

Natural Language Processing

CS221

Christopher Manning
Stanford University

<http://nlp.stanford.edu/~manning/>

Aims and Goals of Computational Linguistics

- Goals can be very far reaching:
 - To be able to fully understand texts like humans
 - To be able to produce fluent texts in human languages
 - Real-time participation in spoken dialogs
 - Machine translation between languages
- Or very down-to-earth:
 - Searching the web
 - Context sensitive spell-checking
 - Recognizing the spoken word “yes” vs. “no”
 - Analyzing reading level or authorship statistically
 - Automatic report generation

NLP

- NLP = NLU + NLG
 - NLU: speech/text → meaning
 - NLG: meaning → text/speech
- Applied NLP
 - Speech recognition/information extraction: speech/text → structured data/actions or words
 - Text/speech generation: structured data/actions or words → text/speech

NLP: A brief history

- 50s: The cold war. Machine translation
- 60s: Machine translation is hopeless
- 70s/80s: Science of the mind
 - Big questions of cognition
 - Successful small simulations (SHRDLU, LUNAR, . . .)
- 1990s: Real problems; rigorous evaluation
 - Big corpora on big hard disks
 - Applications: web, speech, (vertical)
 - Greatly favors statistical techniques
- 2000s: The future is meaning?

Human language is tricky stuff

- Newspaper headlines:
 - “Ban on Nude Dancing on Governor’s Desk” – from a Georgia newspaper column discussing current legislation
 - “Lebanese chief limits access to private parts” – talking about an Army General’s initiative
 - “Death may ease tension” – an article about the death of Colonel Jean-Claude Paul in Haiti

Human language is tricky stuff:

The problem of ambiguity

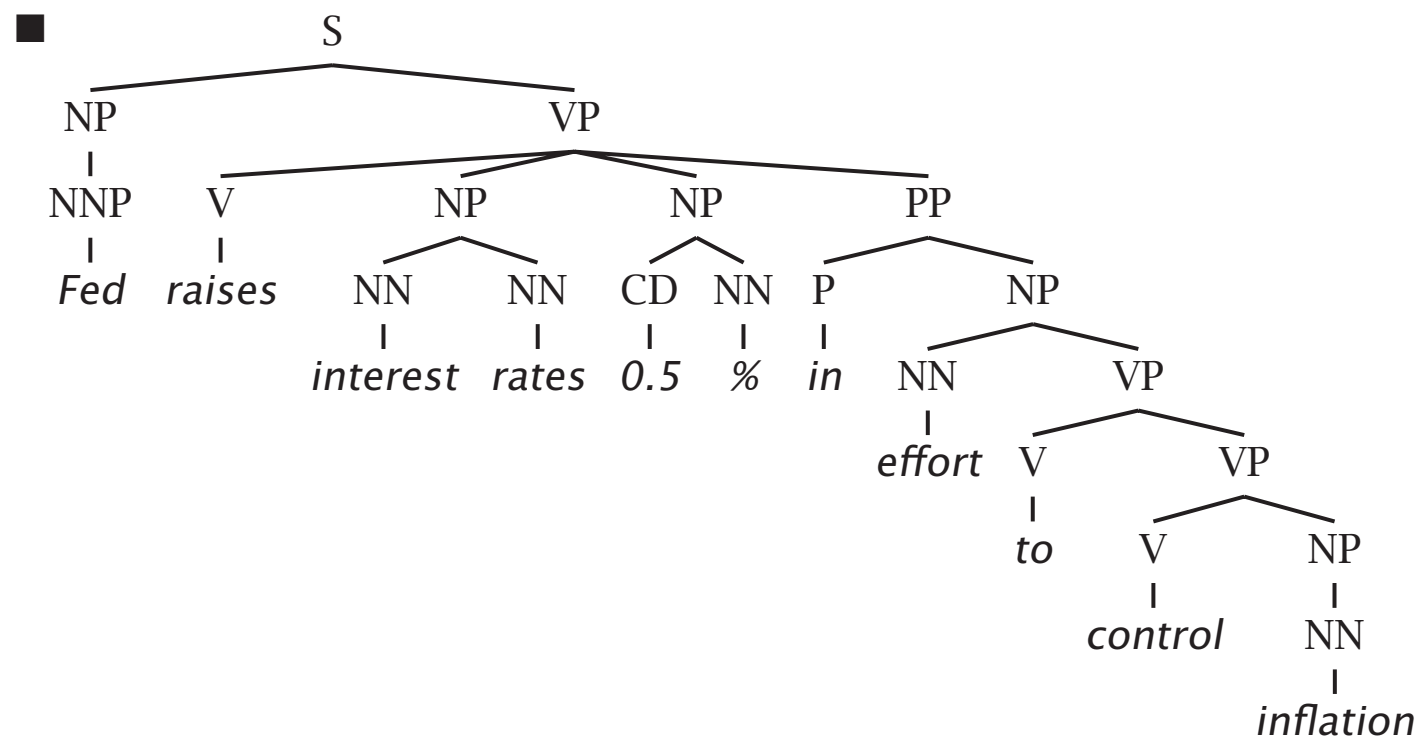
- I saw the Grand Canyon **flying to New York**.
- The man went to the **bank** { to get some cash / and jumped in }.
- He ran the mile **in** { four minutes / the Olympics }.
- I took the cake from the table and { cleaned / ate } **it**.
- **Could you** open the door for me?

Psycholinguistics in one slide

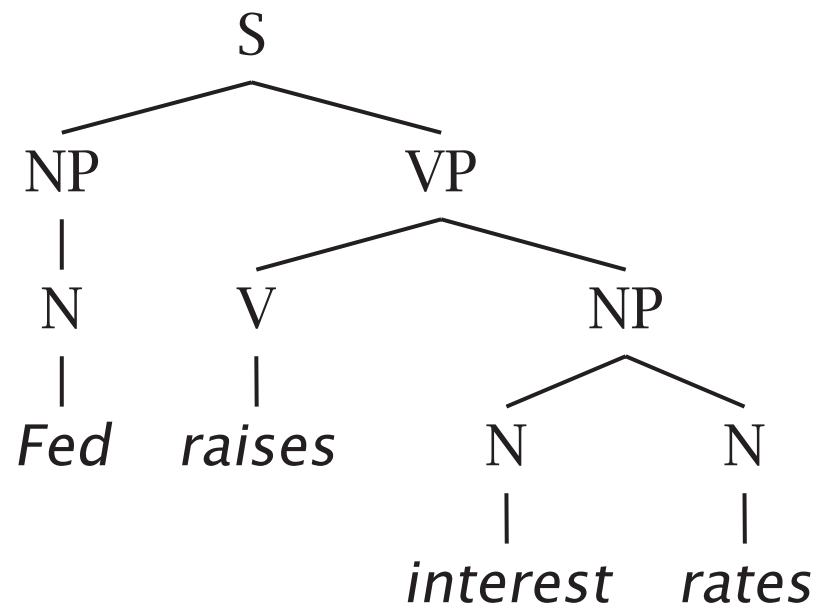
- Humans rapidly and incrementally accumulate and integrate information from world and discourse context and the current utterance so as to interpret what someone is saying in real time. Often commit early.
- They can often finish each other's sentences!
- If a human starts hearing *Pick up the yellow plate* and there is only one yellow item around, they'll already have locked on to it before the word *yellow* is finished
- Our NLP models don't incorporate context into recognition like this, or disambiguate without having heard whole words (and often following context as well)

Why is NLU difficult? The hidden structure of language is hugely ambiguous

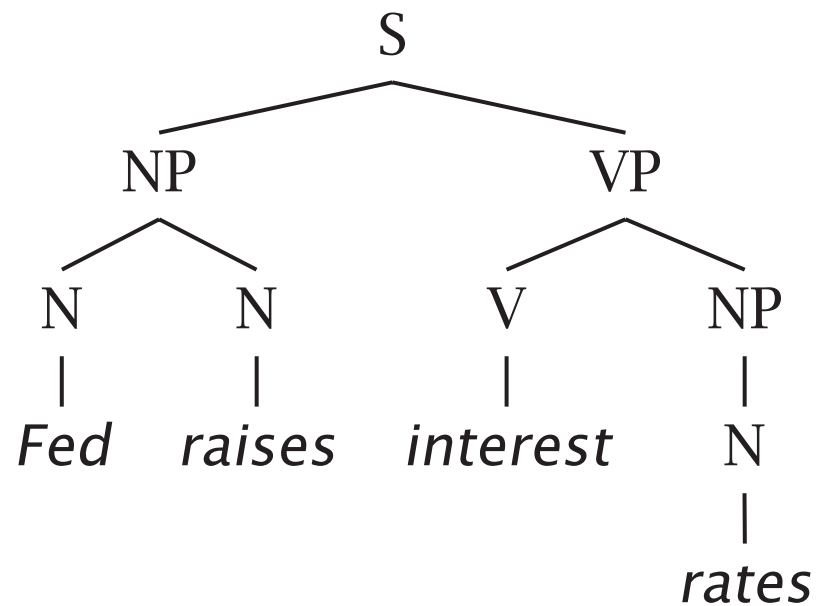
- Structures for: *Fed raises interest rates 0.5% in effort to control inflation* (NYT headline 17 May 2000)



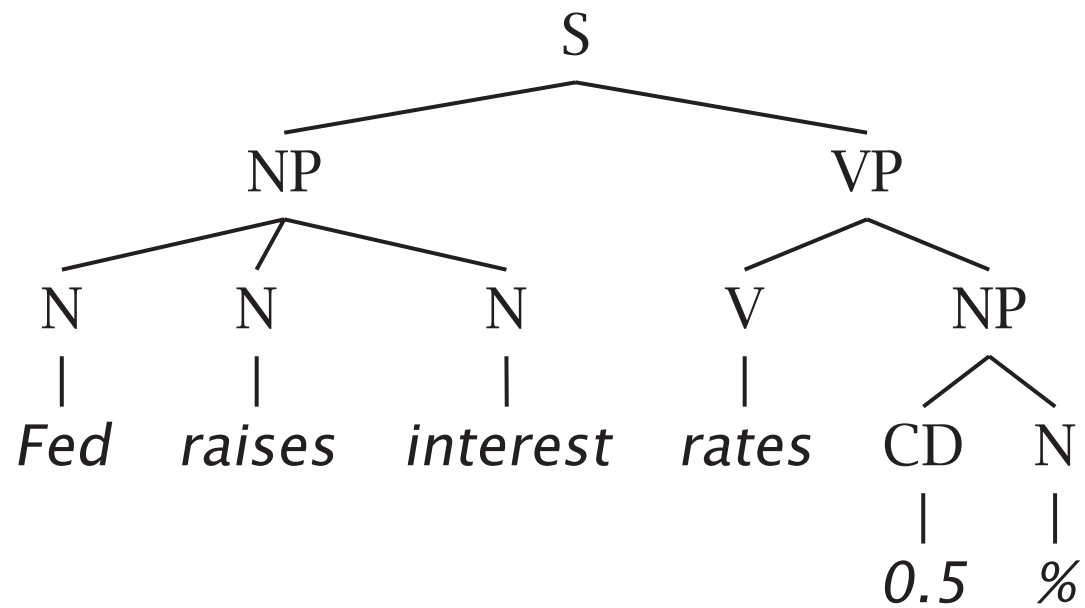
The bad effects of V/N ambiguities (1)



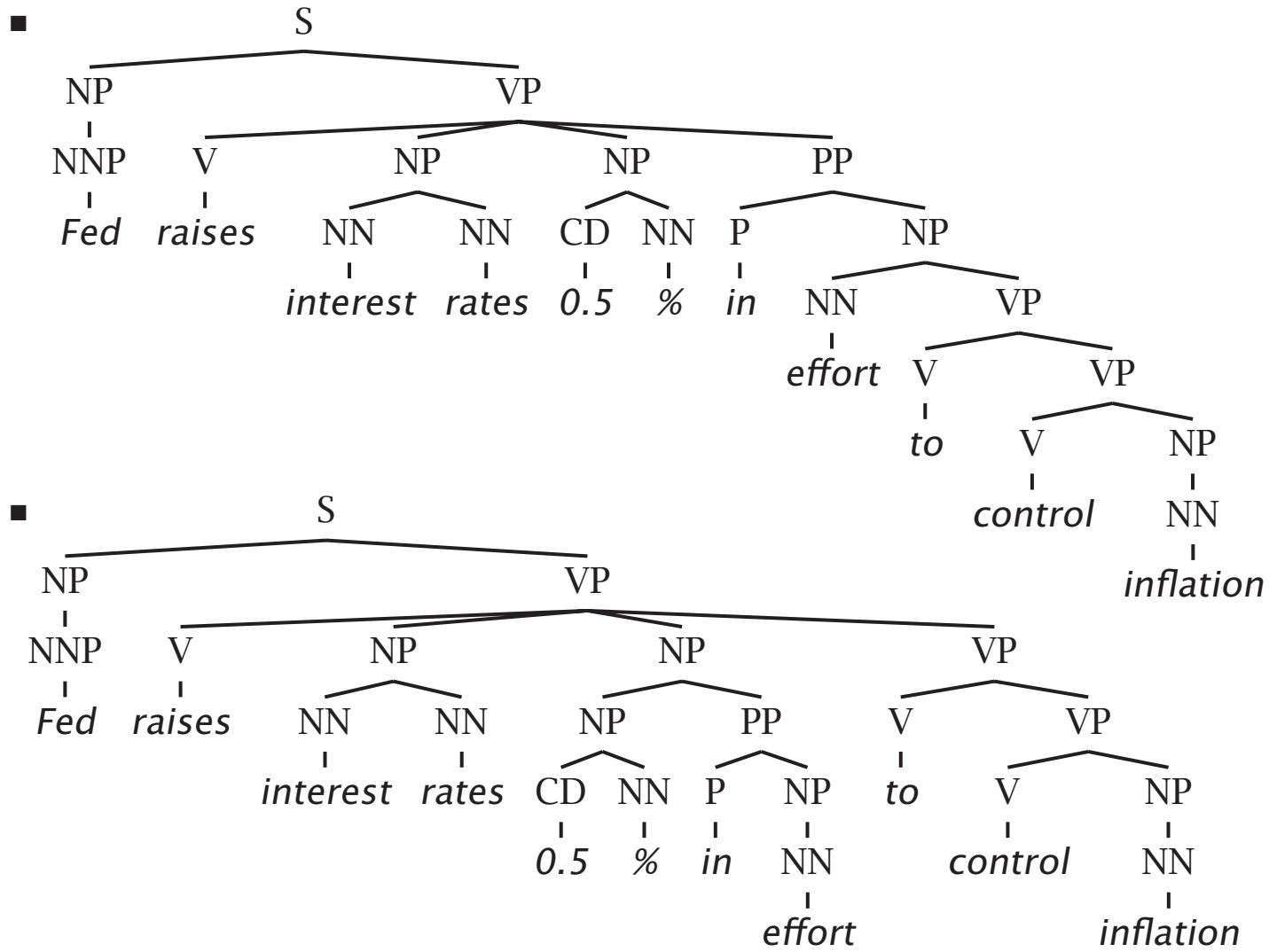
The bad effects of V/N ambiguities (2)



The bad effects of V/N ambiguities (3)



Phrasal attachment ambiguities



The many meanings of *interest* [n.]

- Readiness to give attention to or to learn about something
- Quality of causing attention to be given
- Activity, subject, etc., which one gives time and attention to
- The advantage, advancement or favor of an individual or group
- A stake or share (in a company, business, etc.)
- Money paid regularly for the use of money

Converse: words or senses that mean (almost) the same:
image, likeness, portrait, facsimile, picture

Natural language understanding traditions

- The **logical tradition**

- Gave up the goal of dealing with imperfect natural languages in the development of formal logics
- But the tools were taken and re-applied to natural languages (Lambek 1958, Montague 1973, etc.)
- These tools give rich descriptions of natural language structure, and particularly the construction of sentence meanings (e.g., Carpenter 1999)
 - ▶
$$\frac{NP:\alpha \quad NP \setminus S:\beta}{S:\beta(\alpha)}$$
- They don't tell us about word meaning or use

Natural language understanding traditions

- The **formal language theory tradition** (Chomsky 1957)
 - Languages are generated by a grammar, which defines the strings that are members of the language (others are ungrammatical)
 - ▶ $NP \rightarrow Det\ Adj^*\ N$ $Adj \rightarrow clever$
 - The generation process of the grammar puts structures over these language strings
 - This process is reversed in parsing the language
- These ideas are still usually present in the symbolic backbone of most statistical NLP systems
- Often insufficient attention to meaning

Why Probabilistic Language Understanding?

- Language use is situated in a world context
- People write or say the little that is needed to be understood in a certain discourse situation
- Consequently
 - Language is highly ambiguous
 - Tasks like interpretation and translation involve (probabilistically) reasoning about meaning, using world knowledge not in the source text
- We thus need to explore quantitative techniques that move away from the unrealistic categorical assumptions of much of formal linguistic theory (and earlier computational linguistics)

Why probabilistic linguistics?

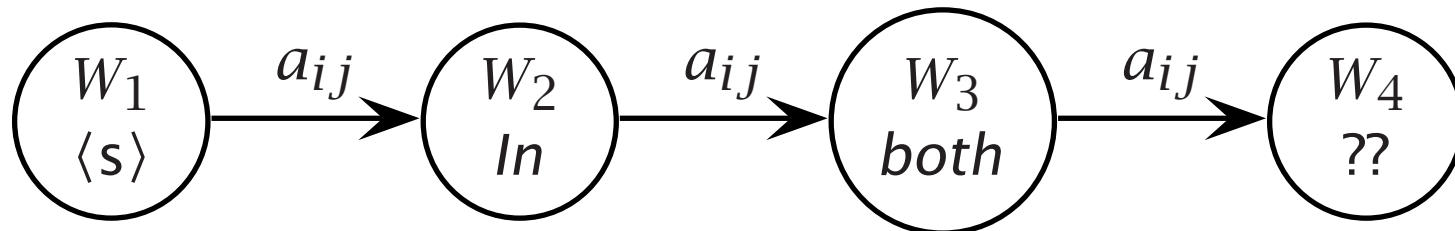
- Categorical grammars aren't predictive: their notions of grammaticality and ambiguity do not accord with human perceptions
 - They don't tell us what "sounds natural"
 - Grammatical but unnatural e.g.: *In addition to this, she insisted that women were regarded as a different existence from men unfairly.*
- Need to account for variation of languages across speech communities and across time
- People are creative: they bend language 'rules' as needed to achieve their novel communication needs
- Consequently "All grammars leak" (Sapir 1921:39)₈

StatNLP: Relation to wider context

- Matches move from logic-based AI to probabilistic AI
 - Knowledge → probability distributions
 - Inference → conditional distributions
- Probabilities give opportunity to unify reasoning, planning, and learning, with communication
- There is now widespread use of machine learning (ML) methods in NLP (perhaps even overuse?)
- Now, an emphasis on empirical validation and the use of approximation for hard problems

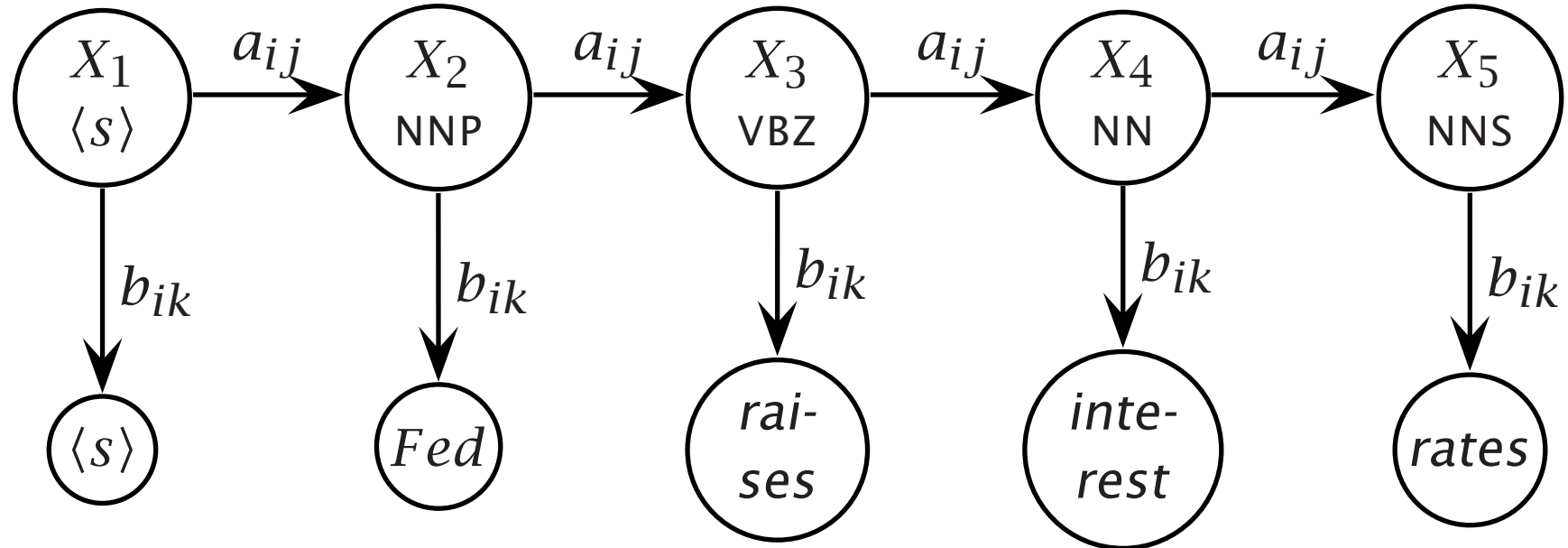
Simple linear models of language

- Markov models a.k.a. n -gram models:



- Word sequence is predicted via a conditional distribution
- Conditional Probability Table (CPT): e.g., $P(X|both)$
 - ▶ $P(of|both) = 0.066$
 - ▶ $P(to|both) = 0.041$
 - ▶ $P(in|both) = 0.038$
- Amazingly successful as a simple engineering model
- From 1940s onward (or even 1910s)

Simple statistical models of language: Hidden Markov Models

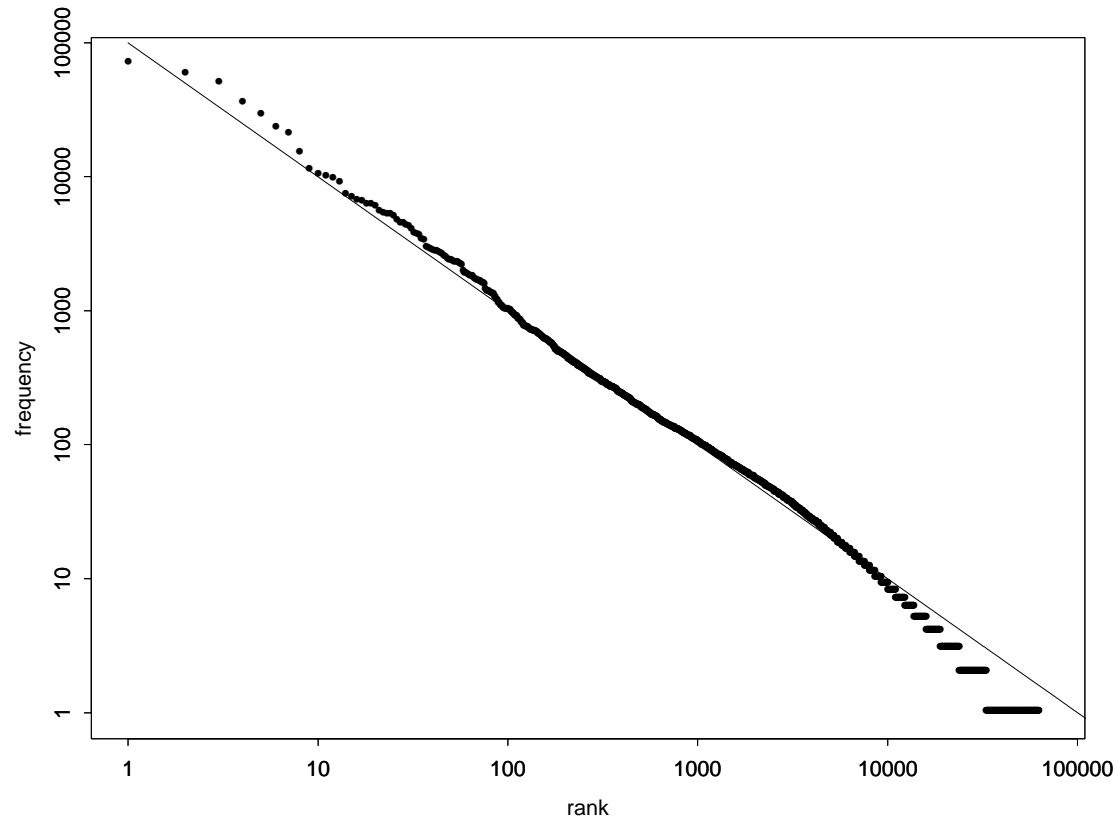


- Top row is unobserved states, interpreted as POS tags
- Bottom row is observed output observations

The problem of data sparseness

Word Frequency	Frequency of Frequency	Frequencies of frequencies in <i>Tom Sawyer</i>	
1	3993	71,730	word tokens
2	1292	8,018	word types
3	664		
4	410		
5	243		
6	199		
7	172		
8	131		
9	82		
10	91		
11-50	540		
51-100	99		
> 100	102		

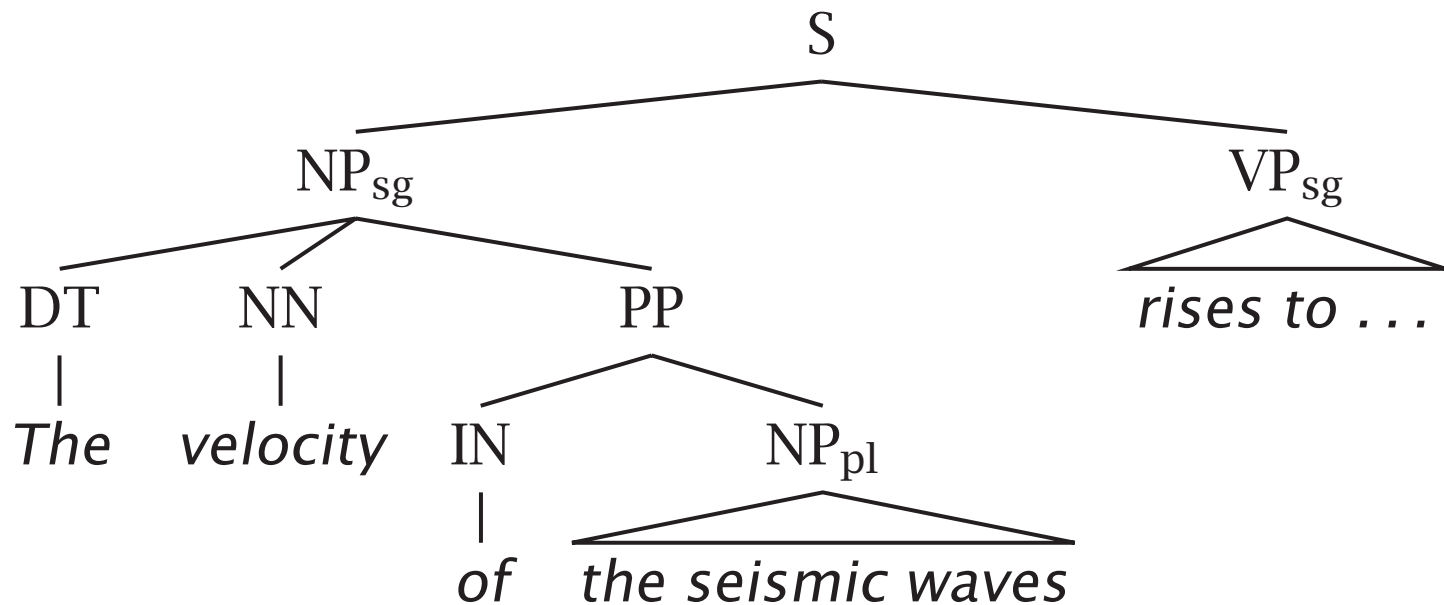
The obligatory Zipf's law slide: Zipf's law for the Brown corpus



$$f \propto \frac{1}{r} \quad \text{or, there is a } k \text{ such that } f \cdot r = k$$

Why we need recursive structure

- Linear models were panned by Chomsky (1957)
- *The velocity of the seismic waves rises to ...*



Probabilistic context-free grammars (PCFGs)

A PCFG G consists of:

- A set of terminals, $\{w^k\}$
- A set of nonterminals, $\{N^i\}$, with a start symbol, N^1
- A set of rules, $\{N^i \rightarrow \zeta^j\}$, (where ζ^j is a sequence of terminals and nonterminals)
- A set of probabilities on rules such that:

$$\forall i \quad \sum_j P(N^i \rightarrow \zeta^j) = 1$$

- A generalization of HMMs to tree structures (branching processes)

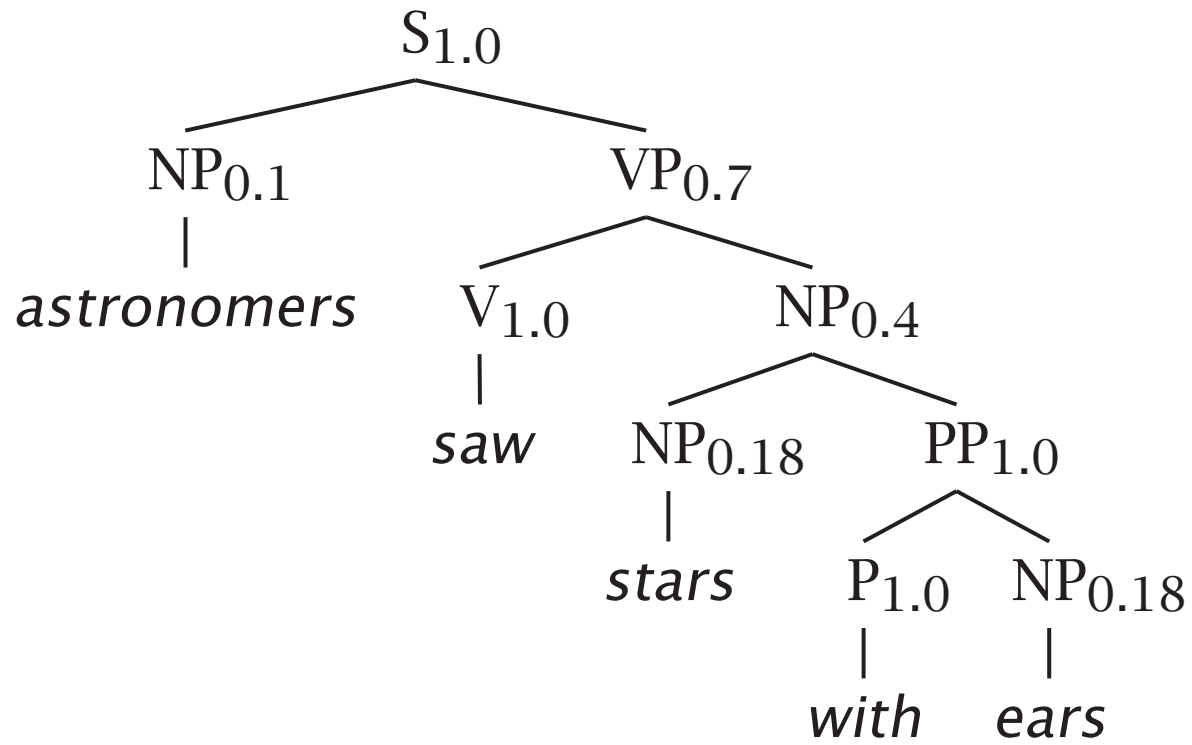
PCFGs

- Like a regular CFG but put probability on each rule
- *Key independence assumption*: Probabilities are also completely context-free, depending just on parent node
- Allow probabilistic inference:
 - $P(w_1 \cdot \cdot \cdot w_m | G)$
 - $\arg \max_t P(t | w_1 \cdot \cdot \cdot w_m, G)$
 - Find G such that $P(w_1 \cdot \cdot \cdot w_m | G)$ is maximized
- Give a partial solution for resolving grammar ambiguities – but not too good, as not lexicalized
- Better for grammar induction (Gold (1967) vs. Horning (1969)) and robustness

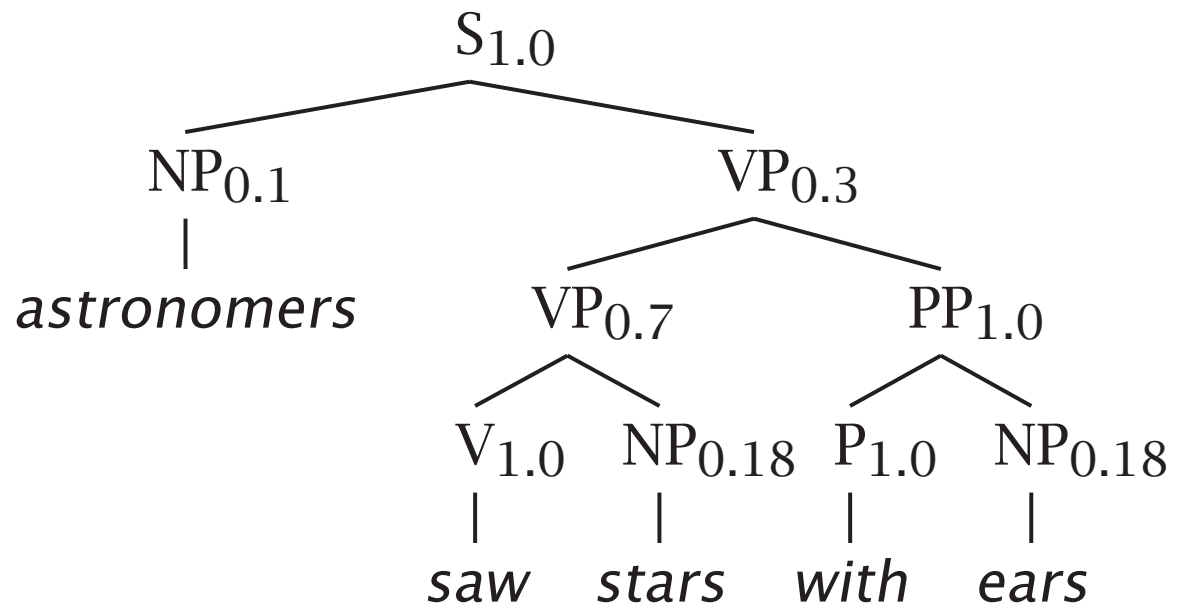
A simple PCFG (in CNF)

$S \rightarrow NP VP$	1.0	$NP \rightarrow NP PP$	0.4
$PP \rightarrow P NP$	1.0	$NP \rightarrow \textit{astronomers}$	0.1
$VP \rightarrow V NP$	0.7	$NP \rightarrow \textit{ears}$	0.18
$VP \rightarrow VP PP$	0.3	$NP \rightarrow \textit{saw}$	0.04
$P \rightarrow \textit{with}$	1.0	$NP \rightarrow \textit{stars}$	0.18
$V \rightarrow \textit{saw}$	1.0	$NP \rightarrow \textit{telescopes}$	0.1

t_1 :



t_2 :



The two parse trees' probabilities and the sentence probability

$$\begin{aligned}P(t_1) &= 1.0 \times 0.1 \times 0.7 \times 1.0 \times 0.4 \\ &\quad \times 0.18 \times 1.0 \times 1.0 \times 0.18 \\ &= 0.0009072\end{aligned}$$

$$\begin{aligned}P(t_2) &= 1.0 \times 0.1 \times 0.3 \times 0.7 \times 1.0 \\ &\quad \times 0.18 \times 1.0 \times 1.0 \times 0.18 \\ &= 0.0006804\end{aligned}$$

$$P(w_{15}) = P(t_1) + P(t_2) = 0.0015876$$

Parsing as search

- How does one actually find these structures for the sentence?
- How does one find the most likely one?
- One starts with the (probabilistic) grammar and does *search*
- All the usual stuff: depth-first, breadth-first, A*
- *Dynamic programming* methods are used to make the search reasonably efficient (so one avoids rebuilding the same structures)

Attachment ambiguities: The key parsing decision

- The main problem in parsing is working out how to ‘attach’ various kinds of constituents – PPs, adverbial or participial phrases, coordinations, and so on
- Prepositional phrase attachment
 - *I saw the man with a telescope*
- What does *with a telescope* modify?
 - The verb *saw*?
 - The noun *man*?
- Is the problem ‘AI-complete’? Yes, but . . .

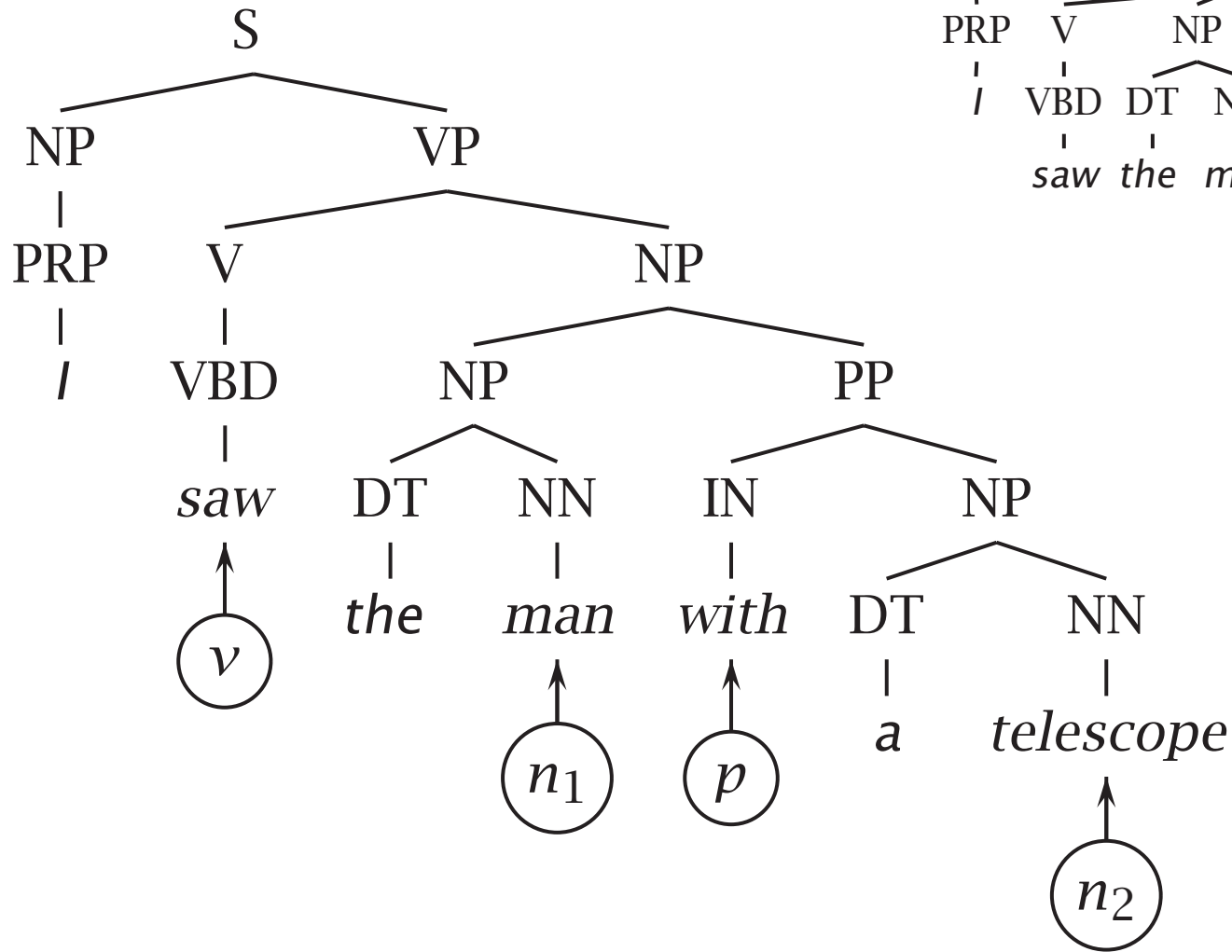
Attachment ambiguities (2)

- Proposed simple structural factors
 - Right association (Kimball 1973) = ‘low’ or ‘near’ attachment = ‘late closure’ (of NP) [NP → NP PP]
 - Minimal attachment (Frazier 1978) [depends on grammar] = ‘high’ or ‘distant’ attachment = ‘early closure’ (of NP) [VP → V NP PP]
- Such simple structural factors dominated in early psycholinguistics, and are still widely invoked.
- In the V NP PP context, right attachment gets right 55–67% of cases.
- But that means it gets wrong 33–45% of cases

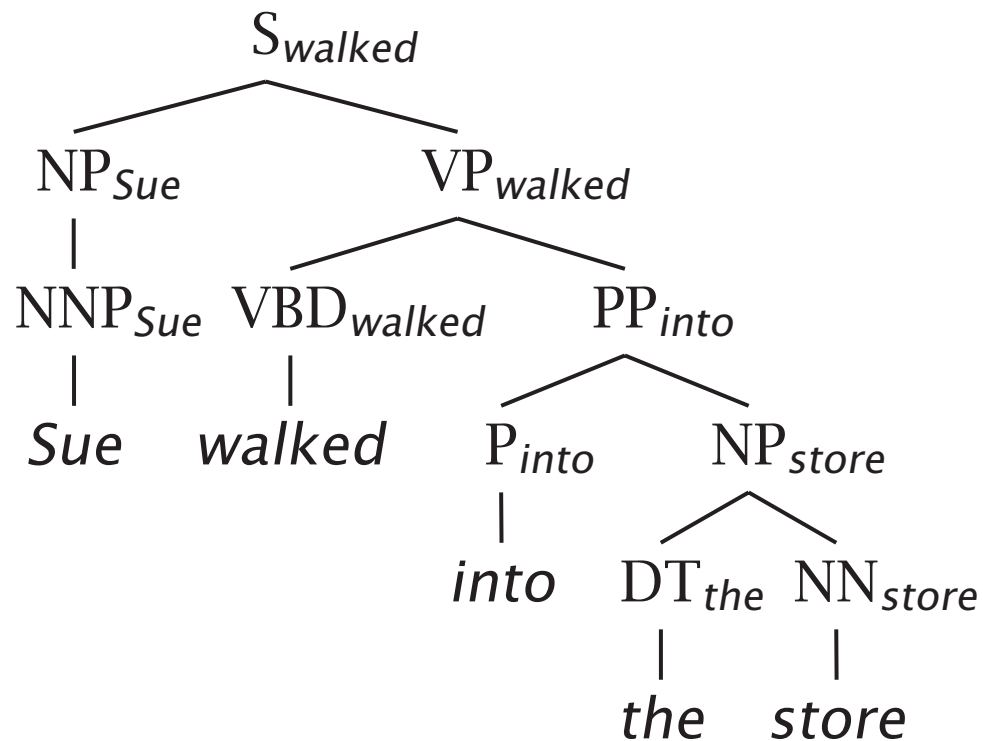
Importance of lexical factors

- Words are good predictors (or even inducers) of attachment (even absent understanding):
 - The children ate the cake with a spoon.
 - The children ate the cake with frosting.
 - Moscow sent more than 100,000 soldiers into Afghanistan
 - Sydney Water breached an agreement with NSW Health
- Ford et al. (1982):
 - Ordering is jointly determined by strengths of alternative lexical forms, alternative syntactic rewrite rules, and the sequence of hypotheses in parsing

Attachment ambiguities



Lexicalization: From a standard tree ... we form a lexicalized tree



Probabilistic models for parsing

- Parsing model: We estimate directly the probability of parses of a sentence

$$\hat{t} = \arg \max_t P(t|s, G) \quad \text{where} \quad \sum_t P(t|s, G) = 1$$

(Magerman 1995, Collins 1996)

- We don't learn from the distribution of sentences we see (but nor do we assume some distribution for them)
- But in effect we're always generalizing over sentences in estimating rules

Language model

- Language model:

$$\sum_{\{t: \text{yield}(t) \in \mathcal{L}\}} P(t) = 1$$

- Sentence probability

$$P(s) = \sum_t P(s, t) = \sum_{\{t: \text{yield}(t) = s\}} P(t)$$

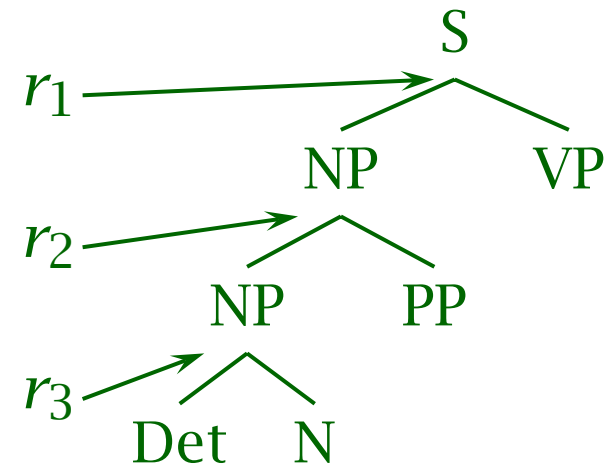
- Most likely tree

$$\hat{t} = \arg \max_t P(t|s) = \arg \max_t \frac{P(t, s)}{P(s)} = \arg \max_t P(t, s)$$

- (Collins 1997, Charniak 1997, Charniak 2000)

Derivational model

$$P(t) = \sum_{\{d: d \text{ is a derivation of } t\}} P(d)$$



Or: $P(t) = P(d)$ where d is the canonical derivation of t

$$d = P(S \xrightarrow{r_1} \alpha_1 \xrightarrow{r_2} \dots \xrightarrow{r_m} \alpha_m = s) = \prod_{i=1}^m P(r_i | r_1, \dots, r_{i-1})$$

■ History-based grammars

$$P(d) = \prod_{i=1}^m P(r_i | \pi(h_i))$$

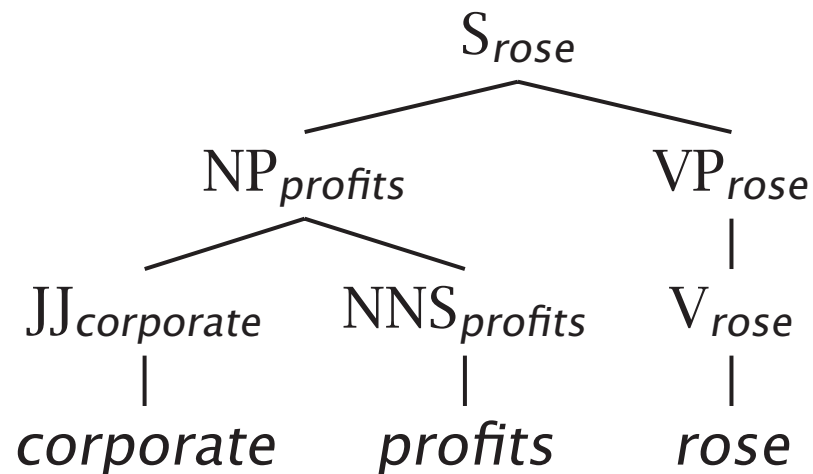
NLP as a classification problem

- Central to recent advances in NLP has been reconceptualizing NLP as a statistical classification problem
- We – preferably someone else – hand-annotate data, and then learn using standard ML methods
- Annotated data items are feature vectors \vec{x}_i with a classification c_i .
- Our job is to assign an unannotated data item \vec{x} to one of the classes c_k .
- We can then use decision trees, Bayes nets, neural nets,

Parsing as classification decisions

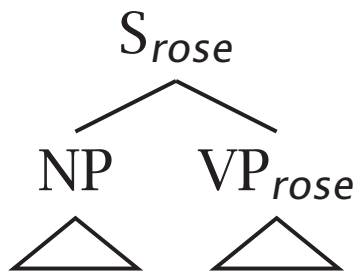
E.g., Charniak (1997)

- A very simple, conservative model of a lexicalized PCFG



- Probabilistic conditioning is “top-down” (but actual computation is bottom-up)

Charniak (1997) example

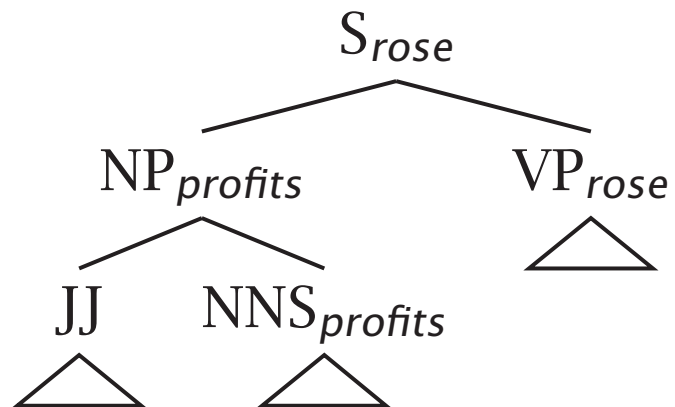
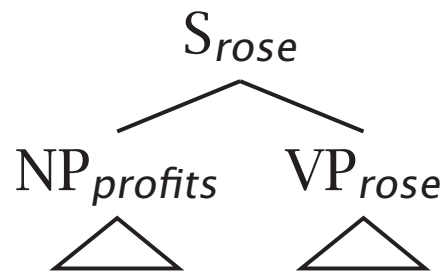


A. $h = profits; c = NP$

B. $ph = rose; pc = S$

C. $P(h|ph, c, pc)$

D. $P(r|h, c, pc)$



Charniak (1997) linear interpolation/shrinkage

$$\begin{aligned}\hat{P}(h|ph, c, pc) &= \lambda_1(e)P_{\text{MLE}}(h|ph, c, pc) \\ &\quad + \lambda_2(e)P_{\text{MLE}}(h|C(ph), c, pc) \\ &\quad + \lambda_3(e)P_{\text{MLE}}(h|c, pc) + \lambda_4(e)P_{\text{MLE}}(h|c)\end{aligned}$$

- $\lambda_i(e)$ is here a function of how much one would expect to see a certain occurrence, given the amount of training data, word counts, etc.
- $C(ph)$ is semantic class of parent headword
- Techniques like these for dealing with data sparseness are vital to successful model construction

Charniak (1997) shrinkage example

	$P(\text{prft} \text{rose, NP, S})$	$P(\text{corp} \text{prft, JJ, NP})$
$P(h ph, c, pc)$	0	0.245
$P(h C(ph), c, pc)$	0.00352	0.0150
$P(h c, pc)$	0.000627	0.00533
$P(h c)$	0.000557	0.00418

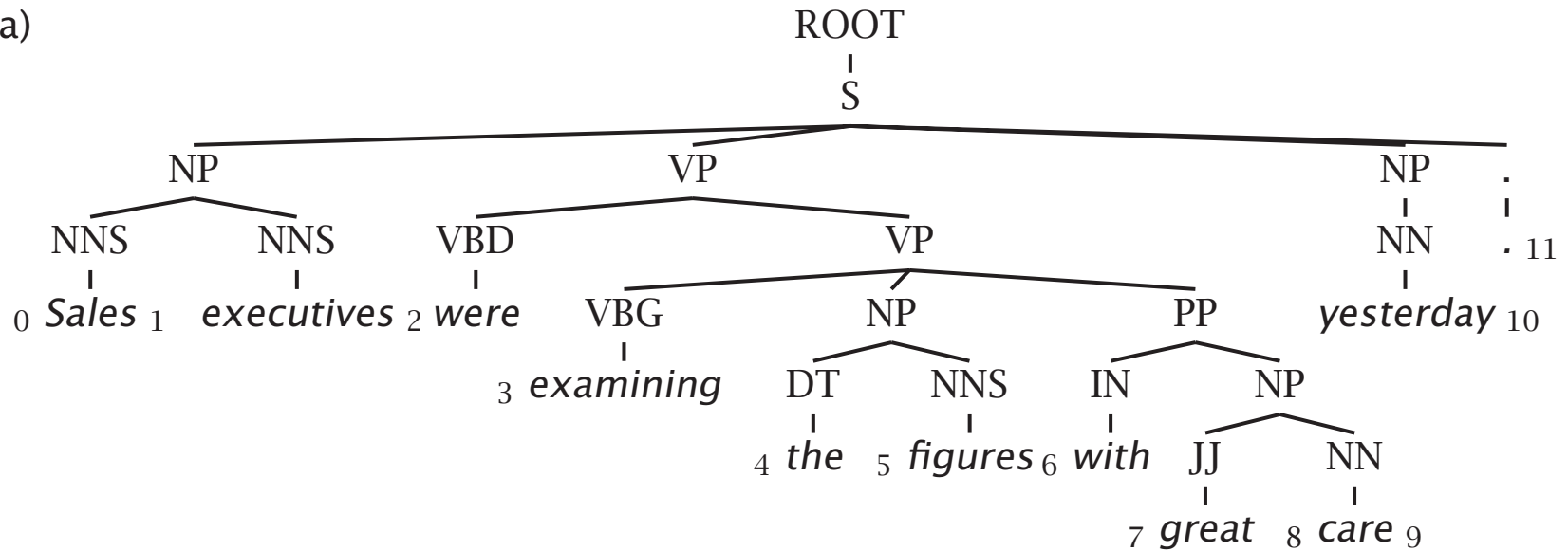
- Allows utilization of rich highly conditioned estimates, but smoothes when sufficient data is unavailable
- One can't just use MLEs: one commonly sees previously unseen events, which would have probability 0.

Modern Statistical Parsers

- A greatly increased ability to do accurate, robust, broad coverage parsing (Charniak 1997, Collins 1997, Ratnaparkhi 1997, Charniak 2000)
- Achieved by converting parsing into a classification task and using statistical/machine learning methods
- Statistical methods (fairly) accurately resolve structural and real world ambiguities
- Much faster: rather than being cubic in the sentence length or worse, for modern statistical parsers parsing time is made linear (by using beam search)
- Provide probabilistic language models that can be integrated with speech recognition systems.

Evaluation

(a)



(b) Brackets in gold standard tree (a.):

S-(0:11), **NP-(0:2)**, **VP-(2:9)**, **VP-(3:9)**, **NP-(4:6)**, **PP-(6-9)**, **NP-(7,9)**, ***NP-(9:10)**

(c) Brackets in candidate parse (b.):

S-(0:11), **NP-(0:2)**, **VP-(2:10)**, **VP-(3:10)**, **NP-(4:10)**, **NP-(4:6)**, **PP-(6-10)**, **NP-(7,10)**

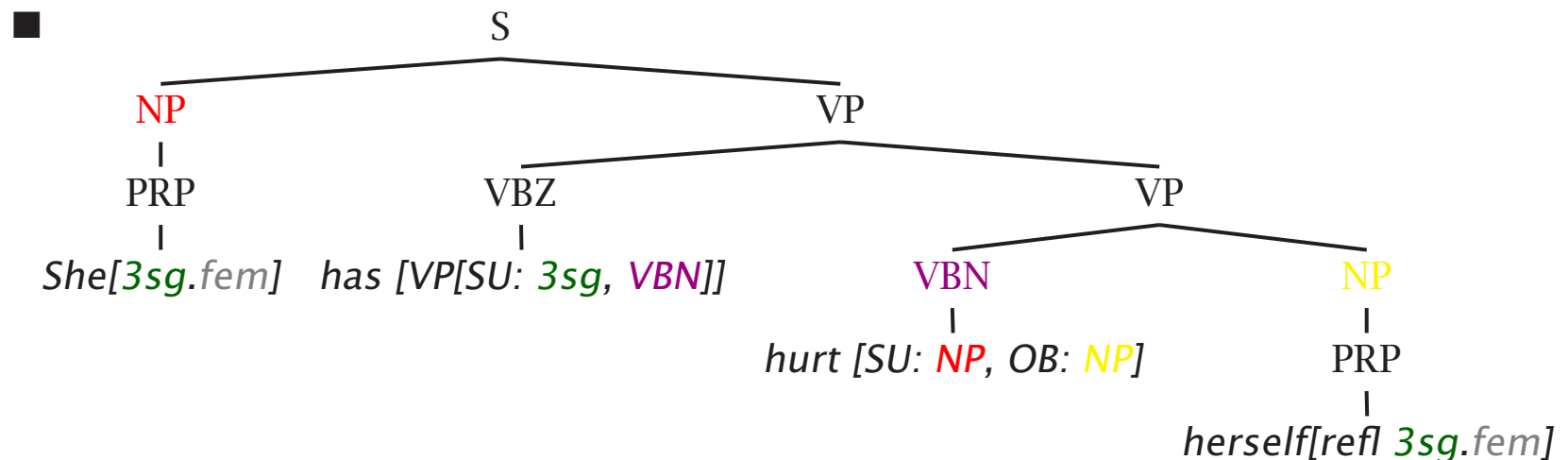
(d) Precision:	3/8 = 37.5%	Crossing Brackets:	0
Recall:	3/8 = 37.5%	Crossing Accuracy:	100%
Labeled Precision:	3/8 = 37.5%	Tagging Accuracy:	10/11 = 90.9%
Labeled Recall:	3/8 = 37.5%		

Parser results

- Normally people present systems balanced on precision/recall
- Normally evaluate on sentences of 40 words or less
- Magerman (1995) got about 85% labeled precision and recall
- Charniak (2000) gets 90.1% labeled precision and recall on sentences less than 40 words
- Impressive! Still steady progress in error reduction
- At some point size of and errors in treebank must become limiting factor

Beyond augmented PCFGs

- For branching process models, relative frequency probability estimates give ML estimates on observed data
- But because of the rich feature dependencies in language, linguists like to use richer constraint models:



- Abney (1997) and Johnson et al. (1999) develop log-linear Markov Random Field/Gibbs models

Extracting facts

- *The problem with IR:* You search for “soldiers attacking rebels” and the top matches are:
 - Hutu rebels attacked soldiers in one of Bujumbura’s suburbs (Google 2000/10/03)
 - Sudanese rebels say they have killed or wounded more than 300 government soldiers (Hotbot 2000/10/03)
 - [Altavista: a Confederate soldier’s memorial!]
- We need to be able to match relationships like:
 - attack(soldiers, rebels)
- Models that see sentence structure, like dependency parsers, let us capture these relations, though we still need to deal with synonymy and polysemy

From structure to meaning

- Syntactic structures aren't meanings, but having heads and dependents essentially gives one relations:
 - orders(president, review(spectrum(wireless)))
- We don't yet resolve (noun phrase) scope, but that's probably too hard for robust broad-coverage NLP
- Main remaining problems: synonymy and polysemy:
 - Words have multiple meanings
 - Several words can mean the same thing
- But there are well-performing methods of also statistically disambiguating and clustering words as well
- So the goal of transforming a text into meaning relations or "facts" is close

Summary

- This has been a quick overview of what NLP is, the problems of NLU, and the methods used to approach that problem
- I've focussed on statistical methods, and in particular on statistical parsing methods
- Statistical methods have brought a new level of performance in robust, accurate, broad-coverage NLP
- They provide a fair degree of disambiguation and interpretation, integrable with other systems
- The time seems ripe to combine sophisticated yet robust NLP models (which do more with meaning) with richer probabilistic contextual models

The End

Thanks for listening!

Bibliography

Abney, S. P. 1997. Stochastic attribute-value grammars. *Computational Linguistics* 23(4):597–618.

Allen, J. 1995. *Natural Language Understanding*. Redwood City, CA: Benjamin Cummings.

Carpenter, B. 1999. *Type-Logical Semantics*. Cambridge, MA: MIT Press.

Charniak, E. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI '97)*, 598–603.

Charniak, E. 2000. A maximum-entropy-inspired parser. In *NAACL 1*, 132–139.

Chomsky, N. 1957. *Syntactic Structures*. The Hague: Mouton.

Collins, M. J. 1996. A new statistical parser based on bigram lexical dependencies. In *ACL 34*, 184–191.

Collins, M. J. 1997. Three generative, lexicalised models for statistical parsing. In *ACL 35/EACL 8*, 16–23.

Ford, M., J. Bresnan, and R. M. Kaplan. 1982. A competence-based theory of syntactic closure. In J. Bresnan (Ed.), *The Mental Representation of Grammatical Relations*, 727–796. Cambridge, MA: MIT Press.

Gold, E. M. 1967. Language identification in the limit. *Information and Control* 10:447–474.

Horning, J. J. 1969. *A study of grammatical inference*. PhD thesis, Stanford.

Johnson, M., S. Geman, S. Canon, Z. Chi, and S. Riezler. 1999. Estimators for stochastic “unification-based” grammars. In *ACL 37*, 535–541.

Jurafsky, D., and J. H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Englewood Cliffs, NJ: Prentice Hall.

Lambek, J. 1958. The mathematics of sentence structure. *American Mathematical Monthly* 65:154–170. Also in Buzkowski, W., W. Marciszewski and J. van Benthem, eds., *Categorical Grammar*. Amsterdam: John Benjamin.

Magerman, D. M. 1995. Review of ‘Statistical language learning’. *Computational Linguistics* 11(1):103–111.

Manning, C. D., and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. Boston, MA: MIT Press.

Montague, R. 1973. The proper treatment of quantification in ordinary English. In J. Hintikka, J. Moravcsik, and P. Suppes (Eds.), *Approaches to Natural Language*. Dordrecht: D. Reidel.

Ratnaparkhi, A. 1997. A simple introduction to maximum entropy models for natural language processing. Technical Report IRCS Report 97–08, Institute for Research in Cognitive Science, Philadelphia, PA.

Sapir, E. 1921. *Language: an introduction to the study of speech*. New York: Harcourt Brace.