Intelligent Systems (AI-2)

Computer Science cpsc422, Lecture ~~28~~ 27

Nov, 16, 2016

Lecture Overview

- **Recap Probabilistic Context Free Grammars (PCFG)**
- CKY parsing for PCFG (only key steps)
- PCFG in practice: Modeling Structural and Lexical Dependencies

Sample PCFG

$S \rightarrow NP\ VP$	[.80]	$Det \rightarrow$ *that* [.05] | *the* [.80] | *a* [.15]	
$S \rightarrow Aux\ NP\ VP$	[.15]	$Noun \rightarrow$ *book*	[.10]
$S \rightarrow VP$	[.05]	$Noun \rightarrow$ *flights*	[.50]
$NP \rightarrow Det\ Nom$	[.20]	$Noun \rightarrow$ *meal*	[.40]
$NP \rightarrow Proper\text{-}Noun$	[.35]	$Verb \rightarrow$ *book*	[.30]
$NP \rightarrow Nom$	[.05]	$Verb \rightarrow$ *include*	[.30]
$NP \rightarrow Pronoun$	[.40]	$Verb \rightarrow$ *want*	[.40]
$Nom \rightarrow Noun$	[.75]	$Aux \rightarrow$ *can*	[.40]
$Nom \rightarrow Noun\ Nom$	[.20]	$Aux \rightarrow$ *does*	[.30]
$Nom \rightarrow Proper\text{-}Noun\ Nom$	[.05]	$Aux \rightarrow$ *do*	[.30]
$VP \rightarrow Verb$	[.55]	$Proper\text{-}Noun \rightarrow$ *TWA*	[.40]
$VP \rightarrow Verb\ NP$	[.40]	$Proper\text{-}Noun \rightarrow$ *Denver*	[.40]
$VP \rightarrow Verb\ NP\ NP$	[.05]	$Pronoun \rightarrow$ *you* [.40] | *I* [.60]	

PCFGs are used to….

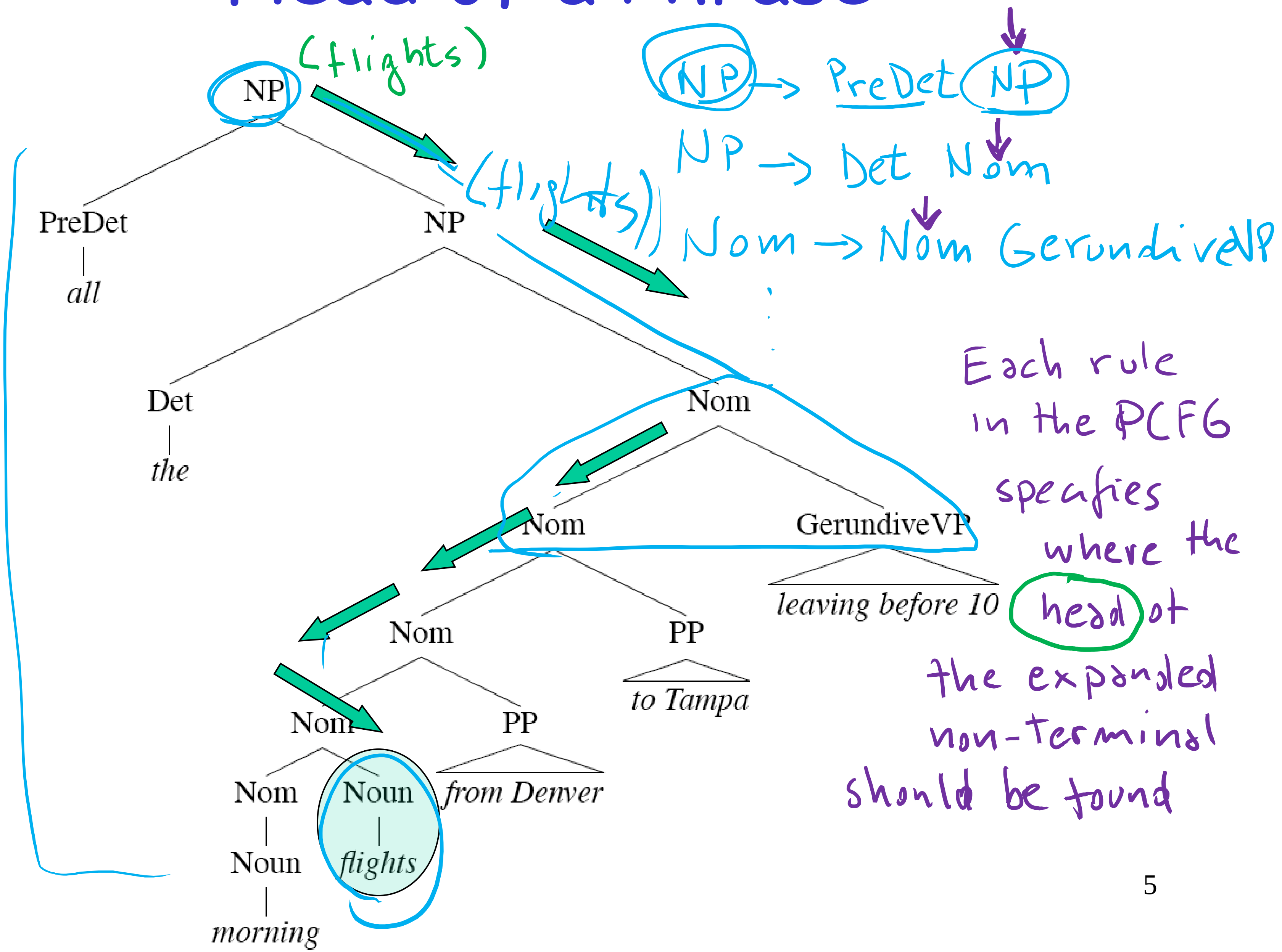
- **Estimate Prob. of parse tree**

$$P(Tree) = \prod_{node \in Tree} P(\text{expansion for node})$$

- **Estimate Prob. to sentences**

$$P(Sentence) = \sum_{Trees \in \text{Parse Trees of Sentence}} P(Tree)$$

Head of a Phrase



Each rule in the PCFG specifies where the head of the expanded non-terminal should be found

Acquiring Grammars and Probabilities

Manually parsed text corpora (e.g., PennTreebank)

- **Grammar: read it off the parse trees**

  Ex: if an NP contains an ART, ADJ, and NOUN then we create the rule *NP -> ART ADJ NOUN*.

- **Probabilities:**

$$P(A \rightarrow \alpha \mid A) = \frac{count(A \rightarrow \alpha)}{\sum_{\forall \gamma} count(A \rightarrow \gamma)} = \frac{count(A \rightarrow \alpha)}{count(A)}$$

Lecture Overview

- Recap Probabilistic Context Free Grammars (PCFG)
- **CKY parsing for PCFG (only key steps)**
- PCFG in practice: Modeling Structural and Lexical Dependencies

Probabilistic Parsing:

- (Restricted) Task is to find the max probability tree for an input

$$\hat{Tree}(Sentence) = \underset{Tree \in Parse-trees(Sentence)}{\operatorname{argmax}} P(Tree)$$

Probabilistic CKY Algorithm

Ney, 1991
Collins, 1999

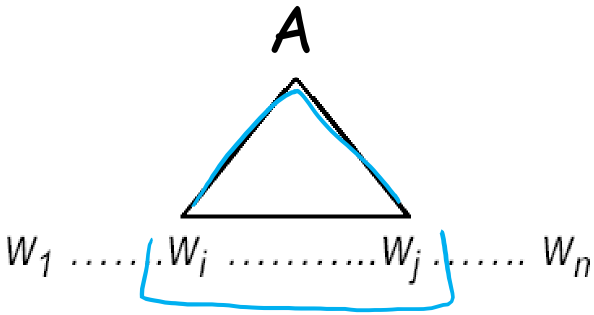
CYK (Cocke-Kasami-Younger) algorithm

- A **bottom-up** parser using dynamic programming
- Assume the PCFG is in Chomsky normal form (CNF)

A -> BC   A -> w

Definitions

- $w_1 \ldots w_n$ an input string composed of n words
- w_{ij} a string of words from word i to word j
- $\mu[i, j, A]$: a table entry holds the maximum probability for a constituent with non-terminal A spanning words $w_i \ldots w_j$

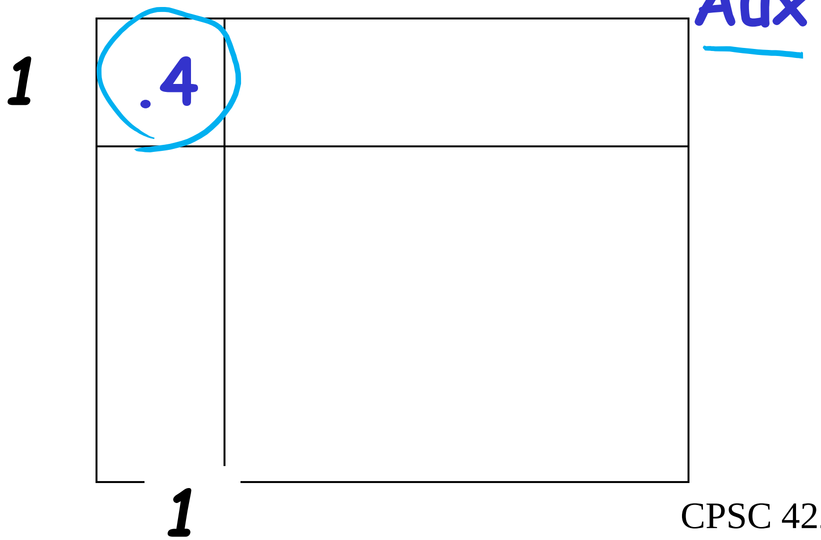


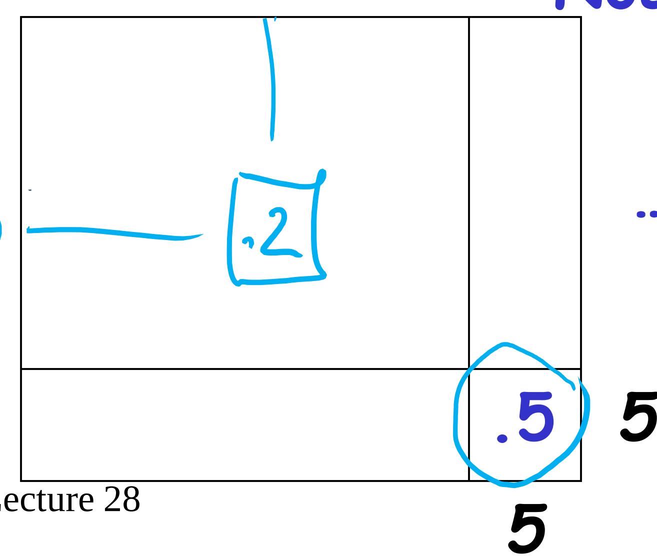
CKY: Base Case

Fill out the table entries by induction: **Base case**

- Consider the input strings of length one (i.e., each individual word w_i)
- Since the grammar is in CNF: $A \stackrel{*}{\Rightarrow} w_i$ iff $A \rightarrow w_i$
- So $\mu[i, i, A] = P(A \rightarrow w_i)$

"Can$_1$ you$_2$ book$_3$ TWA$_4$ flights$_5$?"





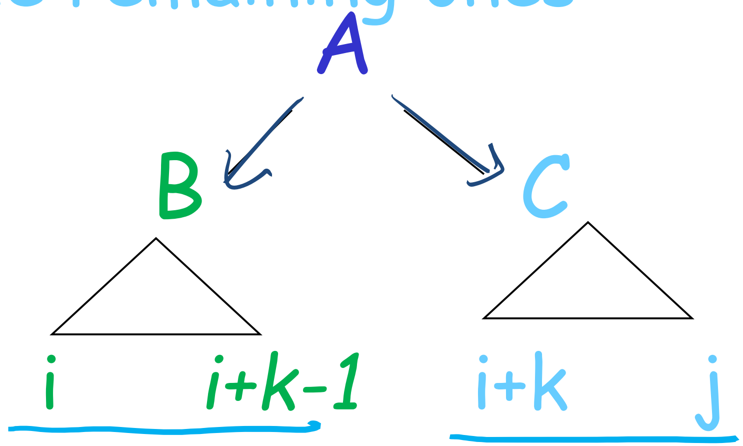
CKY: Recursive Case

Recursive case

- For strings of words of length ≥ 2,

$A \overset{*}{\Rightarrow} w_{ij}$ iff there is at least one rule $A \rightarrow BC$ where B derives the first *k* words (between i and *i+k-1*) and C derives the remaining ones (between i+k and *j*)

- $\mu[i, j, A] = \mu[i, i+k-1, B] * \mu[i+k, j, C] * P(A \rightarrow BC)$

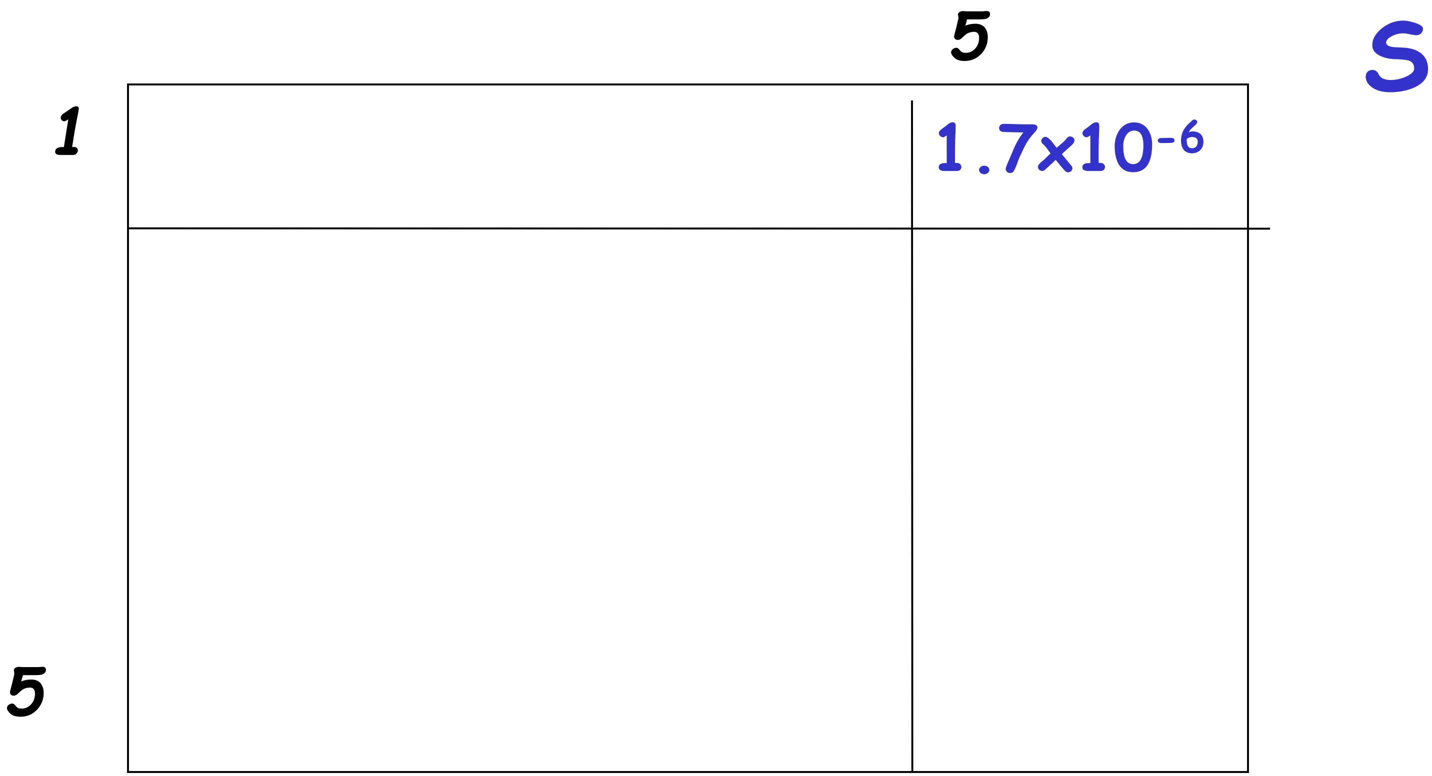


- (for each non-terminal)Choose the max among all possibilities

CKY: Termination

The max prob parse will be $\mu[1, n, S]$

"**Can$_1$ you$_2$ book$_3$ TWA$_4$ flight$_5$?**"



CKY: Termination

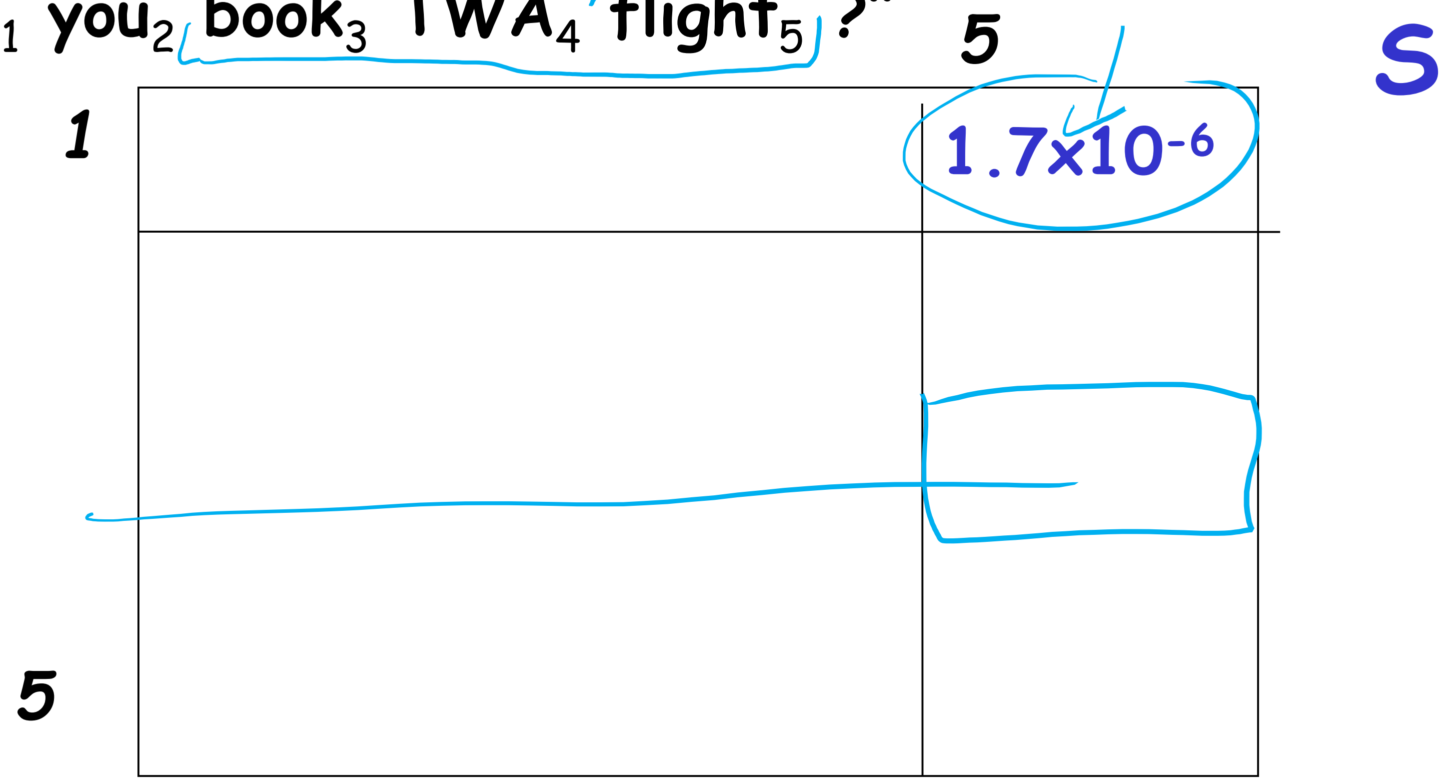
Any other entry in this matrix for S?

A. Yes

B. No

C. Cannot Tell

"Can$_1$ you$_2$ book$_3$ TWA$_4$ flight$_5$?"

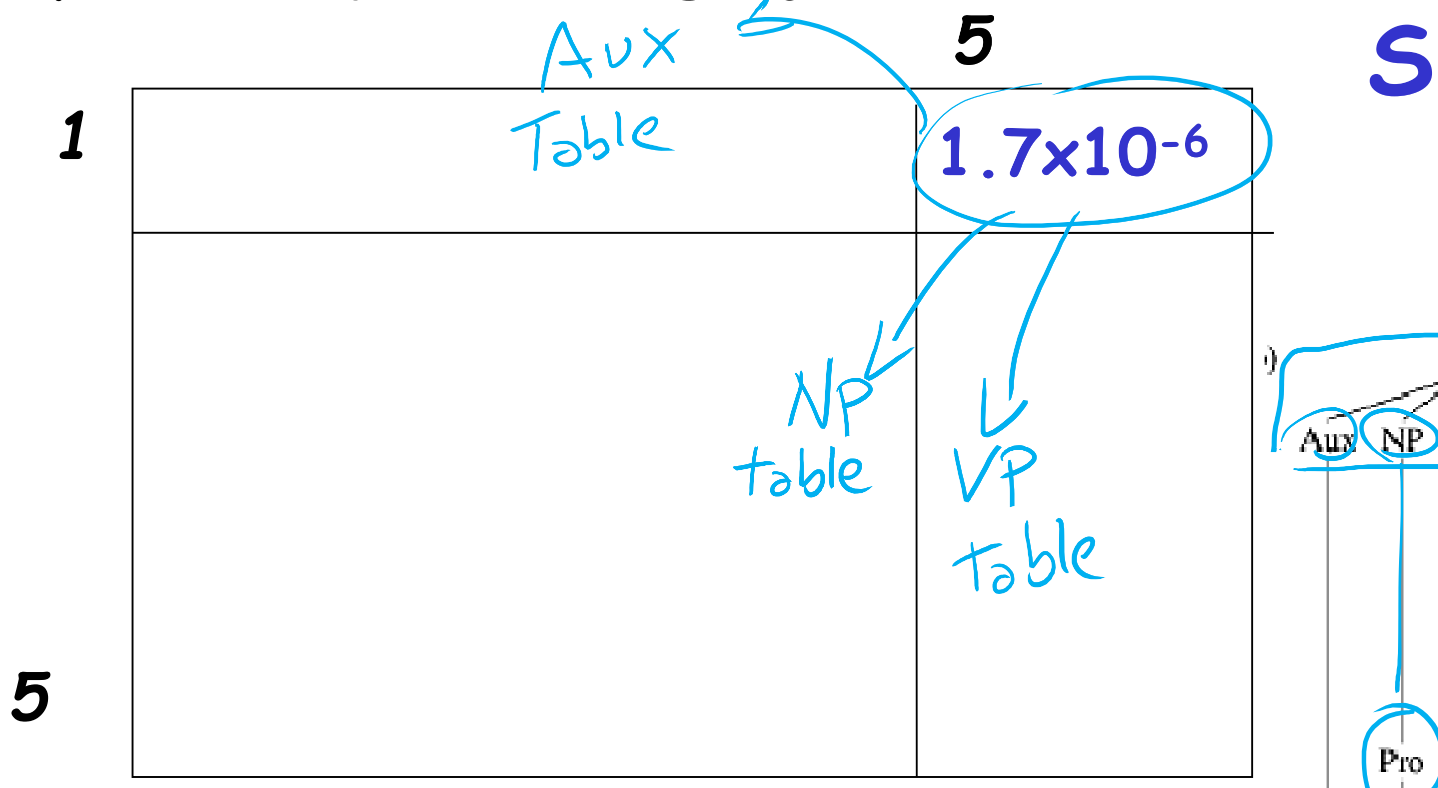


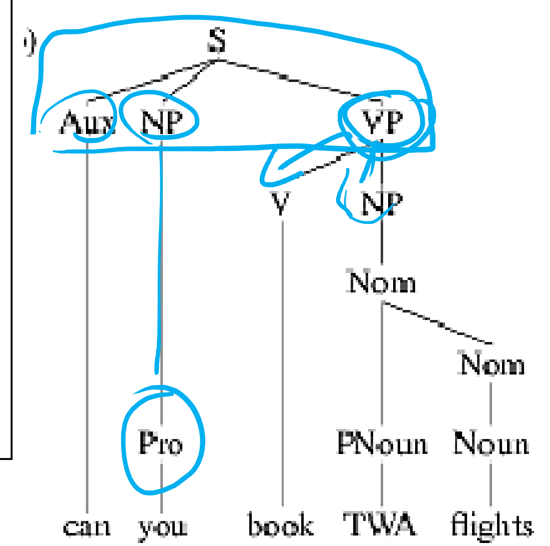
CKY: anything missing?

The parse tree!!

The max prob parse will be $\mu[1, 4, S]$

"Can$_1$ you$_2$ book$_3$ TWA$_4$ flight$_5$?"





Lecture Overview

- Recap Probabilistic Context Free Grammars (PCFG)
- CKY parsing for PCFG (only key steps)
- **PCFG in practice: Modeling Structural and Lexical Dependencies**

Problems with PCFGs

- Most current PCFG models are not vanilla PCFGs
  - Usually augmented in some way
- Vanilla PCFGs assume independence of non-terminal expansions
- But statistical analysis shows this is not a valid assumption
  - Structural and lexical dependencies

Structural Dependencies: Problem

E.g. Syntactic subject of a sentence tends to be a pronoun

- **Subject tends to realize "old information"**
- ***"Mary bought a new book for her trip. She didn't like the first chapter. So she decided to watch a movie."***

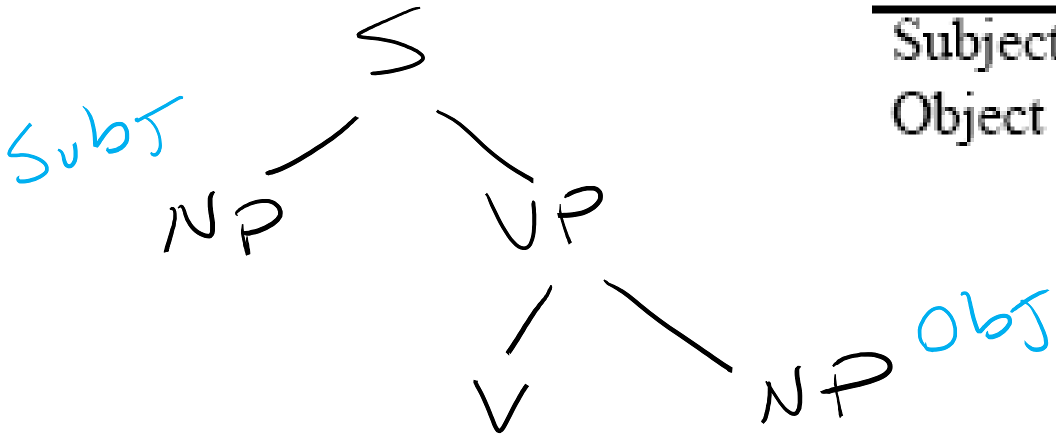
In *Switchboard corpus*:

	Pronoun	Non-Pronoun
Subject	91%	9%
Object	34%	66%

All data

Pronoun	Non-Pronoun
62.5 %	37.5 %

How would you address this problem?

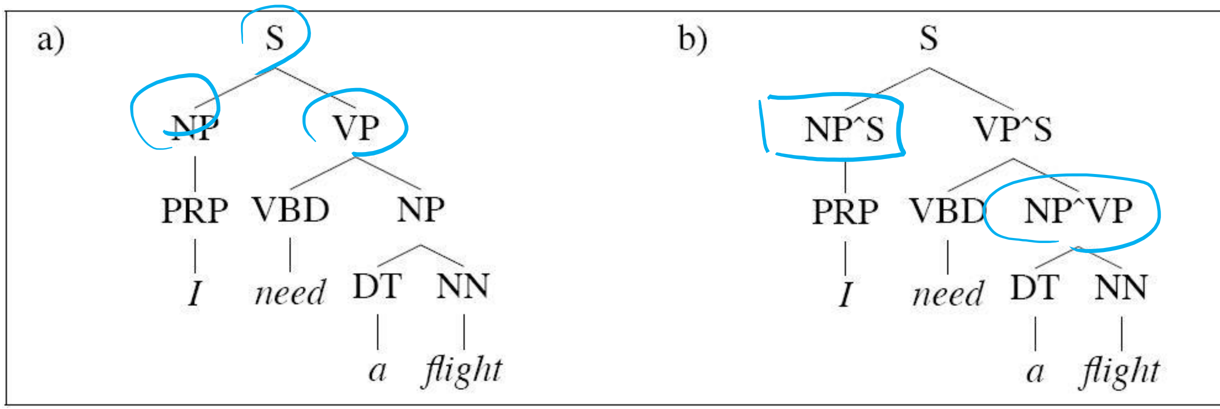


	Pronoun	Non-Pronoun
Subject	91%	9%
Object	34%	66%

Structural Dependencies: Solution

Split non-terminal. E.g., NPsubject and NPobject

Parent Annotation:



Hand-write rules for more complex struct. dependencies

Splitting problems?

- Automatic/Optimal split – Split and Merge algorithm [Petrov et al. 2006- COLING/ACL]

Lexical Dependencies: Problem

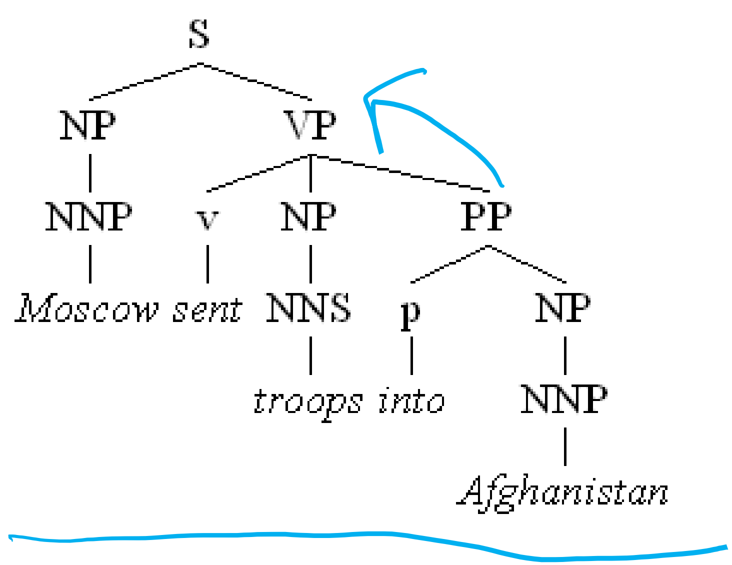
Local tree	Verb *come*	*take*	*think*	*want*
VP → V	9.5%	2.6%	4.6%	5.7%
VP → V NP	1.1%	32.1%	0.2%	13.9%
VP → V PP	34.5%	3.1%	7.1%	0.3%
VP → V SBAR	6.6%	0.3%	73.0%	0.2%
VP → V S	2.2%	1.3%	4.8%	70.8%
VP → V NP S	0.1%	5.7%	0.0%	0.3%
VP → V PRT NP	0.3%	5.8%	0.0%	0.0%
VP → V PRT PP	6.1%	1.5%	0.2%	0.0%

Table 12.2 Frequency of common subcategorization frames (local trees expanding VP) for selected verbs. The data show that the rule used to expand VP is highly dependent on the lexical identity of the verb. The counts ignore distinctions in verbal form tags. Phrase names are as in table 12.1, and tags are Penn Treebank tags (tables 4.5 and 4.6).

Lexical Dependencies: Problem

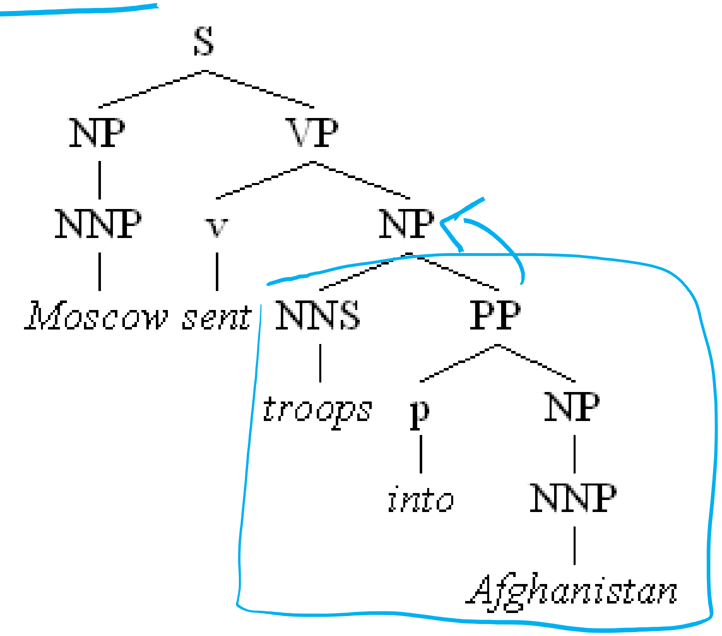
Two parse trees for the sentence
"Moscow sent troops into Afghanistan"

(a)



VP-attachment

(b)



NP-attachment

Typically NP-attachment more frequent than VP-attachment

Lexical Dependencies: Solution

- **Add lexical dependencies to the scheme…**
  - **Infiltrate the influence of particular words into the probabilities of the rules**

All the words?

(a) - P(VP -> V NP PP | VP = "sent troops into Afg.")
(b) - P(VP -> V NP | VP = "sent troops into Afg.")

A. Good Idea

B. Bad Idea

C. Cannot Tell

Lexical Dependencies: Solution

- **Add lexical dependencies to the scheme…**
  - **Infiltrate the influence of particular words into the probabilities of the rules**
  - **All the words?**

(a) - P(VP -> V NP PP | VP = "sent troops into Afg.")

(b) - P(VP -> V NP | VP = "sent troops into Afg.")

Not likely to have significant counts in any treebank!

A. Good Idea

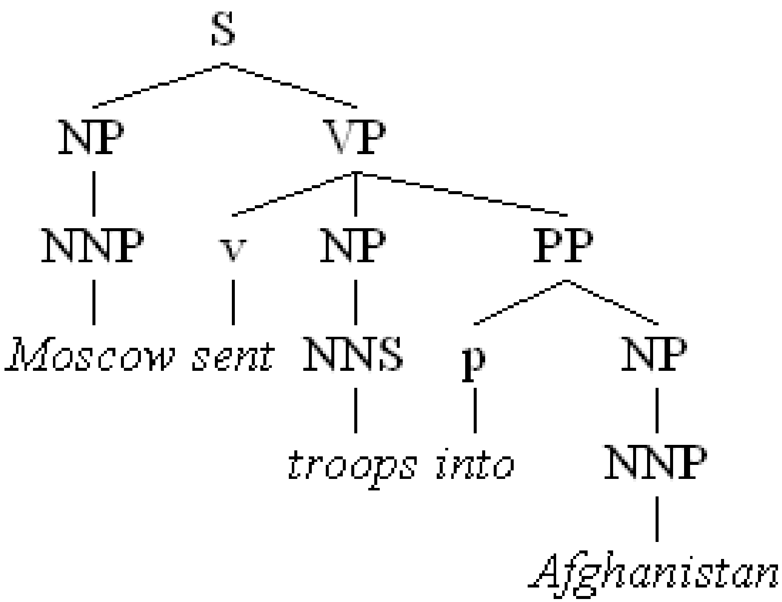
B. Bad Idea

C. Cannot Tell

iclicker.

Use only the Heads

- **To do that we're going to make use of the notion of the head of a phrase**
  - **The head of an NP is its noun**
  - **The head of a VP is its verb**
  - **The head of a PP is its preposition**



More specific rules

- We used to have rule r
  - VP -> V NP PP   P(r|VP)
    - That's the count of this rule divided by the number of VPs in a treebank
- Now we have rule r
  - VP(h(VP))-> V(h(VP)) NP PP   P(r | VP, h(VP))

eg. - VP(sent)-> V(sent) NP PP   P(r | VP, *sent*)

What is the estimate for P(r | VP, *sent*) ?

How many times was this rule used with ***sent***, divided by the number of VPs that ***sent*** appears in total

For each verb reflect probabilities like the ones in slide 21

NLP Practical Goal for FOL (and Prob. Parsing) the ultimate Web question-answering system?

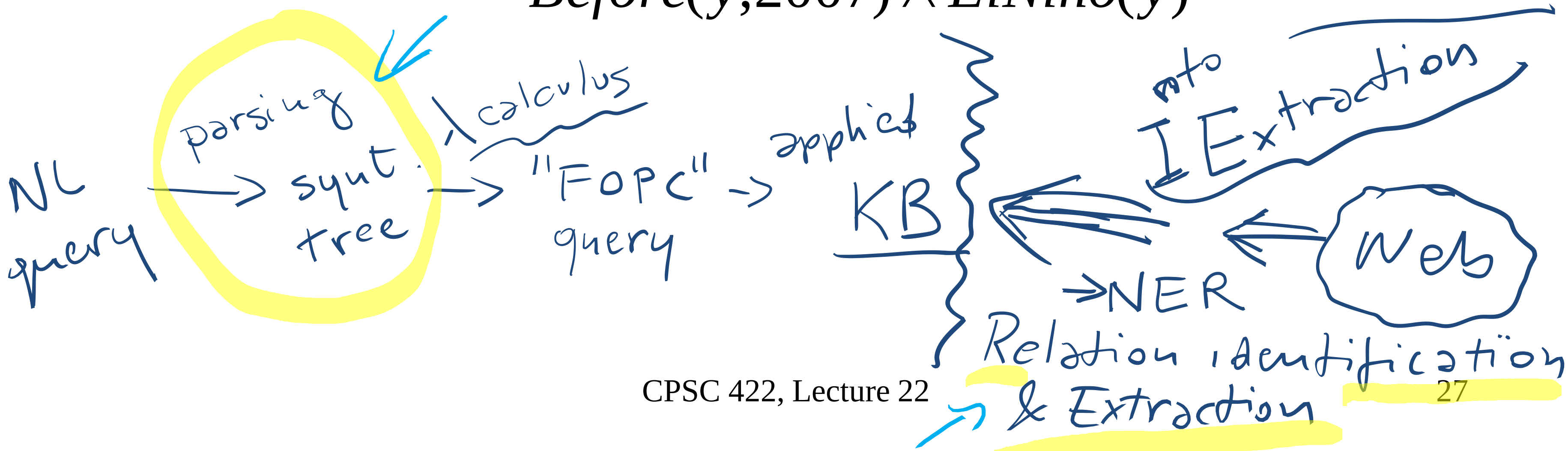
Map NL queries into FOPC so that answers can be effectively computed

- *What African countries are not on the Mediterranean Sea?*

$$\exists c\ Country(c) \wedge \neg Borders(c, Med.Sea) \wedge In(c, Africa)$$

- *Was 2007 the first El Nino year after 2001?*

$$ElNino(2007) \wedge \neg \exists y\ Year(y) \wedge After(y, 2001) \wedge Before(y, 2007) \wedge ElNino(y)$$



Beyond syntax……. Discourse parsing…..

- **CKY Probabilistic parsing Paper on Fri.**

Beyond NLP……. Planning…..

- Li, N., Cushing, W., Kambhampati, S., & Yoon, S. (2012). Learning probabilistic hierarchical task networks as probabilistic context-free grammars to capture user preferences. ACM Transactions on Intelligent Systems and Technology. (CMU+Arizona State)

Discovering Discourse Structure: Computational Tasks

The bank was hamstrung in its efforts to face the challenges of a changing market by its links to the government, analysts say.

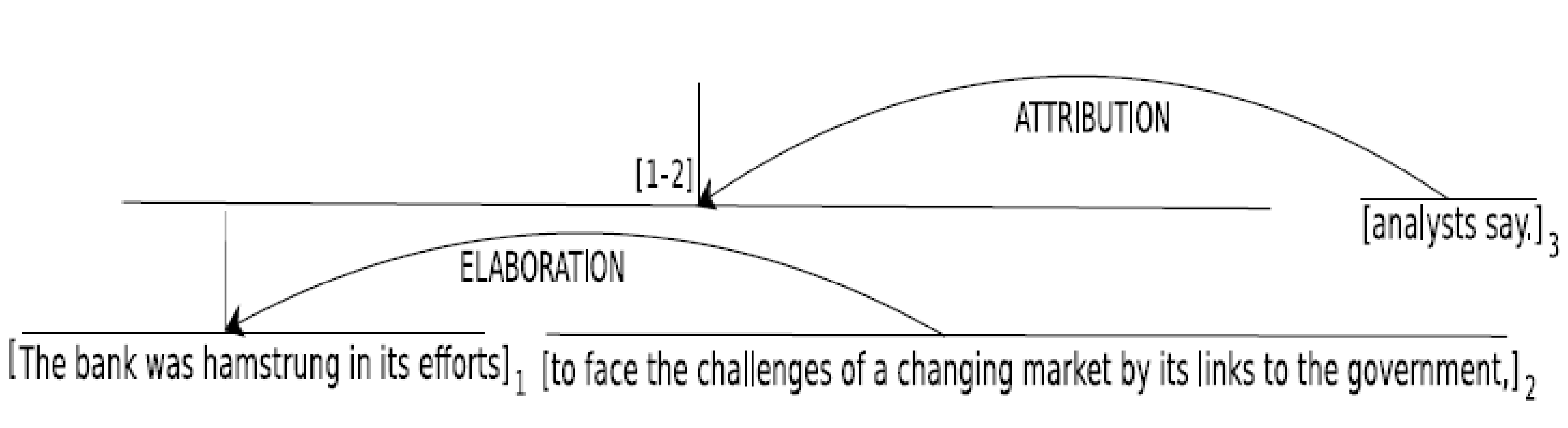
Discourse Segmentation

The bank was hamstrung in its efforts 1

to face the challenges of a changing market by its links to the government, 2

analysts say. 3

Discourse Parsing



422 big picture: Where are we?

StarAI (statistical relational AI)

Hybrid: Det +Sto

Prob CFG

Prob Relational Models

Markov Logics

	Deterministic	Stochastic
Query	*Logics* *First Order Logics* *Ontologies* • Full Resolution • SAT	*Belief Nets* Approx. : Gibbs *Markov Chains and HMMs* Forward, Viterbi…. Approx. : Particle Filtering *Undirected Graphical Models* *Markov Networks* *Conditional Random Fields*
Planning		*Markov Decision Processes and Partially Observable MDP* • Value Iteration • Approx. Inference *Reinforcement Learning*

Applications of AI

Representation

Reasoning Technique

Learning Goals for today's class

You can:

- Describe the key steps of CKY probabilistic parsing
- Motivate introduction of structural and lexical dependencies
- Describe how to deal with these dependencies within the PCFG framework

Next class on Fri : paper discussion

- **Portions of our Journal of Computational Linguistics paper** only sections 1, 3 and 4 are mandatory
- **CODRA: A Novel Discriminative Framework for Rhetorical Analysis**

Assignment-3 due on Mon
Assignment-4 will be out on the same day