Cognitive Modeling

Lecture 7: Models of Syntactic Processing

Sharon Goldwater

School of Informatics
University of Edinburgh
sgwater@inf.ed.ac.uk

February 1, 2010

Reading: Cooper (2002: Ch. 7).

Introduction

Linguistics deals with:

- **phonology:** the sounds of the language;
- **syntax:** the structure of sentences (word order, etc.);
- **semantics:** the meaning of sentences;
- **pragmatics:** the use of language in context; non-literal meaning.

Psycholinguistics studies the comprehension and production of language on all these level.

Here we will focus on *syntactic processing* (aka sentence processing, parsing); assume words are known.

A Small Grammar of English

Phrase markers:

S: sentence, NP: noun phrase, VP: verb phrase

Syntactic categories (aka parts of speech):

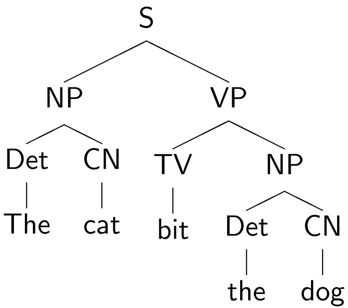
Det: determiner, CN: common noun, TV: transitive verb

Phrase structure rules:

S	$\rightarrow$	NP VP	Det	$\rightarrow$	the
NP	$\rightarrow$	Det CN	CN	$\rightarrow$	cat
VP	$\rightarrow$	TV NP	TV	$\rightarrow$	bit
			CN	$\rightarrow$	dog

Syntax Tree

The grammar is used to generate syntax trees for input sentences:



Crucially, the tree is assumed to be necessary for interpretation, and *different structures lead to different semantic interpretations*.

More Phrase Structure Rules

NP	→	Pro	*pronoun* (I, him)
NP	→	PN	*proper name* (Sarah, Edinburgh)
PP	→	Prep NP	*prepositional phrase* (on the table)
VP	→	IV	*intransitive verb* (sleep, dance)
VP	→	DV NP NP	*ditransitive verb* (give, pronounce)
VP	→	DV NP PP	*ditransitive verb with PP complement* (give, put)
VP	→	V_{inf1} VP(inf)	*verb with infinitival complement*
VP	→	V_{inf2} NP VP(inf)	*verb with NP and infinitival complement* (want, ask)
VP(inf)	→	INF VP	*infinitival VP* (to go)
S(comp)	→	Comp S	*complement sentence* (that S)

Competence vs. Performance

Competence: the linguistic knowledge that a speaker has; formalized, e.g., using phrase structure rules.

Performance: the application of the linguistic knowledge in comprehending and producing language.

Competence is idealized, while performance is subject to cognitive constraints (e.g., memory limitations, fatigue).

Psycholinguistics deals with performance (competence is the domain of linguistic theory).

We will focus on the *Human Sentence Processing Mechanism* (HPSM), i.e., the cognitive device that assigns a syntactic structure to a string of words.

Competence vs. Performance as different levels of analysis?

Recall Marr's (1982) *three levels of analysis*:

- **Computational theory:** What is the goal of the computation and the logical strategy needed to carry it out?
- **Representation and algorithm:** How can the computation be implemented, and what input/output representations are needed?
- **Hardware implementation:** What is the physical realization of the algorithm?

Can view linguistic theory (competence) as making claims about representation and computational level; psycholinguistics (performance) as more concerned with algorithmic processes.

Incrementality

Parsing: extracting syntactic structure from a string; prerequisite for assigning a meaning to the string.

The sentence processor builds structures *incrementally* (word by word) as the input comes in (Tanenhaus et al. 1995).

This can lead to *local ambiguity.*

Example:

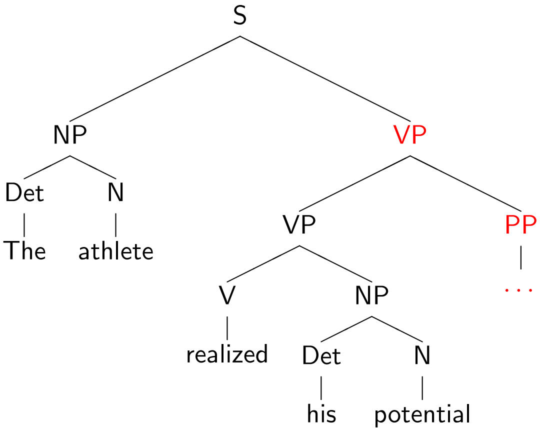
(1) The athlete realized his potential . . .

a. . . . at the competition.

b. . . . could make him a world-class sprinter.

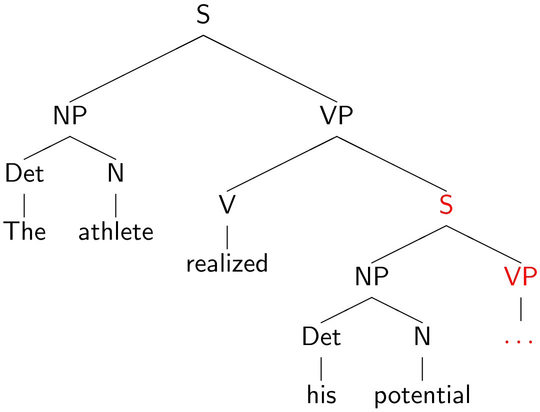
Incrementality

Structure 1 (NP reading):



Incrementality

Structure 2 (S reading):



Garden Paths

- *Early commitment:* when it reaches *potential*, the processor has to decide which structure to build.
- If the parser makes the wrong choice (e.g., NP reading for sentence (1-b)) it needs to backtrack and revise the structure.
- A *garden path* occurs, which typically results in longer reading times (and reverse eye-movements).
- Some garden paths are so strong that they parser fails to recover from them.

Garden Paths

More examples of garden paths:

(2)
a. I convinced her children are noisy.
b. Until the police arrest the drug dealers control the street.
c. The old man the boat.
d. We painted the wall with cracks.
e. Fat people eat accumulates.
f. The cotton clothing is usually made of grows in Mississippi.
g. The prime number few.

Dimensions of Parsing

In addition to incrementality, a number of properties are important when designing a model of the HPSM:

- **Directionality:** the parser can process a sentence bottom-up (from the words up) or top-down (from the phrase markers down). *Evidence that the HPSM combines both strategies.*
- **Parallelism:** a serial parser maintains only one structure at a time; a parallel parser pursues all possible structures. *Controversial issue; proponents for both serialism and limited parallelism.*
- **Interactivity:** the parser can be encapsulated (only access to syntactic information) or interactive (access to semantic information, context). *Evidence for limited interactivity.*

An Incremental Input Module

We first need to create an input module that presents one stimulus word at a time:

Rule 1: Select a sentence to parse for the Stimuli buffer:

IF the current cycle is 1
  once `WordList` is in **Stimuli**
THEN delete `WordList` from **Stimuli**
  add `words(WordList)` to **Current Stimulus**

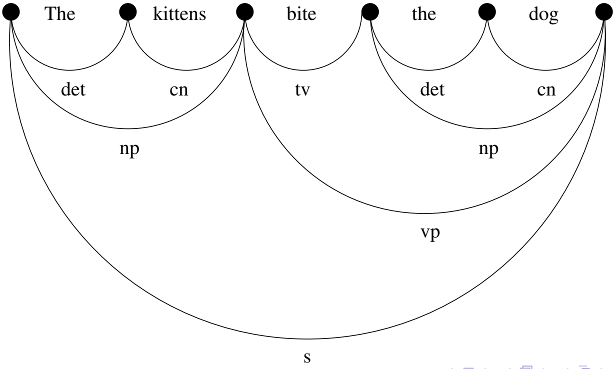
Rule 2: When quiescent, feed one more word to the subject:

TRIGGER `system_quiescent`
IF `words([Head|Tail])` is in **Current Stimulus**
THEN delete `words([Head|Tail])` from **Current Stimulus**
  add `words(Tail)` to **Current Stimulus**
  send `word(Head)` to **Subject:Input/Output**

A Bottom-Up Parallel Parser

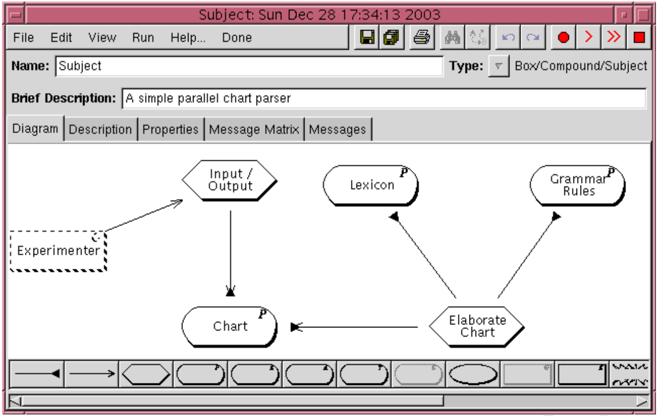
The parser constructs a *chart,* a compact representation of all the analyses of a sentence.

Goal: find an S edge that spans the whole sentence. Example:



A Bottom-Up Parallel Parser

Architecture of a simple parser that constructs the chart bottom-up:



Chart, Lexicon, Grammar Rules

The chart edges are represented as predicates of the form:

`edge(LeftVertex,RightVertex,Content,Level)`

where `LeftVertex` and `RightVertex` are integer vertex labels, `Content` is the content of the edge (e.g., `word(cat)`) and `Level` is formatting information (not discussed here).

Examples for items in the lexicon:

```
category(the,det)
category(kittens,cn)
```

Examples for grammar rules:

```
rule(s,[np,vp])
rule(np,[pn])
```

Input/Output Process

Rule 1: Add a word to the first position of the chart:

TRIGGER `word(W)`
IF not `edge(_,_,_,_)` is in **Chart**
THEN add `edge(0,1,word(W),0)` to **Chart**

Rule 2: Add a word to the next position of the chart:

TRIGGER `word(W)`
IF `edge(N0,N1,word(W1),Y)` is in **Chart**
  not `edge(N1,N2,word(W2),Y)` is in **Chart**
  `N2` is `N1 + 1`
THEN add `edge(N1,N2,word(W),Y)` to **Chart**

Elaborate Chart Process

Rule 1: Lexical look-up:

IF `edge(N0,N1,word(W),L1)` is in **Chart**
  `category(W,C)` is in **Lexicon**
  `L` is `L1 + 1`
THEN add `edge(N0,N1,cat(C),L)` to **Chart**

Rule 2: Apply unary grammar rules:

IF `edge(N0,N1,cat(C1),L1)` is in **Chart**
  `rule(C,[C1])` is in **Grammar Rules**
  `L` is `L1 + 1`
THEN add `edge(N0,N1,cat(C),L)` to **Chart**

Elaborate Chart Process

Rule 3: Apply binary grammar rules:

IF `edge(N0,N1,cat(C1),L1)` is in **Chart**
`edge(N1,N2,cat(C2),L2)` is in **Chart**
`rule(C,[C1,C2])` is in **Grammar Rules**
`L` is `max(L1,L2) + 1`
THEN add `edge(N0,N2,cat(C),L)` to **Chart**

Similar rules for grammar rules with more than two categories.

Properties of the Model

Simple, but complete chart parser with the following properties:

- **bottom-up:** parsing is driven by the addition of words to the chart; chart is expended upwards from lexical to phrasal categories;
- **limited incrementality:** when a new word appears, all possible edges are added to the chart; then the system quiesces and waits for the next word;
- **parallelism:** all chart edges are added at the same time (default Cogent behavior); multiple analyses are pursued.

Summary

- The human parser builds syntactic structure in response to strings of words;
- parsing models have to capture the incrementality of human parsing and account for ambiguity resolution (garden paths);
- parsing models can be implemented in Cogent using a chart (representing partial syntactic structure);
- simple parsing model based on Cogent's default behavior;
- assumes limited incrementality, full parallelism: not cognitively plausible;
- *next lecture:* serial left corner parsing model.

References

Cooper, Richard P. 2002. *Modelling High-Level Cognitive Processes*. Lawrence Erlbaum Associates, Mahwah, NJ.

Marr, D. 1982. *Vision: A Computational Approach*. Freeman & Co., San Francisco.

Tanenhaus, Michael K., Michael J. Spivey-Knowlton, Kathleen M. Eberhard, and Julie C. Sedivy. 1995. Integration of visual and linguistic information in spoken language comprehension. *Science* 268:1632–1634.