Speech Acts

Speech acts achieve the speaker’s goals:

- Inform: “There’s a pit in front of you.”
- Query: “Can you see the gold?”
- Command: “Pick it up.”
- Promise: “I’ll share the gold with you.”
- Acknowledge: “OK.”

Speech act planning requires knowledge of

- Situation
- Semantic and syntactic conventions
- Hearer’s goals, knowledge base, and rationality

Stages in communication (informing)

- Intention: S wants to inform H that P
- Generation: S selects words W to express P in context C
- Synthesis: S utters words W
- Perception: H perceives W in context C’
- Analysis: H infers possible meanings P1, …, Pn
- Disambiguation: H infers intended meaning Pi
- Incorporation: H incorporates Pi into KB

How could this go wrong?

- Insincerity: S doesn’t believe P
- Speech wreck ignition failure
- Ambiguous utterance
- Differing understanding of current context C<>C’
Stages in communication (informing)

Speech Acts

- Animals use isolated symbols for sentences
  - restricted set of communicable propositions, no generative capacity
- Grammar specifies the compositional structure of complex message (speech, language, music)
- A formal language is a set of strings of terminal symbols
- Each string in the language can be analyzed/generated by the grammar
- The grammar is a set of rewrite rules,
  - e.g., $S \rightarrow NP \ VP$
  - $Article \rightarrow \text{the} \ | \ a \ | \ an \ | \ ...$
  - Here $S$ is the sentence symbol, $NP$ and $VP$ are non-terminals, and the, a ... terminals

Grammar Types

Regular: $\text{nonterminal} \rightarrow \text{terminal} | \text{nonterminal}$

- $S \rightarrow aS$
- $S \rightarrow \Lambda$

Context-free: $\text{nonterminal} \rightarrow \text{anything}$

- $S \rightarrow aSB$

Context-sensitive: more nonterminals on right-hand side

- $ASB \rightarrow AAaBB$

Recursively enumerable: no constraints

Natural language probably context-free

The Wumpus Lexicon

- Noun $\rightarrow$ stench | breeze | glitter | nothing
  - wumpus | pit | pits | gold | east | ...
- Verb $\rightarrow$ is | see | smell | shoot | feel | stinks
  - go | grab | carry | kill | turn | ...
- Adjective $\rightarrow$ right | left | east | south | back | smelly | ...
- Adverb $\rightarrow$ here | there | nearby | ahead
  - right | left | east | south | back | ...
- Pronoun $\rightarrow$ me | you | I | it | $S/HE$ | $Y'ALL$
- Name $\rightarrow$ John | Mary | Boston | UCB | PAJC | ...
- Article $\rightarrow$ the | a | an | ...
- Preposition $\rightarrow$ to | in | on | near | ...
- Conjunction $\rightarrow$ and | or | but | ...
- Digit $\rightarrow$ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Divided into open and closed classes
The Wumpus Grammar

\[
S \rightarrow NP \ VP \\
| S\ Conjunction\ S \\
NP \rightarrow \text{Pronoun} \\
| \text{Noun} \\
| \text{Article\ Noun} \\
| \text{Digit\ Digit} \\
| NP\ PP \\
| NP\ RelClause \\
VP \rightarrow \text{Verb} \\
| VP\ NP \\
| VP\ Adjective \\
| VP\ PP \\
| VP\ Adverb \\
PP \rightarrow \text{Preposition\ NP} \\
RelClause \rightarrow \text{that} \ VP \\
\]

Grammatical Judgments

Formal language $L_1$ may well differ from natural language $L_2$.

Adjusting is a learning problem.

Parse Trees

Exhibit the grammatical structure of a sentence.

Syntax in NLP

Most view syntactic structure as an essential step towards meaning.

“Mary hit John” <> “John hit Mary”

“And since I was not informed---as a matter of fact, since I did not know that there were excess funds until we, ourselves, in that checkup after the whole thing blew up, and that was, if you’ll remember, that was the incident in which the attorney general came to me and told me that he had seen a memo that indicated that there were no more funds.”
**Parsing CFGs**

Many different parsing algorithms

- Top-down
- Bottom-up
- Chart-parsing (aka CYK algorithm)
  - Dynamic programming
  - \(O(n^3)\)

\[\text{S: [NP:[Pronoun:I]]} \]
\[\text{[VP:[VP:[Verb:shoot]] [NP:[Article:the][Noun:wumpus]]]}\]

**Top-down parsing**

- Initial state: \([S: ?]\)
- Successor function:
  - Select rules for leftmost node in tree with unknown children and apply
  - \([S:[S:?][Conjunction:?][S:?]]\)
  - \([S:[NP:?][VP:?]]\)
- Goal Test:
  - Check whether leaves of the parse tree correspond to (complete) input

Problem for Top-down: left-recursive rules \(S \rightarrow S\) Conjunction \(S\)

infinite loops

**Bottom-up parsing**

- Initial state
  - \([\text{the, wumpus, is, dead}]\)
- Successor function
  - If subsequence at pos \(i\) matches right-hand of rule then replace by left hand
  - \([[\text{Art:the}],\text{wumpus, is, dead}]]\)
- Goal state:
  - A single state with root \(S\)

Problem for Bottom up:

Art \(\rightarrow []\)
Bottom-up parsing

- [the, wumpus, is, dead]
- [[Art:the], [Noun:wumpus], [is,dead]]
- [[NP:[Art:the][Noun:wumpus]], [is,dead]]
- [[NP:[Art:the][Noun:wumpus]], [Verb:is], [dead]]
- [[NP:[Art:the][Noun:wumpus]], [VP:[Verb:is]], [Adj:dead]]
- [[NP:[Art:the][Noun:wumpus]], [VP:[VP:[Verb:is]]], [Adj:dead]]
- S: [[NP:[Art:the][Noun:wumpus]], [VP:[VP:[Verb:is]]], [Adj:dead]]

Chart parsing

- Have the students in B.Sc. Informatik take the exam of AI.
- Have the students in B.Sc. Informatik taken the exam of AI?
- Double work
  - Dynamic programming - combine bottom-up and top-down
- Chart :
  - N+1 vertices
  - Labeled edges, e.g., [0,2 S -> NP * VP]
    - Denotes that from 0 to 2 we have a NP, and if we find a VP, from 2 to k
      then we have an S from 0 to k

Chart parsing

- Initialization
  - Add [0,0 S' -> * S] non-terminal - upper case
- Add edge [i,j A -> B * c] sequence - lower case
  - if c is empty then call extender
  - else call predictor
- Predictor [i,j] A -> b * C e]
  - With (all) rules C -> d
    - Add edge [i,j A -> b * d e]
- Extender [j,k B -> c *]
  - With (all) edges [i,j A -> e * B f]
  - Add edge [i,k A -> e c * f]
- Scanner [j,k A -> c * D e]
  - Word of type D at position k
  - Add edge [j,k+1 A -> c D * e]
Definite Clause Grammars

- A form of unification based grammar
- Non-terminals become atomic expressions
  - s(Num) → np(Num), vp(Num)
  - Num is a variable
  - Binds to singular and plural
- Employ unification during parsing.
- Directly executable in Prolog
- Rules directly translate to (definite) clause logic
- Next slides employ Prolog notation
  - Running examples with, e.g., YAP Prolog or SWI Prolog
Non-terminals with arguments

sentence  --> noun_phrase(N), verb_phrase(N).
noun_phrase  --> article(N), noun(N).
verb_phrase(N)  --> intransitive_verb(N).
article(singular)  --> [a].
article(plural)  --> [the].
noun(singular)  --> [turtle].
noun(plural)  --> [turtles].
intransitive_verb(singular)  --> [sleeps].
intransitive_verb(plural)  --> [sleep].

Case Marking

pronoun(singular,nominative)  --> [he]; [she]
pronoun(singular,accusative)  --> [him]; [her]
pronoun(plural,nominative)  --> [they]
pronoun(plural,accusative)  --> [them]
sentence  --> np(Number,nominative), vp(Number)
vp(Number)  --> v(Number), np(Number,accusative)
np(Number,Case)  --> pronoun(Number,Case)
np(Number,X)  --> det, n(Number)

He sees her. She sees him. They see her. But not Them see he.

Top-down parsing with DCG

sentence  [He,sees,her]
np(Num,nom), vp(Num)  [He,sees,her]
pronoun(Num,nom), vp(Num)  [He,sees,her]
    Num = sing !!!  pronoun(sing,nom)  --> [he]
vp(sing)  [sees,her]
v(sing), np(sing,accusative)  [sees,her]
np(sing,accusative)  [her]
pronoun(sing,accusative]  [her]
[/]  [/]

Sub-categorization

vp  --> v(1).  v(1)  --> [sleep]
vp  --> v(2), np.  v(2)  --> [chase]
vp  --> v(3), np, np.  ...
vp  --> v(4), s.

Verb  Complement  Example
Sleep  None  The cat slept
Chase  One NP  The cat chased the dog
Give  Two NP  John gave Bill the book
Say  sentence  John said he loved Mary.
Constructing parse trees

sentence(s(NP,VP)) --> noun_phrase(NP), verb_phrase(VP).
noun_phrase(np(N)) --> proper_noun(N).
noun_phrase(np(Art,Adj,N)) --> article(Art), adjective(Adj), noun(N).
verb_phrase(np(Art,N)) --> article(Art), noun(N).
verb_phrase(vp(IV)) --> intransitive_verb(IV).
verb_phrase(vp(TV,NP)) --> transitive_verb(TV), noun_phrase(NP).
article artikel (the) --> the.
adjective (adj) (laz) --> lazy.
adjective (adj) (rap) --> rapid.
proper_noun (pn) (achilles) --> achilles.
noun (n) (turtle) --> turtle.
intransitive_verb (iv) (sleeps) --> sleeps.
transitive_verb (tv) (beats) --> beats.

Semantic Interpretation

np(fido) --> [fido].
np(felix) --> [felix].
v(X^slept(X)) --> [slept].
v(Y^X^chased(X,Y)) --> [chased].
s(Pred) --> np(Subj), vp(Subj^Pred).
vp(Subj^Pred) --> v(Subj^Pred).
np(Subj) --> np(Subj^Pred), vp(Obj^Pred), np(Obj).

Lambda Expressions

X^slept(X) stands for λX. slept(X)
Y^X^chased(X,Y) stands for λY λX. chased(X,Y)

These lambda-expressions denote relationships
They can be instantiated by unification

With unification
X^slept(X) with X=fido yields fido^slept(fido)
unify fido^slept(fido) with X^R
Yields R=slept(fido)

In lambda-calculus
(λX. slept(X)) fido gives slept(fido)

Natural Language Processing

- Complex process
  - Requiring knowledge, reasoning and AI in general
- Grammars to represent natural languages
- Parsing techniques for syntactic analysis
- DCG add expressive power to CFGs
  - Convenient in practice
  - Unification based grammars