CS 6501 Natural Language Processing

Probabilistic Context-free Grammars

Yangfeng Ji

October 1, 2019

Department of Computer Science
University of Virginia

UNIVERSITY of VIRGINIA | ENGINEERING

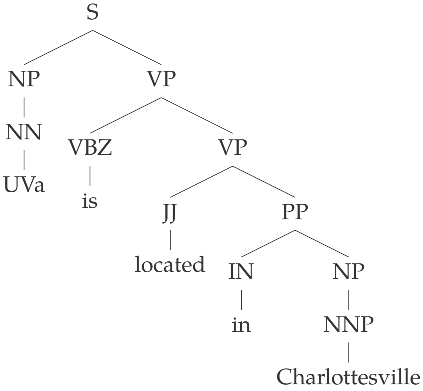
Overview

1. Introduction
2. Context-Free Grammars
3. Ambiguity
4. Probabilistic CFGs

Introduction

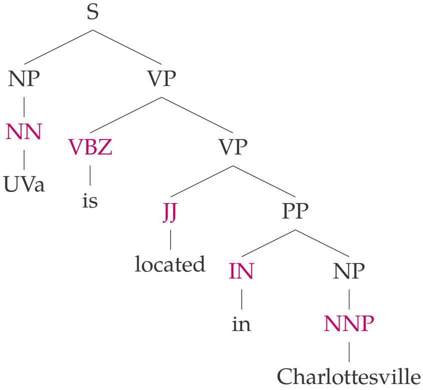
Syntactic Parse Tree

- Input: `UVa is located in Charlottesville`
- Output:



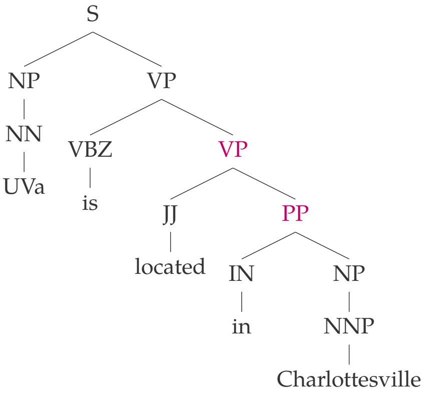
Information Conveyed by Parse Trees

Part of speech for each word



Information Conveyed by Parse Trees (II)

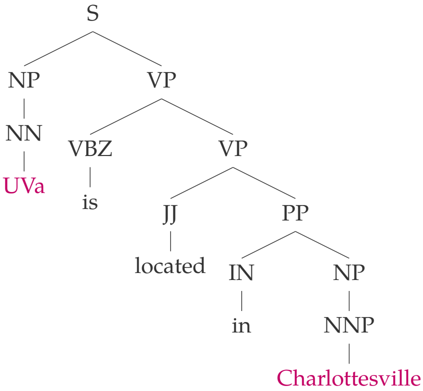
Phrase



- PP: *in Charlottesville*
- VP: *located in Charlottesville*

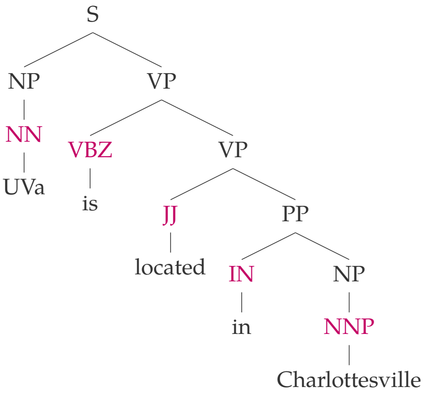
Information Conveyed by Parse Trees (III)

Useful relationships



Relationship between UVa and Charlottesville

Example Applications (I): Question answering



- Question answering: *where is UVa?*

Example Applications (II): Machine translation

- English word order: SUBJECT - VERB - OBJECT
- Japanese word order: SUBJECT - OBJECT - VERB

Example Applications (II): Machine translation

- English word order: SUBJECT - VERB - OBJECT
- Japanese word order: SUBJECT - OBJECT - VERB

Example I

- English: `IBM bought Lotus`
- Japanese: `IBM Lotus bought`

Example Applications (II): Machine translation

- English word order: SUBJECT - VERB - OBJECT
- Japanese word order: SUBJECT - OBJECT - VERB

Example I

- English: `IBM bought Lotus`
- Japanese: `IBM Lotus bought`

Example II

- English: `Sources said that IBM bought Lotus yesterday`
- Japanese: `Sources yesterday IBM Lotus bought that said`

[Collins, 2017]

Example Applications (III)

Other applications:

- Grammar checking
- Dialogue understanding
- Text generation

Context-Free Grammars

Formal Definition

A context free grammar $G = (N, \Sigma, R, S)$ where

- N: a set of non-terminal symbols
- $S \in N$: a distinguished start symbol
- Σ: a set of terminal symbols
- R: a set of rules of the form $X \rightarrow Y_1 Y_2 \cdots Y_n$ for $n \geq 0$, $X \in N$, $Y_i \in N \cup \Sigma$

[Collins, 2017]

A Context-Free Grammar for English

- $N = \{\mathrm{S, NP, VP, PP, DT, Vi, Vt, NN, IN}\}$
- $S = \mathrm{S}$
- $\Sigma =$ {sleeps, saw, man, woman, telescope, the, with, in}
- R[1]

S	→	NP	VP
VP	→	Vi	
VP	→	Vt	NP
VP	→	VP	PP
NP	→	DT	NN
NP	→	NP	PP
PP	→	IN	NP

Vi	→	sleeps
Vt	→	saw
NN	→	man
NN	→	woman
NN	→	telescope
DT	→	the
IN	→	with
IN	→	in

[1] S=sentence, VP=verb phrase, NP=noun phrase, PP=prepositional phrase, DT=determiner, Vi=intransitive verb, Vt=transitive verb, NN=noun, IN=preposition

(Left-Most) Derivations

S	→	NP	VP
VP	→	Vi	
VP	→	Vt	NP
VP	→	VP	PP
NP	→	DT	NN
NP	→	NP	PP
PP	→	IN	NP

Vi	→	sleeps
Vt	→	saw
NN	→	man
NN	→	woman
NN	→	telescope
DT	→	the
IN	→	with
IN	→	in

- Left-most derivation: always pick the left-most non-terminal symbol for replacement

An Example

S	→	NP	VP
VP	→	Vi	
VP	→	Vt	NP
VP	→	VP	PP
NP	→	DT	NN
NP	→	NP	PP
PP	→	IN	NP

Vi	→	sleeps
Vt	→	saw
NN	→	man
NN	→	woman
NN	→	telescope
DT	→	the
IN	→	with
IN	→	in

Derivation	Rules used
S	S → NP VP
NP VP	

An Example

S	→	NP	VP
VP	→	Vi	
VP	→	Vt	NP
VP	→	VP	PP
NP	→	DT	NN
NP	→	NP	PP
PP	→	IN	NP

Vi	→	sleeps
Vt	→	saw
NN	→	man
NN	→	woman
NN	→	telescope
DT	→	the
IN	→	with
IN	→	in

Derivation	Rules used
S	S → NP VP
NP VP	NP → DT NN
DT NN VP	

An Example

S	→	NP	VP
VP	→	Vi	
VP	→	Vt	NP
VP	→	VP	PP
NP	→	DT	NN
NP	→	NP	PP
PP	→	IN	NP

Vi	→	sleeps
Vt	→	saw
NN	→	man
NN	→	woman
NN	→	telescope
DT	→	the
IN	→	with
IN	→	in

Derivation	Rules used
S	S → NP VP
NP VP	NP → DT NN
DT NN VP	DT → the
the NN VP	

An Example

S	→	NP	VP
VP	→	Vi	
VP	→	Vt	NP
VP	→	VP	PP
NP	→	DT	NN
NP	→	NP	PP
PP	→	IN	NP

Vi	→	sleeps
Vt	→	saw
NN	→	man
NN	→	woman
NN	→	telescope
DT	→	the
IN	→	with
IN	→	in

Derivation	Rules used
S	S → NP VP
NP VP	NP → DT NN
DT NN VP	DT → the
the NN VP	NN → man
the man VP	

An Example

S	→	NP	VP
VP	→	Vi	
VP	→	Vt	NP
VP	→	VP	PP
NP	→	DT	NN
NP	→	NP	PP
PP	→	IN	NP

Vi	→	sleeps
Vt	→	saw
NN	→	man
NN	→	woman
NN	→	telescope
DT	→	the
IN	→	with
IN	→	in

Derivation	Rules used
S	S → NP VP
NP VP	NP → DT NN
DT NN VP	DT → the
the NN VP	NN → man
the man VP	VP → Vi
the man Vi	

An Example

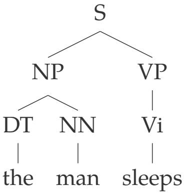
S	→	NP	VP
VP	→	Vi	
VP	→	Vt	NP
VP	→	VP	PP
NP	→	DT	NN
NP	→	NP	PP
PP	→	IN	NP

Vi	→	sleeps
Vt	→	saw
NN	→	man
NN	→	woman
NN	→	telescope
DT	→	the
IN	→	with
IN	→	in

Derivation	Rules used
S	S → NP VP
NP VP	NP → DT NN
DT NN VP	DT → the
the NN VP	NN → man
the man VP	VP → Vi
the man Vi	Vi → sleeps
the man sleeps	

From Derivations to Parse Tree

Derivation	Rules used
S	S → NP VP
NP VP	NP → DT NN
DT NN VP	DT → the
the NN VP	NN → man
the man VP	VP → Vi
the man Vi	Vi → sleeps
the man sleeps	



Properties of CFGs

- A CFG defines a set of possible derivations

Properties of CFGs

- A CFG defines a set of possible derivations
- A string $s \in \Sigma^*$ is in the language defined by the CFG if there is at least one derivation that yield s

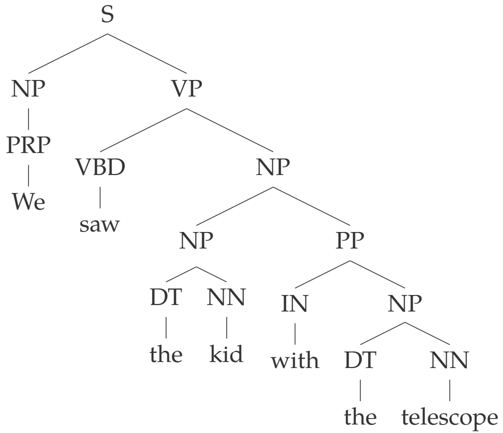
Properties of CFGs

- A CFG defines a set of possible derivations
- A string $s \in \Sigma^*$ is in the language defined by the CFG if there is at least one derivation that yield s
- **Ambiguity**: each string in the language generated by the CFG may have more than one derivation

Ambiguity

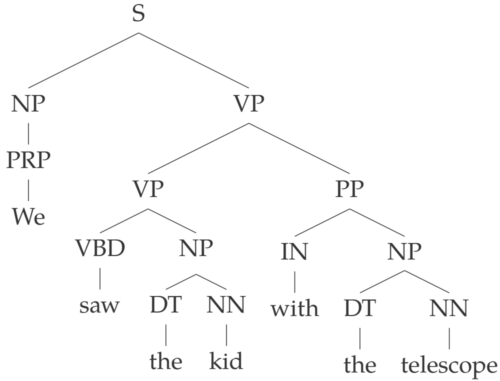
An Example of Ambiguity

Sentence: `We saw the kid with the telescope`



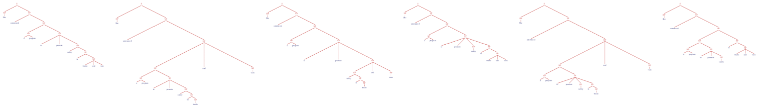
An Example of Ambiguity (II)

Sentence: `We saw the kid with the telescope`



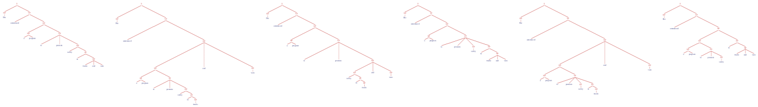
Problem with Parsing: Ambiguity

Sentence: `She announced a program to promote safety in trucks and vans`



Problem with Parsing: Ambiguity

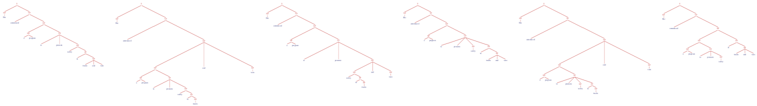
Sentence: She announced a program to promote safety in trucks and vans



- She announced a program to promote (safety in (trucks and vans))
- She announced a program to promote ((safety in trucks) and (vans))

Problem with Parsing: Ambiguity

Sentence: She announced a program to promote safety in trucks and vans

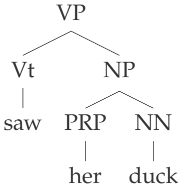


- She announced a program to promote (safety in (trucks and vans))
- She announced a program to promote ((safety in trucks) and (vans))
- She announced ((a program to promote safety in trucks) and (vans))

Sources of Ambiguity (I)

Part-of-Speech ambiguity

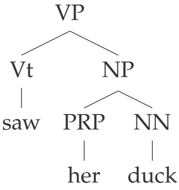
- NN → duck

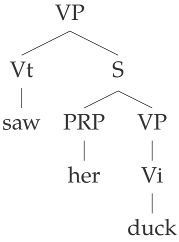


Sources of Ambiguity (I)

Part-of-Speech ambiguity

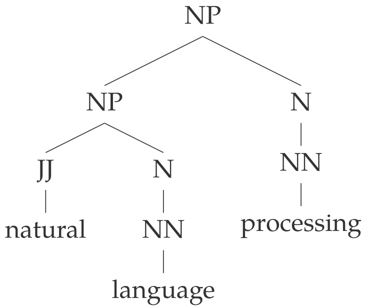
- NN → duck
- Vi → duck

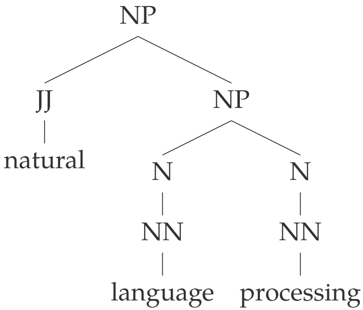




Sources of Ambiguity (II)

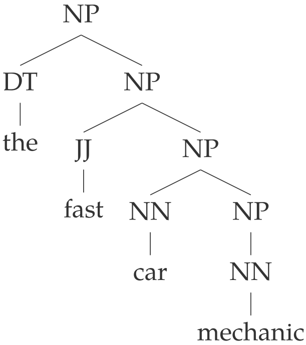
Noun premodifiers

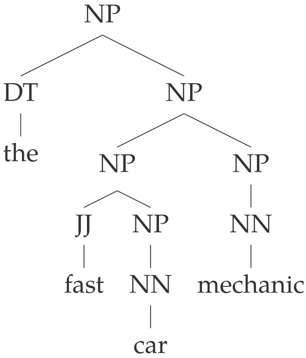




Sources of Ambiguity (III)

Noun premodifiers





Probabilistic CFGs

A Probabilistic Context-Free Grammar (PCFG)

- N: a set of non-terminal symbols
- $S \in N$: a distinguished start symbol
- Σ: a set of terminal symbols
- R:

S	$\Rightarrow$	NP	VP	1.0
VP	$\Rightarrow$	Vi		0.4
VP	$\Rightarrow$	Vt	NP	0.4
VP	$\Rightarrow$	VP	PP	0.2
NP	$\Rightarrow$	DT	NN	0.3
NP	$\Rightarrow$	NP	PP	0.7
PP	$\Rightarrow$	P	NP	1.0

Vi	$\Rightarrow$	sleeps	1.0
Vt	$\Rightarrow$	saw	1.0
NN	$\Rightarrow$	man	0.7
NN	$\Rightarrow$	woman	0.2
NN	$\Rightarrow$	telescope	0.1
DT	$\Rightarrow$	the	1.0
IN	$\Rightarrow$	with	0.5
IN	$\Rightarrow$	in	0.5

Probability of a Tree

The probability of a tree $\boldsymbol{t}$ with rules

$$\alpha_1 \rightarrow \beta_1, \alpha_2 \rightarrow \beta_2, \ldots, \alpha_n \rightarrow \beta_n$$

is

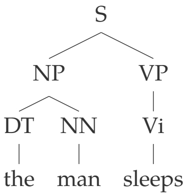
$$\begin{aligned} p(\boldsymbol{t}) &= \prod_{i=1}^{n} p(\alpha_i \rightarrow \beta_i) && (1) \\ &= \prod_{i=1}^{n} p(\beta_i \mid \alpha_i) && (2) \end{aligned}$$

where $p(\alpha \rightarrow \beta)$ is the *generative* rule of $\alpha \rightarrow \beta$

An Example

S	⇒	NP	VP	1.0
VP	⇒	Vi		0.4
VP	⇒	Vt	NP	0.4
VP	⇒	VP	PP	0.2
NP	⇒	DT	NN	0.3
NP	⇒	NP	PP	0.7
PP	⇒	P	NP	1.0

Vi	⇒	sleeps	1.0
Vt	⇒	saw	1.0
NN	⇒	man	0.7
NN	⇒	woman	0.2
NN	⇒	telescope	0.1
DT	⇒	the	1.0
IN	⇒	with	0.5
IN	⇒	in	0.5



$$\begin{aligned} p(\boldsymbol{t}) &= p(\text{S} \rightarrow \text{NP VP}) \cdot p(\text{NP} \rightarrow \text{DT NN}) \cdot p(\text{DT} \rightarrow \text{the}) \\ & \quad \cdot p(\text{NN} \rightarrow \text{man}) \cdot p(\text{VP} \rightarrow \text{Vi}) \cdot p(\text{Vi} \rightarrow \text{sleeps}) \\ &= 1.0 \times 0.3 \times 1.0 \times 0.7 \times 0.4 \times 1.0 \end{aligned}$$

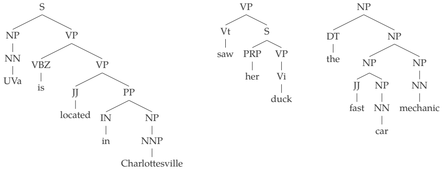
Properties of PCFGs

- Assigns a probability to each derivation, or parse-tree, allowed by the underlying CFG
- If one sentence has more than one derivations, we can rank them based on their probabilities
- The most likely parse tree for a sentence is

$$\operatorname*{argmax}_{\boldsymbol{t} \in \mathcal{T}(s)} p(\boldsymbol{t}) \tag{3}$$

Deriving a PCFG from a Corpus

- Given a set of example trees (a treebank), the underlying CFG can simply be all rules seen in the corpus

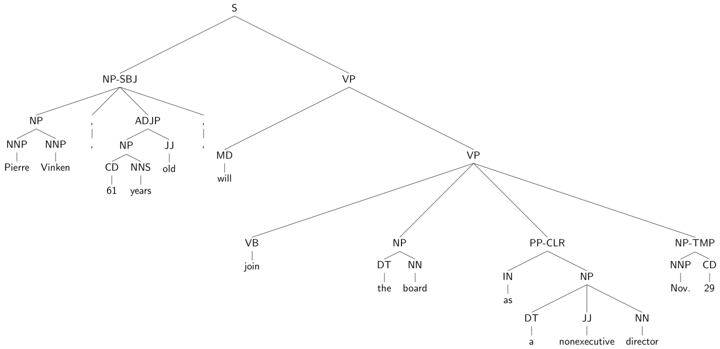


- Maximum likelihood estimates:

$$p(\alpha \rightarrow \beta) = p(\beta \mid \alpha) = \frac{\#(\alpha \rightarrow \beta)}{\#(\alpha)} \tag{4}$$

Penn Treebank

- 50,000 sentences with associated trees
- Usual setup: 40,000 training sentences, 2,400 test sentences



Some Penn Treebank Rules with Counts

40717 PP → IN NP
33803 S → NP-SBJ VP
22513 NP-SBJ → -NONE-
21877 NP → NP PP
20740 NP → DT NN
14153 S → NP-SBJ VP .
12922 VP → TO VP
11881 PP-LOC → IN NP
11467 NP-SBJ → PRP
11378 NP → -NONE-
11291 NP → NN
. . .
989 VP → VBG S
985 NP-SBJ → NN
983 PP-MNR → IN NP

100 VP → VBD PP-PRD
100 PRN → : NP :
100 NP → DT JJS
100 NP-CLR → NN
99 NP-SBJ-1 → DT NNP
98 VP → VBN NP PP-DIR
98 VP → VBD PP-TMP
98 PP-TMP → VBG NP
97 VP → VBD ADVP-TMP VP
. . .
10 WHNP-1 → WRB JJ
10 VP → VP CC VP PP-TMP
10 VP → VP CC VP ADVP-MNR
10 VP → VBZ S , SBAR-ADV
10 VP → VBZ S ADVP-TMP

Reference

Collins, M. (2017).
Natural language processing: Lecture notes.