Intelligent Systems (AI-2)

Computer Science cpsc422, Lecture 27

Nov, 13, 2019

Call For Participants

Explainable AI in Intelligent Tutoring Systems

3h | $30 | ICICS/CS Building

Email obarral@cs.ubc.ca
If you have any questions

Doodle Sign-up

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

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

PCFGs are used to....

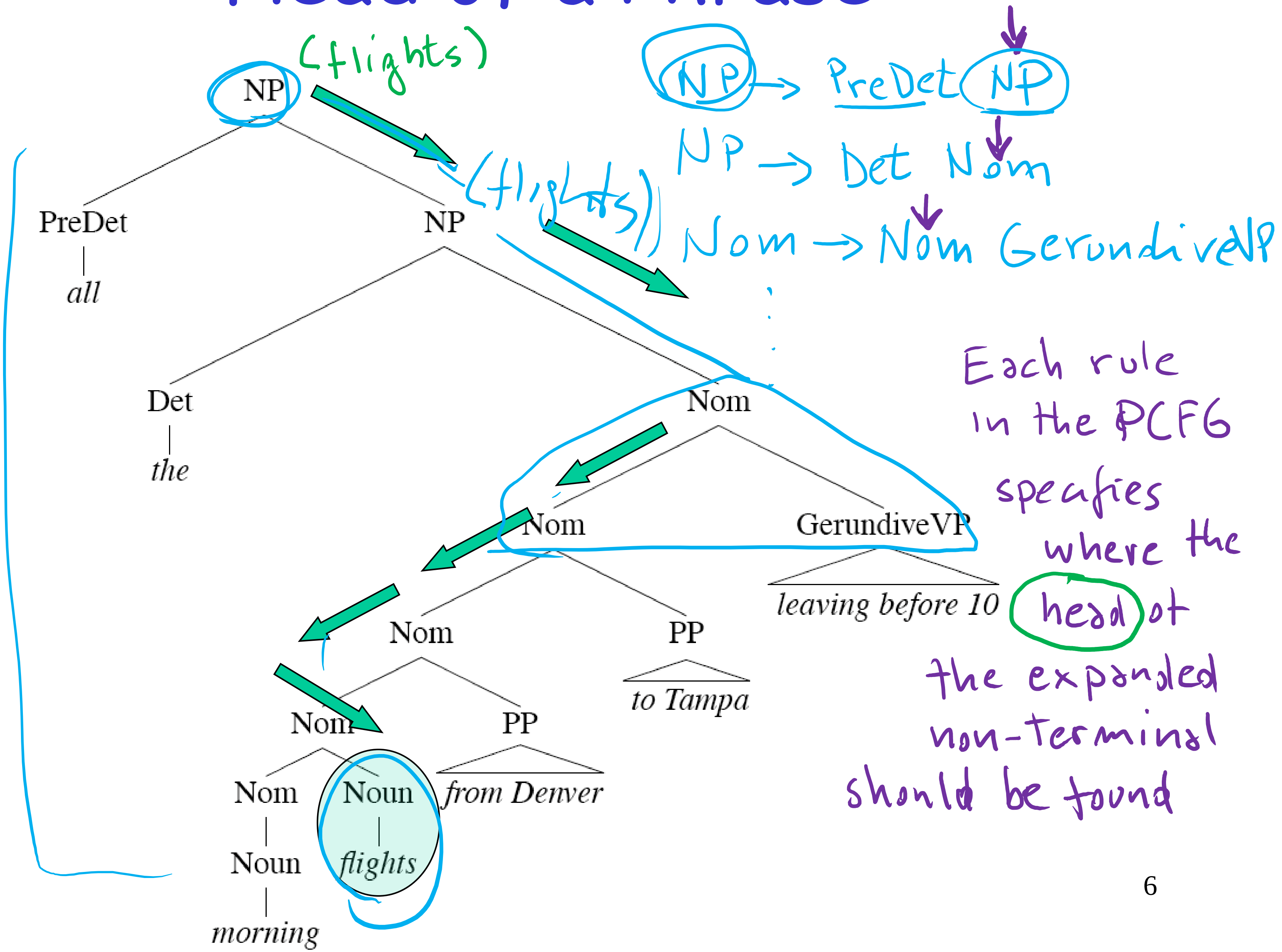
- **Estimate Prob. of parse tree**

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

- **Estimate Prob. to sentences**

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

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

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

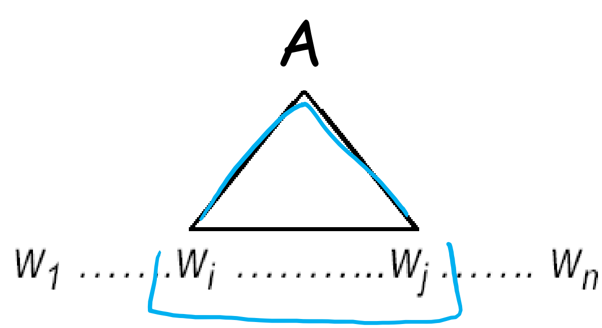
Probabilistic CKY Algorithm

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)

$A \to BC$ $\quad A \to w$

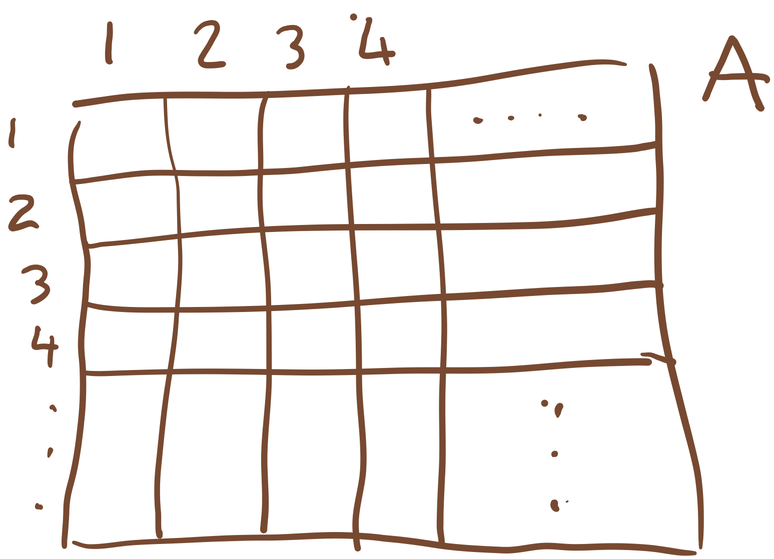
Definitions
- $w_1 \ldots w_n$ an input string composed of n words
- w_{ij} a string of words from word i to word j
- $\mu[i, j, A]$: a table entry holds the maximum probability for a constituent with non-terminal A spanning words $w_i \ldots w_j$

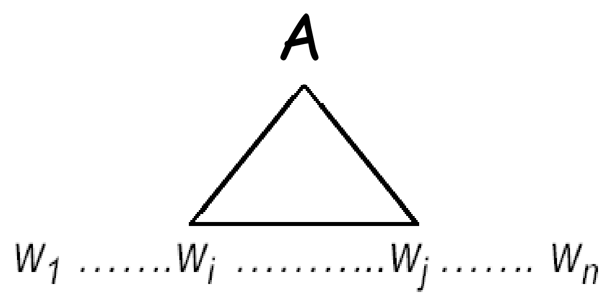


Probabilistic CKY Algorithm

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 $w_i \ldots w_j$

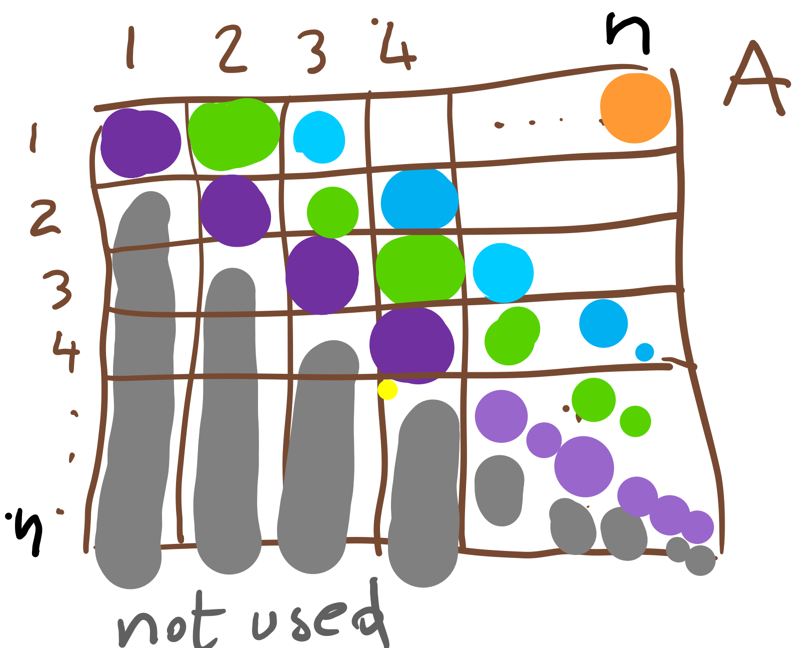


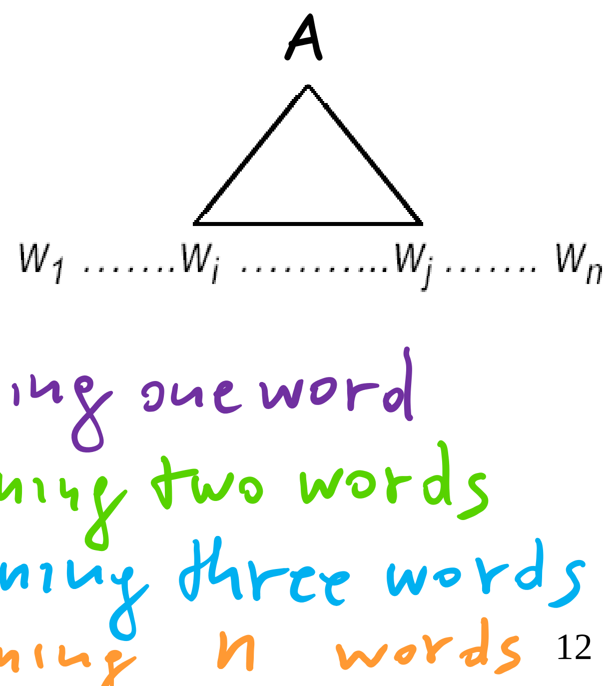


Probabilistic CKY Algorithm

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 $w_i \ldots w_j$



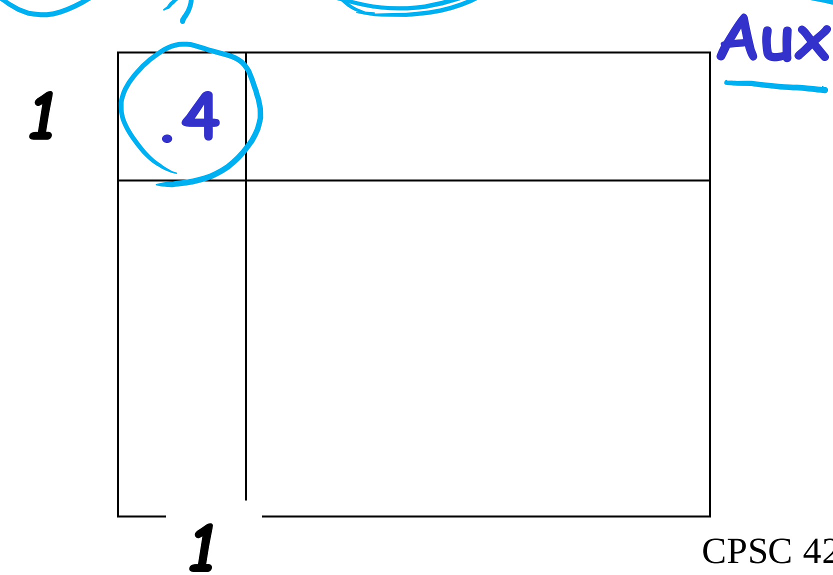


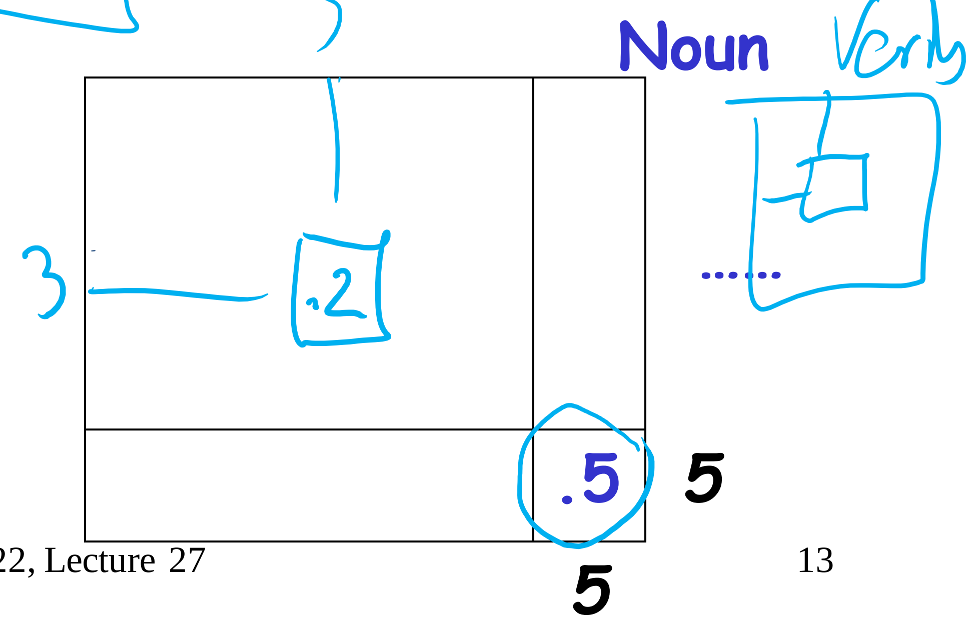
CKY: Base Case

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

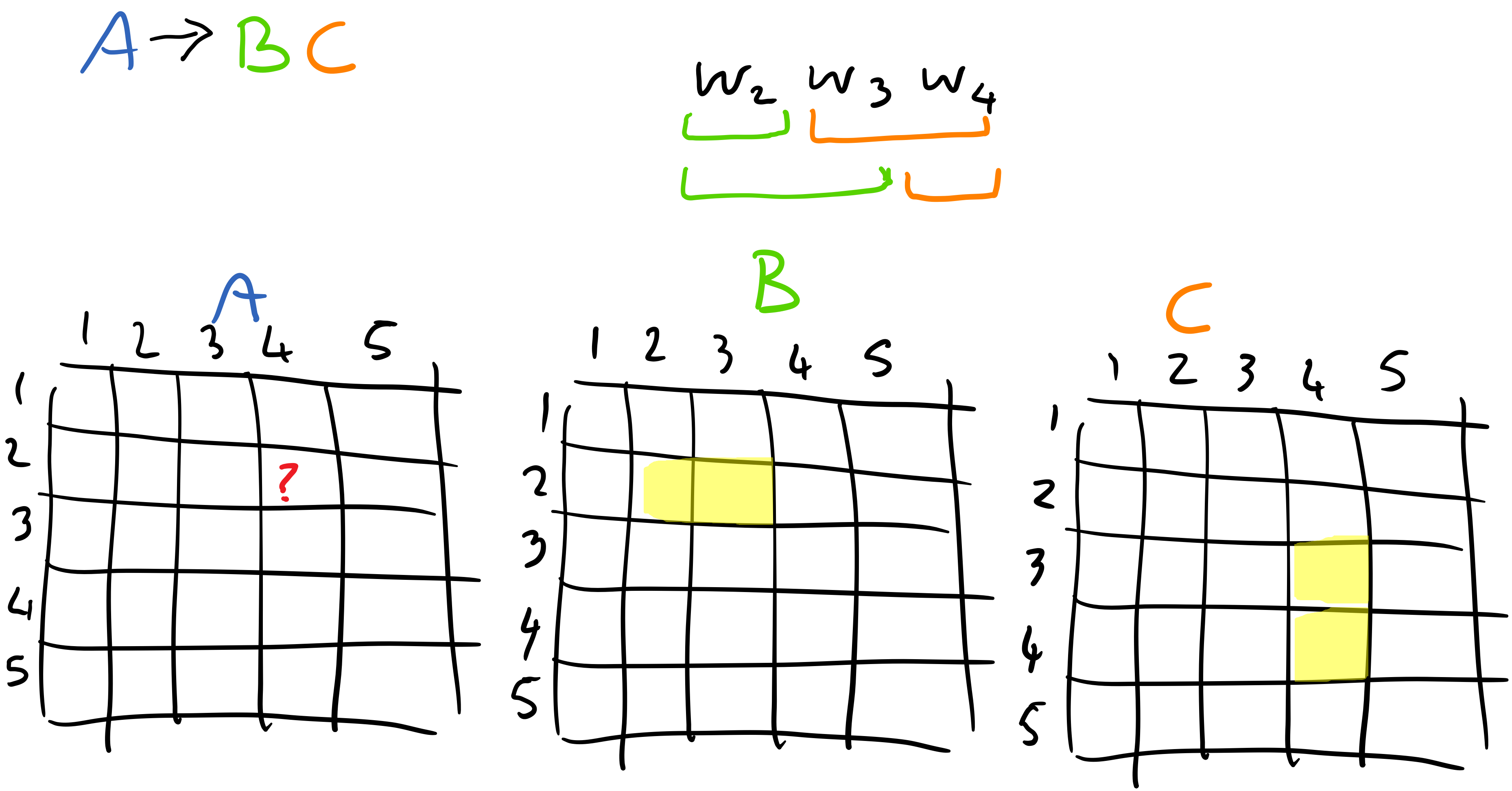
- Consider the input strings of length one (i.e., each individual word w_i)
- Since the grammar is in CNF: $A \overset{*}{\Rightarrow} w_i$ iff $A \rightarrow w_i$
- So $\mu[i, i, A] = P(A \rightarrow w_i)$

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





Probabilistic CKY Algorithm: recursive case (visual)

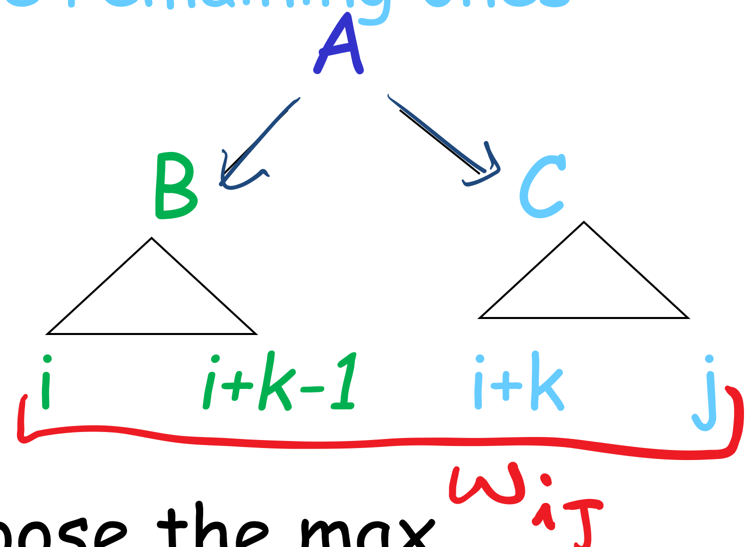


CKY: Recursive Case

Recursive case

- For strings of words of length = 2, 3 ... n

  $A \overset{*}{\Rightarrow} w_{ij}$ 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)



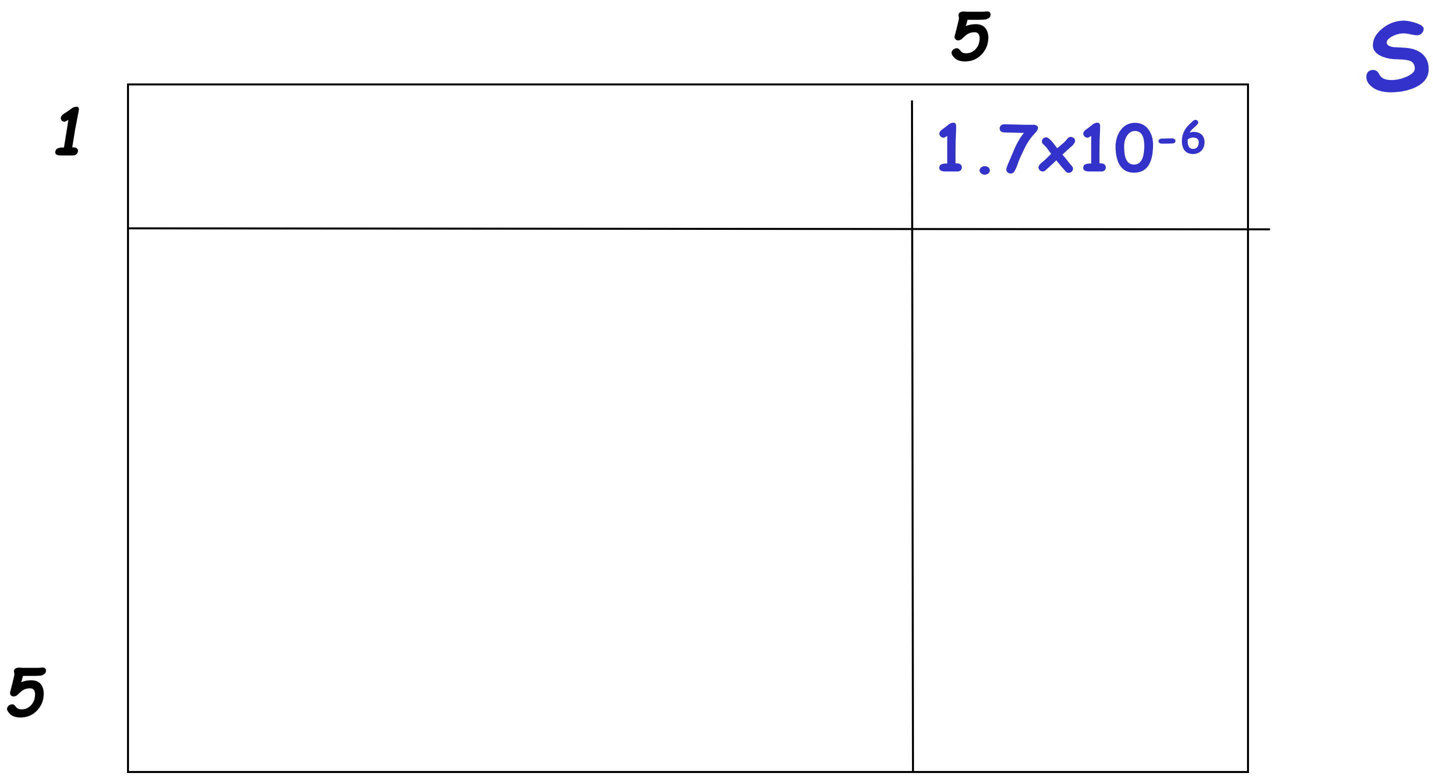
- $\mu[i, j, A] = \mu[i, i+k-1, B] *$
  $\mu[i+k, j, C] *$
  $P(A \rightarrow BC)$

- (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 $flight_5$?"

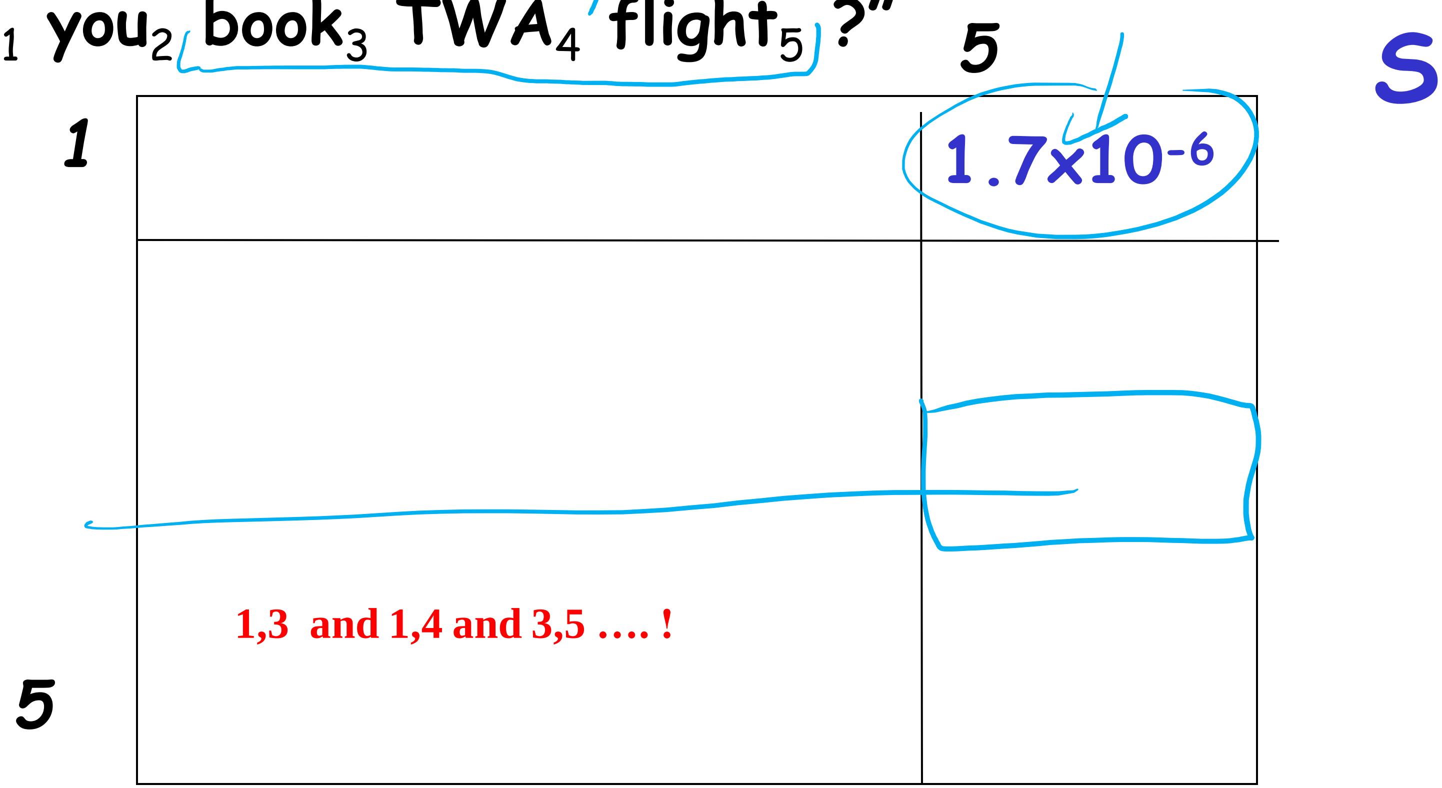


CKY: Termination

Any other entry in this matrix for S?

A. Yes
B. No
C. Cannot Tell

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

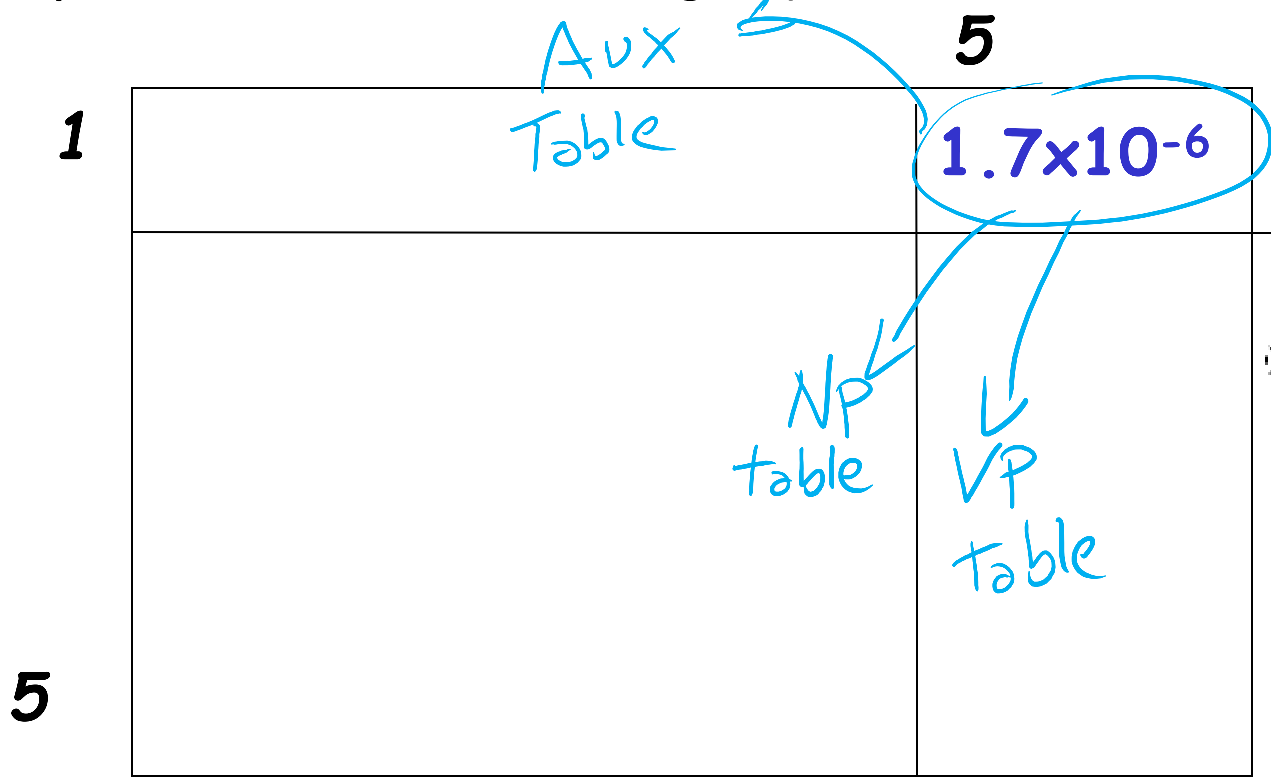


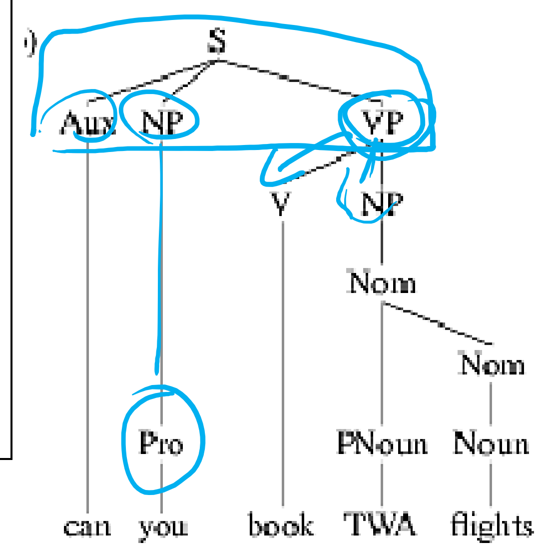
CKY: anything missing?

The parse tree!!

The max prob parse will be μ[1, 4, S]

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





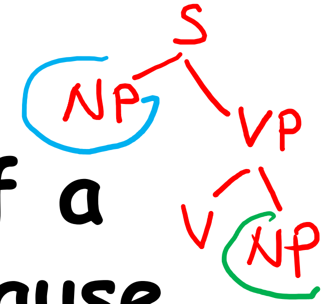
Lecture Overview

- 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 (vs. object) of a sentence tends to be a <u>pronoun</u> because

- Subject tends to realize the topic of a sentence
- Topic is usually old information (expressed in previous sentences)
- Pronouns are usually used to refer to old information

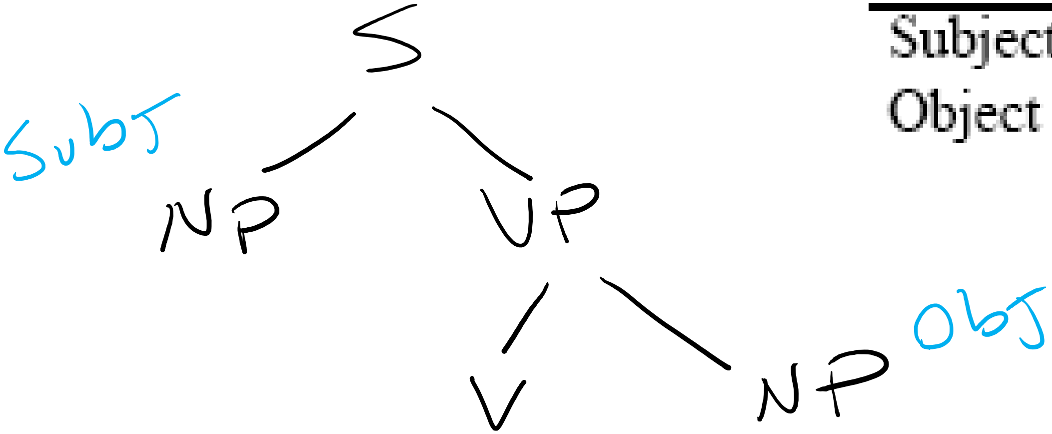
Mary bought a new book for her trip. She didn't like the first chapter. So she decided to watch a movie.

In *Switchboard corpus*:

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

All data
Pronoun 62.5%
Non-Pronoun 37.5%

How would you address this problem?

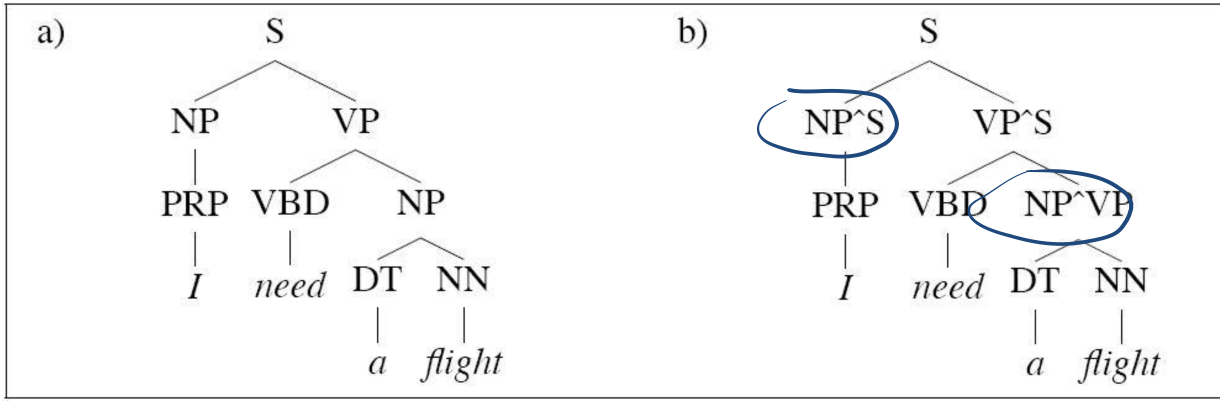


	Pronoun	Non-Pronoun
Subject	91%	9%
Object	34%	66%

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

	Verb			
Local tree	*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%

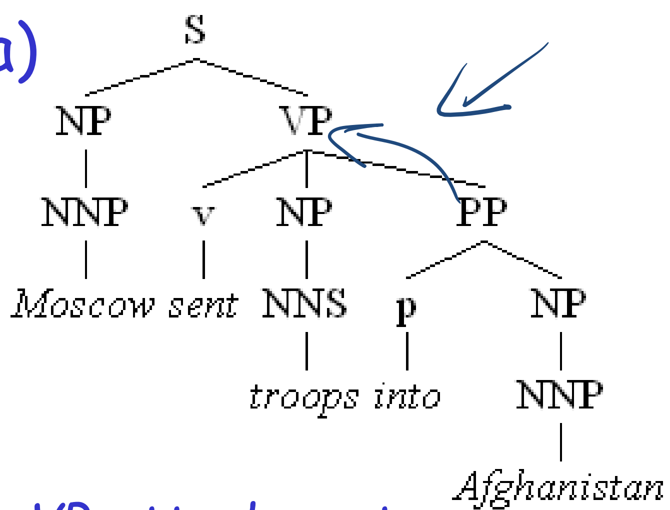
Table 12.2 Frequency of common subcategorization frames (local trees expanding VP) for selected verbs. The data show that the rule used to expand VP is highly dependent on the lexical identity of the verb. The counts ignore distinctions in verbal form tags. Phrase names are as in table 12.1, and tags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

SBAR = subordinate clause

Lexical Dependencies: Problem

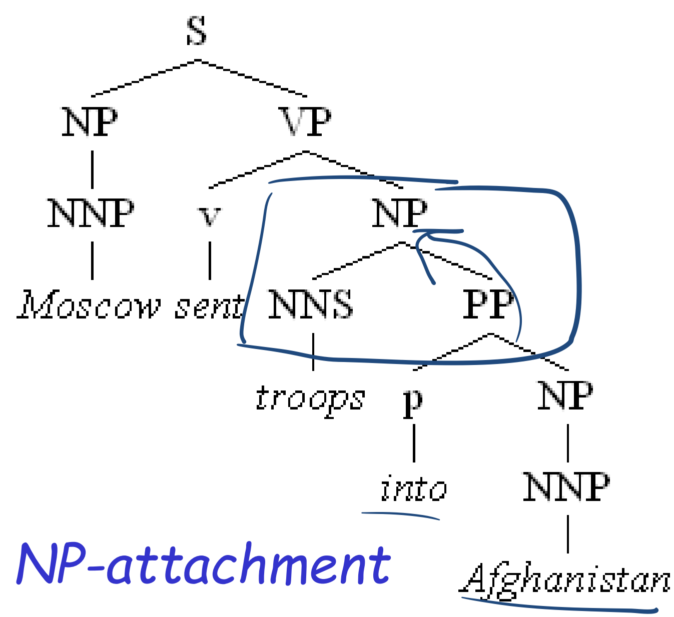
Two parse trees for the sentence
"Moscow sent troops into Afghanistan"

(a)



VP-attachment

(b)



NP-attachment

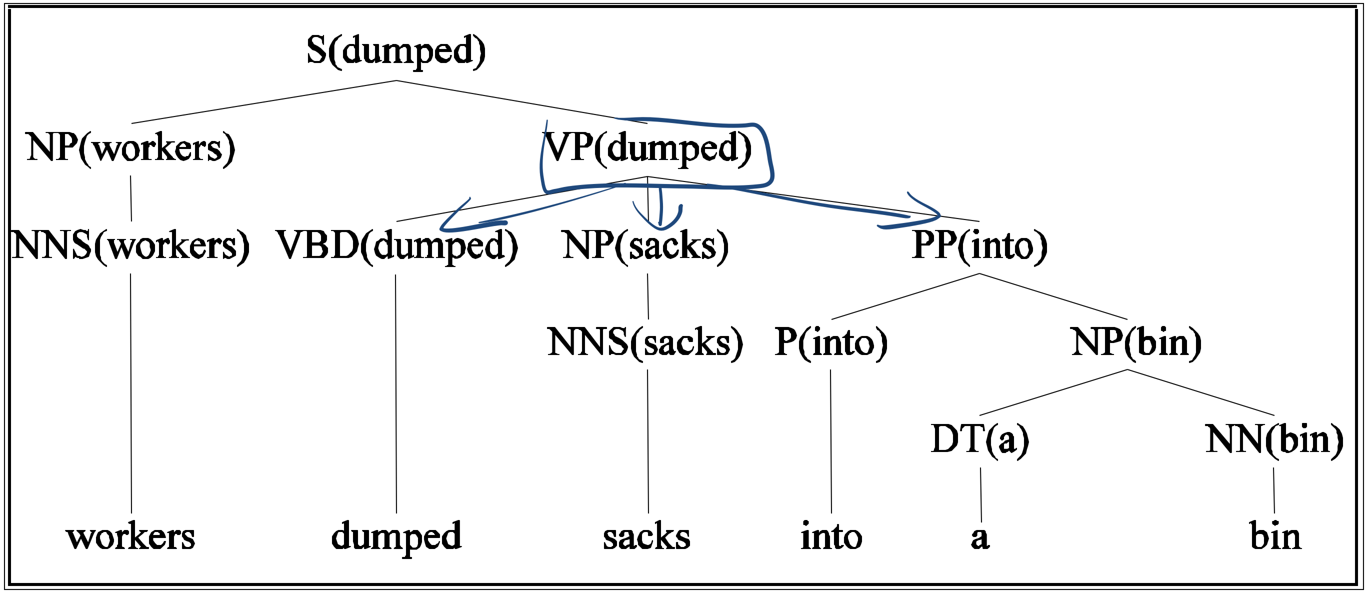
Typically NP-attachment more frequent than VP-attachment

VP → V NP
VP → V NP PP

VP(send) → V NP
VP(send) → V NP PP ← higher prob!

Attribute grammar for Lexicalized PCFG : each non-terminal is annotated with its lexical head… many more rules!

(Collins 1999)



- **We used to have rules like**
  VP -> V NP PP

- **Now we have much more specific rules like**
  VP(dumped)-> V(dumped) NP(sacks) PP(into)

PCFG Parsing State of the art(~2010)

Parser	F1 ≤ 40 words	F1 all words
Klein & Manning unlexicalized 2003	86.3	85.7
Matsuzaki et al. simple EM latent states 2005	86.7	86.1
Charniak generative, lexicalized ("maxent inspired") 2000	90.1	89.5
Petrov and Klein NAACL 2007	90.6	90.1
Charniak & Johnson discriminative reranker 2005	92.0	91.4
Fossum & Knight 2009 combining constituent parsers		92.4

sentence length

hand crafted "states"

no limit on sentence length

From C. Manning (Stanford NLP)

Parser	Training Set	WSJ 22	WSJ 23
baseline LSTM+D	WSJ only	< 70	< 70
LSTM+A+D	WSJ only	88.7	88.3
LSTM+A+D ensemble	WSJ only	90.7	90.5
baseline LSTM	BerkeleyParser corpus	91.0	90.5
LSTM+A	high-confidence corpus	93.3	92.5
LSTM+A ensemble	high-confidence corpus	**93.5**	**92.8**
Petrov et al. (2006) [12]	WSJ only	91.1	90.4
Zhu et al. (2013) [13]	WSJ only	N/A	90.4
Petrov et al. (2010) ensemble [14]	WSJ only	92.5	91.8
Zhu et al. (2013) [13]	semi-supervised	N/A	91.3
Huang & Harper (2009) [15]	semi-supervised	N/A	91.3
McClosky et al. (2006) [16]	semi-supervised	92.4	92.1
Huang & Harper (2010) ensemble [17]	semi-supervised	92.8	92.4

Neural

Table 1: F1 scores of various parsers on the development and test set. See text for discussion.

Grammar as a Foreign Language

Computation and Language [cs.CL] Published 24 Dec 2014
Updated 9 Jun 2015

O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, G. Hinton *Google*

Fast and Accurate Shift-Reduce Constituent Parsing by Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang and Jingbo Zhu (ACL - 2013)

Very recent paper (NAACL 2018)

What's Going On in Neural Constituency Parsers? An Analysis,

D.Gaddy, M. Stern, D. Klein, Computer Science ., Univ. of California, Berkeley

- Abstractly, our model consists of a single scoring function ***s(i, j, l)*** that assigns a real-valued score to every label ***l*** for each ***span(i, j)*** in an input sentence.
- We take the set of available labels to be the collection of **all non-terminals** … in the training data,
- To build up to spans, we first run a **bidirectional LSTM** over the sequence of word representations for an input sentence
- we implement the label scoring function by feeding the span representation through a one **layer feedforward network** whose output dimensionality equals the number of possible labels
- …. we can still employ a **CKY-style algorithm** for efficient globally optimal inference …..
- "We find that our model implicitly learns to encode much of the same information that was explicitly provided by grammars and lexicons in the past, indicating that this scaffolding can largely be subsumed by powerful general-purpose neural machinery
- Also this one does **(92.08 F1 on PTB)**

CKY/PCFG Beyond syntax……. Discourse Parsing….. And Dialog

- **CKY Probabilistic parsing Paper in Reading**

- **Conversation Trees: A Grammar Model for Topic Structure in Forums, Annie Louis and Shay B. Cohen, EMNLP 2015. [corpus]**

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.

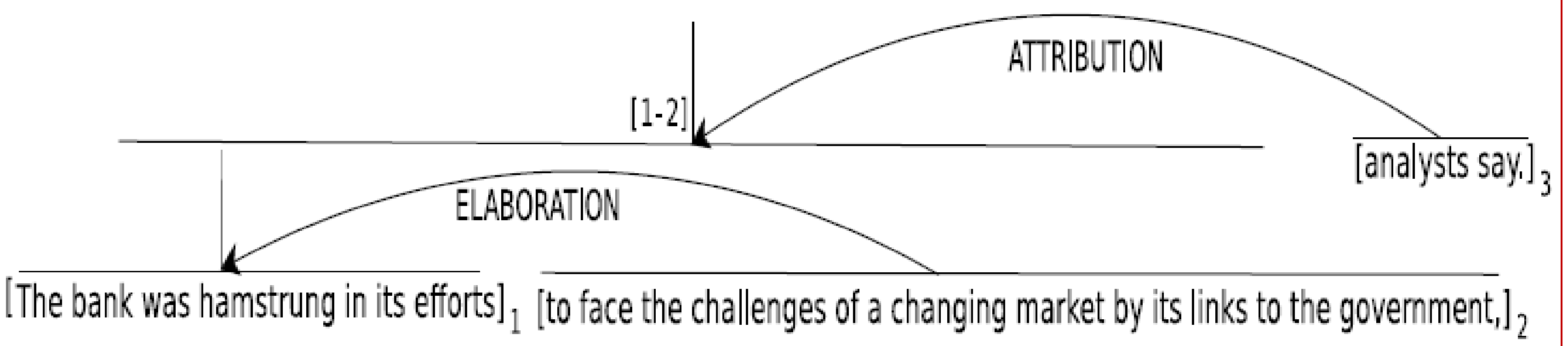
Discourse Segmentation

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

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

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

e.g.

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?

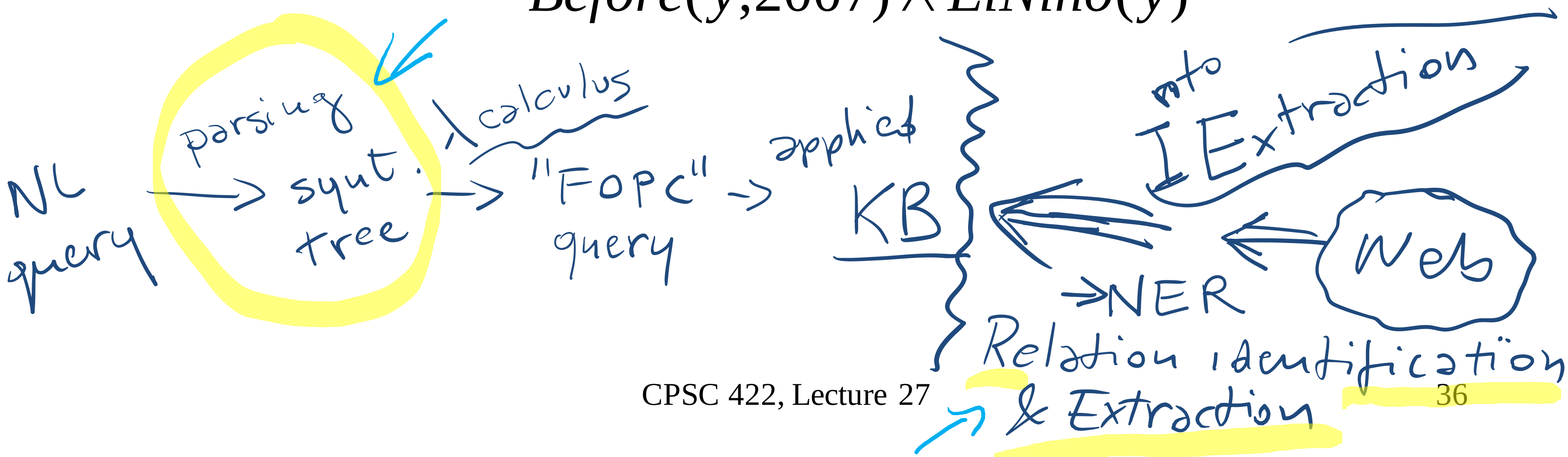
Map NL queries into FOPC so that answers can be effectively computed

- *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?*

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



Parser	Training Set	WSJ 22	WSJ 23
baseline LSTM+D	WSJ only	< 70	< 70
LSTM+A+D	WSJ only	88.7	88.3
LSTM+A+D ensemble	WSJ only	90.7	90.5
baseline LSTM	BerkeleyParser corpus	91.0	90.5
LSTM+A	high-confidence corpus	93.3	92.5
LSTM+A ensemble	high-confidence corpus	**93.5**	**92.8**
Petrov et al. (2006) [12]	WSJ only	91.1	90.4
Zhu et al. (2013) [13]	WSJ only	N/A	90.4
Petrov et al. (2010) ensemble [14]	WSJ only	92.5	91.8
Zhu et al. (2013) [13]	semi-supervised	N/A	91.3
Huang & Harper (2009) [15]	semi-supervised	N/A	91.3
McClosky et al. (2006) [16]	semi-supervised	92.4	92.1
Huang & Harper (2010) ensemble [17]	semi-supervised	92.8	92.4

Table 1: F1 scores of various parsers on the development and test set. See text for discussion.

Grammar as a Foreign Language

Computation and Language [cs.CL] Published 24 Dec 2014 Updated 9 Jun 2015

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, Geoffrey Hinton *Google*

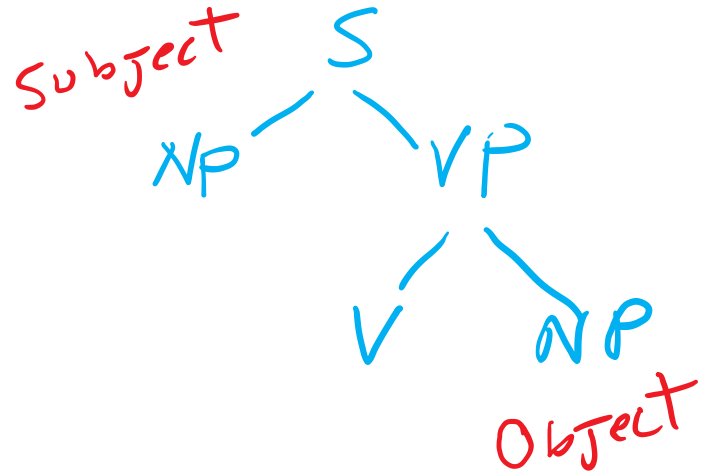
Fast and Accurate Shift-Reduce Constituent Parsing by Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang and Jingbo Zhu (ACL - 2013)

- *Announcing SyntaxNet:* **The World's Most Accurate Parser Goes Open Source, 2016; Posted by Slav Petrov, Senior Staff Research Scientist (different parsing framework)**

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."***



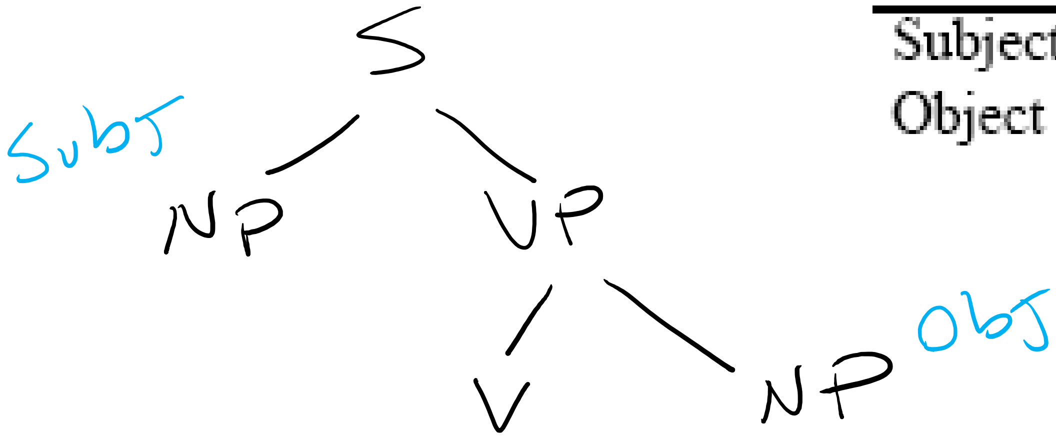
In *Switchboard corpus*:

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

All data

Pronoun	Non-Pronoun
62.5%	37.5%

How would you address this problem?

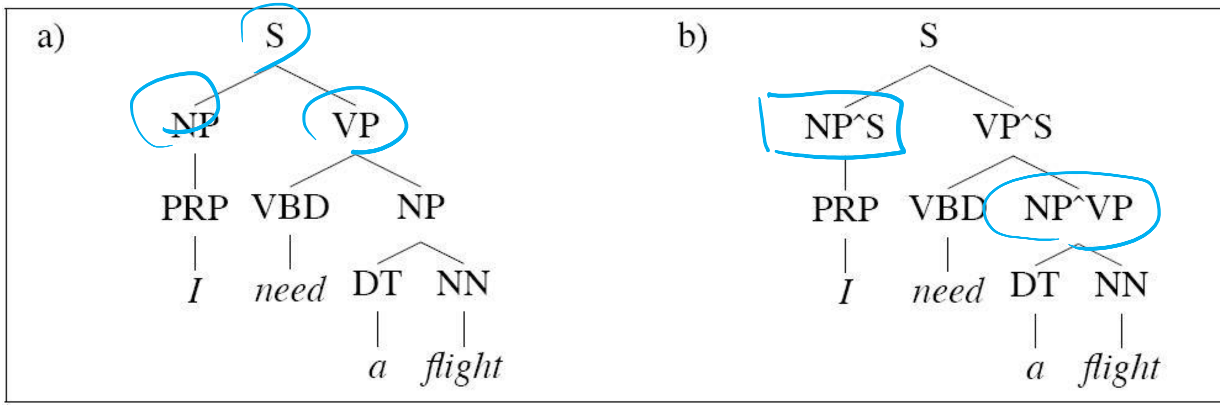


	Pronoun	Non-Pronoun
Subject	91%	9%
Object	34%	66%

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

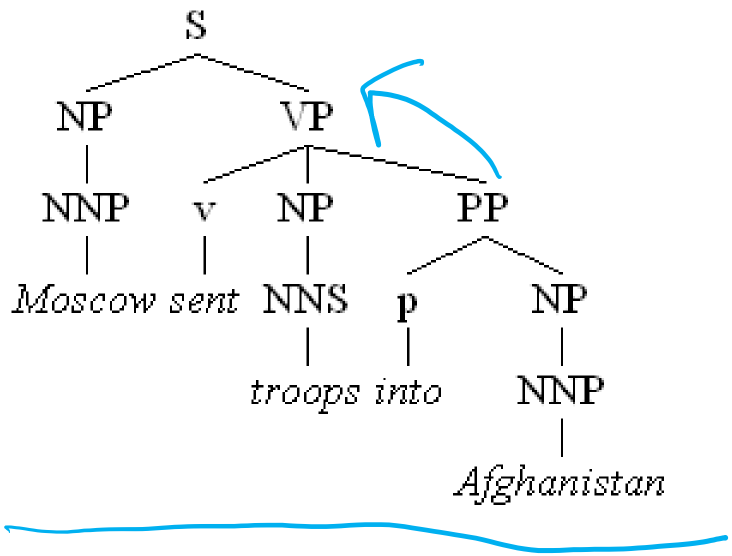
Local tree	Verb *come*	*take*	*think*	*want*
VP → V	9.5%	2.6%	4.6%	5.7%
VP → V NP	1.1%	32.1%	0.2%	13.9%
VP → V PP	34.5%	3.1%	7.1%	0.3%
VP → V SBAR	6.6%	0.3%	73.0%	0.2%
VP → V S	2.2%	1.3%	4.8%	70.8%
VP → V NP S	0.1%	5.7%	0.0%	0.3%
VP → V PRT NP	0.3%	5.8%	0.0%	0.0%
VP → V PRT PP	6.1%	1.5%	0.2%	0.0%

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

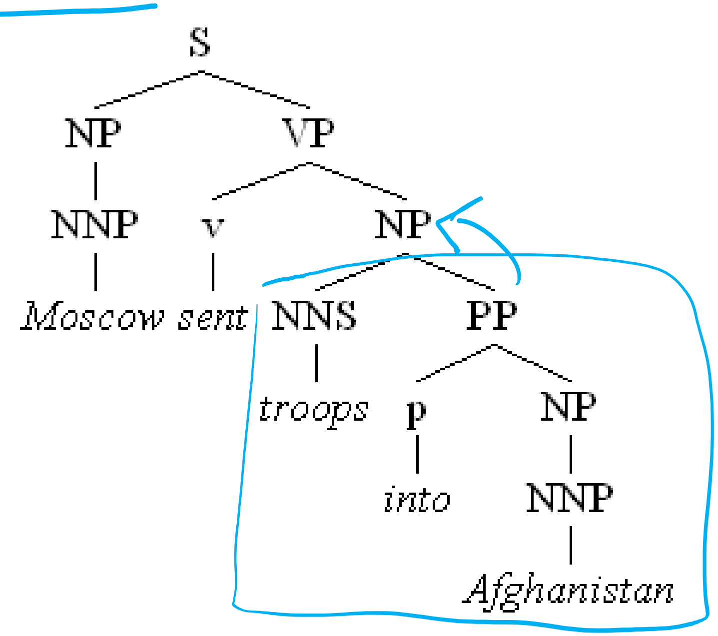
Two parse trees for the sentence
"Moscow sent troops into Afghanistan"

(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