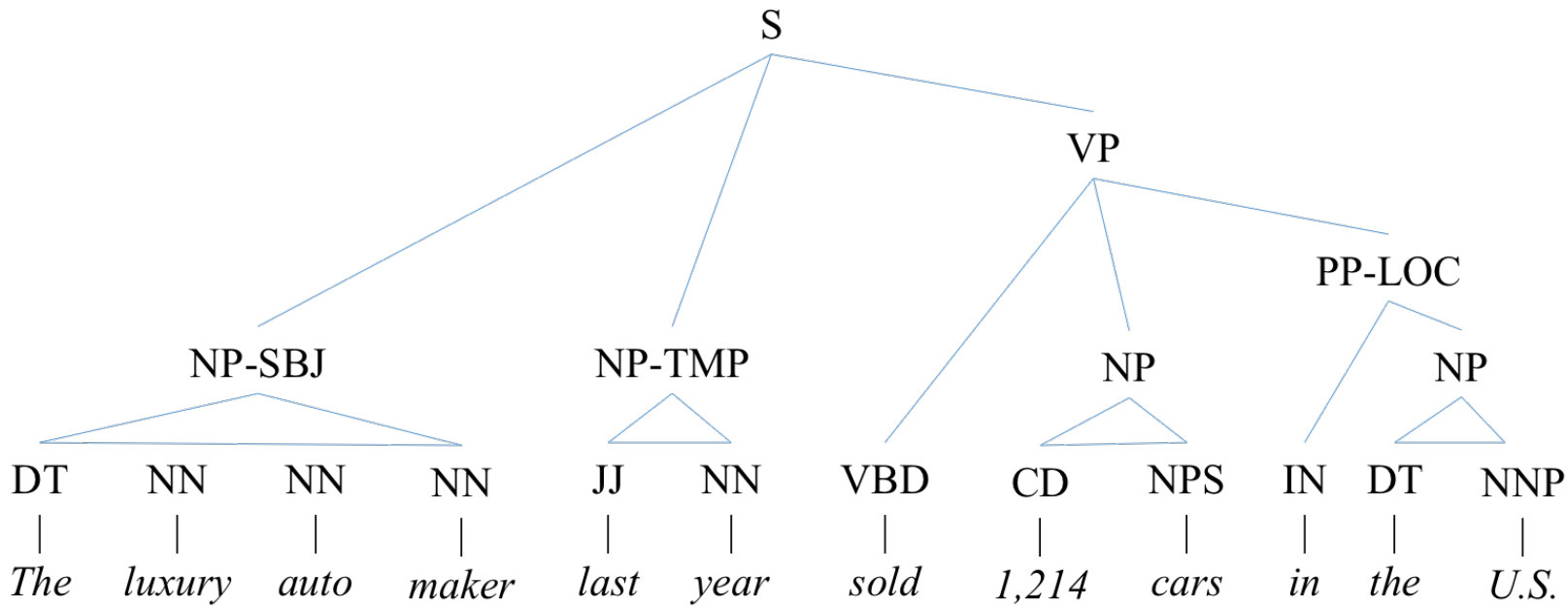

S
VP
PP-LOC
NP-SBJ
NP-TMP
NP
NP
DT
NN
NN
NN
JJ
NN
VBD
CD
NPS
IN
DT
NNP
The
luxury
auto
maker
last
year
sold
1,214
cars
in
the
U.S.


S
VP
PP-LOC
NP-SBJ
NP-TMP
NP
NP
DT
NN
NN
NN
JJ
NN
VBD
CD
NPS
IN
DT
NNP
The
luxury
auto
maker
last
year
sold
1,214
cars
in
the
U.S.

Dependencies



Chunking: Shallow parsing



Abney, Steven (1991), Parsing By Chunks. *Principle-Based Parsing* Kluwer Academic Publishers, pp. 257–278.