

# For Friday

- Finish chapter 22
- Homework
  - Chapter 22, exercises 1, 7, 9, 14
  - Allocate some time for this one

# Program 5

# Learning mini-project

- Worth 2 homeworks
- Due Wednesday
- Foil6 is available in `/home/mecalif/public/itk340/foil`
- A manual and sample data files are there as well.
- Create a data file that will allow FOIL to learn rules for a `sister/2` relation from background relations of `parent/2`, `male/1`, and `female/1`. You can look in the `prolog` folder of my `327` folder for sample data if you like.
- Electronically submit your data file—which should be named `sister.d`, and turn in a hard copy of the rules FOIL learns.

# Input/Output Coding

- Appropriate coding of inputs and outputs can make learning problem easier and improve generalization.
- Best to encode each binary feature as a separate input unit and for multi-valued features include one binary unit per value rather than trying to encode input information in fewer units using binary coding or continuous values.

# I/O Coding cont.

- Continuous inputs can be handled by a single input by scaling them between 0 and 1.
- For disjoint categorization problems, best to have one output unit per category rather than encoding  $n$  categories into  $\log n$  bits. Continuous output values then represent certainty in various categories. Assign test cases to the category with the highest output.
- Continuous outputs (regression) can also be handled by scaling between 0 and 1.

# Neural Net Conclusions

- Learned concepts can be represented by networks of linear threshold units and trained using gradient descent.
- Analogy to the brain and numerous successful applications have generated significant interest.
- Generally much slower to train than other learning methods, but exploring a rich hypothesis space that seems to work well in many domains.
- Potential to model biological and cognitive phenomenon and increase our understanding of real neural systems.
  - Backprop itself is not very biologically plausible

# Natural Language Processing

- What's the goal?

# Communication

- Communication for the speaker:
  - Intention: Decided why, when, and what information should be transmitted. May require planning and reasoning about agents' goals and beliefs.
  - Generation: Translating the information to be communicated into a string of words.
  - Synthesis: Output of string in desired modality, e.g. text on a screen or speech.

# Communication (cont.)

- Communication for the hearer:
  - Perception: Mapping input modality to a string of words, e.g. optical character recognition or speech recognition.
  - Analysis: Determining the information content of the string.
    - Syntactic interpretation (parsing): Find correct parse tree showing the phrase structure
    - Semantic interpretation: Extract (literal) meaning of the string in some representation, e.g. FOPC.
    - Pragmatic interpretation: Consider effect of overall context on the meaning of the sentence
  - Incorporation: Decide whether or not to believe the content of the string and add it to the KB.

# Ambiguity

- Natural language sentences are highly ambiguous and must be disambiguated.

I saw the man on the hill with the telescope.

I saw the Grand Canyon flying to LA.

I saw a jet flying to LA.

Time flies like an arrow.

Horse flies like a sugar cube.

Time runners like a coach.

Time cars like a Porsche.

# Syntax

- Syntax concerns the proper ordering of words and its effect on meaning.

The dog bit the boy.

The boy bit the dog.

\* Bit boy the dog the

Colorless green ideas sleep furiously.

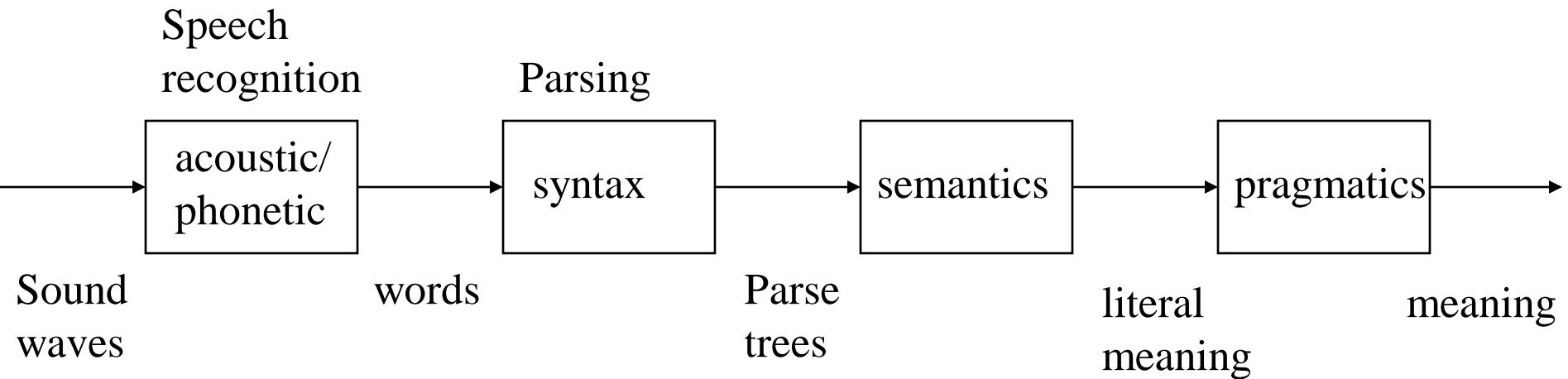
# Semantics

- Semantics concerns of meaning of words, phrases, and sentences. Generally restricted to “literal meaning”
  - “plant” as a photosynthetic organism
  - “plant” as a manufacturing facility
  - “plant” as the act of sowing

# Pragmatics

- Pragmatics concerns the overall communicative and social context and its effect on interpretation.
  - Can you pass the salt?
  - Passerby: Does your dog bite?  
Clouseau: No.  
Passerby: (pets dog) Chomp!  
I thought you said your dog didn't bite!!  
Clouseau: That, sir, is not my dog!

# Modular Processing



# Examples

- Phonetics

“grey twine” vs. “great wine”

“youth in Asia” vs. “euthanasia”

“yawanna” -> “do you want to”

- Syntax

I ate spaghetti with a fork.

I ate spaghetti with meatballs.

# More Examples

- Semantics

I put the plant in the window.

Ford put the plant in Mexico.

The dog is in the pen.

The ink is in the pen.

- Pragmatics

The ham sandwich wants another beer.

John thinks vanilla.

# Formal Grammars

- A **grammar** is a set of **production rules** which generates a set of strings (a language) by **rewriting** the top symbol S.
- **Nonterminal** symbols are intermediate results that are not contained in strings of the language.

$S \rightarrow NP VP$

$NP \rightarrow Det N$

$VP \rightarrow V NP$

- **Terminal** symbols are the final symbols (words) that compose the strings in the language.
- Production rules for generating words from part of speech categories constitute the lexicon.
- $N \rightarrow \text{boy}$
- $V \rightarrow \text{eat}$

# Context-Free Grammars

- A context-free grammar only has productions with a single symbol on the left-hand side.
- CFG:  
     $S \rightarrow NP V$   
     $NP \rightarrow Det N$   
     $VP \rightarrow V NP$
- not CFG:  
     $AB \rightarrow C$   
     $BC \rightarrow FG$

# Simplified English Grammar

S -> NP VP

S -> VP

NP -> Det Adj\* N

NP -> ProN

NP -> PName

VP -> V

VP -> V NP

VP -> VP PP

PP -> Prep NP

Adj\* -> e

Adj\* -> Adj Adj\*

## Lexicon:

ProN -> I; ProN -> you; ProN -> he; ProN -> she

Name -> John; Name -> Mary

Adj -> big; Adj -> little; Adj -> blue; Adj -> red

Det -> the; Det -> a; Det -> an

N -> man; N -> telescope; N -> hill; N -> saw

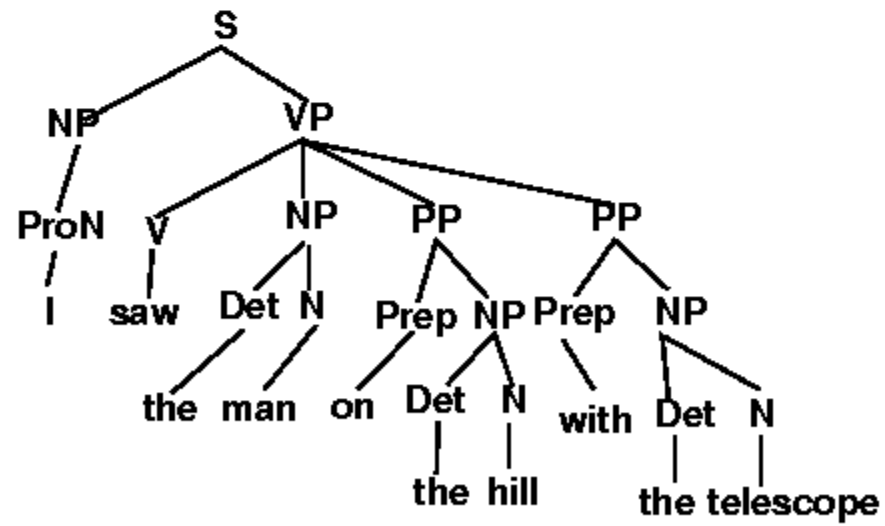
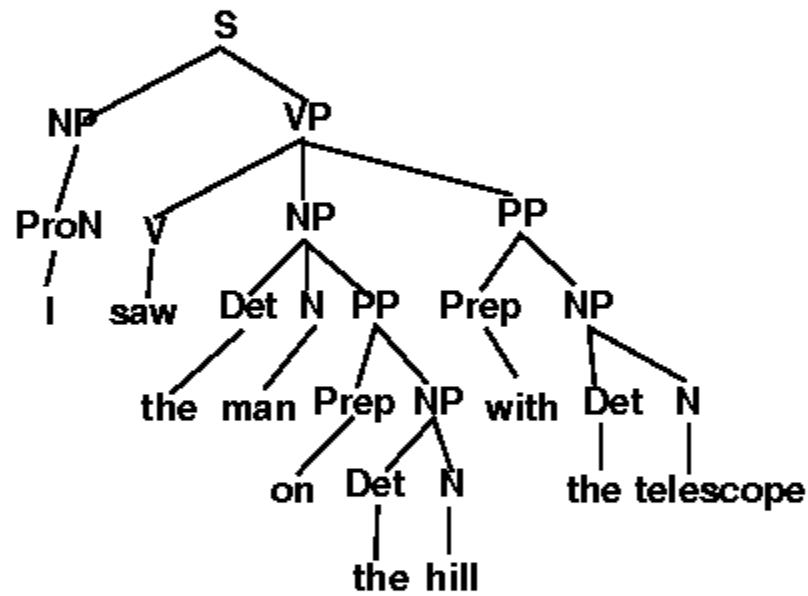
Prep -> with; Prep -> for; Prep -> of; Prep -> in

V -> hit; V -> took; V -> saw; V -> likes

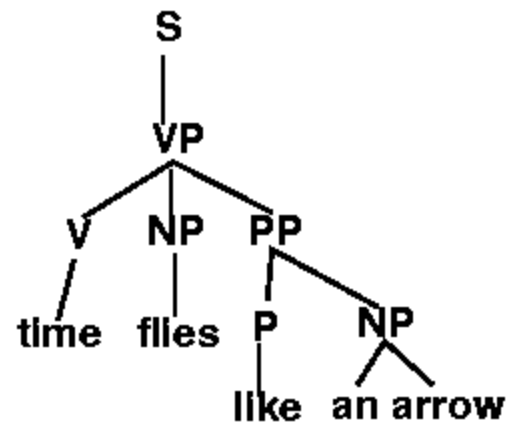
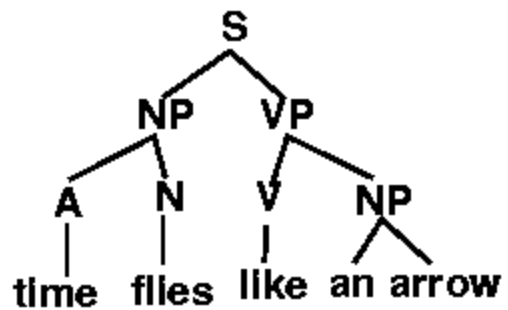
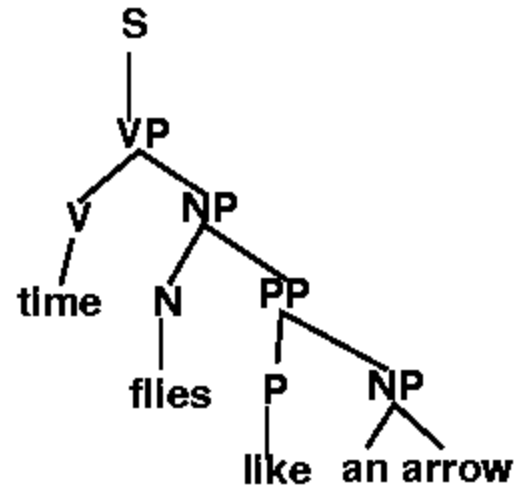
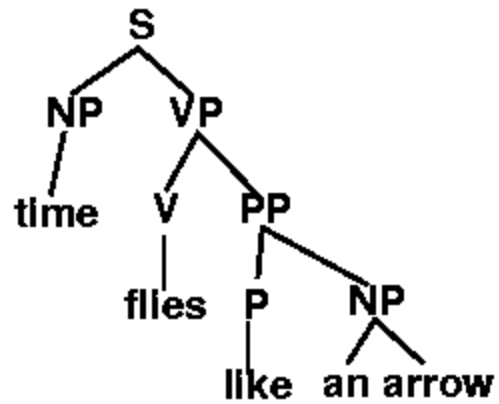
# Parse Trees

- A parse tree shows the **derivation** of a sentence in the language from the start symbol to the terminal symbols.
- If a given sentence has more than one possible derivation (parse tree), it is said to be **syntactically ambiguous**.

(part 2/3), it is said to be syntactically ambiguous.



# Spurious Parses



# Syntactic Parsing

- Given a string of words, determine if it is grammatical, i.e. if it can be derived from a particular grammar.
- The derivation itself may also be of interest.
- Normally want to determine all possible parse trees and then use semantics and pragmatics to eliminate spurious parses and build a semantic representation.

# Parsing Complexity

- **Problem:** Many sentences have many parses.
- An English sentence with  $n$  prepositional phrases at the end has at least  $2^n$  parses.

I saw the man on the hill with a telescope on Tuesday in Austin...

- The actual number of parses is given by the Catalan numbers:  
1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796...

# Parsing Algorithms

- Top Down: Search the space of possible derivations of S (e.g. depth-first) for one that matches the input sentence.

I saw the man.

S -> NP VP	VP -> V NP
NP -> Det Adj* N	V -> hit
Det -> the	V -> took
Det -> a	V -> saw
Det -> an	NP -> Det Adj* N
NP -> ProN	Det -> the
ProN -> I	Adj* -> e
	N -> man

# Parsing Algorithms (cont.)

- Bottom Up: Search upward from words finding larger and larger phrases until a sentence is found.

I saw the man.

ProN saw the man

NP saw the man

NP N the man

NP V the man

NP V Det man

NP V Det Adj\* man

NP V Det Adj\* N

NP V NP

NP VP

S

ProN -> I

NP -> ProN

N -> saw (dead end)

V -> saw

Det -> the

Adj\* -> e

N -> man

NP -> Det Adj\* N

VP -> V NP

S -> NP VP

# Bottom-up Parsing Algorithm

**function** BOTTOM-UP-PARSE(*words*, *grammar*) **returns** a parse tree

*forest*  $\leftarrow$  *words*

**loop do**

**if** LENGTH(*forest*) = 1 and CATEGORY(*forest*[1]) = START(*grammar*) **then**  
    **return** *forest*[1]

**else**

*i*  $\leftarrow$  **choose** from {1...LENGTH(*forest*)}

*rule*  $\leftarrow$  **choose** from RULES(*grammar*)

*n*  $\leftarrow$  LENGTH(RULE-RHS(*rule*))

*subsequence*  $\leftarrow$  SUBSEQUENCE(*forest*, *i*, *i+n-1*)

**if** MATCH(*subsequence*, RULE-RHS(*rule*)) **then**

*forest*[*i*...*i+n-1*] / [MAKE-NODE(RULE-LHS(*rule*), *subsequence*)]

**else fail**

**end**

# Augmented Grammars

- Simple CFGs generally insufficient:  
“The dogs bites the girl.”
- Could deal with this by adding rules.
  - What’s the problem with that approach?
- Could also “augment” the rules: add constraints to the rules that say number and person must match.

# Verb Subcategorization

# Semantics

- Need a semantic representation
- Need a way to translate a sentence into that representation.
- Issues:
  - Knowledge representation still a somewhat open question
  - Composition  
“He kicked the bucket.”
  - Effect of syntax on semantics

# Dealing with Ambiguity

- Types:
  - Lexical
  - Syntactic ambiguity
  - Modifier meanings
  - Figures of speech
    - Metonymy
    - Metaphor

# Resolving Ambiguity

- Use what you know about the world, the current situation, and language to determine the most likely parse, using techniques for uncertain reasoning.

# Discourse

- More text = more issues
- Reference resolution
- Ellipsis
- Coherence/focus