Probabilistic Models in Computational Linguistics

Christopher Manning

Depts of Computer Science and Linguistics
Stanford University

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

Aims and Goals of Computational Linguistics

- To be able to understand and act on human languages
- To be able to fluently produce human languages
- *Applied goals:* machine translation, question answering, information retrieval, speech-driven personal assistants, text mining, report generation, . . .

The big questions for linguistic science

- What kinds of things do people say?
- What do these things say/ask/request about the world?

I will argue that answering these involves questions of frequency, probability, and likelihood

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)
    - $\dfrac{\text{NP}:\alpha \qquad \text{NP}\backslash\text{S}:\beta}{\text{S}:\beta(\alpha)}$
  - 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 → Det Adj* N Adj → 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)

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)

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

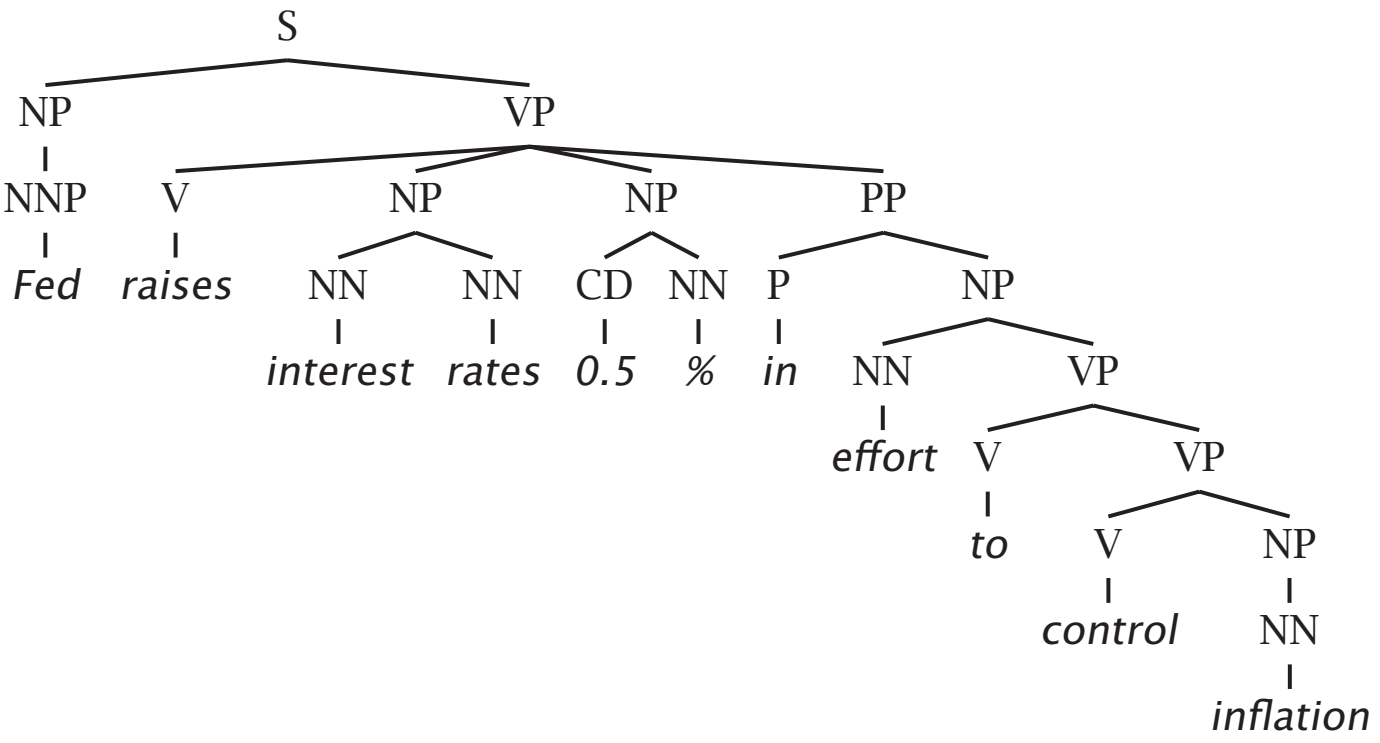
Speech and NLP: A probabilistic view

- A acoustic signal
- W words
- T syntactic (tree) structures
- M meanings
- In spoken language use, we have a distribution:

$$P(A, W, T, M)$$

- In written language, just: $P(W, T, M)$
- Speech people have usually looked at: $P(W|A)$ – the rest of the hidden structure is ignored
- NLP people interested in the 'more hidden' structure – T and often M – but sometimes W is observable
- E.g., there is much work looking at the parsing problem $P(T|W)$. Language generation is $P(W|M)$.

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)
- 

Where are the ambiguities?

Part of speech ambiguities

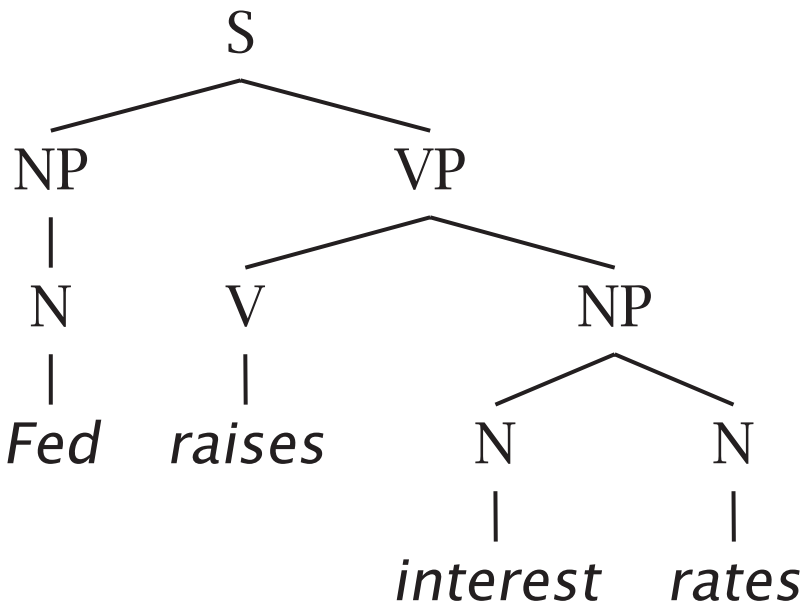
		VB					
	VBZ	VBP	VBZ				
NNP	NNS	NN	NNS	CD	NN		
Fed	*raises*	*interest*	*rates*	*0.5*	*%*	*in*	*effort*
						to	*control*
							inflation

Syntactic attachment ambiguities

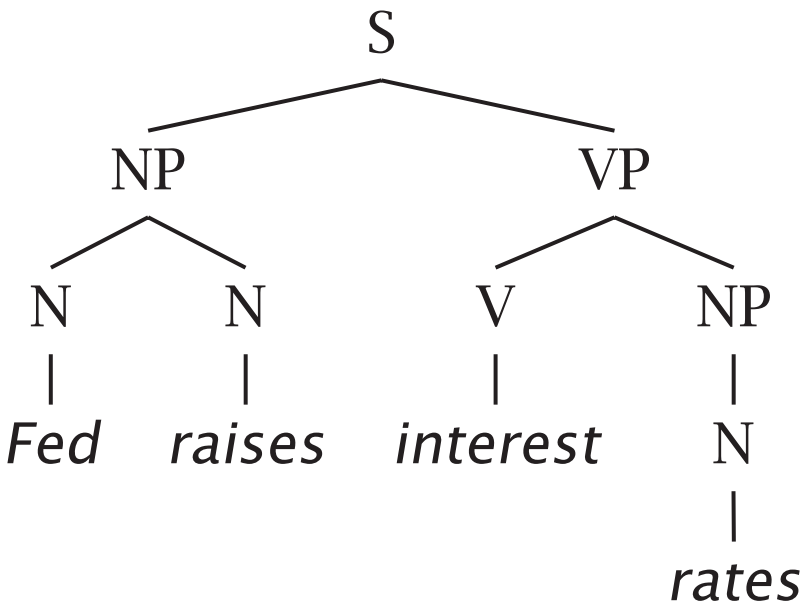
Word sense ambiguities: Fed → *"federal agent"*
interest → *a feeling of wanting to know or learn more*

Semantic interpretation ambiguities above the word level

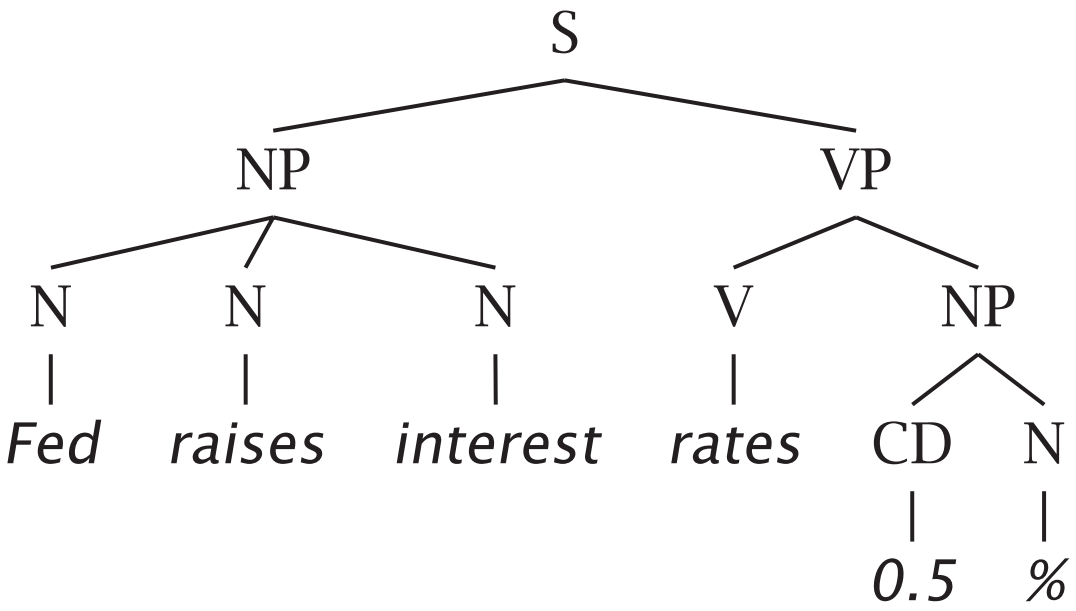
The bad effects of V/N ambiguities (1)



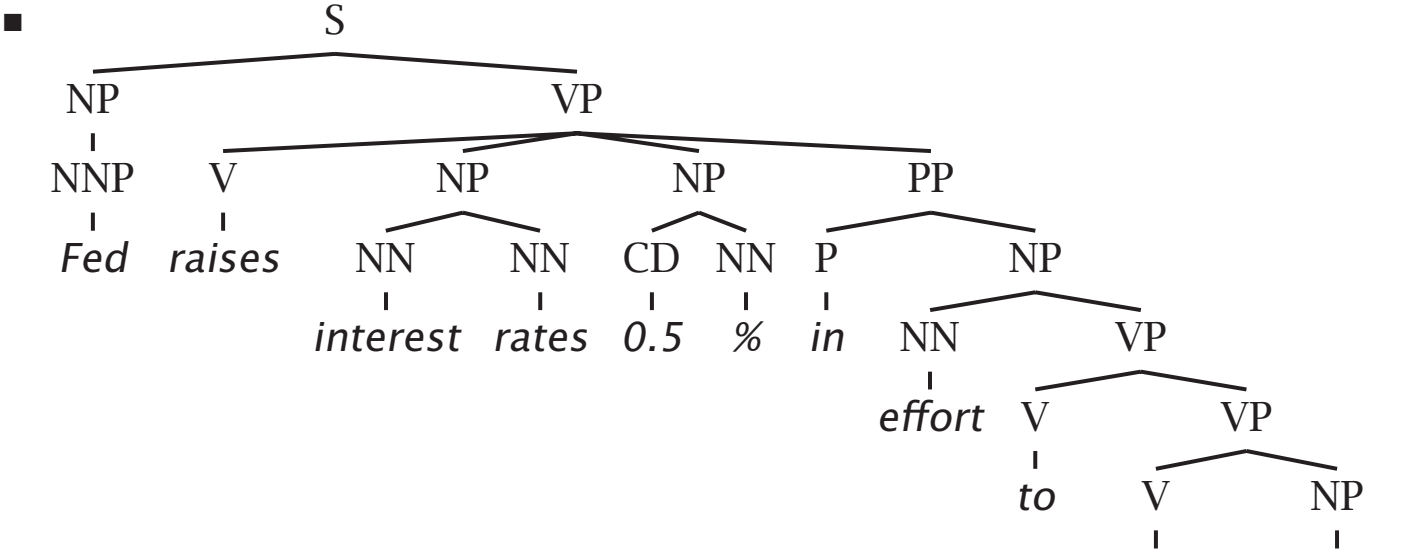
The bad effects of V/N ambiguities (2)

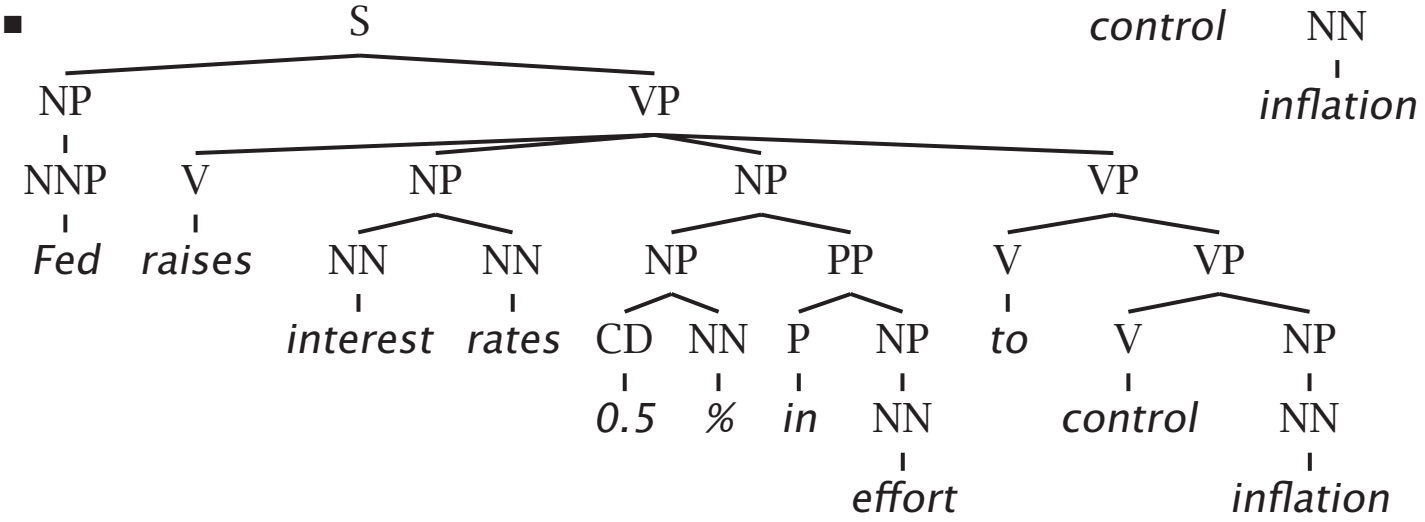


The bad effects of V/N ambiguities (3)



Phrasal attachment ambiguities

- 

- 

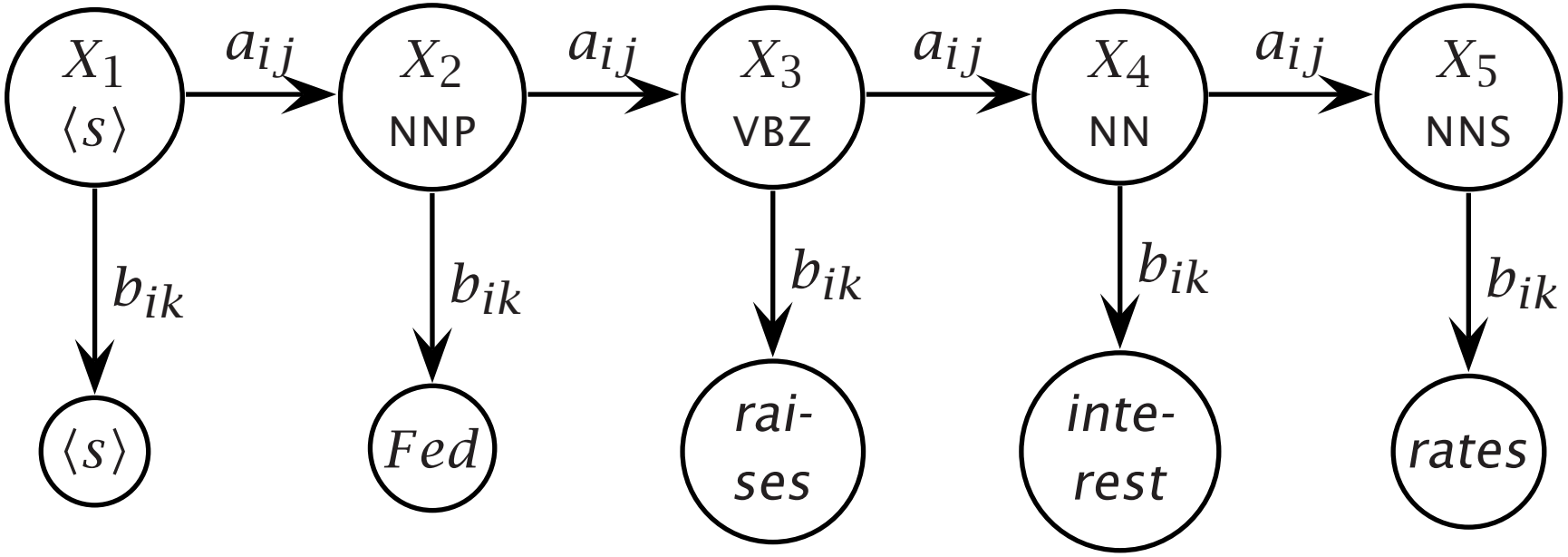
The many meanings of *interest* [n.]

- 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:

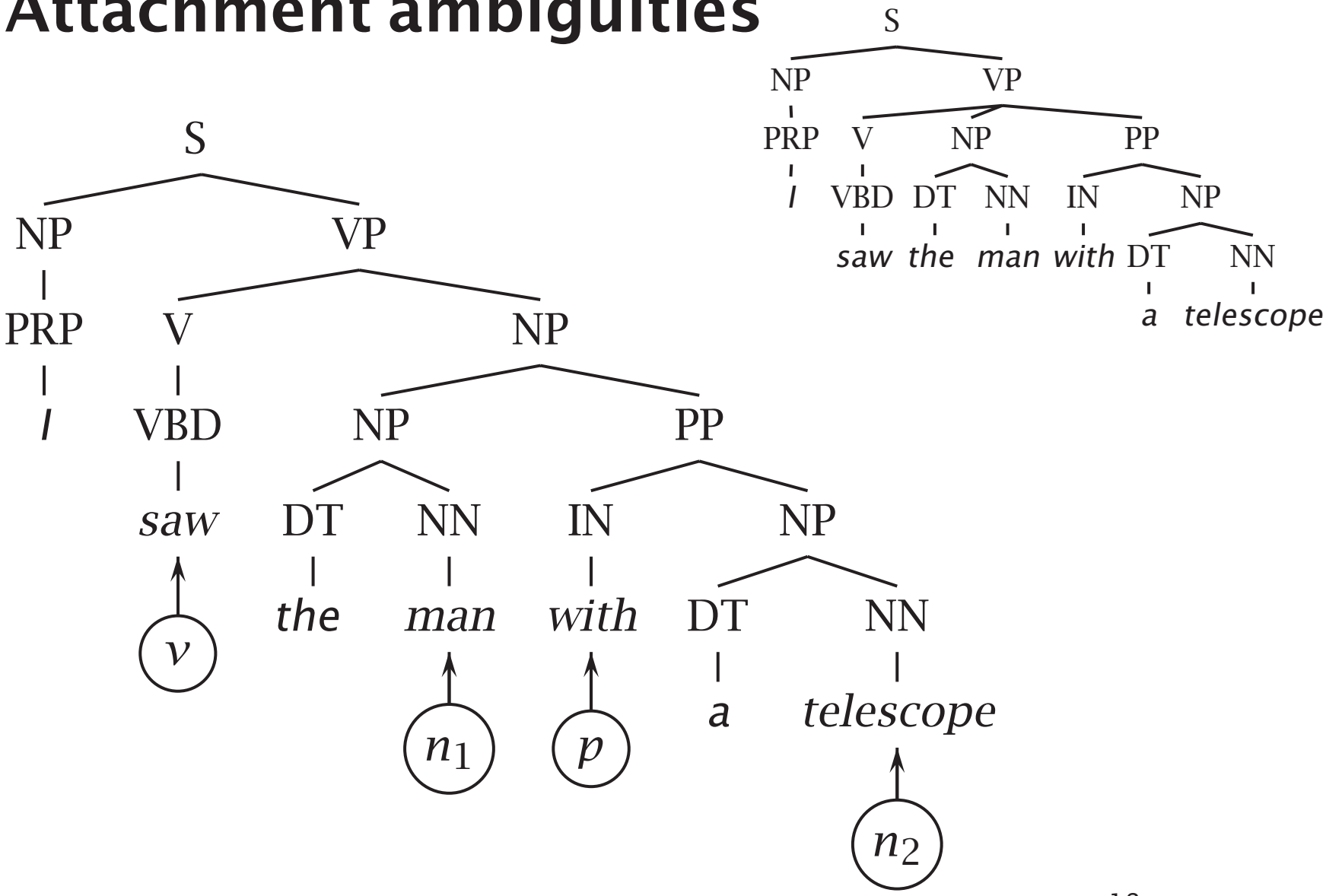
image, likeness, portrait, facsimile, picture

Hidden Markov Models – POS example



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

Attachment ambiguities



Likelihood ratios for PP attachment

- Likely attachment chosen by a (log) likelihood ratio:

$$\lambda(v,n,p) = \log_2 \frac{P(\text{Attach}(p)=v|v,n)}{P(\text{Attach}(p)=n|v,n)}$$

$$= \log_2 \frac{P(\text{VA}_p=1|v)P(\text{NA}_p=0|v)}{P(\text{NA}_p=1|n)}$$

  If (large) positive, decide verb attachment [e.g., below]; if (large) negative, decide noun attachment.

- *Moscow sent more than 100,000 soldiers into Afghanistan*

$$\lambda(\textit{send},\textit{soldiers},\textit{into}) \approx \log_2 \frac{0.049 \times 0.9993}{0.0007} \approx 6.13$$

  Attachment to verb is about 70 times more likely.

(Multinomial) Naive Bayes classifiers for WSD

- $\vec{x}$ is the context (something like a 100 word window)
- c_k is a sense of the word to be disambiguated

$$
\begin{aligned}
\text{Choose } c' &= \arg\max_{c_k} P(c_k|\vec{x}) \\
&= \arg\max_{c_k} \frac{P(\vec{x}|c_k)}{P(\vec{x})} P(c_k) \\
&= \arg\max_{c_k} [\log P(\vec{x}|c_k) + \log P(c_k)] \\
&= \arg\max_{c_k} [\sum_{v_j \text{ in } \vec{x}} \log P(v_j|c_k) + \log P(c_k)]
\end{aligned}
$$

- An effective method in practice, but also an example of a structure-blind 'bag of words' model

Statistical Computational Linguistic Methods

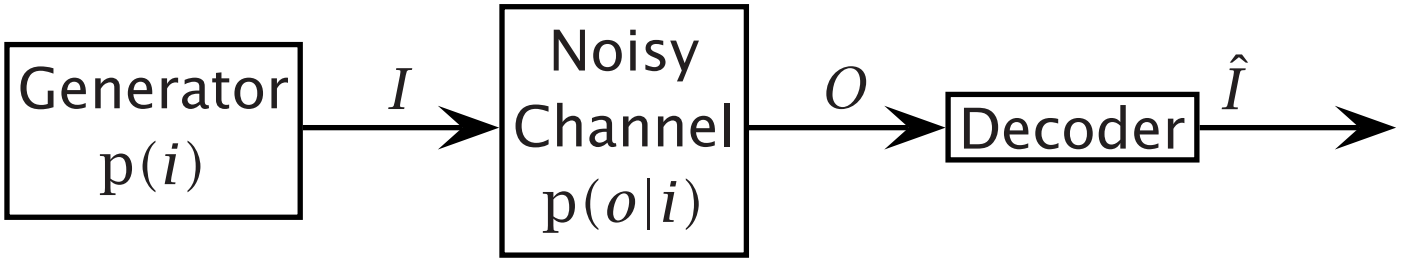
- Many (related) techniques are used:
  - n-grams, history-based models, decision trees / decision lists, memory-based learning, loglinear models, HMMs, neural networks, vector spaces, graphical models, decomposable models, PCFGs, Probabilistic LSI, . . .
- Predictive and robust
- Good for learning (well, supervised learning works well; unsupervised learning is still hard)
- The list looks pretty similar to speech work . . .
  because we copied from them

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 (or possibly to the doubt $\mathcal{D}$ or outlier $\mathcal{O}$ categories, though in practice rarely used).

Simple Bayesian Inference for NLP

- Central conception in early work: The "noisy channel" model. We want to determine English text given acoustic signal, OCRed text, French text, . . .



words	speech	words
POS tags	words	POS tags
L_1 words	L_2 words	L_1 words

- $\hat{\imath} = \arg\max_i P(i|o) = \arg\max_i P(i) \times P(o|i)$

Probabilistic inference in more generality

- Overall there is a joint distribution of all the variables
  - e.g., $P(s, t, m, d)$
- We assume a generative or causal model that factorizes the joint distribution:
  - e.g., $P(t)P(s|t)P(m|t)P(d|m)$
- This allows the distribution to be represented compactly
- Some items in this distribution are observed
- We do inference to find other parts:
  - $P(Hidden|Obs = o_1)$

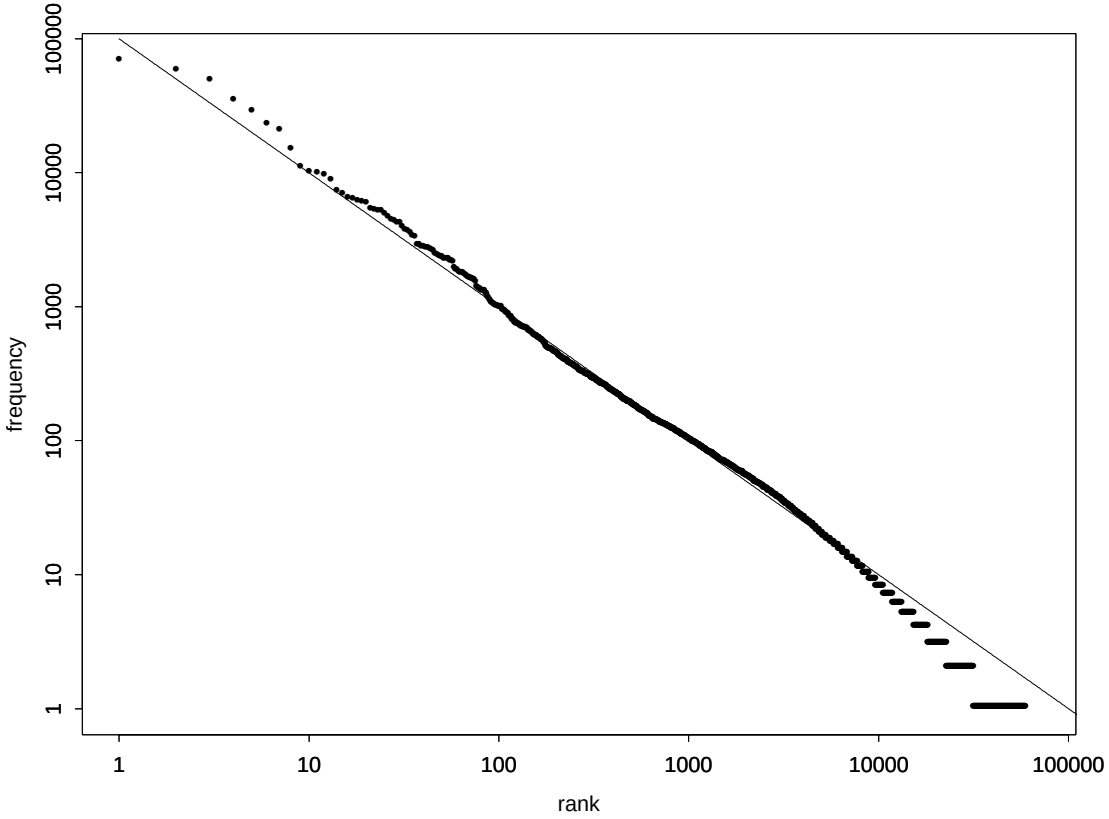
Machine Learning for NLP

Method \ Problem	POS tagging	WSD	Parsing
Naive Bayes		Gale et al. (1992)	
(H)MM	Charniak et al. (1993)		
Decision Trees	Schmid (1994)	Mooney (1996) Ringuette (1994)	Magerman (1995)
Decision List/TBL	Brill (1995)		Brill (1993)
kNN/MBL	Daelemans et al. (1996)	Ng and Lee (1996)	Zavrel et al. (1997)
Maximum entropy	Ratnaparkhi (1996)		Ratnaparkhi et al. (1994)
Neural networks	Benello et al. (1989)		Henderson and Lane (1998)

Distinctiveness of NLP as an ML problem

- Language allows the complex compositional encoding of thoughts, ideas, feelings, . . . , intelligence.
- We are minimally dealing with hierarchical structures (branching processes), and often want to allow more complex forms of information sharing (dependencies).
- Enormous problems with data sparseness
- Both features and assigned classes regularly involve multinomial distributions over huge numbers of values (often in the tens of thousands)
- Generally dealing with discrete distributions though!
- The distributions are very uneven, and have fat tails

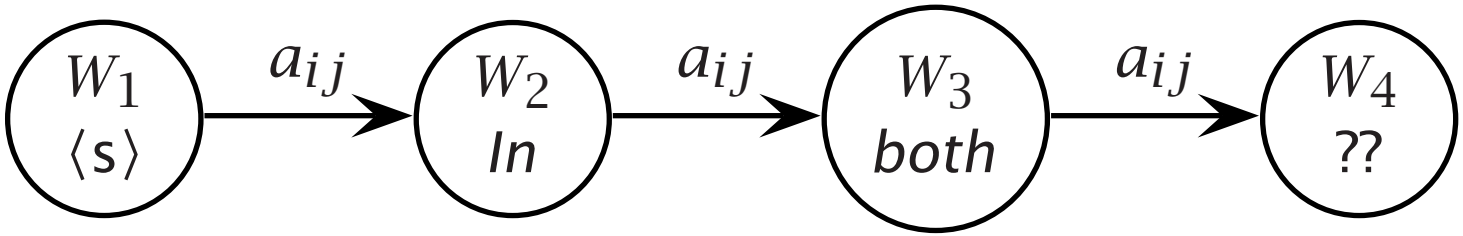
The obligatory Zipf's law slide: Zipf's law for the Brown corpus



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

Simple linear models of language

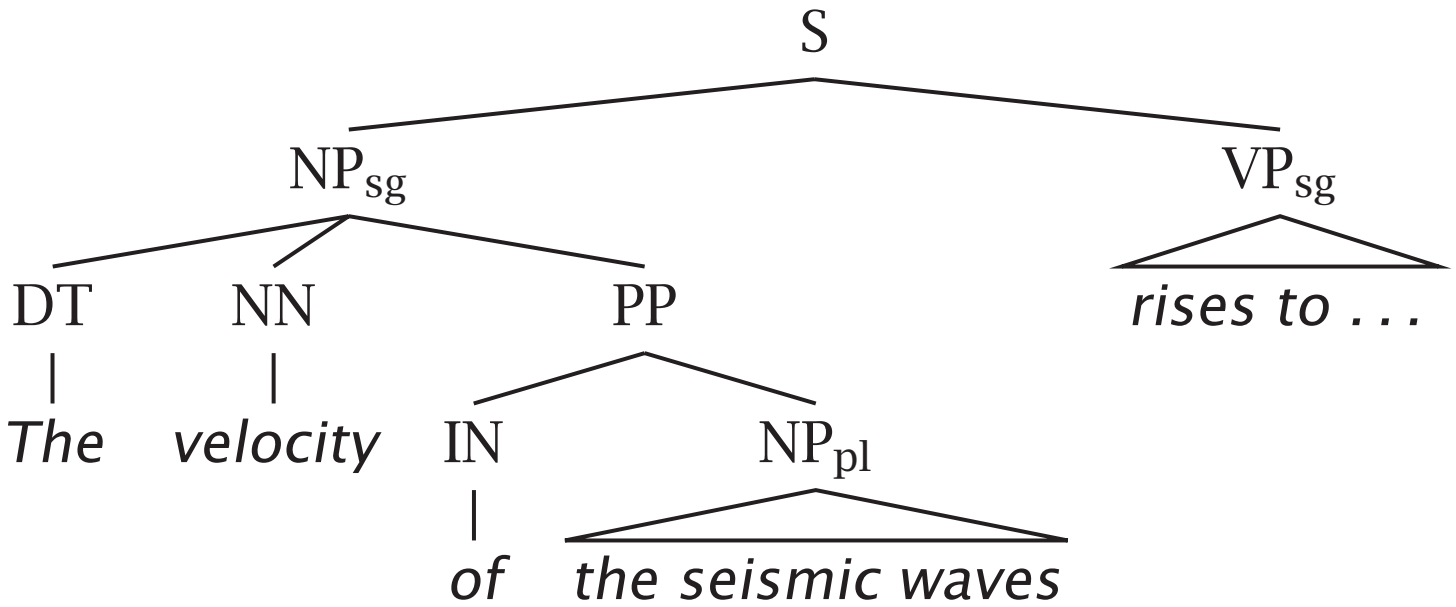
- Markov models a.k.a. n-gram models:



  - Word sequence is predicted via a conditional distribution
  - Conditional Probability Table (CPT): e.g., $P(X|\textit{both})$
    - $P(\textit{of}|\textit{both}) = 0.066 \qquad P(\textit{to}|\textit{both}) = 0.041$
  - Amazingly successful as a simple engineering model
- Hidden Markov Models (above, for POS tagging)
- Linear models panned by Chomsky (1957)

Why we need recursive structure

- The velocity of the seismic waves rises to . . .



- Or you can use dependency grammar representations – isomorphisms exist. (Ditto link grammar.)

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
- A similar algorithm to the Viterbi algorithm is used for finding the most probable parse

Expectation Maximization (EM) algorithm

- For both HMMs and PCFGs, we can use EM estimation to learn the 'hidden' structure from plain text data
- We start with initial probability estimates
- E-step: We work out the expectation of the hidden variables, given the current parameters for the model
- M-step: (Assuming these are right), we calculate the maximum likelihood estimates for the parameters
- Repeat until convergence... (Dempster et al. 1977)
- It's an iterative hill-climbing algorithm that can get stuck in local maxima
- Frequently not effective if we wish to imbue the hidden states with meanings the algorithm doesn't know

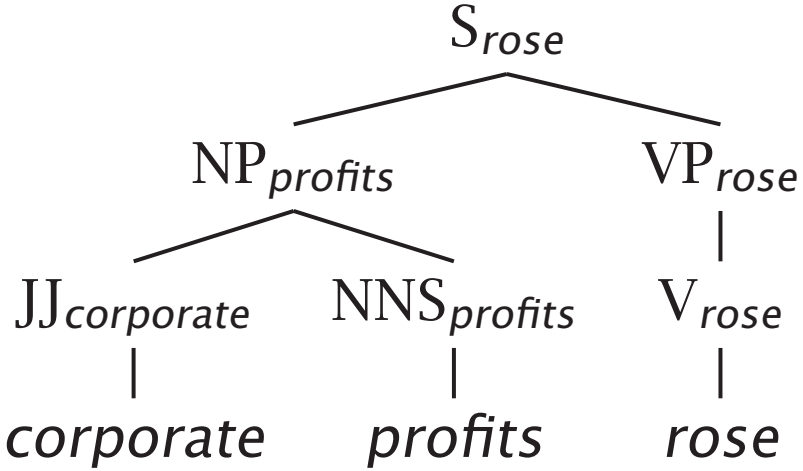
Modern Statistical Parsers

- 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.

Parsing as classification decisions

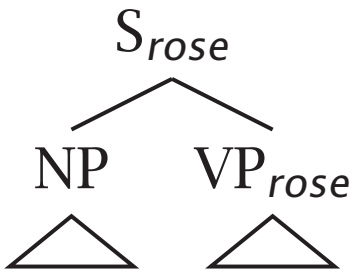
E.g., Charniak (1997)

- A very simple, conservative model of lexicalized PCFG



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

Charniak (1997) example

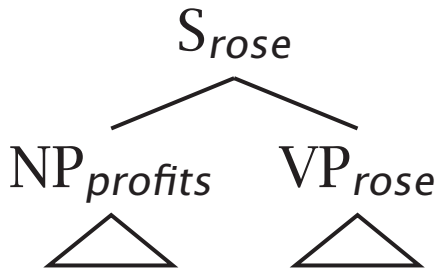


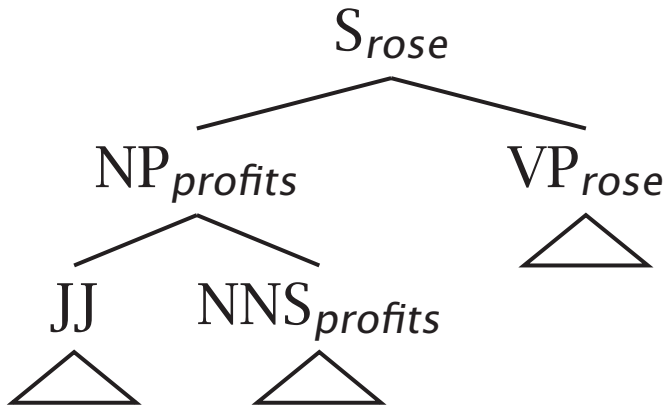
A. h = *profits*; c = NP

B. ph = *rose*; pc = S

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

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





Charniak (1997) linear interpolation/shrinkage

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

- $\lambda_i(e)$ 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

	$P(\text{prft}\|\text{rose}, \text{NP}, \text{S})$	$P(\text{corp}\|\text{prft}, \text{JJ}, \text{NP})$
$P(h\|ph, c, pc)$	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.

Unifying different approaches

- Most StatNLP work is using loglinear/exponential models
- For discrete distributions – common in NLP! – we can build a contingency table model of the joint distribution of the data.
- Example contingency table: predicting POS JJ ($N = 150$)

		f_1		
		+hyphen	−hyphen	
f_2	+ *-al*	Y: 8 N: 2	Y: 18 N: 27	Y: 26 N: 29
	− *-al*	Y: 10 N: 20	Y: 3 N: 62	Y: 13 N: 82
		Y: 18 N: 22	Y: 21 N: 89	Y: 39 N: 111

Loglinear/exponential ("maxent") models

- Most common modeling choice is a loglinear model:

$$\log P(X_1 = x_1, \ldots, X_p = x_p) = \sum_C \lambda_C(x_C)$$

  where $C \subset \{1, \ldots, p\}$.

- Maximum entropy loglinear models

$$\mathrm{p}(\vec{x}, c) = \frac{1}{Z} \prod_{i=1}^{K} \alpha_i^{f_i(\vec{x}, c)}$$

  K is the number of features, α_i is the weight for feature f_i and Z is a normalizing constant. Log form:

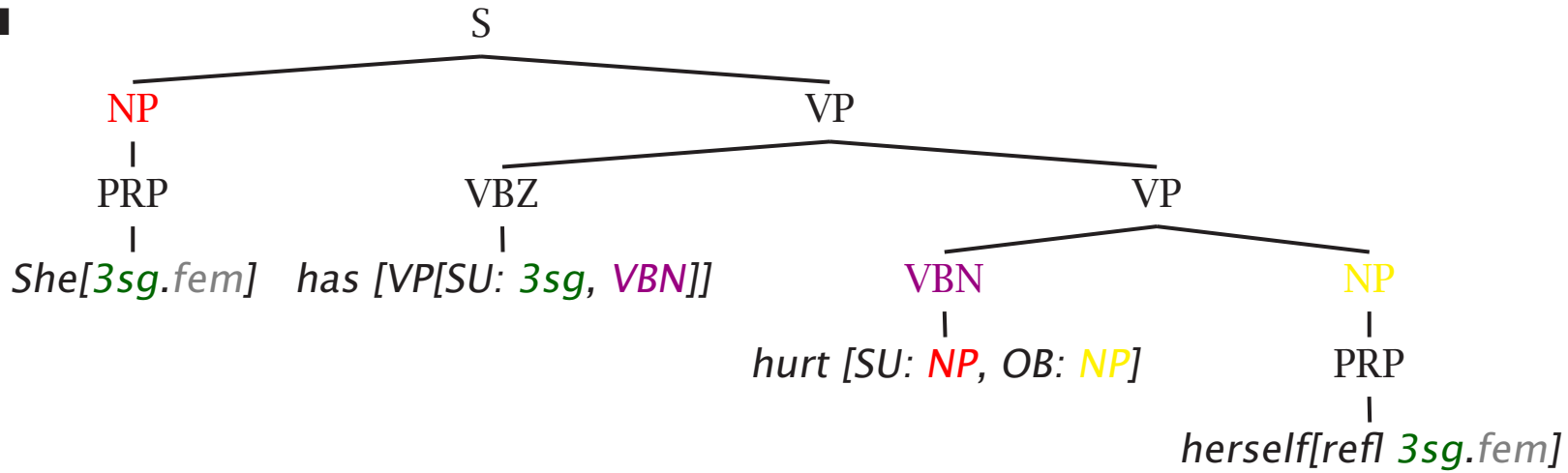
$$\log \mathrm{p}(\vec{x}, c) = -\log Z + \sum_{i=1}^{K} f_i(\vec{x}, c) \times \log \alpha_i$$

- Generalized iterative scaling gives unique ML solution

The standard models are loglinear

- *All* the widely used generative probability models in StatNLP are loglinear, because they're done as a product of probabilities decomposed by the chain rule (Naive Bayes, HMMs, PCFGs, decomposable models, Charniak (1997), Collins (1997) . . .)
- The simpler ones (Naive Bayes, HMMs, . . .) can also easily be interpreted as Bayes Nets/"graphical models" (Pearl 1988), as in the pictures earlier

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:
- 

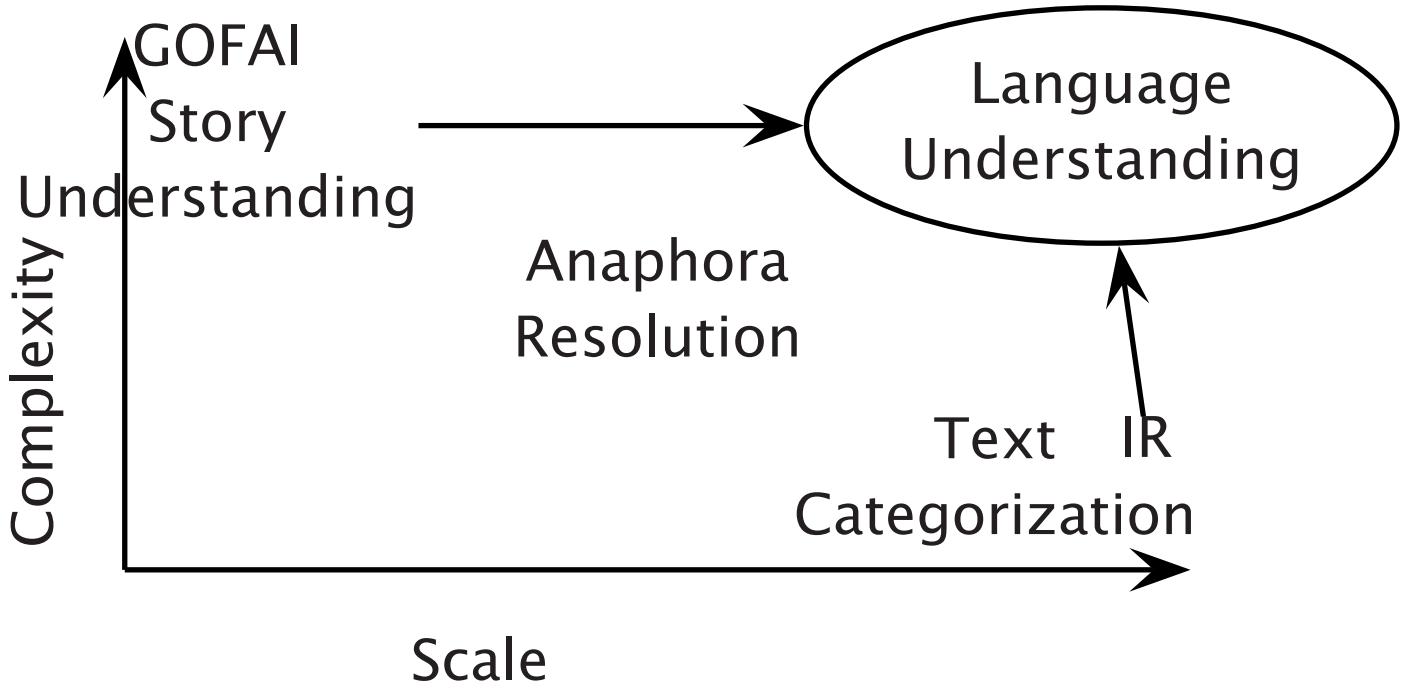
- Abney (1997) and Johnson et al. (1999) develop log-linear Markov Random Field/Gibbs models

But the whole NLP world isn't loglinear

- Other methods, e.g., non-parametric instance-based learning methods are also used
  - The Memory-Based Learning approach (Daelemans et al. 1996) has achieved good results for many NLP problems
- Also quantitative but non-probabilistic methods, such as vector spaces
- Latent semantic indexing via singular value decomposition is often effective for dimensionality reduction and unsupervised clustering
  - E.g., Schütze (1997) for learning parts of speech, word clusters, and word sense clusters

What we don't know how to do yet:

Where are we at on meaning?



Miller et al. (1996) [BBN]

- System over ATIS air travel domain
- Discourse-embedded meaning processing:
  - U: I want to fly from Boston to Denver
  - S: OK ⟨flights are displayed⟩
  - U: Which flights are available on Tuesday
  - [interpret as flights from Boston to Denver]
  - S: ⟨displays appropriate flights⟩
- End-to-end statistical model from words to discourse-embedded meaning: cross-sentence discourse model.

Miller et al. (1996) [BBN]

- Three stage n-best pipeline:
- Pragmatic interpetation D from words W and discourse history H via sentence meaning M and parse tree T

$$\begin{aligned}
\hat{D} &= \arg\max_D P(D|W,H) \\
&= \arg\max_D \sum_{M,T} P(D|W,H,M,T)P(M,T|W,H) \\
&= \arg\max_D \sum_{M,T} P(D|H,M)P(M,T|W)
\end{aligned}$$

- Possible because of annotated language resources that allow supervised ML at all stages (and a rather simple slot-filler meaning representation)

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

Integrating probabilistic reasoning about context with probabilistic language processing

- Paek and Horvitz (2000) treats conversation as inference and decision making under uncertainty
- Quartet: A framework for spoken dialog which models and exploits uncertainty in:
  - conversational control
  - intentions
  - maintenance (notices lack of understanding, etc.)
- Attempts to model the development of mutual understanding in a dialog
- But language model is very simple
- Much more to do in incorporating knowledge into NLP models

Learning and transfering knowledge

- We can do well iff we can train our models on supervised data from the same domain
- We have adequate data for very few domains/genres
- In general, there have been modest to poor results in learning rich NLP models from unannotated data
- It is underexplored how one can adapt or bootstrap with knowledge from one domain to another where data is more limited or only available unannotated
- Perhaps we need to more intelligently design models that use less parameters (but the right conditioning)?
- These are vital questions for making StatNLP interesting to cognitive science

Sometimes an approach where probabilities annotate a symbolic grammar isn't sufficient

- There's lots of evidence that our representations should also be squishy. E.g.:
  - What part of speech do "marginal prepositions" have? *concerning*, *supposing*, *considering*, *regarding*, *following*
  - Transitive verb case: *Asia's other cash-rich countries* ***are following*** *Japan's lead.*
  - Marginal preposition (VP modifier, sense of *after*): *U.S. chip makers are facing continued slack demand* ***following*** *a traditionally slow summer.*
  - Penn Treebank tries to mandate that they are verbs

- But some have already moved to become only prepositions: *during* (originally a verb, cf. *endure*) and *notwithstanding* (a compound from a verb)
- And others seem well on their way:
- ***According** to this, industrial production declined*
- They're in between being verbs and prepositions

- Conversely standard probabilistic models don't explain why language is 'almost categorical': categorical grammars have been used for thousands of years because they just about work. . . .
- In many places there is a very steep drop-off between 'grammatical' and 'ungrammatical' strings that our probabilistic models often don't model well

Envoi

- 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
- To avoid plateauing, we need to keep developing richer and more satisfactory representational models
- The time seems ripe to combine sophisticated yet robust NLP models (which do more with meaning) with richer probabilistic contextual models

Thanks for listening!

Bibliography

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

Benello, J., A. W. Mackie, and J. A. Anderson. 1989. Syntactic category disambiguation with neural networks. *Computer Speech and Language* 3:203–217.

Brill, E. 1993. Automatic grammar induction and parsing free text: A transformation-based approach. In *ACL 31*, 259–265.

Brill, E. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics* 21(4):543–565.

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.

Charniak, E., C. Hendrickson, N. Jacobson, and M. Perkowitz. 1993. Equations for part-of-speech tagging. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 784–789, Menlo Park, CA.

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

Chomsky, N. 1969. Quine's empirical assumptions. In D. Davidson and J. Hintikka (Eds.), *Words and Objections: Essays on the Work of W.V. Quine*, 53–68. Dordrecht: D. Reidel.

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

Daelemans, W., J. Zavrel, P. Berck, and S. Gillis. 1996. MBT: A memory-based part of speech tagger generator. In *WVLC 4*, 14–27.

Dempster, A., N. Laird, and D. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society Series B* 39:1–38.

Gale, W. A., K. W. Church, and D. Yarowsky. 1992. A method for disambiguating word senses in a large corpus. *Computers and the Humanities* 26:415–439.

Henderson, J., and P. Lane. 1998. A connectionist architecture for learning to parse. In *ACL 36/COLING 17*, 531–537.

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

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., *Categorial Grammar*. Amsterdam: John Benjamin.

Lyons, J. 1968. *Introduction to Theoretical Linguistics*. Cambridge: Cambridge University Press.

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.

Miller, S., D. Stallard, and R. Schwartz. 1996. A fully statistical approach to natural language interfaces. In *ACL 34*, 55–61.

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.

Mooney, R. J. 1996. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *EMNLP 1*, 82–91.

Ng, H. T., and H. B. Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *ACL 34*, 40–47.

Paek, T., and E. Horvitz. 2000. Conversation as action under uncertainty. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI-2000)*.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inferenc*e. San Mateo, CA: Morgan Kaufmann.

Ratnaparkhi, A. 1996. A maximum entropy model for part-of-speech tagging. In *EMNLP 1*, 133–142.

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.

Ratnaparkhi, A., J. Reynar, and S. Roukos. 1994. A maximum entropy model for prepositional phrase attachment. In *Proceedings*

of the ARPA Workshop on Human Language Technology, 250–255, Plainsboro, NJ.

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

Schmid, H. 1994. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, 44–49, Manchester, England.

Schütze, H. 1997. *Ambiguity Resolution in Language Learning*. Stanford, CA: CSLI Publications.

Weaver, W. 1955. Translation. In W. N. Locke and A. D. Booth (Eds.), *Machine Translation of Languages: Fourteen Essays*, 15–23. New York: John Wiley & Sons.

Zavrel, J., W. Daelemans, and J. Veenstra. 1997. Resolving PP attachment ambiguities with memory-based learning. In *Proceedings of the Workshop on Computational Natural Language Learning*, 136–144, Somerset, NJ. Association for Computational Linguistics.