
# Statistical Natural Language Parsing

Two views of syntactic structure

---

Christopher Manning

## Two views of linguistic structure: 1. Constituency (phrase structure)

- Phrase structure organizes words into nested constituents.

the cat
a dog
large in the crate
barking on the table
cuddly by the door
large barking
to
for

---

Christopher Manning

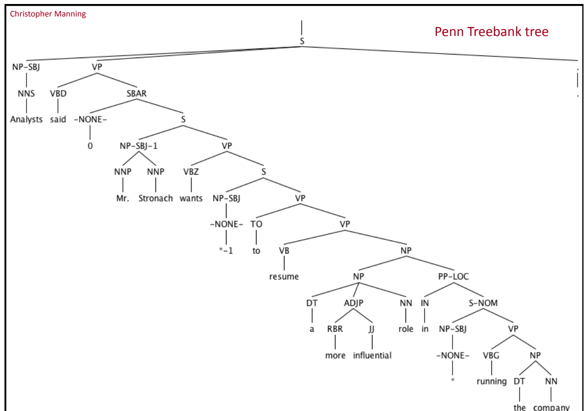
## Two views of linguistic structure: 1. Constituency (phrase structure)

- How do we know what is a constituent? (Not that linguists don't argue about some cases.)
  - Substitution/expansion/pro-forms:
    - I sat [on the box/right on top of the box/there].
  - Distribution: a constituent behaves as a unit that can appear in different places:
    - John talked [to the children] [about drugs].
    - John talked [about drugs] [to the children].
    - *John talked drugs to the children about
  - Coordination, regular internal structure, no intrusion, fragments, semantics, …
    - John [drove to the store] and [bought a bike].

---

Christopher Manning

Penn Treebank tree




---

Christopher Manning

## Headed phrase structure

- VP → … VB* …
- NP → … NN* …
- ADJP → … JJ* …
- ADVP → … RB* …

- SBAR(Q) → S|SINV|SQ → … NP VP …

- Plus minor phrase types:
  - QP (quantifier phrase in NP), CONJP (multi word constructions: *as well as*), INTJ (interjections), etc.

---

Christopher Manning

## Two views of linguistic structure: 2. Dependency structure

- Dependency structure shows which words depend on (modify or are arguments of) which other words.

*The boy put the tortoise on the rug*


# Statistical Natural Language Parsing

Parsing: ambiguity and the rise of data and statistics

---

Christopher Manning

## Pre 1990 ("Classical") NLP Parsing

- Wrote symbolic grammar (CFG or often richer) and lexicon

| | |
|---|---|
| S → NP VP | NN → *interest* |
| NP → (DT) NN | NNS → *rates* |
| NP → NN NNS | NNS → *raises* |
| NP → NNP | VBP → *interest* |
| VP → V NP | VBZ → *rates* |

- Used grammar/proof systems to prove parses from words
- This scaled very badly and didn't give coverage. For sentence:

  *Fed raises interest rates 0.5% in effort to control inflation*

  - Minimal grammar: 36 parses
  - Simple 10 rule grammar: 592 parses
  - Real-size broad-coverage grammar: millions of parses

---

Christopher Manning

## Ambiguity: PP attachments

*The boy ate the dessert with a spoon/cherry*

---

Christopher Manning

## Attachment ambiguities

- A key parsing decision is how we 'attach' various constituents
  - PPs, adverbial or participial phrases, infinitives, coordinations, etc.

The board approved [its acquisition] [by Royal Trustco Ltd.]
[of Toronto]
[for $27 a share]
[at its monthly meeting].

- Catalan numbers: $C_n = (2n)!/[(n+1)!n!]$
- An exponentially growing series, which arises in many tree-like contexts:
  - E.g., the number of possible triangulations of a polygon with $n$+2 sides
    - Turns up in triangulation of probabilistic graphical models….

---

Christopher Manning

## Classical NLP Parsing: The problem and its solution

- Categorical constraints can be added to grammars to limit unlikely/weird parses for sentences
  - But the attempt makes the grammars not robust
    - In traditional systems, commonly 30% of sentences in even an edited text would have *no* parse.
- A less constrained grammar can parse more sentences
  - But simple sentences end up with ever more parses with no way to choose between them
- We need mechanisms that allow us to find the most likely parse(s) for a sentence
  - Statistical parsing lets us work with very loose grammars that admit millions of parses for sentences but still quickly find the best parse(s)

---

Christopher Manning

## The rise of annotated data: The Penn Treebank

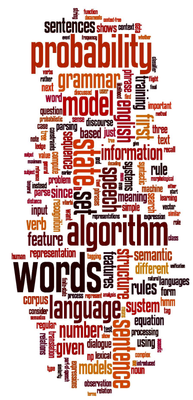
[Marcus et al. 1993, *Computational Linguistics*]

```
( (S
   (NP-SBJ (DT The) (NN move))
   (VP (VBD followed)
    (NP
     (NP (DT a) (NN round))
     (PP (IN of)
      (NP
       (NP (JJ similar) (NNS increases))
       (PP (IN by)
        (NP (JJ other) (NNS lenders)))
       (PP (IN against)
        (NP (NNP Arizona) (JJ real) (NN estate) (NNS loans))))))
    (, ,)
    (S-ADV
     (NP-SBJ (-NONE- *))
     (VP (VBG reflecting)
      (NP
       (NP (DT a) (VBG continuing) (NN decline))
       (PP-LOC (IN in)
        (NP (DT that) (NN market)))))))
   (. .)))
```

Christopher Manning

## The rise of annotated data

- Starting off, building a treebank seems a lot slower and less useful than building a grammar
- But a treebank gives us many things
  - Reusability of the labor
    - Many parsers, POS taggers, etc.
    - Valuable resource for linguistics
  - Broad coverage
  - Frequencies and distributional information
  - A way to evaluate systems

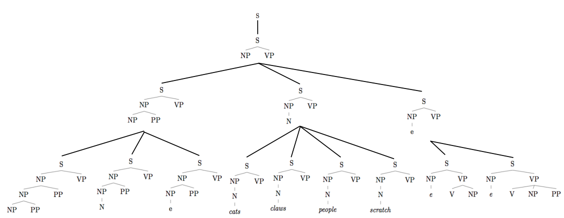


## Statistical Natural Language Parsing

Parsing: solving exponential work and ambiguity

Christopher Manning

## Top-down parsing



Christopher Manning

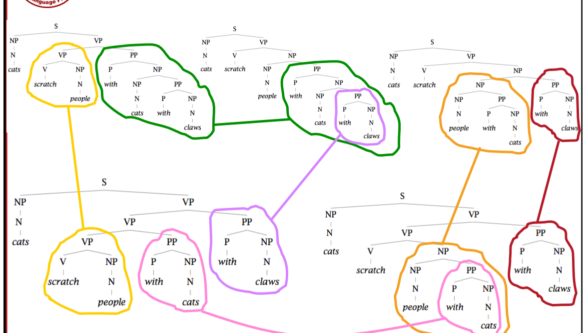
## Shift-reduce parsing: one path

| | | |
|---|---|---|
| | *cats scratch people with claws* | |
| *cats* | *scratch people with claws* | SHIFT |
| N | *scratch people with claws* | REDUCE |
| NP | *scratch people with claws* | REDUCE |
| NP *scratch* | *people with claws* | SHIFT |
| NP V | *people with claws* | REDUCE |
| NP V *people* | *with claws* | SHIFT |
| NP V N | *with claws* | REDUCE |
| NP V NP | *with claws* | REDUCE |
| NP V NP *with* | *claws* | SHIFT |
| NP V NP P | *claws* | REDUCE |
| NP V NP P *claws* | | SHIFT |
| NP V NP P N | | REDUCE |
| NP V NP P NP | | REDUCE |
| NP V NP PP | | REDUCE |
| NP VP | | REDUCE |
| S | | REDUCE |

What other search paths are there for parsing this sentence?

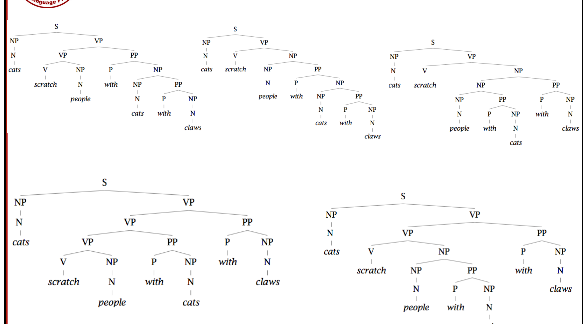
Christopher Manning

## Two problems to solve: 1. Repeated work…



Christopher Manning

## Two problems to solve: 1. Repeated work…



Christopher Manning

## Two problems to solve: 2. Choosing the correct parse

- How do we work out the correct attachment:
  - She saw the man with a telescope
- Is the problem 'AI complete'? Yes, but …
- Words are good predictors of attachment
  - Even absent full understanding
  - Moscow sent more than 100,000 soldiers into Afghanistan …
  - Sydney Water breached an agreement with NSW Health …
- Our statistical parsers will try to exploit such statistics.

Christopher Manning

## A simple prediction

- Use a likelihood ratio:
  - E.g., $LR(v,n,p) = \frac{P(p \mid v)}{P(p \mid n)}$
  - P(*with*|*agreement*) = 0.15
  - P(*with*|*breach*) = 0.02
  - LR(*breach*, *agreement*, *with*) = 0.13
    → Choose noun attachment

Christopher Manning

## A problematic example

- *Chrysler confirmed that it would end its troubled venture with Maserati.*
- Should be a noun attachment but get wrong answer:
  - *w* — *C(w)* — *C(w, with)*
  - *end* — 5156 — 607
  - *venture* — 1442 — 155

$$P(with \mid v) = \frac{607}{5156} \approx 0.118 > P(with \mid n) = \frac{155}{1442} \approx 0.107$$

Christopher Manning

## A problematic example

- What might be wrong here?
  - If you see a V NP PP sequence, then for the PP to attach to the V, then it must also be the case that the NP doesn't have a PP (or other postmodifier)
    - Since, except in extraposition cases, such dependencies can't cross
    - Also, the verb must take an NP object
      - Unlike cases like "end with a bang"
- Parsing allows us to factor in and integrate such constraints.

Christopher Manning

## Human parsing

- Humans often do ambiguity maintenance
  - *Have the police … eaten their supper?*
  - *come in and look around.*
  - *taken out and shot.*
- But humans also commit early and are "garden pathed":
  - *The man who hunts ducks out on weekends.*
  - *The cotton shirts are made from grows in Mississippi.*


## CFGs and PCFGs

(Probabilistic) Context-Free Grammars

Christopher Manning

## A phrase structure grammar

| | |
|---|---|
| S → NP VP | N → people |
| VP → V NP | N → fish |
| VP → V NP PP | N → tanks |
| NP → NP NP | N → rods |
| NP → NP PP | V → people |
| NP → N | V → fish |
| NP → *e* | V → tanks |
| PP → P NP | P → with |

*people fish tanks*
*people fish with rods*

Christopher Manning

## Phrase structure grammars = context-free grammars (CFGs)

- G = (T, N, S, R)
  - T is a set of terminal symbols
  - N is a set of nonterminal symbols
  - S is the start symbol (S ∈ N)
  - R is a set of rules/productions of the form X → γ
    - X ∈ N and γ ∈ (N ∪ T)*
- A grammar G generates a language L.

Christopher Manning

## Phrase structure grammars in NLP

- G = (T, C, N, S, L, R)
  - T is a set of terminal symbols
  - C is a set of preterminal symbols
  - N is a set of nonterminal symbols
  - S is the start symbol (S ∈ N)
  - L is the lexicon, a set of items of the form X → x
    - X ∈ P and x ∈ T
  - R is the grammar, a set of items of the form X → γ
    - X ∈ N and γ ∈ (N ∪ C)*
- By usual convention, S is the start symbol, but in statistical NLP, we usually have an extra node at the top (ROOT, TOP)
- We usually write *e/ε* for an empty sequence, rather than nothing

Christopher Manning

## A phrase structure grammar

| | |
|---|---|
| S → NP VP | N → *people* |
| VP → V NP | N → *fish* |
| VP → V NP PP | N → *tanks* |
| NP → NP NP | N → *rods* |
| NP → NP PP | V → *people* |
| NP → N | V → *fish* |
| NP → *e* | V → *tanks* |
| PP → P NP | P → *with* |

*people fish tanks*
*people fish with rods*

Christopher Manning

## Probabilistic – or stochastic – context-free grammars (PCFGs)

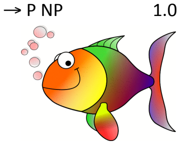
- G = (T, N, S, R, P)
  - T is a set of terminal symbols
  - N is a set of nonterminal symbols
  - S is the start symbol (S ∈ N)
  - R is a set of rules/productions of the form X → γ
  - P is a probability function
    - P: R → [0,1]
    - $\forall X \in N, \sum_{X \to \gamma \in R} P(X \to \gamma) = 1$
- A grammar G generates a language model L.

$$\sum_{\gamma \in T^*} P(\gamma) = 1$$

Christopher Manning

## A PCFG

| | | | |
|---|---|---|---|
| S → NP VP | 1.0 | N → *people* | 0.5 |
| VP → V NP | 0.6 | N → *fish* | 0.2 |
| VP → V NP PP | 0.4 | N → *tanks* | 0.2 |
| NP → NP NP | 0.1 | N → *rods* | 0.1 |
| NP → NP PP | 0.2 | V → *people* | 0.1 |
| NP → N | 0.7 | V → *fish* | 0.6 |
| PP → P NP | 1.0 | V → *tanks* | 0.3 |
| | | P → *with* | 1.0 |



[With empty NP removed so less ambiguous]

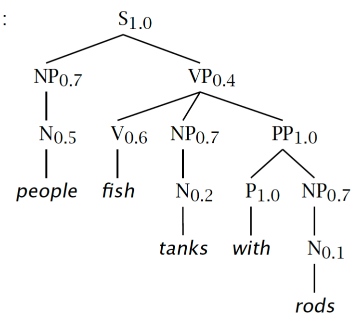
## The probability of trees and strings

- P($t$) – The probability of a tree $t$ is the product of the probabilities of the rules used to generate it.
- P($s$) – The probability of the string $s$ is the sum of the probabilities of the trees which have that string as their yield

$$P(s) = \Sigma_j P(s, t) \text{ where } t \text{ is a parse of } s$$
$$= \Sigma_j P(t)$$

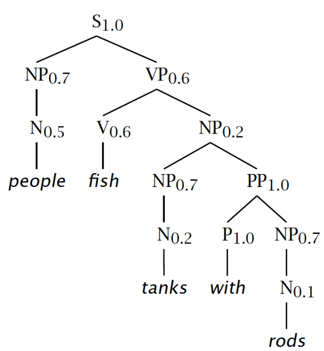


$t_1$:






$t_2$:



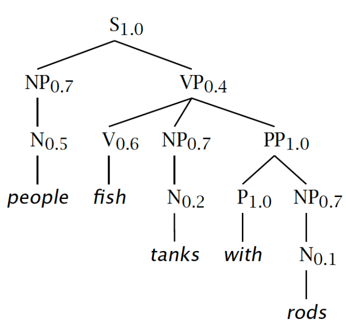



## Tree and String Probabilities

- $s$ = *people fish tanks with rods*
- P($t_1$) = 1.0 × 0.7 × 0.4 × 0.5 × 0.6 × 0.7 × 1.0 × 0.2 × 1.0 × 0.7 × 0.1 = 0.0008232 — Verb attach
- P($t_2$) = 1.0 × 0.7 × 0.6 × 0.5 × 0.6 × 0.2 × 0.7 × 1.0 × 0.2 × 1.0 × 0.7 × 0.1 = 0.00024696 — Noun attach
- P($s$) = P($t_1$) + P($t_2$) = 0.0008232 + 0.00024696 = 0.00107016



$t_1$:






$t_2$: