Communication and Natural Language Processing

CPSC 470 – Artificial Intelligence

Brian Scassellati

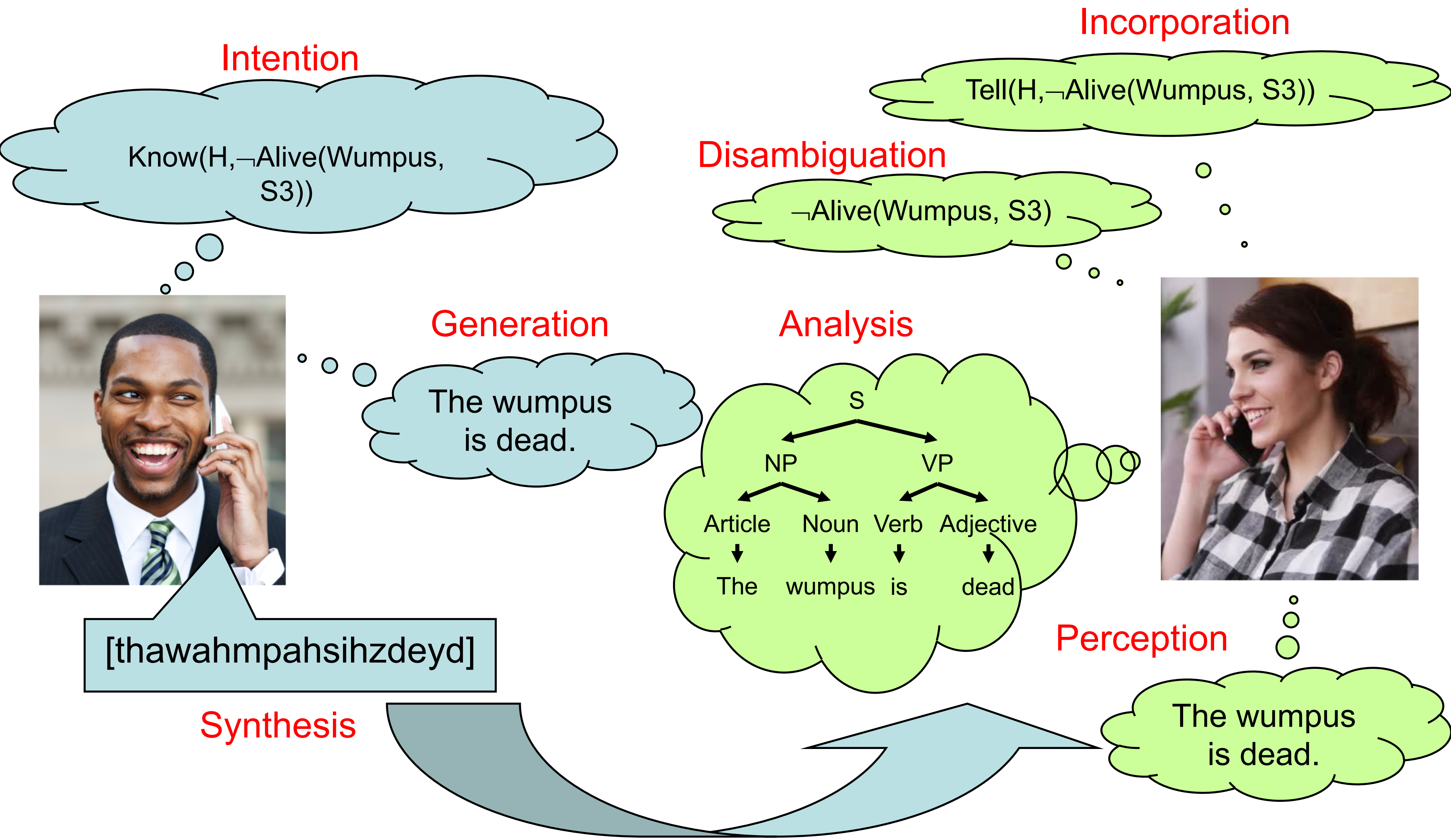
Time to switch gears…

- We started by talking about general-purpose systems
  - Search
  - Logic
  - Planning
  - Machine Learning
- Now we are starting to focus on things that are much more special-purpose task domains

Communication

- Intentional exchange of information brought about by the production and perception of signs drawn from a shared system of conventional signs
- Formal languages: LISP, FOPC, C++
- Natural languages: Danish, German, ASL
  - Vervets, dolphins, bees, and humans

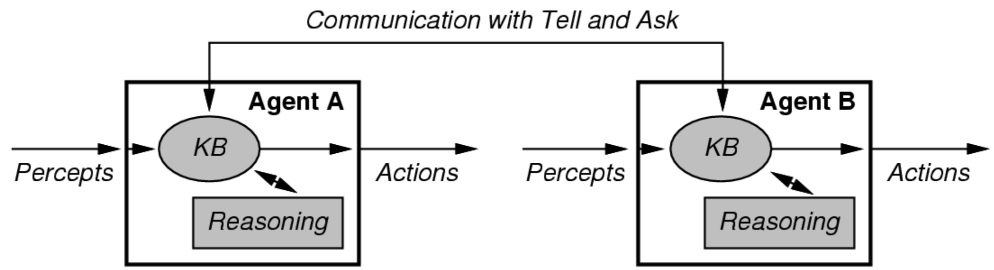
Component Steps of Communication



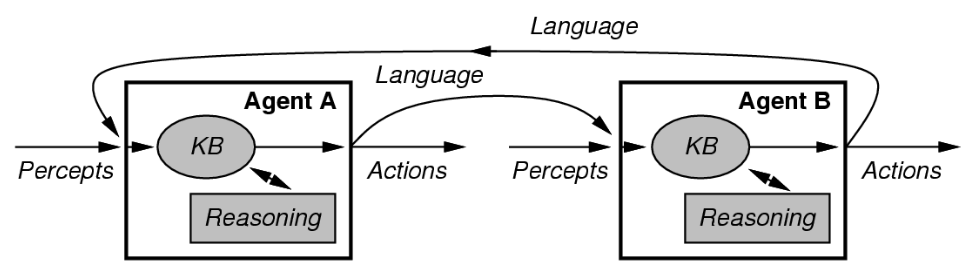
Communication as Perception and Action

- Agents that can share information from one KB to another
  - Difficulties of matching context and background knowledge
- Situated language model: Agents that can share information using formal languages

Telepathic Agents



Agents using a Formal Language



Where AI Traditionally Starts: Syntax and Grammars

- Terminal symbols
  – Could be words, phonemes, letters, etc
  – Traditionally lower-case letters
- Non-terminal symbols
  – Categories that stand for some expansion into terminal symbols
  – Noun phrase (NP), verb phrase (VP), sentence (S)
  – Traditionally upper-case letters
- Combine terminal and non-terminal symbols using rewrite (or production) rules

  $S \rightarrow NP\ VP$

A sample English-based grammar for the Wumpus world

- Backus-Naur Form
- Lexicon (list of vocabulary words)
  - Terminals

Noun → *stench* | *breeze* | *glitter* | *pit* | *wumpus* | …
Verb → *is* | *see* | *smell* | *shoot* | *grab* | *turn* | *kill* | …
Adjective → *right* | *left* | *east* | *south* | *back* | *smelly* | …
Adverb → *here* | *there* | *nearby* | *ahead* | *east* | *south* | …
Pronoun → *me* | *you* | *it* | …
Article → *the* | *an* | *a*
Preposition → *to* | *in* | *on* | *over* | *under* | …
Conjunction → *and* | *or* | *but* | …

A sample English-based grammar for the Wumpus world

- Categories (phrase structure)
- Non-terminals

S → NP VP
S → S Conjunction S
NP → Pronoun
NP → Noun
NP → NP PP
NP → NP RelClause
…
VP → Verb
VP → Verb Adjective
VP → VP PP
…
RelClause → *that* VP

I + feel a breeze
I feel a breeze + and + I smell a wumpus
I
Wumpus
The wumpus to the east
The wumpus that is smelly
…
Stinks
Is + smelly
Turn + to the east
…
That + is smelly

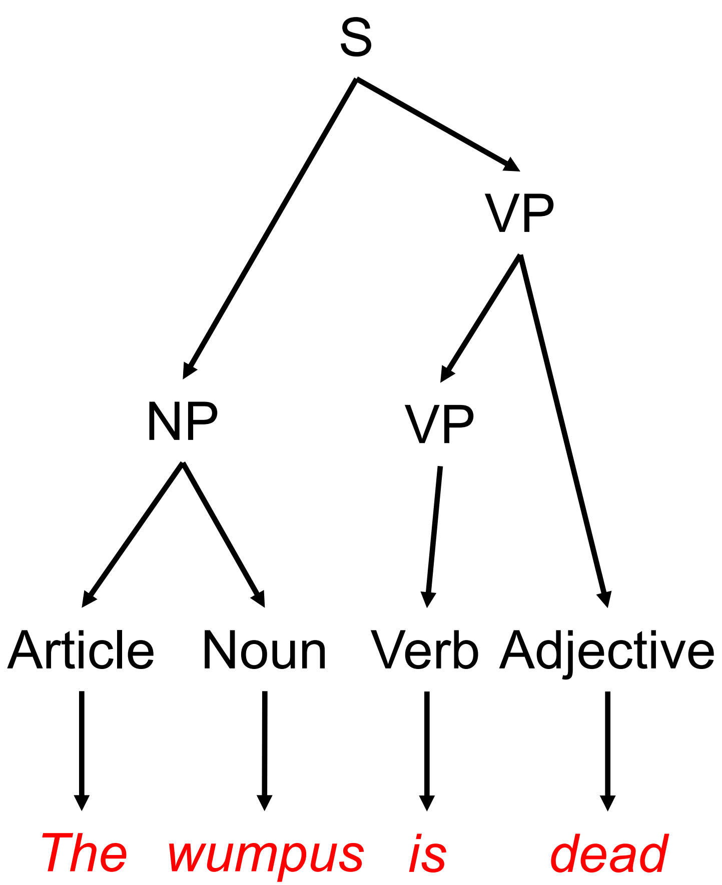
Bottom-Up Parsing

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

  forest ← words
  loop do
     if LENGTH(forest) = 1 and CATEGORY(forest[1]) = START(grammar) then
        return forest[1]
     else
        i ← choose from {1...LENGTH(forest)}
        rule ← choose from RULES(grammar)
        n ← LENGTH(RULE-RHS(rule))
        subsequence ← 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
```

- Treat the list of words as a parse forest (an ordered list of parse trees)
- Non-deterministically find some rule that matches a subsequence of words/symbols

Bottom-Up Parsing Example



Forest	**Rule being applied**
The wumpus is dead	Article → *the*
Article *wumpus is dead*	Noun → *wumpus*
Article Noun *is dead*	NP → Article Noun
NP *is dead*	Verb → *is*
NP Verb *dead*	Adjective → *dead*
NP Verb Adjective	VP → Verb
NP VP Adjective	VP → VP Adjective
NP VP	S → NP VP
S	

Generative Capacity of a Grammar

Recursively Enumerable

- No restrictions on the grammar
- A B → C
- Sample language: Any

Context-Free Grammars

- LHS consists of only a single non-terminal
- S → a S b
- Sample language: $a^n b^n$

Context-Sensitive

- RHS must have at least as many symbols as LHS
- A B → B A
- Sample language: $a^n b^n c^n$

Regular Grammars

- LHS is single non-terminal, RHS is terminal + optional non-terminal
- S → a S
- Sample language: $a^* b^*$

Adding Meaning to a Syntax

- Backus-Naur form describes the syntax, but tells us nothing about the meaning
- Resort to a logic grammar for semantics

Backus-Naur Form	**First-Order Logic**
$S \rightarrow NP\ VP$	$NP(s1) \wedge VP(s2) \Rightarrow S(Append(s1,s2))$
Noun → *stench* | …	(s="stench" ∨ …) ⇒ Noun(s)

- Unrestricted logical inference is too expensive
- Definite Clause Grammar (DCG): every sentence is a Horn clause with one atom in the consequent $\boldsymbol{A \wedge B \wedge C \wedge \ldots \Rightarrow X}$

Augmenting a Grammar

- Our current grammar overgenerates non-grammatical sentences
  - Me smells a stench
- Handling subjective and objective cases: create more non-terminals

$S \rightarrow NP_S\ VP$

$NP_S \rightarrow Pronoun_S \mid Noun \mid Article\ Noun$

$NP_O \rightarrow Pronoun_O \mid Noun \mid Article\ Noun$

$VP \rightarrow VP\ NP_O \mid \ldots$

$PP \rightarrow Preposition\ NP_O$

$Pronouns_S \rightarrow$ *I* | *you* | *he* | *she* | …

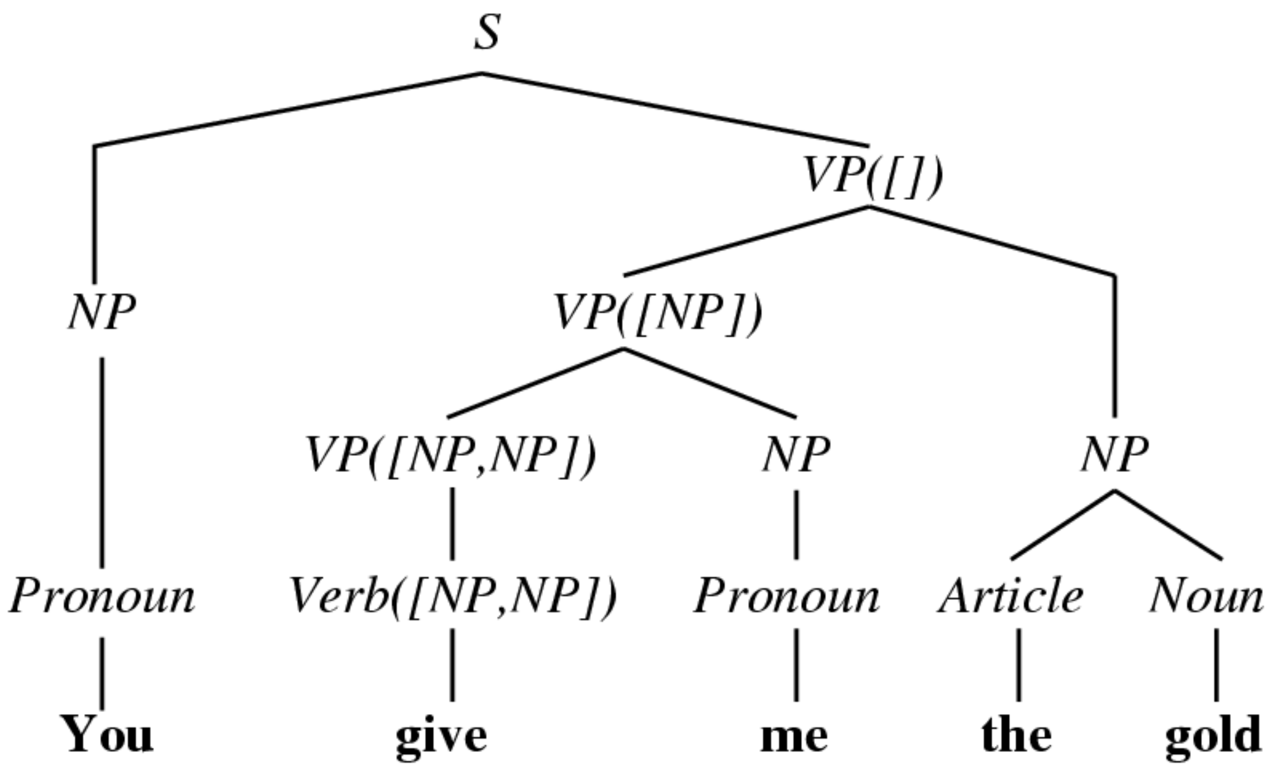
$Pronouns_O \rightarrow$ *me* | *you* | *him* | *her* | …

Augmenting a Grammar

Verb	Subcats	Example Verb Phrase
give	[*NP*, *PP*] [*NP*, *NP*]	give the gold in 3 3 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 smelly wumpus in 2 2 is dead

- Subcategorization gives the types of structures that follows a symbol
- Example
  - *Give* is followed by a NP and a PP or a NP and a NP

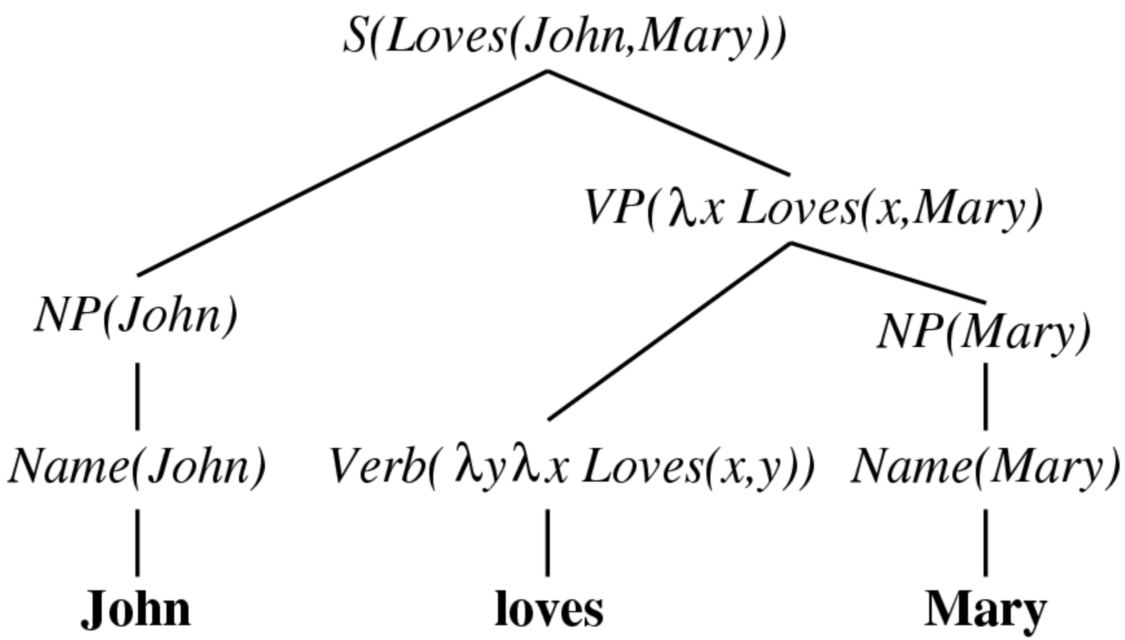
Parsing using Subcategorizations



- Subcats restrict the selection of other symbols, but can just be seen as a specialization of a symbol

  VP([]) → VP([NP]) NP

Parsing with Semantics



- Using lambda-notation, we define the semantic content as a type of subcat

  Verb(λx λy Loves(x,y)) → *loves*

Quasi-Logical Forms

- An intermediary between syntactic structure and first-order logic for semantics that allows parsing
- Includes
  - All of first-order logic notation
  - Lambda expressions
  - Quantified terms
    - Looks like a logical sentence, but treated like a term
    - "every agent" is quantified as [$\forall$a Agent(a)]

Quasi-Logical form for the Wumpus Grammar

Category	Type	Example	Quasi-Logical Form
S	Sentence	I sleep.	$\exists e \;\; e \in (Sleep, Speaker) \wedge During(Now, e)$
Adjective	$object \rightarrow sentence$	smelly	$\lambda x \; Smelly(x)$
Adverb	$event \rightarrow sentence$	today	$\lambda e \; During(e, Today)$
Article	Quantifier	the	$\exists !$
Conjunction	$sentence^2 \rightarrow sentence$	and	$\lambda p, q \; (p \wedge q)$
Digit	Number	7	7
Noun	$object \rightarrow sentence$	wumpus	$\lambda x \; Wumpus(x)$
Preposition	$object^2 \rightarrow sentence$	in	$\lambda x \; \lambda y \; In(x, y)$
Pronoun	Object	I	*Speaker*
Verb	$object^n \rightarrow sentence$	eats	$\lambda y \; \lambda x \; \exists e \;\; e \in Eats(x, y) \wedge During(Now, e)$
NP	Object	a dog	$[\exists d \; Dog(d)]$
PP	$object^2 \rightarrow sentence$	in [2,2]	$\lambda x \; In(x, [2,2])$
RelClause	$object \rightarrow sentence$	that sees me	$\lambda x \; \exists e \;\; e \in Sees(x, Speaker) \wedge During(Now, e)$
VP	$object^n \rightarrow sentence$	sees me	$\lambda x \; \exists e \;\; e \in Sees(x, Speaker) \wedge During(Now, e)$

Parsing with Syntax and Semantics

Every **agent** **smells** **a** **wumpus**

Pragmatics

- The study of language as it is used in a social context, including its effect on the agents involved
- Indexicals: refer directly to the current situation
  - Pragmatics of "*I* am in Boston *today*."
- Anaphora: refer to something mentioned previously
  - Pragmatics of "John was hungry. *He* ate a carrot."

Ambiguity

- Ambiguous newspaper headlines
  - Squad helps dog bite victim.
  - Red-hot star to wed astronomer.
  - Helicopter powered by human flies.
  - American pushes bottle up Germans.
- Many places that ambiguity can arise
  - Lexical ambiguity (*star* has more than one meaning)
  - Syntactic ambiguity (is *dog* an adjective or a noun)
  - Semantic ambiguity (A *coast road* can either lead to the coast or run along the coast)
  - Pragmatic ambiguity (*I'll meet you next Wednesday*… is *Wednesday* two days or nine days away?)

Disambiguation

- Two approaches
- Model-based
  - Rely upon the contents of the knowledge base and a model of how the world works (most of the time) to disambiguate
- Statistical
  - Probabilistic context-free grammar

    S → NP VP   (90%)

    S → S Conjunction S   (10%)
- In general, we don’t have a good way to do this

Current Systems

- Syntax
  – Not too hard
  – Requires large grammars for natural languages
- Semantics
  – Difficult, but possible
  – Works best in restricted domains
- Pragmatics
  – Somewhat ignored, very difficult

Google Duplex

May 8, 2018 – Sundar Pichai https://www.youtube.com/watch?v=D5VN56jQMWM

Reasons to be Skeptical

- Lack of ambient background noise
- Odd conversation
  - The businesses *never* identify themselves
  - The humans picking up the phone never give their names
  - The reservation-takers never request information. No contact phone number. No name.
- California’s 2-party consent laws
- No response from Google

Administrivia

- PS 6 out now
- Friday: HMMs and more!