NATURAL LANGUAGE PROCESSING FOR COMMUNICATION

Sections 23.1 – 23.3
(not covering 23.2.1-2)

Please set your mobile devices to silent.

Last Time

Introduction to Learning

- Supervised Learning - Induction from observations
- Trading model fit for simplicity
- Algorithms
  - KNN
  - Naïve Bayes
  - Decision Trees / Information Gain

Outline

- Formal Grammar
- Parsing: Syntactic Analysis
- Augmented Grammars
- The larger context
  - Communication as Action
  - Semantic Interpretation
  - Ambiguity and Disambiguation
  - Discourse Understanding

Language

- Formal language: A (possibly infinite) set of strings
- Grammar: A finite set of rules that specifies a language
- Rewrite rules
  - Non-terminal symbols [not observed] (S, NP, etc.)
  - Terminal symbols [observed] (“he”)

  - S → NP VP
  - NP → Pronoun
  - Pronoun → “he”

Convention: Uppercase are for non-terminals, lowercase for terminals

Generative Capacity

Noam Chomsky described four grammatical formalisms:

- Recursively enumerable grammars
  - Unrestricted rules: both sides of the rewrite rules can have any number of terminal and non-terminal symbols; full Turing machines

    ABd → CaE
- Context-sensitive grammars
  - The RHS must contain at least as many symbols as the LHS

    ASB → AXB
- Context-free grammars (CFG)
  - LHS is a single non-terminal symbol

    S → XYa
- Regular grammars
  - LHS is single non-terminal; RHS a terminal plus optional non-terminal

    X → a X → aY

Formal Grammar

- The lexicon for ε_o:

Noun → stench | breeze | glitter | wumpus | pit | pits | gold | …
Verb → is | see | smell | shoot | stinks | go | grab | turn | …
Adjective → right | left | east | dead | back | smelly | …
Adverb → here | there | nearby | ahead | right | left | east | …
Pronoun → me | you | I | it | …
Name → John | Mary | Boston | Aristotle | …
Article → the | a | an | …
Preposition → to | in | on | near | …
Conjunction → and | or | but | …
Digit → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Formal Grammar

- The grammar for ε_o:

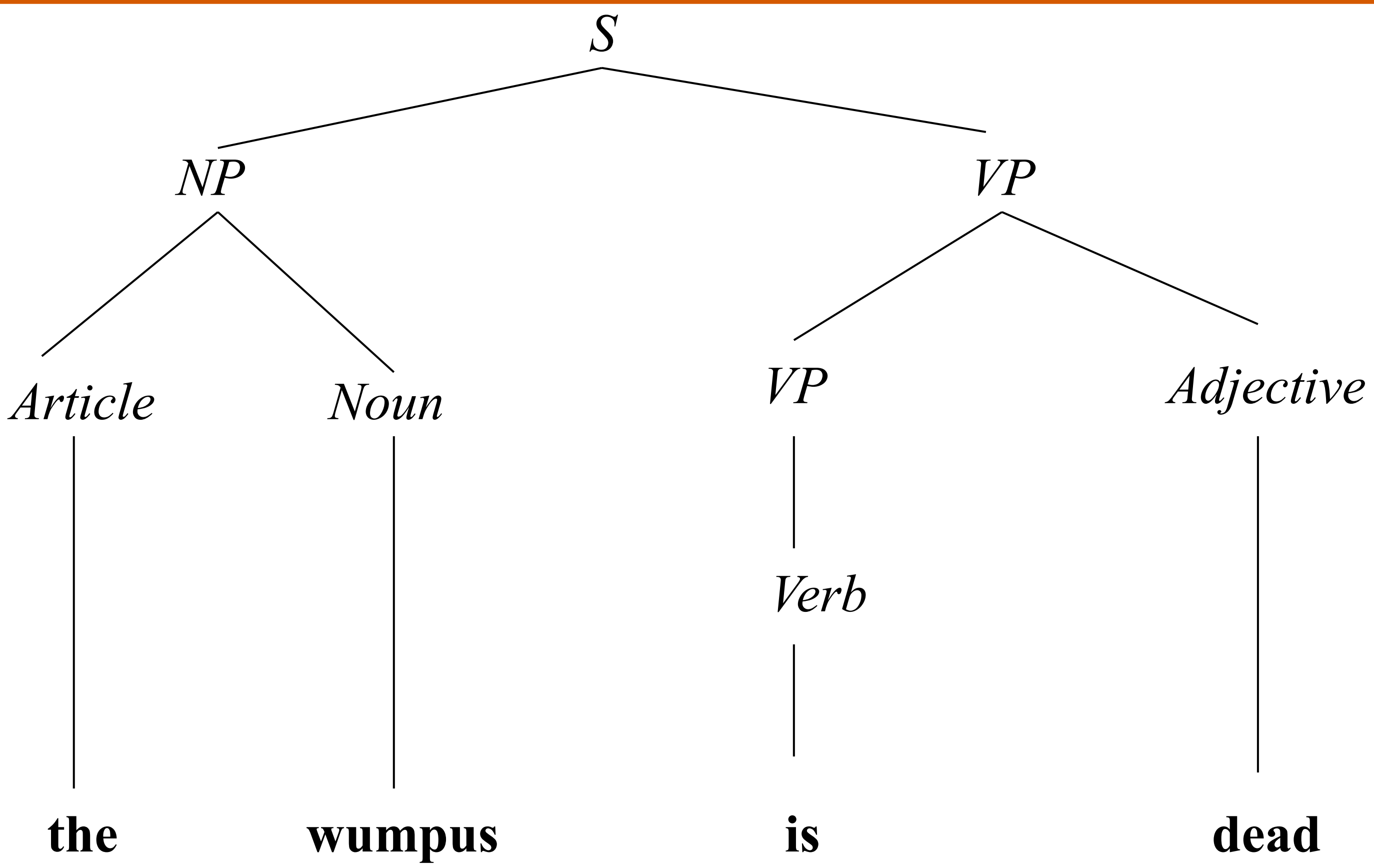
S	→	NP VP	I + feel a breeze
	|	S Conjunction S	I feel a breeze + and + I smell a wumpus
NP	→	Pronoun	I
	|	Name	John
	|	Noun	pits
	|	Article Noun	the + wumpus
	|	Digit Digit	3 4
	|	NP PP	the wumpus + to the east
	|	NP RelClause	the wumpus + that is smelly

We'll deal with probabilistic rules later

Formal Grammar

- Parts of speech
  - Open class: noun, verb, adjective, adverb
  - Closed class: pronoun, article, preposition, conjunction, …
- Shortcomings of our grammar
  - Overgenerate: “Me go Boston”
  - Undergenerate: “I think the wumpus is smelly”

Parse Tree



Syntactic Analysis (Parsing)

- Parsing: The process of finding a parse tree for a given input string
- Top-down parsing
  - Start with the S symbol and search for a tree that has the words as its leaves
- Bottom-up parsing
  - Start with the words and search for a tree with root S

Trace of Bottom-up Parsing

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

- Left to Right processing per token

Blank spaces to fill in on this slide

Intrasentence ambiguity

"Have the students of CS 3243

Artificial Intelligence …

Probabilistic Grammars

- Probabilistic lexicon

Noun → stench [.05] | breeze [.1] | wumpus [.15] | pits [.05] | …
Verb → is [.1] | feel [.1] | stinks [.05] | …
…
Digit → 0 [.1] | 1 [.1] | 2 [.1] | …

- Probabilistic Grammar

VP→	Verb	[.4] stinks
|	VP NP	[.35] feel + a breeze
|	VP Adjective	[.05] is + smelly
|	VP PP	[.1] turn + to the east
|	VP Adverb	[.1] go + ahead

Where each category (e.g., VP) rule has probabilities that sum to one.

CYK Chart Parsing

- Uses dynamic programming to memoize intermediate results, saving to a **chart**.
- Bottom Up Iterative Processing
- Converts context free grammar into a special form: Chomsky Normal Form
  - X → “a”
  - X → YZ
- Uses space $O(n^2m) \cong O(n^3)$, despite $O(2^n)$ possible parses.
- Suitable for probabilistic CFGs (PCFGs).

CYK Parsing

```
function CYK-PARSE(words, grammar) returns P, a table of probabilities
  N ← LENGTH(words)
  M ← the number of nonterminal symbols in grammar
  P ← 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 → words_i [p]) do
      P[X, i, 1] ← 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 ← length − len1
        for each rule of the form (X → Y Z [p]) do
          P[X, start, length] ← MAX(P[X, start, length],
                                   P[Y, start, len1] × P[Z, start + len1, len2] × p)
  return P
```

An Ambiguous Example

N,A	N,V	N,V	P	A	N
British	left	waffles	on	Falkland	Islands

$S \rightarrow NP\ VP$
$S \rightarrow NP\ VP\ PP$
$PP \rightarrow P\ NP$
$NP \rightarrow A\ NP$
$NP \rightarrow NP\ PP$
$VP \rightarrow V\ PP$
$VP \rightarrow V\ NP$
$VP \rightarrow V$
$NP \rightarrow N$

See whether you understand what each of the “S”s stand for

Blank spaces to fill in on this slide

Dealing with Probabilities

S[3.6864E-4], S[4.608E-4], S[5.37E-3]					
	S[.00576], VP[.00384]				
		?			
S[.0144]			PP [.16]		
S[.048], NP[.16]	?			NP[.16]	
N,A	N,V	N,V	P	A	N
British	left	waffles	on	Falkland	Islands

S → NP VP [.3]
S → NP VP PP [.7]
PP → P NP [1]
NP → A NP [.4]
NP → NP PP [.2]
NP → N [.4]
VP → V PP [.4]
VP → V NP [.3]
VP → V [.4]

We ignored the probabilistic lexicon for this example which should also be used.

Subjective & Objective Cases

- Overgeneration:
  - S → NP VP → NP VP NP → NP Verb NP
  - Pronoun Verb NP → Pronoun Verb Pronoun

She loves him

*her loves he

She ran towards him

*She ran towards he

Handling Subjective & Objective Cases

S	→	NPs VP | …
NP_s	→	$Pronoun_s$ | Name | Noun | …
NP_o	→	$Pronoun_o$ | Name | Noun | …
VP	→	VP NPo | …
PP	→	Preposition NP_o
$Pronoun_s$	→	**I** | **you** | **he** | **she** | **it** | …
$Pronoun_o$	→	**me** | **you** | **him** | **her** | **it** | …

- Disadvantage: Grammar size grows exponentially

Augmented Grammars

- Handling case, agreement, etc
- Augment grammar rules to allow parameters on nonterminal categories
  - NP(Subjective)
  - NP(Objective)
  - NP(case)

Definite Clause Grammar (DCG)

- The grammar for ε1:

S	→	NP(Subjective) VP | …
NP(case)	→	Pronoun(case) | Name | Noun | …
VP	→	VP NP(Objective) | …
PP	→	Preposition NP(Objective)
Pronoun(Subjective)	→	**I** | **you** | **he** | **she** | **it** | …
Pronoun(Objective)	→	**me** | **you** | **him** | **her** | **it** | …

Definite Clauses, where did we hear that before?

Definite Clause Grammar (DCG)

- Each grammar rule is a definite clause in logic:
  - S → NP VP
  - NP(s1) ∧ VP(s2) ⇒ S(s1 + s2)
  - NP(case) → Pronoun(case)
  - Pronoun(case, s1) ⇒ NP(case, s1)

Prolog as a suitable language for deterministic NLP

- DCG enables parsing as logical inference:
  - Top-down parsing is backward chaining
  - Bottom-up parsing is forward chaining

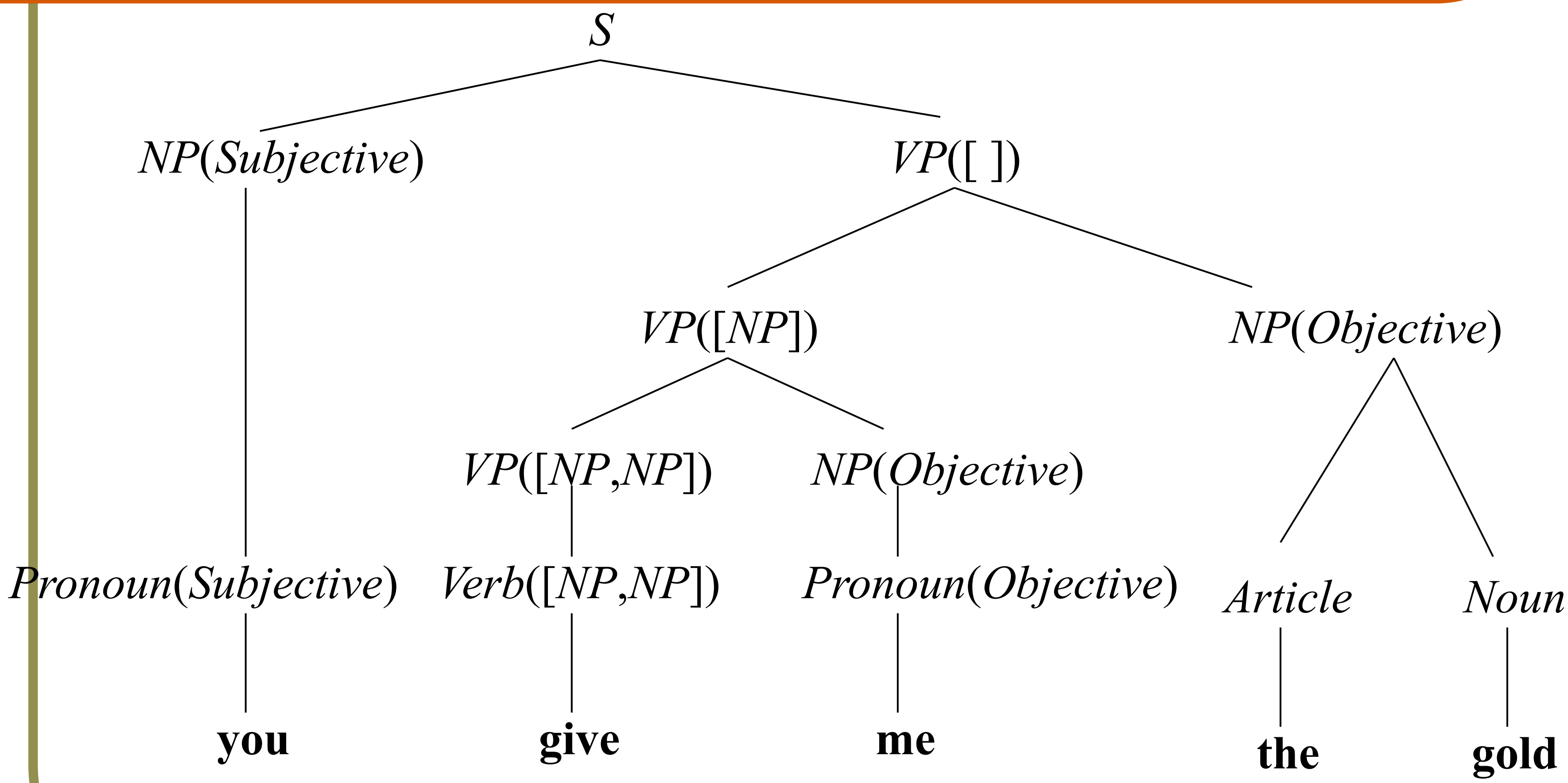
Verb Subcategorization

Verb	Subcats	Example Verb Phrase
give	[*NP*,*PP*] [*NP*,*NP*]	give the gold to me give me the gold
smell	[*NP*] [*Adjective*] [*PP*]	smell a wumpus smell awful smell like a wumpus
is	[*Adjective*] [*PP*] [*NP*]	is smelly is in 2 2 is a pit
died	[]	died
believe	[*S*]	believe the wumpus is dead

Verb Subcategorization

S	→	NP(Subjective) VP([ ])
VP(subcat)	→	Verb(subcat)
	|	VP(subcat + [NP]) NP(Objective)
	|	VP(subcat + [Adjective]) Adjective
	|	VP(subcat + [PP]) PP
VP(subcat)	→	VP(subcat) PP
	|	VP(subcat) Adverb
Verb([NP,NP])	→	**give** | **hand** | …

Parsing Using Verb Subcategorization



The larger context – communication

- Communication
  - Intentional exchange of information brought about by the production and perception of signs drawn from a shared system of conventional signs
- Humans use language to communicate most of what is known about the world
- The Turing test is based on language

Communication as Action

- Speech act
  - Language production viewed as an action
  - Speaker, hearer, utterance
- Examples:
  - Query: “Have you smelled the wumpus anywhere?”
  - Inform: “There’s a breeze here in 3 4.”
  - Request: “Please help me carry the gold.” “I could use some help carrying this.”
  - Acknowledge: “OK”
  - Promise: “I’ll shoot the wumpus.”

Component Steps of Communication

SPEAKER:

- Intention

  Know(H,¬Alive(Wumpus,S_3))

- Generation

  "The wumpus is dead"

- Synthesis

  [thaxwahmpaxsihzdehd]

Component Steps of Communication

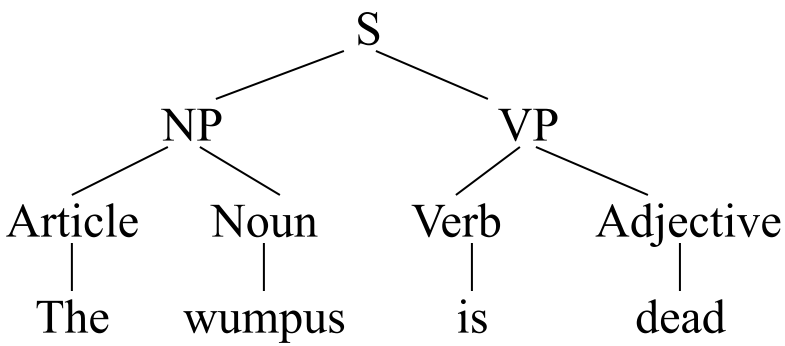
HEARER:

- Perception:

  "The wumpus is dead"

- Analysis

  (Parsing):



(Semantic Interpretation): ¬Alive(Wumpus, Now)

Tired(Wumpus, Now)

(Pragmatic Interpretation): $\neg Alive(Wumpus_1, S_3)$

$Tired(Wumpus_1, S_3)$

Component Steps of Communication

HEARER:

- Disambiguation:

  $\neg Alive(Wumpus_1, S_3)$

- Incorporation:

  $TELL(KB, \neg Alive(Wumpus_1, S_3))$

Semantic Interpretation

- Semantics: meaning of utterances
- First-order logic as the representation language
- Compositional semantics: meaning of a phrase is composed of meaning of the constituent parts of the phrase

Semantic Interpretation

$Exp(x) \rightarrow Exp(x_1)\ Operator(op)\ Exp(x_2)$

$\{ x = Apply(op, x_1, x_2) \}$

$Exp(x) \rightarrow (Exp(x))$

$Exp(x) \rightarrow Number(x)$

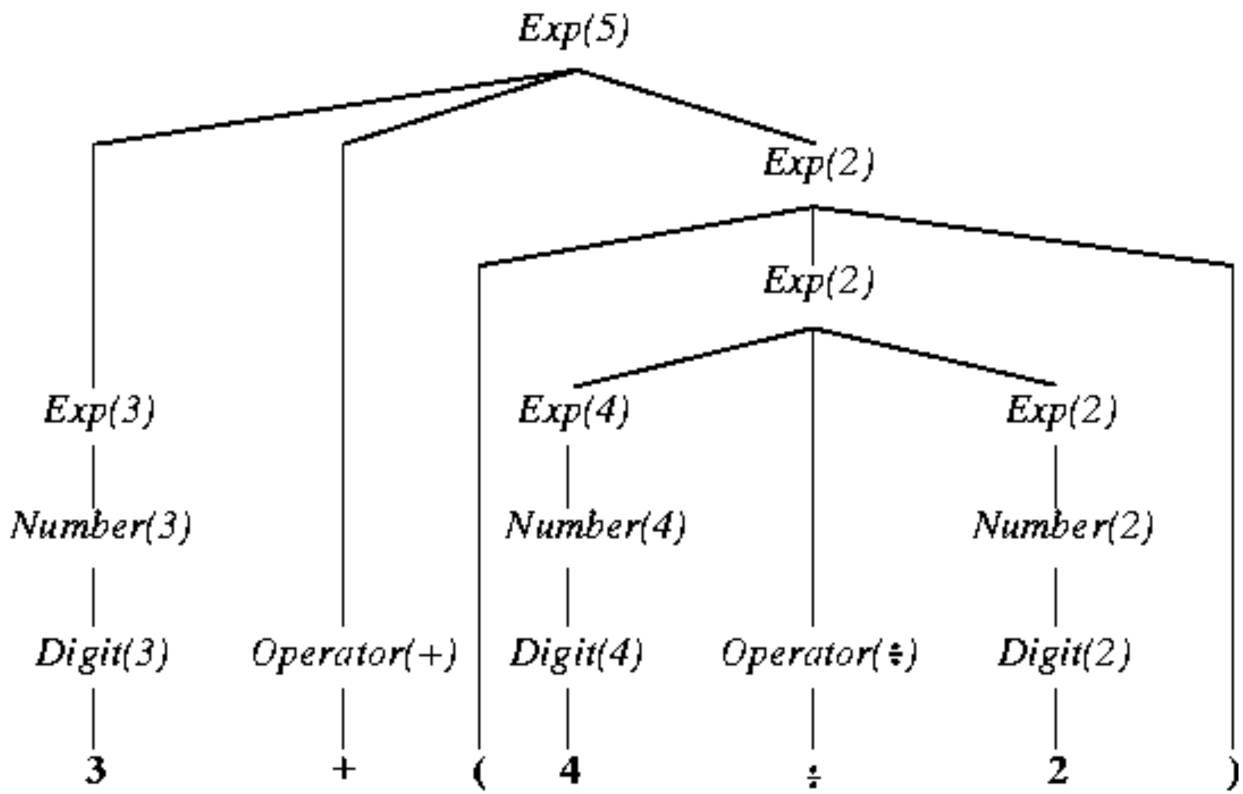
$Number(x) \rightarrow Digit(x)$

$Number(x) \rightarrow Number(x_1)\ Digit(x_2)\ \{ x = 10 \times x_1 + x_2 \}$

$Digit(x) \rightarrow x\ \{ 0 \leq x \leq 9 \}$

$Operator(x) \rightarrow x\ \{ x \in \{ +, -, \times, \div \}\}$

Semantic Interpretation



Semantic Interpretation

John loves Mary

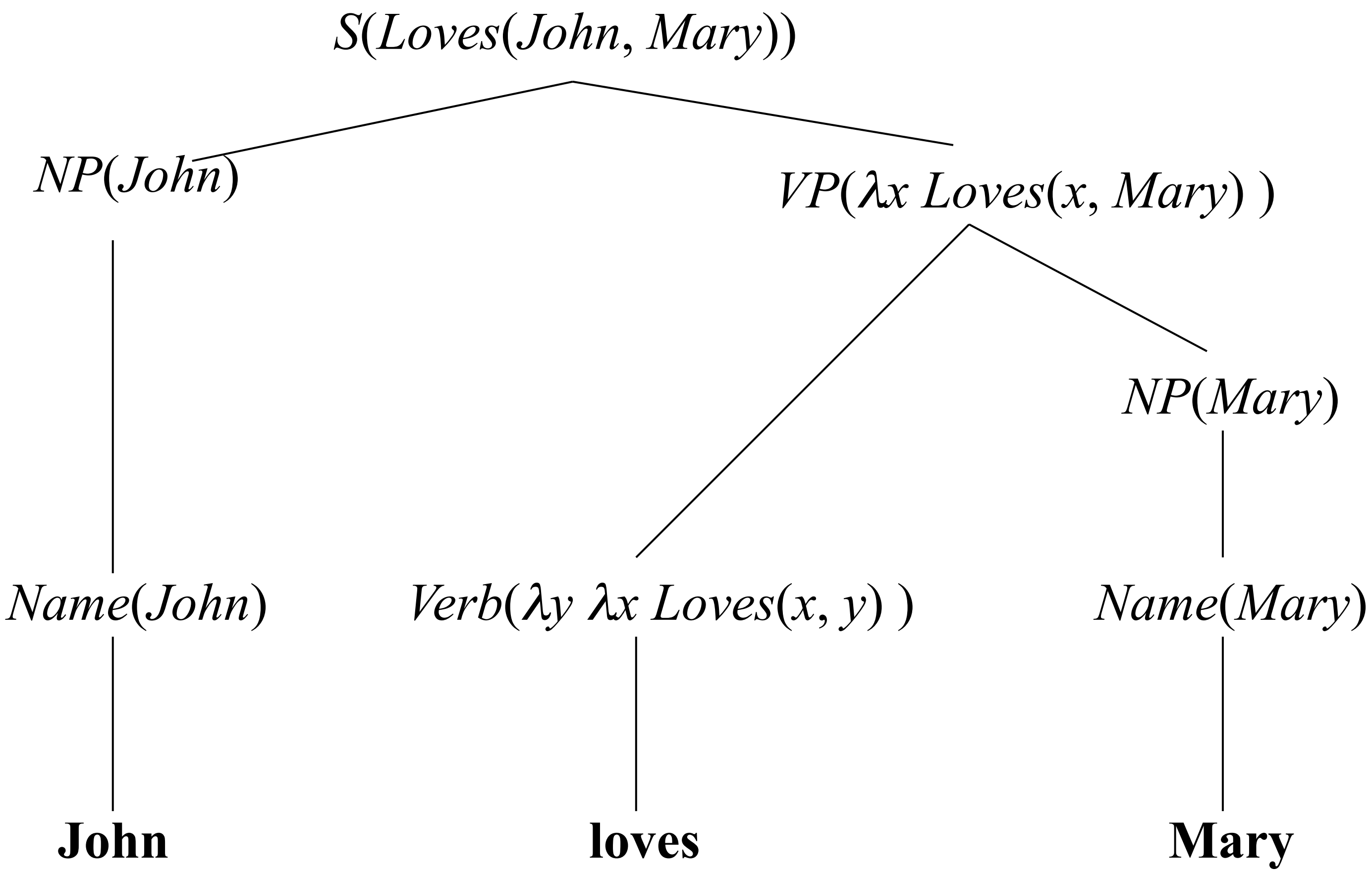
Loves(John, Mary)

$(\lambda y\ \lambda x\ Loves(x,y))\ (Mary) \equiv \lambda x\ Loves(x, Mary)$

$(\lambda x\ Loves(x, Mary))\ (John) \equiv Loves(John, Mary)$

S(rel(obj))	→	NP(obj) VP(rel)
VP(rel(obj))	→	Verb(rel) NP(obj)
NP(obj)	→	Name(obj)
Name(John)	→	**John**
Name(Mary)	→	**Mary**
Verb(λy λx Loves(x,y))	→	**loves**

Semantic Interpretation



Pragmatic Interpretation

- Adding context-dependent information about the current situation to each candidate semantic interpretation
- Indexicals: phrases that refer directly to the current situation
  - "I am in Boston today"

    ("I" refers to speaker and "today" refers to now)

Language Generation

The same DCG can be used for parsing and generation

- Parsing:
  - Given: S(sem, [John, loves, Mary])
  - Return: sem = Loves(John, Mary)
- Generation:
  - Given: S(Loves(John, Mary), words)
  - Return: words = [John, loves, Mary]

Ambiguity

- Lexical ambiguity
  - “the **back** of the room” vs. “**back** up your files”
  - “In the **interest** of stimulating the economy, the government lowered the **interest** rate.”
- Syntactic ambiguity (structural ambiguity)
  - “I smelled a wumpus in 2,2”
- Semantic ambiguity
  - “the IBM lecture”
- Pragmatic ambiguity
  - “I’ll meet you next Friday”

Metonymy

Denotes a concept by naming some other concept closely *related* to it

- Examples:
  - Company for company's spokesperson ("IBM announced a new model")
  - Author for author's works ("I read Shakespeare")
  - Producer for producer's product ("I drive a Honda")

Metonymy

- Representation of "IBM announced"

$$\exists m,x,e \;\; x = IBM \wedge e \in Announce(m) \wedge After(Now,e) \wedge Metonymy(m,x)$$

$$\forall m,x \;\; (m = x) \Rightarrow Metonymy(m,x)$$

$$\forall m,x \;\; x \in Organizations \wedge Spokesperson(m,x) \Rightarrow Metonymy(m,x)$$

Metaphor

Refer to concepts using words whose meanings are appropriate to other completely different kinds of concepts

- Example: corporation-as-person metaphor:
  - Speak of a corporation as if it is a person and can experience emotions, has a mind, etc.
  - "That doesn't **scare** Digital, which has grown to be the world's second-largest computer maker."
  - "But if the company changed its **mind**, however, it would do so for investment reasons, the filing said."

Disambiguation

$$\arg\max_{intent} Likelihood(intent \mid words, situation)$$

- Disambiguation is like diagnosis
- The speaker's intent to communicate is an unobserved cause of the words in the utterance
- The hearer's job is to work backwards from the words and from knowledge of the situation to recover the most likely intent of the speaker

Discourse Understanding

- Discourse: multiple sentences
- Reference resolution: The interpretation of a pronoun or a definite noun phrase that refers to an object in the world
- "John flagged down the waiter. He ordered a ham sandwich."
  - "He" refers to "John"
- "After John proposed to Mary, they found a preacher and got married. For the honeymoon, they went to Hawaii."
  - "they"? "the honeymoon"?

Discourse Understanding

- Structure of coherent discourse: Sentences are joined by coherence relations
- Examples of coherence relations between S1 and S2:
  - Enable or cause: S1 brings about a change of state that causes or enables S2
    - "I went outside. I drove to school."
  - Explanation: the reverse of enablement, S2 causes or enables S1 and is an explanation for S1
    - "I was late for school. I overslept."
  - Exemplification: S2 is an example of the general principle in S1
    - "This algorithm reverses a list. The input [A,B,C] is mapped to [C,B,A]."
  - Etc.

Summary

- NLP is full of ambiguity
  - Natural languages a testament to human intelligence, creativity
- NLP largely processes one utterance at a time
- Deterministic methods can follow context free grammars, (DCGs, hence Prolog)
- PCFGs add probabilistic interpretation
- Both parsable using DP (CYK algorithm)