Models of Syntactic Processing

LaLoCo, Fall 2013

Karl DeVries, Adrian Brasoveanu

[based on slides by Sharon Goldwater & Frank Keller]

A Small Phrase Structure (PS) Grammar of English

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

Syntactic categories (aka Parts of Speech):
Det: determiner, CN: common noun, TV: transitive verb

Phrase structure (PS) rules:

S	→	NP VP	Det	→	the
NP	→	Det CN	CN	→	kitten
VP	→	TV NP	CN	→	dog
			TV	→	bit

Derivations and Parse Trees

A derivation is the sequence of strings that results from applying a sequence of PS rules, starting from a start symbol, here S.

In a PSG derivation, only one symbol is rewritten per step.

Derivation 1

S ⇒ NP VP ⇒ NP TV NP ⇒ NP TV Det CN ⇒ NP bit Det CN ⇒ NP bit Det dog ⇒ NP bit the dog ⇒ . . .

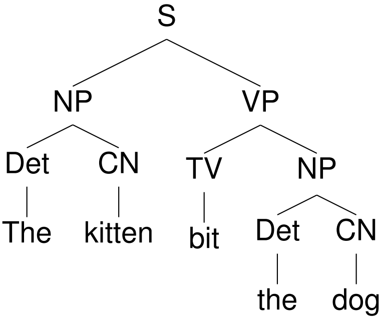
Derivation 2

S ⇒ NP VP ⇒ NP TV NP ⇒ NP TV Det CN ⇒ Det CN TV Det CN ⇒ The CN TV Det CN ⇒ The CN TV the CN ⇒ . . .

The order in which symbols are rewritten does not matter in these PSG derivations.

Syntax Tree

Syntactic trees allow us to represent such equivalent derivations in a simple way.



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

Competence vs. Performance

With respect to language structure, we can distinguish between

- Competence: The linguistic knowledge that an (ideal) speaker and/or hearer has; formalized using e.g. phrase structure rules.
- Performance: The application of linguistic knowledge in processing (comprehending or producing) language.

The distinction comes from Noam Chomsky's seminal 1957 monograph *Syntactic Structures*.

Competence vs. Performance

Competence:

- Typically studied by formal linguists.
- Which sentences are grammatical/ungrammatical and what is the grammar that captures these facts?
- An idealization: grammatical sentences can be so long or convoluted that no real person would have enough memory or time to process them.

Performance:

- Typically studied by psycholinguists.
- Which sentences are hard to understand/generate and why?
- How does linguistic knowledge interact with other cognitive factors (memory, attention, age, etc.)?

Competence vs. Performance as different levels of analysis?

Recall Marr (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.

Human sentence processing

In syntax, performance can be studied with respect to either:

- Understanding: How humans infer syntactic structure (eventually, meaning) from a string of words.
- Generation: How humans go from meanings and/or syntactic structures to produce sentences (along with errors such as false starts, hesitations, fillers, spoonerisms, etc.)

Both are rich areas; here we focus on modeling the Human Sentence Processing Mechanism (HSPM), the cognitive device involved in syntactic parsing for understanding.

Incrementality

Parsing: Computing one or more structures for a string, given a grammar.

Like word recognition, parsing is incremental: the HSPM must build structures word by word as the input arrives (Tanenhaus et al., 1995).

Problems occur if more than one structure is compatible with the input either

- at the current point but not later (local ambiguity);
- for the input overall (global ambiguity).

Global ambiguity

Given a grammar, strings that have more than one complete syntax tree (parse) are said to have global structural ambiguity.

Examples:

1. She sat on the chair covered in dust.
2. He saw the man with the telescope.
3. Kids make nutritious snacks.
4. Milk drinkers are turning to powder.
5. Old school pillars are replaced by alumni.

Global ambiguity is a problem even for non-incremental parsers.

Exs. 3-5 from http://www.fun-with-words.com/ambiguous_garden_path.html

Local ambiguity

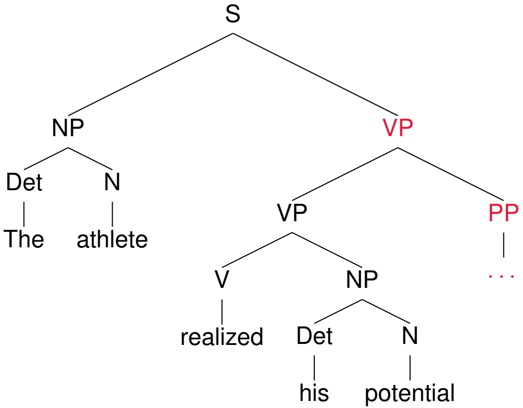
When only an initial substring is structurally ambiguous, the sentence is said to have local structural ambiguity.

- Once the remainder of the string is known, only one tree remains possible.
- Local ambiguity is a problem only for incremental parsers.

Example:

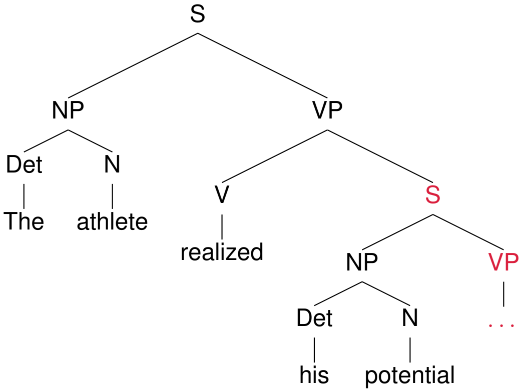
1. The athlete realized his potential . . .
   a. . . . at the competition.
   b. . . . could make him a world-class sprinter.

Local ambiguity: Structure 1 (VP → V NP)



The athlete realized his potential at the competition.

Local ambiguity: Structure 2 (VP → V S)



The athlete realized his potential could make him a world-class sprinter.

Garden Paths

- Both structures are compatible with the input from realized through potential; only the next word disambiguates.
- In many cases local ambiguity causes no apparent difficulty; other times it causes a garden path.
- A garden path is said to occur when the processor apparently commits to a single (wrong) structure early on, causing a “dead end” parse when later input is inconsistent with that structure. Processor must backtrack and revise the structure.
- Garden path sentences result in longer reading times and reverse eye-movements.
- Some garden paths are so strong that the parser fails to recover from them.

Garden Paths

More examples of garden paths:

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

Examples from http://www.fun-with-words.com/ambiguous_garden_path.html

Dimensions of Parsing

In addition to incrementality, three properties common to all parsers are important when designing a model of the HSPM:

- Directionality: The parser can process a sentence bottom-up (from the words up) or top-down (from the phrasal categories down). Evidence that the HSPM combines both strategies.
- Parallelism: A serial parser maintains only one structure at a time; a parallel parser pursues all/several. Still controversial: Evidence for both serialism and limited parallelism.
- Interactivity: The parser can be encapsulated (with access to only syntactic information) or interactive (with access to semantics and context). Evidence for limited interactivity.

Review: Types of Parsers

There are several types of parsers:

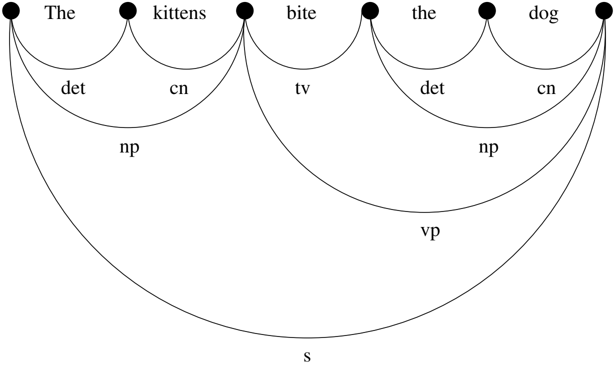
- Recursive descent parser: Top-down, serial (depth-first)
- Shift-reduce parser: Bottom-up, serial (depth-first)
- CKY chart parser: Bottom-up, parts of which can be parallelized
- Various chart parsers that combine TD and BU features, with some serial and some parallizable sub-processes.

We consider two types of parsers as models for the HSPM: a bottom-up parallel chart parser and a left-corner chart parser.

A Bottom-Up Parallel Parser

The parser constructs a chart, a compact representation of all the analyses of a sentence. Edges correspond to recognized phrases.

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



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 expanded 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 (ie, no more rules fire) and Experimenter is triggered to send the next word;
- parallelism: all chart edges are added at the same time (default Cogent behavior); multiple analyses are pursued.

Incrementality

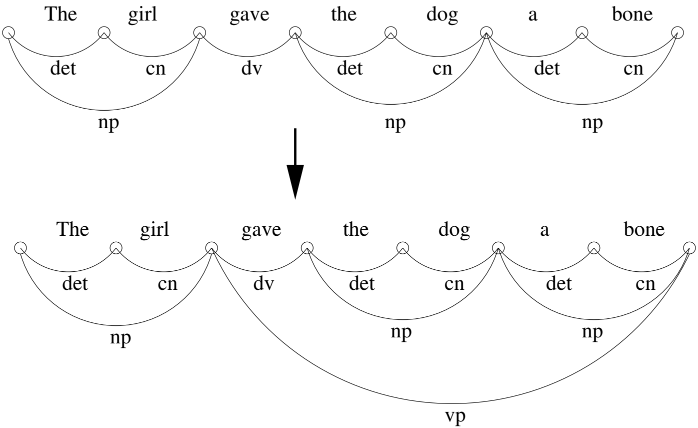
- The bottom-up parser processes each word as it appears—a limited form of incrementality.
- A fully incremental parser should maintain a fully connected parse structure at all times. This is not guaranteed by the bottom-up parser.

Consider processing “The girl gave the dog a bone”:

$$\begin{array}{rcl@{\qquad\qquad}rcl}
\text{S} & \rightarrow & \text{NP VP} & \text{Det} & \rightarrow & \text{the} \mid \text{a} \\
\text{NP} & \rightarrow & \text{Det CN} & \text{CN} & \rightarrow & \text{girl} \mid \text{dog} \mid \text{bone} \\
\text{VP} & \rightarrow & \text{TV NP} & \text{TV} & \rightarrow & \text{bit} \\
\text{VP} & \rightarrow & \text{DV NP NP} & \text{DV} & \rightarrow & \text{gave}
\end{array}$$

Disconnected structures

There are **4** disconnected structures before the rule VP → DV NP NP applies, reducing the number to **2.**



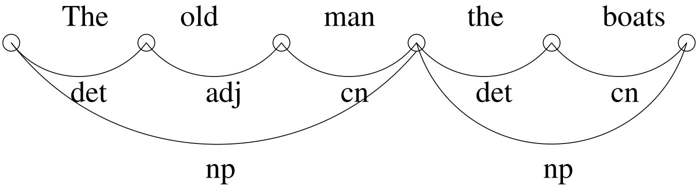
Argument for connectivity

Consider the garden path sentence “the old man the boats”.

- Assume a serial bottom-up parser (or limited parallel—key is that the correct structure is not considered initially).
- At what point does this parser realize its initial analysis is incorrect?
- At what point (intuitively) does a human realize this?

Argument for connectivity

The bottom-up parser doesn’t realize its mistake until it reaches the end of the sentence, and cannot create a full parse:



But humans recognize a problem at the second the: they have an expectation about what should come next, and it is violated.

Left Corner Parsing

Left corner parsing is more cognitively plausible: each word is immediately integrated into a single evolving structure which makes predictions about what will come next.

A left-corner parser's chart contains active edges: incomplete constituents (phrasal categories) representing predictions.

Ex: NP/CN is an incomplete constituent that will become a complete NP if a CN is seen next.

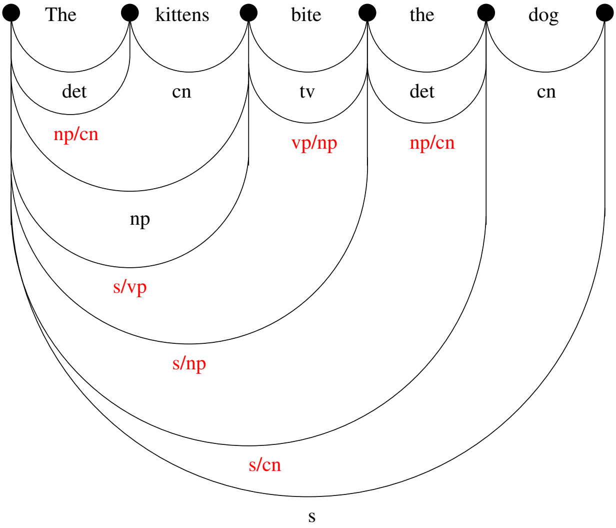
NP/CN $\approx$ dotted rule in *active chart parsing*: NP→Det . CN

Rules of Left Corner Parsing

1. Projection: For a completed edge Y and a grammar rule $X \rightarrow Y\ Z$, add an active edge X/Z, where Y and X/Z span the same part of the string.
2. Completion: For an active edge X/Y and a completed edge Y that are adjacent, add a completed edge X that spans the width of both.
3. Composition: For two adjacent active edges X/Y and Y/Z, add an active edge X/Z that spans the width of both.

Rule 3 is not necessary for LC parsing, but is necessary for a fully incremental version (i.e., to ensure a single connected structure).

Example of a Left Corner Chart



Serial Parsing

If parsing was fully parallel, all analyses of a sentence would be equally available; there would be no garden paths.

Since there **are** garden paths, the literature provides two alternative ways to explain them:

- Ranked parallel models: Multiple structures are pursued in parallel; they are ranked in order of preferences; garden paths occur if a low-ranked structure turns out to be correct;
- Serial models: Only one structure is pursued; if it turns out to be incorrect, then a garden path occurs.

Serial Parsing

Serial left-corner parser with backtracking:

- Single structure evolves over time, following a “left corner” (LC) version of the grammar.
- At each point of ambiguity, the parser has to chose one way that the structure will evolve.
- If the structure turns out to be incorrect, the parser has to backtrack.
- At the last point of ambiguity, the incorrect structure is disassembled, and another alternative is pursued instead.

A Serial Model of Left Corner Parsing

Computational requirements:

- operator selection: At each stage of processing, the parser has to select what to do: elaborate the current structure, read the next word, or backtrack.
- depth-first search: Elaborate the current structure as far as possible before alternatives are considered. This requires inhibition of some edges in the chart.
- backtracking: If this is to occur, previous states of the parser must be recoverable. This requires a buffer to store those choice points and ability to remove edges from the chart.

Properties of the Model

Properties of the left corner model:

- This model can parse garden path sentences such as the old man the boats.
- Extensive backtracking may occur for such sentences; full parse is found only if the **Choice Point** stack size is sufficient.

Potential problems:

- Backtracking requires that parse failure is detected, and that the parser knows where the sentence boundaries are.
- Operator evaluations are fixed; context or experience is not taken into account; no attempt to minimize backtracking.
- “The horse raced past the barn fell.” vs. “The cow milked after the storm fell.”

Summary

- Parsing models must build structure incrementally and account for ambiguity resolution and garden paths.
- A chart can be used to represent partial syntactic structure.
- Left-corner parsing model achieves full incrementality and makes predictions, using operator selection to model serial parsing and backtracking.
- Haven't yet clearly explained why some parses are preferred or some locally ambiguous sentences (but not others) cause garden paths.

Jurafsky (1996)

Presents a probabilistic parallel model of human sentence processing that

explains garden paths and disambiguation.

considers several types of ambiguities and sources of information to resolve them.

Lexical category ambiguity

Ambiguity resolved without trouble (fires = N or V):

(2) .The warehouse fires destroyed all the buildings.
.The warehouse fires a dozen employees each year.

Ambiguity leads to garden path (complex= N or Adj, houses= N or V, etc.):

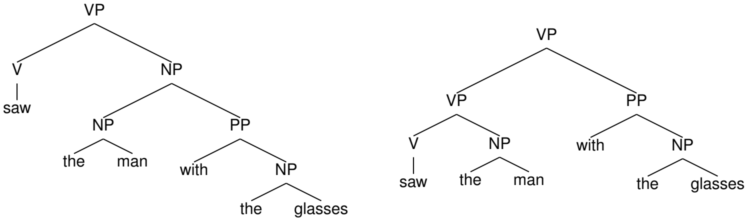
(3) .#The complex houses married and single students.
.#The old man the boats.

Note: # means garden path.

Attachment ambiguity

Prepositional phrase can attach to NP or VP.

(4) I saw the man with the glasses.



(4) #The landlord painted the walls with cracks.

Disambiguation

Main assumptions of Jurafsky (1996):

Observed preferences in interpretation of ambiguous sentences reflect probabilities of different syntactic structures.

Garden path effects are merely extreme cases of processing preferences. Examples from several types of ambiguity:

Lexical category ambiguity

Attachment ambiguity

Main clause vs. reduced relative clause ambiguity

A probabilistic parallel parser

Jurafsky, 1996 adopts methods from statistical natural language processing in a parallel parsing model.

Each full or partial parse is assigned a probability.

Parses are pruned from the search space if their probability is a factor of α below the most probable parse (beam search).

Other pruning methods are possible, e.g., maintain a fixed number of parses at all times.

Computing parse probabilities

Jurafsky, 1996 focuses on two sources of information:

Construction probabilities: probability of syntactic tree.

Valence probabilities: probability of particular syntactic categories as arguments for specific verbs.

Assumes that construction probabilities and valence probabilities

are independent, so

$$P(parse) = P(constructions) * P(valence)$$

can be estimated from a large treebank using relative frequencies.

Probabilistic Context-free Grammars

P(constructions) is computed as $P_{pcfg}(parse)$.

Example (Manning and Schütze, 1999)

S → NP VP	1.0	NP → NP PP	0.4
PP → P NP	1.0	NP → astronomers	0.1
VP → V NP	0.7	NP → ears	0.18
VP → VP PP	0.3	NP → saw	0.04
P → with	1.0	NP → stars	0.18
V → saw	1.0	NP → telescopes	0.1

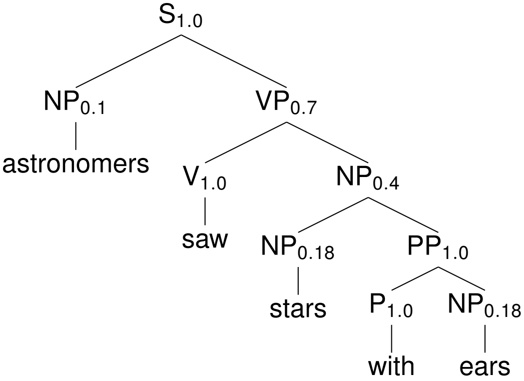
- The rule A → B C with probability *p* means

$$P(\textit{expansion is B C} \mid \textit{left-hand side is A}) = p$$

- so, probabilities of all rules with the same LHS sum to one;
- $P_{pcfg}(parse) = \prod P_{pcfg}(rule_i)$ of all rules applied in the parse.

Probabilistic Context-free Grammars

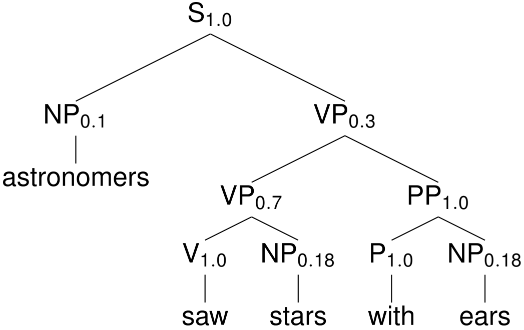
Example (Manning and Schütze, 1999)



$$P(t_1) = 1.0 \cdot 0.1 \cdot 0.7 \cdot 1.0 \cdot 0.4 \cdot 0.18 \cdot 1.0 \cdot 1.0 \cdot 0.18 = 0.0009072$$

Probabilistic Context-free Grammars

Example (Manning and Schütze, 1999)



$P(t_2) = 1.0 \cdot 0.1 \cdot 0.3 \cdot 0.7 \cdot 1.0 \cdot 0.18 \cdot 1.0 \cdot 1.0 \cdot 0.18 = 0.0006804$
t_1 more probable than t_2 according to construction probabilites.

Subcategorization frames

Construction probabilities account for different tree shapes being (dis)preferred overall. But: rating studies show different verbs have different attachment preferences (“A Competence-based Theory of Syntactic Closure”).

(1) The women discussed the dogs on the beach.

  a. The women discussed the dogs that were on the beach. (90%)
  b. The women discussed the dogs while on the beach. (10%)

(2) The women kept the dogs on the beach.

  a. The women kept the dogs that were on the beach. (5%)
  b. The women kept them (the dogs) on the beach. (95%)

The arguments required by a verb are its subcategorization frame or valence. Different valence preferences create different attachment preferences.

Valence Probabilities

Note: For Jurafsky, *arguments* and *valence* are actually defined semantically, not syntactically; the semantic arguments of the verb are those phrases that are semantically necessary to complete the verb phrase. But in most cases (except, e.g., passive sentences), these are just the same as the syntactic arguments of the verb, i.e., the other syntactic categories inside the verb phrase.

Valence Probabilities

Verb subcat frame: the other categories in the verb phrase.

- Ex. In VP → V NP PP, the subcat frame for the V is NP PP.
- Ex. Subcategorization frames of the verb keep:

NP AP	keep the pricesNP reasonableAP
NP VP	keep his foesNP guessingVP
NP VP	keep their eyesNP peeledVP
NP PRT	keep the peopleNP inPRT
NP PP	keep his nervesNP from janglingPP

Valence probabilities tell us how likely each of these frames is.

Valence Probabilities

Like PCFG probs, valence probs are estimated from treebank.

discuss	⟨NP PP⟩	.24
	⟨NP⟩	.76
keep	⟨NP XP[pred +]⟩	.81
	⟨NP⟩	.19

Proportion of cases of ‘discuss’ where arguments are

NP PP: ’discuss the dogsNP with gustoPPVP’

NP: ’discuss the dogs with fleasPPNPVP’

Modeling Results

Garden path or disambiguation can be caused

by construction probabilities alone (if no verb arguments involved)

primarily by valence probabilities

by both Model assumes that

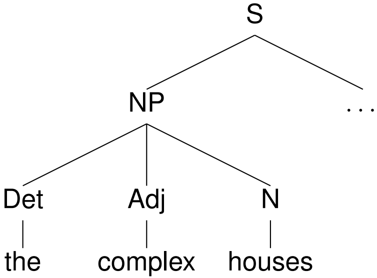
probabilities of tree fragments are computed incrementally (left-to-right)

garden paths caused when incorrect structure is much more probable; weak disambiguation preferences when both structures are similar in probability.

Modeling Garden Path Effects

Garden path caused by construction probabilities:

S → NP …	0.92	N → houses	0.00055
NP → Det Adj N	0.28	Adj → complex	0.00086
Det → the	0.71		

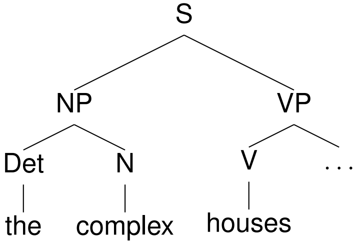


$p(t_1) = 8.5 \cdot 10^{-8}$ (preferred)

Modeling Garden Path Effects

Garden path caused by construction probabilities:

NP → Det N	0.63	V → houses	0.000052
S → [NP $_{VP}$[V …	0.48	Det → the	0.71
N → complex	0.000029		

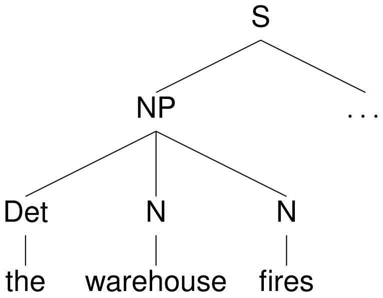


$p(t_2) = 3.2 \cdot 10^{-10}$ (grossly dispreferred)

Modeling Ambiguity

Ambiguous construction, no garden path:

S → NP . . .	0.92	N → fires	0.00017
NP → Det N N	0.28		

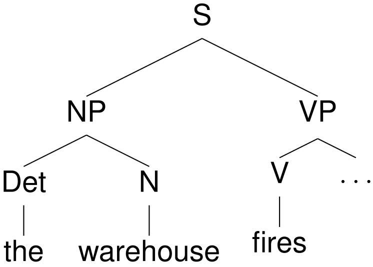


$p(t_1) = 4.2 \cdot 10^{-5}$ (preferred)

Modeling Ambiguity

Ambiguous construction, no garden path:

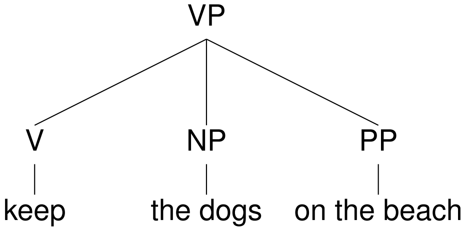
NP → Det N	0.63	V → fires	0.000036
S → [NP $_{\mathrm{VP}}$[V ...	0.48		



$p(t_2) = 1.1 \cdot 10^{-5}$ (mildly dispreferred)

Modeling Valence Preferences

Disambiguation using valence probabilities, no garden path:
$p(\text{keep}, \langle \text{NP XP[pred} +] \rangle) = 0.81$
VP → V NP XP 0.15



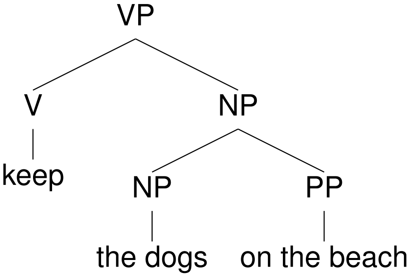
$p(t_1) = 0.15 \cdot 0.81 = 0.12$ (preferred)

Modeling Valence Preferences

Disambiguation using valence probabilities, no garden path:

$p(\text{keep}, \langle \text{NP} \rangle) = 0.19$ $\quad$ $\text{VP} \rightarrow \text{V NP}$ $\quad$ 0.39

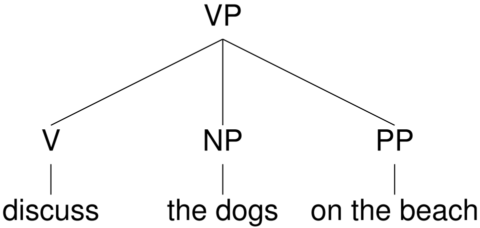
$\text{NP} \rightarrow \text{NP XP}$ $\quad$ 0.14



$p(t_2) = 0.19 \cdot 0.39 \cdot 0.14 = 0.01$ (mildly dispreferred)

Modeling Valence Preferences

Disambiguation using valence probabilities, no garden path:
$p(\text{discuss}, \langle\text{NP PP}\rangle) = 0.24$
VP → V NP XP   0.15

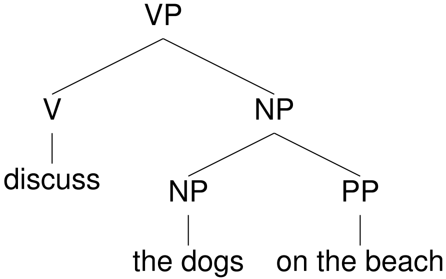


$p(t_1) = 0.15 \cdot 0.24 = 0.036$ (mildly dispreferred)

Modeling Valence Preferences

Disambiguation using valence probabilities, no garden path:

$p(\text{discuss}, \langle \text{NP} \rangle) = 0.76$	$\text{VP} \rightarrow \text{V NP}$	0.39
	$\text{NP} \rightarrow \text{NP XP}$	0.14



$p(t_2) = 0.76 \cdot 0.39 \cdot 0.14 = 0.041$ (preferred)

Combining valence and construction probabilities

Consider the following examples:

(5) .#The horse raced past the barn fell.
(= ‘The horse that was raced past the barn fell.’)
.The horse found in the woods died.
(= ‘The horse that was found in the woods died.’)

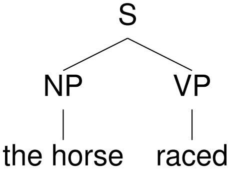
Another case of different subcategorization preferences:

X raced $>>$ X raced Y

X found Y $>>$ X found

Combining valence and construction probabilities

Garden path caused by construction probabilities and valence probabilities:[1]
$p(\text{race}, \langle\text{agent}\rangle) = 0.92$

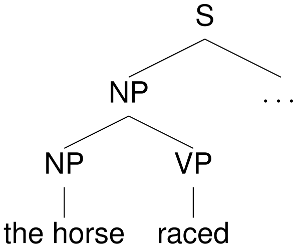


$p(t_1) = 0.92$ (preferred)

[1] Since the upcoming examples require passive sentences, we're now using the semantic notion of valence.

Combining valence and construction probabilities

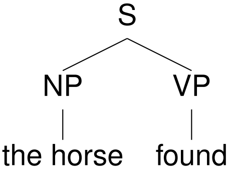
Garden path caused by construction probabilities and valence probabilities:
$p(\text{race}, \langle \text{agent}, \ \text{theme} \rangle) = 0.08$
NP → NP XP   0.14



$p(t_2) = 0.0112$ (grossly dispreferred)

Combining valence and construction probabilities

Disambiguation using construction probabilities and valence probabilities, no garden path:
$p(\text{find}, \langle\text{agent}\rangle) = 0.38$



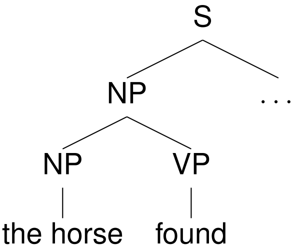
$p(t_1) = 0.38$ (preferred)

Combining valence and construction probabilities

Disambiguation using construction probabilities and valence probabilities, no garden path:

$p(\text{find}, \langle \text{agent}, \ \text{theme} \rangle) = 0.62$

NP → NP XP 0.14



$p(t_2) = 0.0868$ (mildly dispreferred)

Setting the Beam Width

Crucial assumption: if the relative probability of a tree falls below a certain value, then it will be pruned.

sentence	probability ratio
the complex houses . . .	267:1
the horse raced . . .	82:1
the warehouse fires . . .	3.8:1
the horse found . . .	3.7:1

Assumption: a garden path occurs if the probability ratio is higher than 5:1.

Open Issues

- Incrementality: Can we make more fine-grained predictions of the time course of ambiguity resolution?
- Coverage: Jurafsky used hand-crafted examples. Will this model work when considering the fully array of sentences in a real corpus?
- Crosslinguistics: does this model work for languages other than English?

Summary

- Different kinds of ambiguity: phrase attachment; lexical category;
- rating studies provide evidence for subcat frame preferences;
- modeling assumptions:
    - parser with bounded parallelism;
    - pruning of improbable analyses (beam search);
    - independent combination of PCFG and valence probabilities;
- Beam width: ratio of the probability of the preferred analysis to the dispreferred analysis; needs to be determined empirically.
- Model accounts for human parse preferences in several well-known examples.

References I

Ford, Marilyn, Joan Bresnan, and Ronald M. Kaplan. “A Competence-based Theory of Syntactic Closure”. In: pp. 727–796.

Jurafsky, Daniel (1996). “A Probabilistic Model of Lexical and Syntactic Access and Disambiguation”. In: 20.2, pp. 137–194.

Manning, Christopher D. and Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

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

Tanenhaus, Michael K. et al. (1995). “Integration of Visual and Linguistic Information in Spoken Language Comprehension”. In: *Science* 268, pp. 1632–1634.