Chapter 23(continued)

Natural Language for Communication

Phrase Structure Grammars

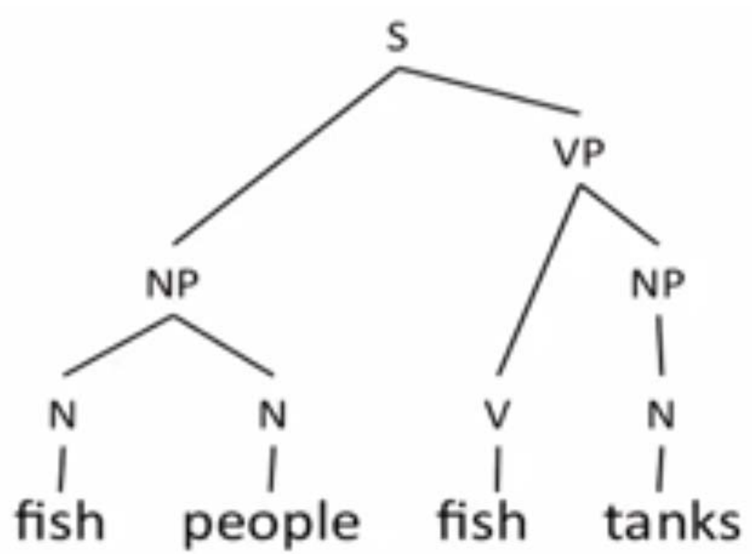
- Probabilistic context-free grammar (PCFG):
  - Context free: the left-hand side of the grammar consists of a single nonterminal symbol
  - Probabilistic: the grammar assigns a probability to every string
  - Lexicon: list of allowable words
  - Grammar: a collection of rules that defines a language as a set of allowable string of words
  - Example: **Fish people fish tanks**

Backus–Naur Form (BNF)

Rule	**Prob θ_i**
S → NP VP	θ_0
NP → NP NP	θ_1
...	
N → fish	θ_{42}
N → people	θ_{43}
V → fish	θ_{44}
...	

Phrase Structure Grammars (continued)

- Probabilistic context-free grammar (PCFG):
  - Context free: the left-hand side of the grammar consists of a single nonterminal symbol
  - Probabilistic: the grammar assigns a probability to every string
  - Lexicon: list of allowable words
  - Grammar: a collection of rules that defines a language as a set of allowable string of words
  - Example: **Fish people fish tanks**

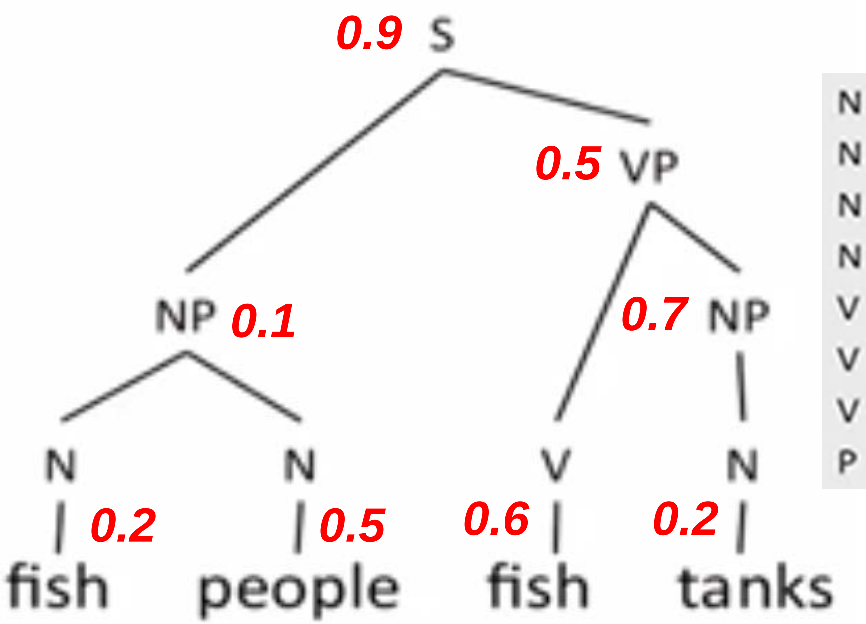


PCFG

Rule	Prob θ_i
S → NP VP	θ_0
NP → NP NP	θ_1
…	
N → fish	θ_{42}
N → people	θ_{43}
V → fish	θ_{44}
…	

Phrase Structure Grammars (continued)

- Example: **<u>Fish people fish tanks</u>**



Lexicon

N → *people*		0.5
N → *fish*		0.2
N → *tanks*		0.2
N → *rods*		0.1
V → *people*		0.1
V → *fish*		0.6
V → *tanks*		0.3
P → *with*		1.0

Grammar

S → NP VP		0.9
S → VP		0.1
VP → V NP		0.5
VP → V		0.1
VP → V @VP_V		0.3
VP → V PP		0.1
@VP_V → NP PP		1.0
NP → NP NP		0.1
NP → NP PP		0.2
NP → N		0.7
PP → P NP		1.0

Probability = 0.2 x 0.5 x 0.6 x 0.2 x 0.1 x 0.7 x 0.5 x 0.9

Parsing

- Objective: analyzing a string of words to uncover its phrase structure, given the lexicon and grammar.
  - The result of parsing is a parse tree
- Top-down parse and bottom-up parse
  - Naïve solutions: left-to-right or right-to-left parse
  - Example: **The wumpus is dead**

List of items	*Rule*
S	
NP VP	*S* → *NP VP*
NP VP Adjective	*VP* → *VP Adjective*
NP Verb Adjective	*VP* → *Verb*
NP Verb **dead**	*Adjective* → **dead**
NP **is dead**	*Verb* → **is**
Article Noun **is dead**	*NP* → *Article Noun*
Article **wumpus is dead**	*Noun* → **wumpus**
the wumpus is dead	*Article* → **the**

Parsing (continued)

- Objective: analyzing a string of words to uncover its phrase structure, given the lexicon and grammar.
  - The result of parsing is a parse tree
- Naïve solutions:
  - Top-down parse and bottom-up parse
  - Example: **<u>The wumpus is dead</u>**

List of items	*Rule*
S	
NP VP	*S* → *NP VP*
NP VP Adjective	*VP* → *VP Adjective*
NP Verb Adjective	*VP* → *Verb*
NP Verb **dead**	*Adjective* → **dead**
NP **is dead**	*Verb* → **is**
Article Noun **is dead**	*NP* → *Article Noun*
Article **wumpus is dead**	*Noun* → **wumpus**
the wumpus is dead	*Article* → **the**

  - Efficient?
  - Example:

<u>Have the students in section 2 of Computer Science 101 take the exam.</u>
<u>Have the students in section 2 of Computer Science 101 taken the exam?</u>

Parsing (continued)

- Efficient solutions: chart parsers
  - Using dynamic programming
- CYK algorithm
  - A bottom-up chart parser:
    (Named after its inventors, John Cocke, Daniel Younger, and Tadeo Kasami)
  - Input: lexicon, grammar and query strings.
  - Output: a parse tree
  - Three major steps:
    - Assign lexicons
    - Compute probability of adjacent phrases
    - Solve grammar conflict by selecting the most probable phrases

Parsing (continued)

- CYK algorithm
  - Three major steps:
    - Assign lexicons
    - Compute probability of adjacent phrases
    - Solve grammar conflict by selecting the most probable phrases

function CYK-PARSE(*words*, *grammar*) **returns** P, a table of probabilities

$N \leftarrow$ LENGTH(*words*)
$M \leftarrow$ the number of nonterminal symbols in *grammar*
$P \leftarrow$ an array of size $[M, N, N]$, initially all 0
/ Insert lexical rules for each word */*
for $i = 1$ **to** N **do**
  for each rule of form $(X \rightarrow words_i\ [p])$ **do**
    $P[X, i, 1] \leftarrow p$
/ Combine first and second parts of right-hand sides of rules, from short to long */*
for $length = 2$ **to** N **do**
  for $start = 1$ **to** $N - length + 1$ **do**
    for $len1 = 1$ **to** $N - 1$ **do**
      $len2 \leftarrow length - len1$
      for each rule of the form $(X \rightarrow Y\ Z\ [p])$ **do**
        $P[X, start, length] \leftarrow$ MAX($P[X, start, length]$,
          $P[Y, start, len1] \times P[Z, start + len1, len2] \times p$)
return P

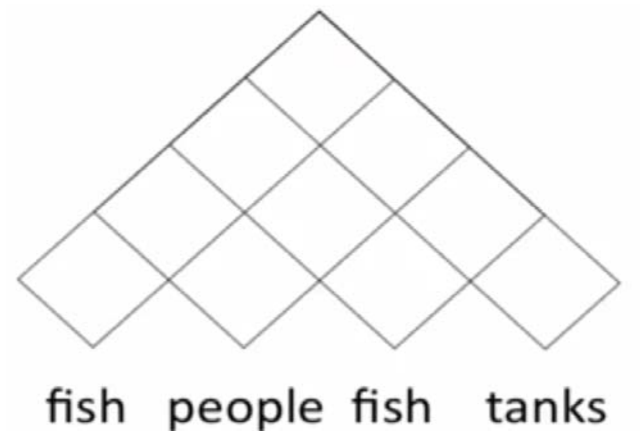
Assign lexicons

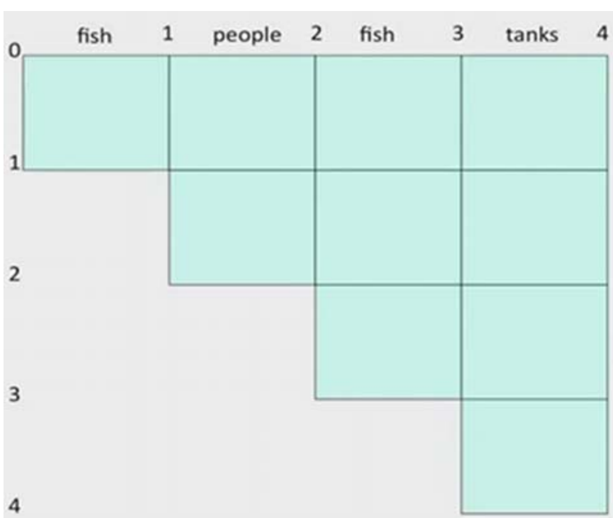
Solve grammar conflict

Compute probability of adjacent phrases

Parsing (continued)

- Example: **Fish people fish tanks**





Lexicon

N → *people*	0.5
N → *fish*	0.2
N → *tanks*	0.2
N → *rods*	0.1
V → *people*	0.1
V → *fish*	0.6
V → *tanks*	0.3
P → *with*	1.0

Grammar

S → NP VP	0.9
S → VP	0.1
VP → V NP	0.5
VP → V	0.1
VP → V @VP_V	0.3
VP → V PP	0.1
@VP_V → NP PP	1.0
NP → NP NP	0.1
NP → NP PP	0.2
NP → N	0.7
PP → P NP	1.0

Parsing (continued)

- Example: by Dr. Christopher Manning from Stanford

Augmented Parsing Methods

- Lexicalized PCFGs
  - BNF notation for grammars too restrictive
  - Augmented grammar
    - adding logical inference
    - to construct sentence semantics

$$
\begin{array}{lrcl}
\mathcal{E}_1 : & S & \rightarrow & NP_S\ VP \mid \ \ldots \\
 & NP_S & \rightarrow & Pronoun_S \mid Name \mid Noun \mid \ldots \\
 & NP_O & \rightarrow & Pronoun_O \mid Name \mid Noun \mid \ldots \\
 & VP & \rightarrow & VP\ NP_O \mid \ \ldots \\
 & PP & \rightarrow & Prep\ NP_O \\
 & Pronoun_S & \rightarrow & \mathbf{I} \mid \mathbf{you} \mid \mathbf{he} \mid \mathbf{she} \mid \mathbf{it} \mid \ \ldots \\
 & Pronoun_O & \rightarrow & \mathbf{me} \mid \mathbf{you} \mid \mathbf{him} \mid \mathbf{her} \mid \mathbf{it} \mid \ \ldots \\
 & & & \ldots
\end{array}
$$

$$
\begin{array}{lrcl}
\mathcal{E}_2 : & S(head) & \rightarrow & NP(Sbj, pn, h)\ VP(pn, head) \mid \ \ldots \\
 & NP(c, pn, head) & \rightarrow & Pronoun(c, pn, head) \mid Noun(c, pn, head) \mid \ \ldots \\
 & VP(pn, head) & \rightarrow & VP(pn, head)\ NP(Obj, p, h) \ \mid \ \ldots \\
 & PP(head) & \rightarrow & Prep(head)\ NP(Obj, pn, h) \\
 & Pronoun(Sbj, 1S, \mathbf{I}) & \rightarrow & \mathbf{I} \\
 & Pronoun(Sbj, 1P, \mathbf{we}) & \rightarrow & \mathbf{we} \\
 & Pronoun(Obj, 1S, \mathbf{me}) & \rightarrow & \mathbf{me} \\
 & Pronoun(Obj, 3P, \mathbf{them}) & \rightarrow & \mathbf{them} \\
 & & & \ldots
\end{array}
$$

Real language

- Real human languages provide many problems for NLP
  - Ambiguity
  - Anaphora
  - Indexicality
  - Vagueness
  - Discourse structure
  - Metonymy
  - Metaphor
  - Noncompositionality

Real language

- Real human languages provide many problems for NLP
  - Ambiguity: can be lexical (polysemy), syntactic, semantic, referential

I ate spaghetti with meatballs

Real language

- Real human languages provide many problems for NLP
  - Ambiguity: can be lexical (polysemy), syntactic, semantic, referential

I ate spaghetti with meatballs
salad

Real language

- Real human languages provide many problems for NLP
  - Ambiguity: can be lexical (polysemy), syntactic, semantic, referential

I ate spaghetti with meatballs
salad
abandon

Real language

- Real human languages provide many problems for NLP
  - Ambiguity: can be lexical (polysemy), syntactic, semantic, referential

I ate spaghetti with meatballs
salad
abandon
a fork

Real language

- Real human languages provide many problems for NLP
  - Ambiguity: can be lexical (polysemy), syntactic, semantic, referential

I ate spaghetti with meatballs

salad

abandon

a fork

a friend

Real language

- Real human languages provide many problems for NLP
  - Ambiguity
  - Anaphora: using pronouns to refer back to entities already introduced in the text

After Mary proposed to John, **they** found a preacher and got married.

Real language

- Real human languages provide many problems for NLP
  - Ambiguity
  - Anaphora: using pronouns to refer back to entities already introduced in the text

After Mary proposed to John, **they** found a preacher and got married.

For the honeymoon, **they** went to Hawaii

Real language

- Real human languages provide many problems for NLP
  - Ambiguity
  - Anaphora: using pronouns to refer back to entities already introduced in the text

After Mary proposed to John, **they** found a preacher and got married.

For the honeymoon, **they** went to Hawaii

Mary saw a ring through the window and asked John for **it**

Real language

- Real human languages provide many problems for NLP
  - Ambiguity
  - Anaphora: using pronouns to refer back to entities already introduced in the text

After Mary proposed to John, **they** found a preacher and got married.

For the honeymoon, **they** went to Hawaii

Mary saw a ring through the window and asked John for **it**

Mary threw a rock at the window and broke **it**

Real language

- Real human languages provide many problems for NLP
  - Ambiguity
  - Anaphora
  - Indexicality: indexical sentences refer to utterance situation (place, time, S/H, etc.)

I am over **here**

Why did **you** do **that**?

Real language

- Real human languages provide many problems for NLP
  - Ambiguity
  - Anaphora
  - Indexicality
  - Vagueness
  - Discourse structure
  - Metonymy: using one noun phrase to stand for another

I've read **Shakespeare**

Chrysler announced record profits

The **ham sandwich** on Table 4 wants another beer

Real language

- Real human languages provide many problems for NLP
  - Ambiguity
  - Anaphora
  - Indexicality
  - Vagueness
  - Discourse structure
  - Metonymy
  - Metaphor: “Non-literal” usage of words and phrases

I've tried **killing** the process but it won't **die**. Its **parent** keeps it alive

Real language

- Real human languages provide many problems for NLP
  - Ambiguity
  - Anaphora
  - Indexicality
  - Vagueness
  - Discourse structure
  - Metonymy
  - Metaphor
  - Noncompositionality

basketball shoes	red book
baby shoes	red pen
alligator shoes	red hair
designer shoes	red herring
brake shoes	

Real language

- Real human languages provide many problems for NLP
  - Ambiguity
  - Anaphora
  - Indexicality
  - Vagueness
  - Discourse structure
  - Metonymy
  - Metaphor
  - Noncompositionality

- Interpreting natural language using computer agents is challenging and still an open problem
(but we are doing better)