# Syntactic Parsing

Dr. Demetrios Glinos
University of Central Florida

CAP6640 – Computer Understanding of Natural Language

# Today

- Structural Ambiguity
- Conversion to CNF
- CKY Recognition
- CKY Parsing
- Partial Parsing

# Syntactic Parsing

- Syntactic parsing
  - recognizing a sentence, and
  - assigning a syntactic structure

- Parse trees
  - grammar checking
    - if a sentence cannot be parsed, it probably has grammatical errors
  - first step in semantic analysis
    - information extraction
    - question answering
    - summarization

# Ambiguity in NLP

- So far, we have examined
  - part-of-speech ambiguity

    *"Time flies."*

    → Is "time" a noun or a verb?
    → Is "flies" a noun or a verb?

  - word sense ambiguity

    *"Peter admired the fly."*

    → Is "fly" the 2-winged insect or a fisherman's lure?

# Structural Ambiguity

- Occurs when a CFG-based grammar can assign more than one parse to a sentence
- Sentence meaning is affected

- Types
  - attachment ambiguity

    *"Peter saw the man on the hill with the telescope"*

    → Who has the telescope?

  - coordination ambiguity

    *"young men and women"* → Are the women young, too?

  - others (we will not cover)

# L1 Miniature Grammar for English

| Grammar | Lexicon |
|---|---|
| $S \rightarrow NP\ VP$ | $Det \rightarrow that \mid this \mid the \mid a$ |
| $S \rightarrow Aux\ NP\ VP$ | $Noun \rightarrow book \mid flight \mid meal \mid money$ |
| $S \rightarrow VP$ | $Verb \rightarrow book \mid include \mid prefer$ |
| $NP \rightarrow Pronoun$ | $Pronoun \rightarrow I \mid she \mid me$ |
| $NP \rightarrow Proper\text{-}Noun$ | $Proper\text{-}Noun \rightarrow Houston \mid NWA$ |
| $NP \rightarrow Det\ Nominal$ | $Aux \rightarrow does$ |
| $Nominal \rightarrow Noun$ | $Preposition \rightarrow from \mid to \mid on \mid near \mid through$ |
| $Nominal \rightarrow Nominal\ Noun$ | |
| $Nominal \rightarrow Nominal\ PP$ | |
| $VP \rightarrow Verb$ | |
| $VP \rightarrow Verb\ NP$ | |
| $VP \rightarrow Verb\ NP\ PP$ | |
| $VP \rightarrow Verb\ PP$ | |
| $VP \rightarrow VP\ PP$ | |
| $PP \rightarrow Preposition\ NP$ | |

**Figure 12.1** The $\mathscr{L}_1$ miniature English grammar and lexicon.

*source: J&M (3d Ed. draft)*

# Example: Attachment Ambiguity

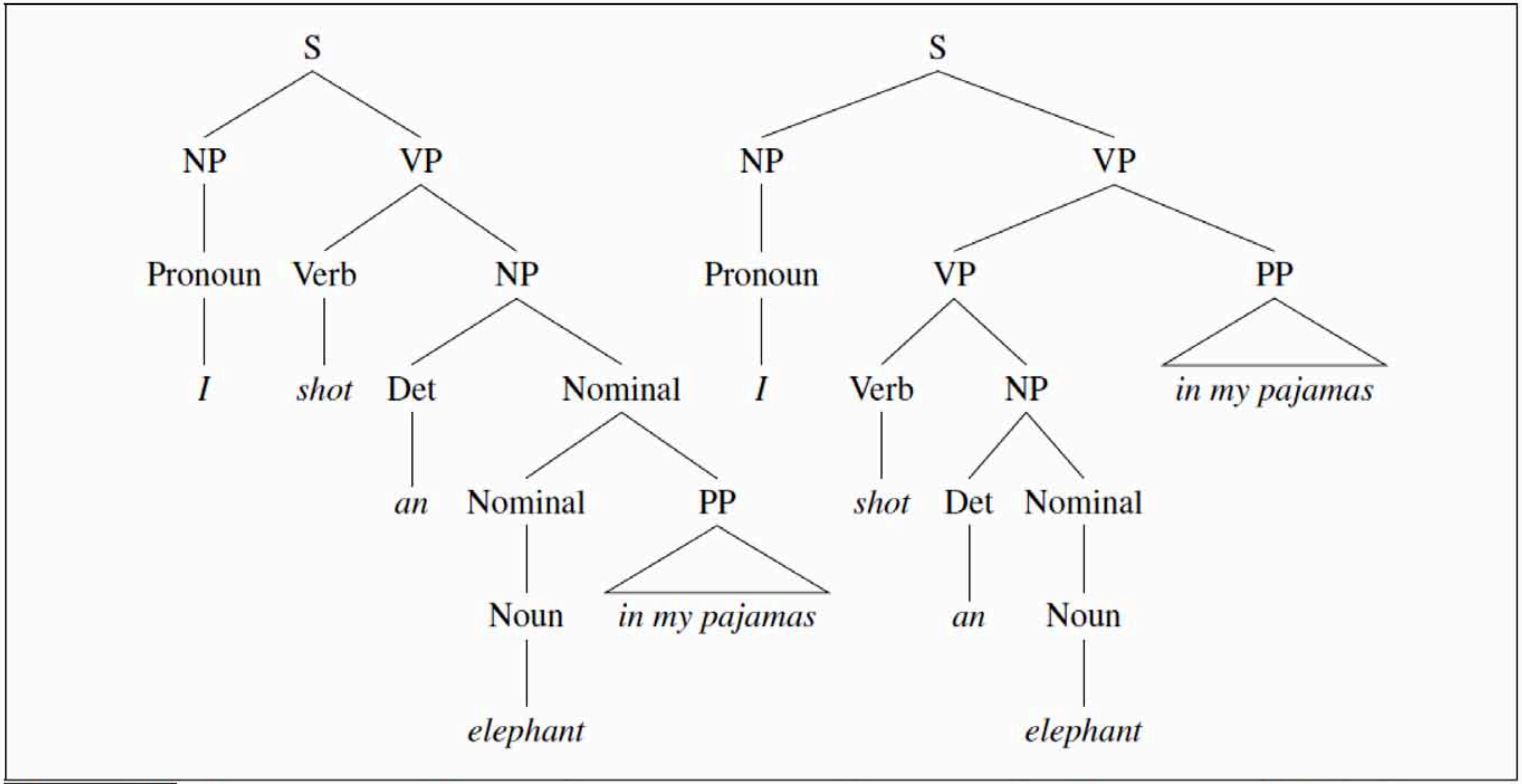



**Figure 12.2** Two parse trees for an ambiguous sentence. The parse on the left corresponds to the humorous reading in which the elephant is in the pajamas, the parse on the right corresponds to the reading in which Captain Spaulding did the shooting in his pajamas.

*source: J&M (3d Ed. draft)*

# Syntactic Disambiguation

- Syntactic ambiguity affects all parsers
  - grammatically correct but sematically unreasonable parses

- Need additional knowledge sources for syntactic disambiguation
  - statistical
  - semantic
  - contextual

- CKY dynamic programming parsing algorithm is designed to handle structural ambiguities
  - can also be extended with statistical techniques to produce highly accurate parsers

# CKY Parser

- Named after Cooke-Kasami-Younger (1965, 1967)
  - also known as CYK parser
- Context-free nature of grammar allows dynamic programming approach
  - once a constituent has been discovered, we can use it in subsequent derivations
  - this provides both time and storage efficiencies
    - subtrees can be looked up in a table, not reanalyzed
- Requirement: the CFG must be in Chomsky Normal Form (CNF)
  - recall: any CFG can be converted into CNF

# Today

- Structural Ambiguity
- Conversion to CNF
- CKY Recognition
- CKY Parsing
- Partial Parsing

# Conversion to CNF

- Recall: all CNF rules must be of form: $A \rightarrow B\ C$

  $A \rightarrow w$

- Given: a generic CFG (assume $\varepsilon$-free)

- We must consider 3 cases
  1. mix terminals with nonterminals
  2. have a single nonterminal on the right-hand side
  3. right-hand side has length > 2

# Rules that mix terminals and nonterminals

- CNF solution: introduce a new dummy nonterminal for each terminal
- Example

Original rule: $INF\_VP \rightarrow to\ \ VP$

CNF version: $INF\_VP \rightarrow TO\ \ VP$

$TO \rightarrow to$

# Rules with single nonterminal at right

- Terminology: A rule of the form $A \rightarrow B$ is called a "unit production"

- CNF solution:

  Eliminate each unit productions by rewriting the original right-hand side (RHS) with the RHS of all the non-unit productions that can be reached from the original RHS

- Example:

| Original rules | CNF version |
|---|---|
| $A \rightarrow B$ | $\left\{ \begin{array}{l} A \rightarrow \gamma \\ A \rightarrow \delta \end{array} \right.$ |
| $B \rightarrow C$ | |
| $B \rightarrow \gamma$ | |
| $C \rightarrow \delta$ | $B \rightarrow \delta$ |

# Rules with RHS longer than 2

- CNF solution: introduce a new dummy nonterminals to spread the longer sequences over several new rules

- Example

Original rule: $S \rightarrow Aux\ \ NP\ \ VP$

CNF version: $S \rightarrow X1\ \ VP$

$X1 \rightarrow Aux\ \ NP$

# $L_1$ in CNF

| $\mathscr{L}_1$ Grammar | $\mathscr{L}_1$ in CNF |
|---|---|
| $S \rightarrow NP\ VP$ | $S \rightarrow NP\ VP$ |
| $S \rightarrow Aux\ NP\ VP$ | $S \rightarrow X1\ VP$ |
| | $X1 \rightarrow Aux\ NP$ |
| $S \rightarrow VP$ | $S \rightarrow book \mid include \mid prefer$ |
| | $S \rightarrow Verb\ NP$ |
| | $S \rightarrow X2\ PP$ |
| | $S \rightarrow Verb\ PP$ |
| | $S \rightarrow VP\ PP$ |
| $NP \rightarrow Pronoun$ | $NP \rightarrow I \mid she \mid me$ |
| $NP \rightarrow Proper\text{-}Noun$ | $NP \rightarrow TWA \mid Houston$ |
| $NP \rightarrow Det\ Nominal$ | $NP \rightarrow Det\ Nominal$ |
| $Nominal \rightarrow Noun$ | $Nominal \rightarrow book \mid flight \mid meal \mid money$ |
| $Nominal \rightarrow Nominal\ Noun$ | $Nominal \rightarrow Nominal\ Noun$ |
| $Nominal \rightarrow Nominal\ PP$ | $Nominal \rightarrow Nominal\ PP$ |
| $VP \rightarrow Verb$ | $VP \rightarrow book \mid include \mid prefer$ |
| $VP \rightarrow Verb\ NP$ | $VP \rightarrow Verb\ NP$ |
| $VP \rightarrow Verb\ NP\ PP$ | $VP \rightarrow X2\ PP$ |
| | $X2 \rightarrow Verb\ NP$ |
| $VP \rightarrow Verb\ PP$ | $VP \rightarrow Verb\ PP$ |
| $VP \rightarrow VP\ PP$ | $VP \rightarrow VP\ PP$ |
| $PP \rightarrow Preposition\ NP$ | $PP \rightarrow Preposition\ NP$ |

**Figure 12.3** $\mathscr{L}_1$ Grammar and its conversion to CNF. Note that although they aren't shown here, all the original lexical entries from $\mathscr{L}_1$ carry over unchanged as well.

*source: J&M (3d Ed. draft)*

# CNF Conversion Summary

1. Copy all rules already in CNF form to the new grammar without change

2. Convert terminals within rules to dummy nonterminals

3. Convert unit-productions

4. Make all rules binary and add them to the new grammar

# Today

- Structural Ambiguity
- Conversion to CNF
- CKY Recognition
- CKY Parsing
- Partial Parsing

# CKY Recognition

- Given CNF form for CFG
  - each nonterminal above POS level will have exactly 2 daughters
  - so, we can use a 2-D matrix to encode the structure of the entire parse tree
- Notation
  - for sentence of length n
  - we will work with the upper-triangular portion of an (n+1) x (n+1) matrix
  - each cell [ i, j ] will contain the set of nonterminals that represent all the constituents that span positions i through j of the input.
  - indexing scheme begins with 0
    - so, we can view the indices as pointing to the gaps between the input words
    - e.g., $_0$ Book $_1$ that $_2$ flight $_3$
  - the cell that represents the entire sentence is matrix entry [ 0, n ]

# The CKY Algorithm

**function** CKY-PARSE(*words, grammar*) **returns** *table*

**for** $j \leftarrow$ **from** 1 **to** LENGTH(*words*) **do** ← iterate over columns
  **for all** $\{A \mid A \rightarrow words[j] \in grammar\}$
    $table[j-1, j] \leftarrow table[j-1, j] \cup A$
  **for** $i \leftarrow$ **from** $j-2$ **downto** 0 **do** ← iterate over rows
    **for** $k \leftarrow i+1$ **to** $j-1$ **do** ← iterate over all possible places to split into daughters
      **for all** $\{A \mid A \rightarrow BC \in grammar$ **and** $B \in table[i,k]$ **and** $C \in table[k,j]\}$
        $table[i,j] \leftarrow table[i,j] \cup A$

**Figure 12.5** The CKY algorithm.

*source: J&M (3d Ed. draft)*

- Interpretation:
  - Superdiagonal contains POS tags
  - Each nonterminal [ i, j ] entry above superdiagonal is composed of two daughters: 1 is same row to ***left***; and the other in same column ***below***
  - Fill in table 1 column at time, from left to right
  - Within each column, fill in from bottom to top

# Example: CKY Matrix

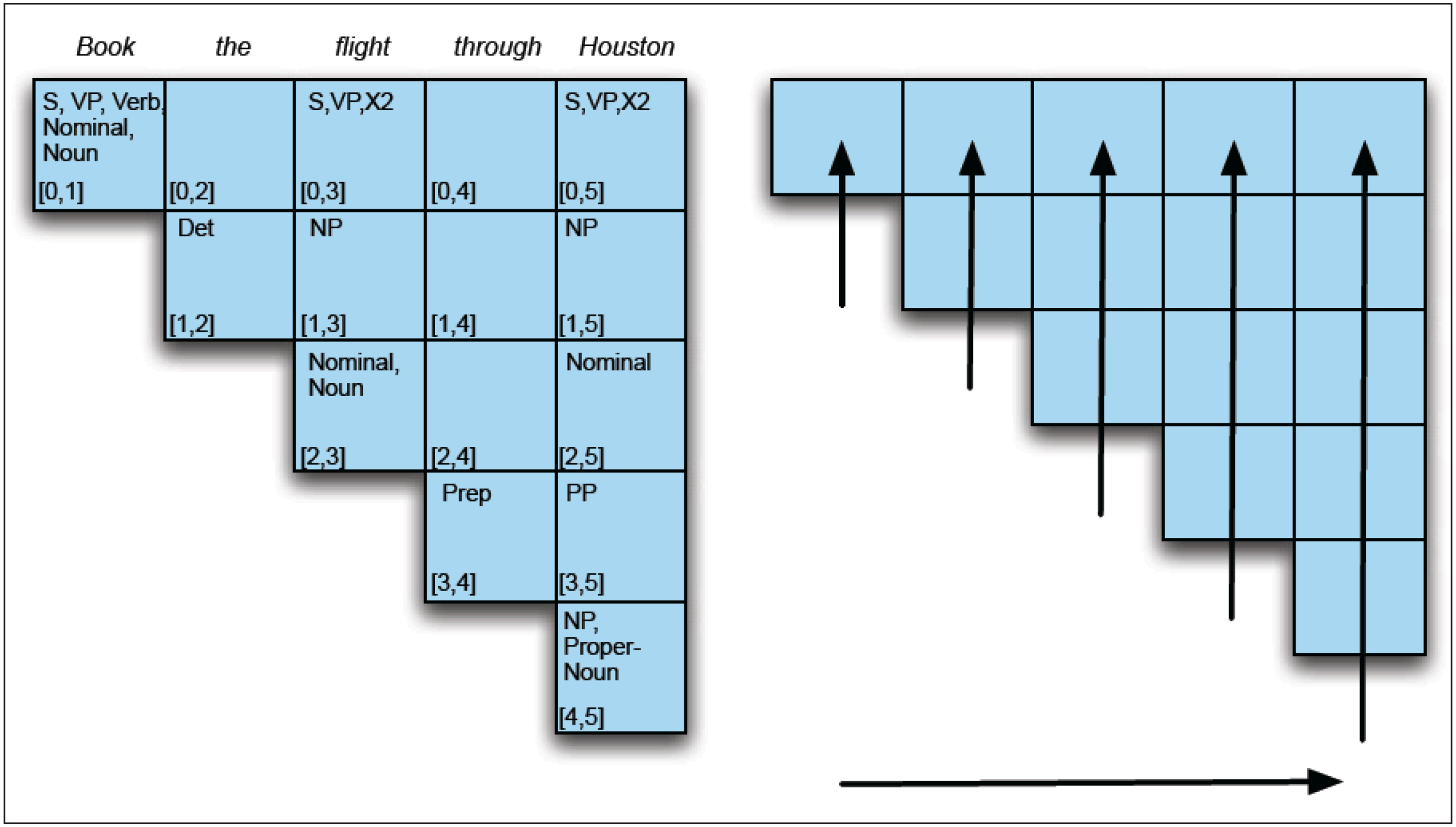



**Figure 12.4** Completed parse table for *Book the flight through Houston.*

*source: J&M (3d Ed. draft)*

# Filling in the [ i, j ]th cell

- The candidate cells for each split move in lockstep
- For each candidate pair, the algorithm checks whether the contents of the cells can be combined according to some rule of the grammar
- If such a rule exists, the nonterminal on its LHS is entered in the table
- Otherwise, the table entry is left empty

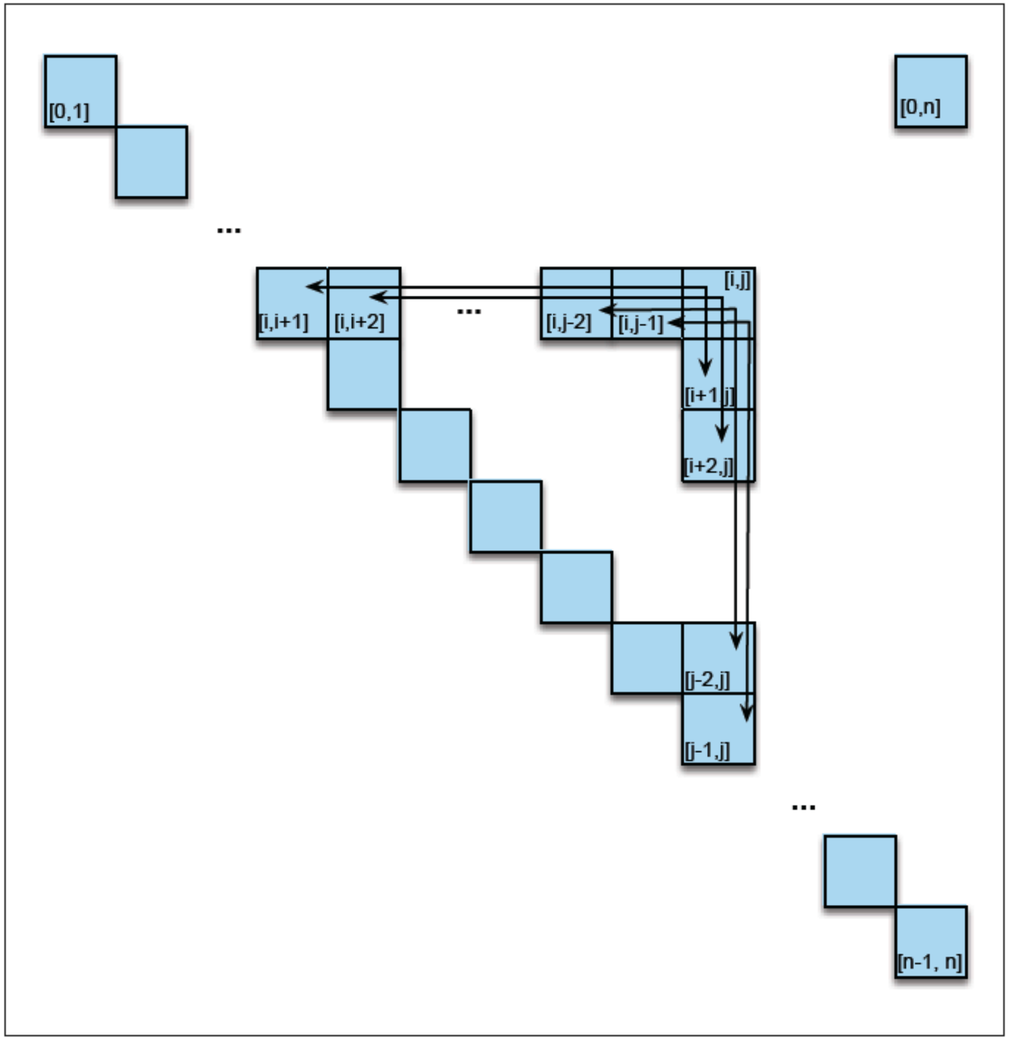



**Figure 12.6** All the ways to fill the [$i$, $j$]th cell in the CKY table.

*source: J&M (3d Ed. draft)*

# Example: Filling last column of table (1)

[ 4, 5 ]: The word "Houston" can be NP or Proper-Noun

| Book | the | flight | through | Houston |
|---|---|---|---|---|
| S, VP, Verb, Nominal, Noun [0,1] | [0,2] | S,VP,X2 [0,3] | [0,4] | [0,5] |
| | Det [1,2] | NP [1,3] | [1,4] | [1,5] |
| | | Nominal, Noun [2,3] | [2,4] | Nominal [2,5] |
| | | | Prep [3,4] | [3,5] |
| | | | | NP, Proper-Noun [4,5] |

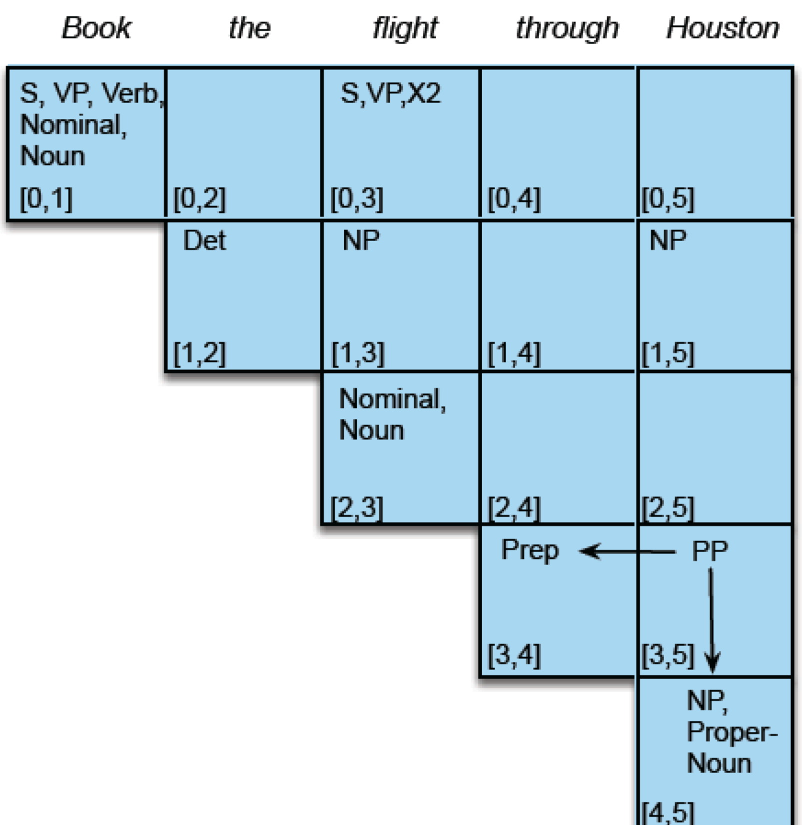
| $\mathscr{L}_1$ **Grammar** | $\mathscr{L}_1$ **in CNF** |
|---|---|
| $S \rightarrow NP\ VP$ | $S \rightarrow NP\ VP$ |
| $S \rightarrow Aux\ NP\ VP$ | $S \rightarrow X1\ VP$ |
| | $X1 \rightarrow Aux\ NP$ |
| $S \rightarrow VP$ | $S \rightarrow book \mid include \mid prefer$ |
| | $S \rightarrow Verb\ NP$ |
| | $S \rightarrow X2\ PP$ |
| | $S \rightarrow Verb\ PP$ |
| | $S \rightarrow VP\ PP$ |
| $NP \rightarrow Pronoun$ | $NP \rightarrow I \mid she \mid me$ |
| $NP \rightarrow Proper\text{-}Noun$ | $NP \rightarrow TWA \mid Houston$ |
| $NP \rightarrow Det\ Nominal$ | $NP \rightarrow Det\ Nominal$ |
| $Nominal \rightarrow Noun$ | $Nominal \rightarrow book \mid flight \mid meal \mid money$ |
| $Nominal \rightarrow Nominal\ Noun$ | $Nominal \rightarrow Nominal\ Noun$ |
| $Nominal \rightarrow Nominal\ PP$ | $Nominal \rightarrow Nominal\ PP$ |
| $VP \rightarrow Verb$ | $VP \rightarrow book \mid include \mid prefer$ |
| $VP \rightarrow Verb\ NP$ | $VP \rightarrow Verb\ NP$ |
| $VP \rightarrow Verb\ NP\ PP$ | $VP \rightarrow X2\ PP$ |
| | $X2 \rightarrow Verb\ NP$ |
| $VP \rightarrow Verb\ PP$ | $VP \rightarrow Verb\ PP$ |
| $VP \rightarrow VP\ PP$ | $VP \rightarrow VP\ PP$ |
| $PP \rightarrow Preposition\ NP$ | $PP \rightarrow Preposition\ NP$ |

**Figure 12.3** $\mathscr{L}_1$ Grammar and its conversion to CNF. Note that although they aren't shown here, all the original lexical entries from $\mathscr{L}_1$ carry over unchanged as well.

*source: J&M (3d Ed. draft)*

# Example: Filling last column of table (2)

[ 3, 5 ] Can form PP from Prep and NP




| $\mathscr{L}_1$ Grammar | $\mathscr{L}_1$ in CNF |
|---|---|
| $S \rightarrow NP\ VP$ | $S \rightarrow NP\ VP$ |
| $S \rightarrow Aux\ NP\ VP$ | $S \rightarrow X1\ VP$ |
| | $X1 \rightarrow Aux\ NP$ |
| $S \rightarrow VP$ | $S \rightarrow book \mid include \mid prefer$ |
| | $S \rightarrow Verb\ NP$ |
| | $S \rightarrow X2\ PP$ |
| | $S \rightarrow Verb\ PP$ |
| | $S \rightarrow VP\ PP$ |
| $NP \rightarrow Pronoun$ | $NP \rightarrow I \mid she \mid me$ |
| $NP \rightarrow Proper\text{-}Noun$ | $NP \rightarrow TWA \mid Houston$ |
| $NP \rightarrow Det\ Nominal$ | $NP \rightarrow Det\ Nominal$ |
| $Nominal \rightarrow Noun$ | $Nominal \rightarrow book \mid flight \mid meal \mid money$ |
| $Nominal \rightarrow Nominal\ Noun$ | $Nominal \rightarrow Nominal\ Noun$ |
| $Nominal \rightarrow Nominal\ PP$ | $Nominal \rightarrow Nominal\ PP$ |
| $VP \rightarrow Verb$ | $VP \rightarrow book \mid include \mid prefer$ |
| $VP \rightarrow Verb\ NP$ | $VP \rightarrow Verb\ NP$ |
| $VP \rightarrow Verb\ NP\ PP$ | $VP \rightarrow X2\ PP$ |
| | $X2 \rightarrow Verb\ NP$ |
| $VP \rightarrow Verb\ PP$ | $VP \rightarrow Verb\ PP$ |
| $VP \rightarrow VP\ PP$ | $VP \rightarrow VP\ PP$ |
| $PP \rightarrow Preposition\ NP$ | $PP \rightarrow Preposition\ NP$ |

**Figure 12.3** $\mathscr{L}_1$ Grammar and its conversion to CNF. Note that although they aren't shown here, all the original lexical entries from $\mathscr{L}_1$ carry over unchanged as well.

*source: J&M (3d Ed. draft)*

# Example: Filling last column of table (3)

[ 2, 5 ] Can form Nominal from Nominal and PP

| Book | the | flight | through | Houston |
|---|---|---|---|---|
| S, VP, Verb, Nominal, Noun [0,1] | [0,2] | S,VP,X2 [0,3] | [0,4] | [0,5] |
| | Det [1,2] | NP [1,3] | [1,4] | NP [1,5] |
| | | Nominal, Noun [2,3] | [2,4] | Nominal [2,5] |
| | | | Prep [3,4] | PP [3,5] |
| | | | | NP, Proper-Noun [4,5] |

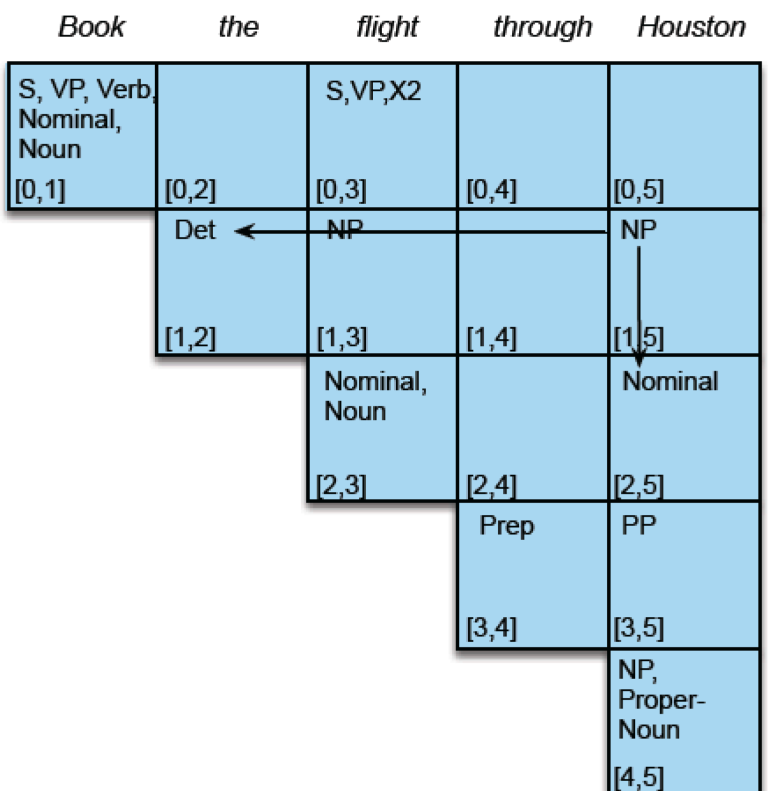
| $\mathscr{L}_1$ Grammar | $\mathscr{L}_1$ in CNF |
|---|---|
| $S \rightarrow NP\ VP$ | $S \rightarrow NP\ VP$ |
| $S \rightarrow Aux\ NP\ VP$ | $S \rightarrow X1\ VP$ |
| | $X1 \rightarrow Aux\ NP$ |
| $S \rightarrow VP$ | $S \rightarrow book \mid include \mid prefer$ |
| | $S \rightarrow Verb\ NP$ |
| | $S \rightarrow X2\ PP$ |
| | $S \rightarrow Verb\ PP$ |
| | $S \rightarrow VP\ PP$ |
| $NP \rightarrow Pronoun$ | $NP \rightarrow I \mid she \mid me$ |
| $NP \rightarrow Proper\text{-}Noun$ | $NP \rightarrow TWA \mid Houston$ |
| $NP \rightarrow Det\ Nominal$ | $NP \rightarrow Det\ Nominal$ |
| $Nominal \rightarrow Noun$ | $Nominal \rightarrow book \mid flight \mid meal \mid money$ |
| $Nominal \rightarrow Nominal\ Noun$ | $Nominal \rightarrow Nominal\ Noun$ |
| $Nominal \rightarrow Nominal\ PP$ | $Nominal \rightarrow Nominal\ PP$ |
| $VP \rightarrow Verb$ | $VP \rightarrow book \mid include \mid prefer$ |
| $VP \rightarrow Verb\ NP$ | $VP \rightarrow Verb\ NP$ |
| $VP \rightarrow Verb\ NP\ PP$ | $VP \rightarrow X2\ PP$ |
| | $X2 \rightarrow Verb\ NP$ |
| $VP \rightarrow Verb\ PP$ | $VP \rightarrow Verb\ PP$ |
| $VP \rightarrow VP\ PP$ | $VP \rightarrow VP\ PP$ |
| $PP \rightarrow Preposition\ NP$ | $PP \rightarrow Preposition\ NP$ |

**Figure 12.3** $\mathscr{L}_1$ Grammar and its conversion to CNF. Note that although they aren't shown here, all the original lexical entries from $\mathscr{L}_1$ carry over unchanged as well.

*source: J&M (3d Ed. draft)*

# Example: Filling last column of table (4)

[ 1, 5 ] Can form NP from Det and Nominal




| $\mathscr{L}_1$ Grammar | $\mathscr{L}_1$ in CNF |
|---|---|
| *S → NP VP* | *S → NP VP* |
| *S → Aux NP VP* | *S → X1 VP* |
| | *X1 → Aux NP* |
| *S → VP* | *S → book \| include \| prefer* |
| | *S → Verb NP* |
| | *S → X2 PP* |
| | *S → Verb PP* |
| | *S → VP PP* |
| *NP → Pronoun* | *NP → I \| she \| me* |
| *NP → Proper-Noun* | *NP → TWA \| Houston* |
| *NP → Det Nominal* | *NP → Det Nominal* |
| *Nominal → Noun* | *Nominal → book \| flight \| meal \| money* |
| *Nominal → Nominal Noun* | *Nominal → Nominal Noun* |
| *Nominal → Nominal PP* | *Nominal → Nominal PP* |
| *VP → Verb* | *VP → book \| include \| prefer* |
| *VP → Verb NP* | *VP → Verb NP* |
| *VP → Verb NP PP* | *VP → X2 PP* |
| | *X2 → Verb NP* |
| *VP → Verb PP* | *VP → Verb PP* |
| *VP → VP PP* | *VP → VP PP* |
| *PP → Preposition NP* | *PP → Preposition NP* |

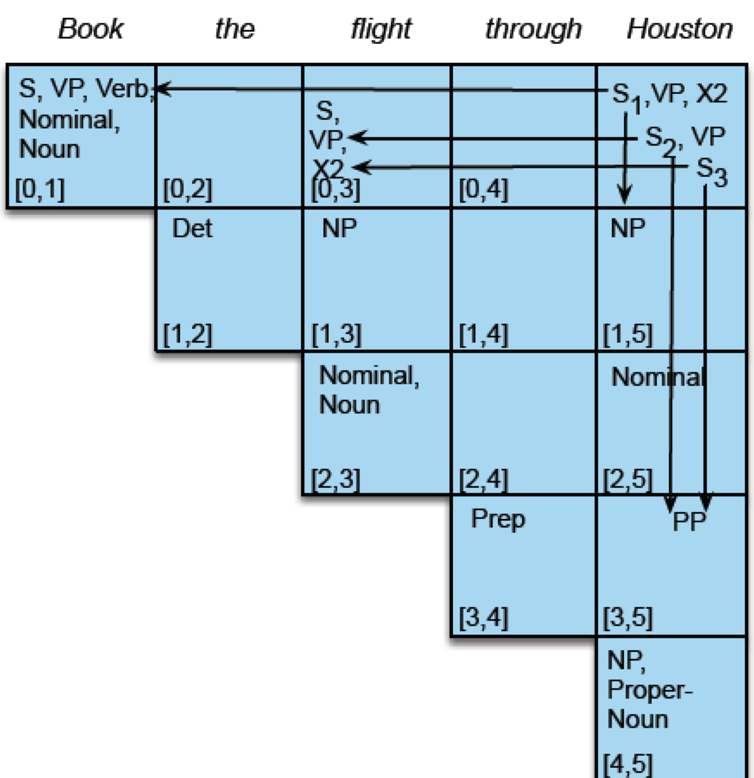
**Figure 12.3** $\mathscr{L}_1$ Grammar and its conversion to CNF. Note that although they aren't shown here, all the original lexical entries from $\mathscr{L}_1$ carry over unchanged as well.

*source: J&M (3d Ed. draft)*

# Example: Filling last column of table (5)

[ 0, 5 ] Can form 3 alternate parses

- the PP modifies the *flight*
- the PP modifies the booking
- also Verb NP (now in rule X2)




| $\mathscr{L}_1$ Grammar | $\mathscr{L}_1$ in CNF |
|---|---|
| $S \rightarrow NP\ VP$ | $S \rightarrow NP\ VP$ |
| $S \rightarrow Aux\ NP\ VP$ | $S \rightarrow X1\ VP$ |
| | $X1 \rightarrow Aux\ NP$ |
| $S \rightarrow VP$ | $S \rightarrow book \mid include \mid prefer$ |
| | $S \rightarrow Verb\ NP$ |
| | $S \rightarrow X2\ PP$ |
| | $S \rightarrow Verb\ PP$ |
| | $S \rightarrow VP\ PP$ |
| $NP \rightarrow Pronoun$ | $NP \rightarrow I \mid she \mid me$ |
| $NP \rightarrow Proper\text{-}Noun$ | $NP \rightarrow TWA \mid Houston$ |
| $NP \rightarrow Det\ Nominal$ | $NP \rightarrow Det\ Nominal$ |
| $Nominal \rightarrow Noun$ | $Nominal \rightarrow book \mid flight \mid meal \mid money$ |
| $Nominal \rightarrow Nominal\ Noun$ | $Nominal \rightarrow Nominal\ Noun$ |
| $Nominal \rightarrow Nominal\ PP$ | $Nominal \rightarrow Nominal\ PP$ |
| $VP \rightarrow Verb$ | $VP \rightarrow book \mid include \mid prefer$ |
| $VP \rightarrow Verb\ NP$ | $VP \rightarrow Verb\ NP$ |
| $VP \rightarrow Verb\ NP\ PP$ | $VP \rightarrow X2\ PP$ |
| | $X2 \rightarrow Verb\ NP$ |
| $VP \rightarrow Verb\ PP$ | $VP \rightarrow Verb\ PP$ |
| $VP \rightarrow VP\ PP$ | $VP \rightarrow VP\ PP$ |
| $PP \rightarrow Preposition\ NP$ | $PP \rightarrow Preposition\ NP$ |

**Figure 12.3** $\mathscr{L}_1$ Grammar and its conversion to CNF. Note that although they aren't shown here, all the original lexical entries from $\mathscr{L}_1$ carry over unchanged as well.

*source: J&M (3d Ed. draft)*

# Today

- Structural Ambiguity
- Conversion to CNF
- CKY Recognition
- CKY Parsing
- Partial Parsing

# From Recognition to Parsing

- Previous algorithm is a recognizer

- Recognition is successful if can find any S in cell [ 0, n ]

- To turn this into a parser, we must make two simple changes

  1. augment the entries in the table so each nonterminal is paired with pointers to the table entries from which it was derived

  2. allow multiple versions of the same nonterminal to be entered in a cell

- Can retrieve each parse by choosing an S at [ 0, n ] and recursively retrieving its component constituents using the pointers

# CKY in Practice

- Basic performance of recognizer is $O(\, n^3 \; \cdot |G|\,)$, where n is length of parsed string and $|G|$ is size of CNF grammar

- Retrieving all parses degrades performance substantially
  - reason: in general, there is an exponential number of parses for a given input

- To retrieve the "best" parse, we must further augment the table with probabilities
  - generally use a modified version of the Viterbi algorithm

- In practice, also often wish to convert results of algorithm back into original grammar before CNF conversion
  - can be done, but takes some work

# Today

- Structural Ambiguity
- Conversion to CNF
- CKY Recognition
- CKY Parsing
- Partial Parsing

# Partial Parsing

- Complete parse trees are not always needed
  - information extraction does not seek to extract ***all*** information, just some predefined interesting information
  - information retrieval systems may index documents according to only a subset of the constituents that may be found in the documents
- Partial (or shallow) parsing approaches
  - cascades of FSTs – tends to produce flat parse trees
  - **chunking**
    - identifying and classifying non-overlapping segments
    - constitute basic non-recursive phrases (e.g., NP, VP, AdjP, PP)
      - these are the phrases that correspond to the content-bearing parts-of-speech
      - some tasks require finding only NP chunks

# Notation Partial Parses

- Since chunks do not overlap, we can use simple bracketed notation

- Example

  [ $_{NP}$ The morning flight ] [ $_{PP}$ from ] [ $_{NP}$ Denver ] [ $_{VP}$ has arrived. ]

- Chunking conventions
  - base phrases will not recursively contain constituents of the same type
  - base phrases include headword, plus any pre-head material
  - base phrases will not include post-head material
    - see, for example, how "from Denver" was chunked above

# Machine Learning Approaches for Chunking

- Chunking, like POS-tagging, is viewed as a ***sequence labeling*** problem for machine learning
  - a "chunk tag" is attached to each token of input
  - use "IOB" scheme with chunk type
    - size of chunk tagset is 2n+1, where n is the number of chunk types

- Example with NP, VP, PP chunk types

| The | morning | flight | from | Denver | has | arrived |
| --- | --- | --- | --- | --- | --- | --- |
| B_NP | I_NP | I_NP | B_PP | B_NP | B_VP | I_VP |

- Same example with only base-NPs tagged

| The | morning | flight | from | Denver | has | arrived |
| --- | --- | --- | --- | --- | --- | --- |
| B_NP | I_NP | I_NP | O | B_NP | O | O |

# Training Data

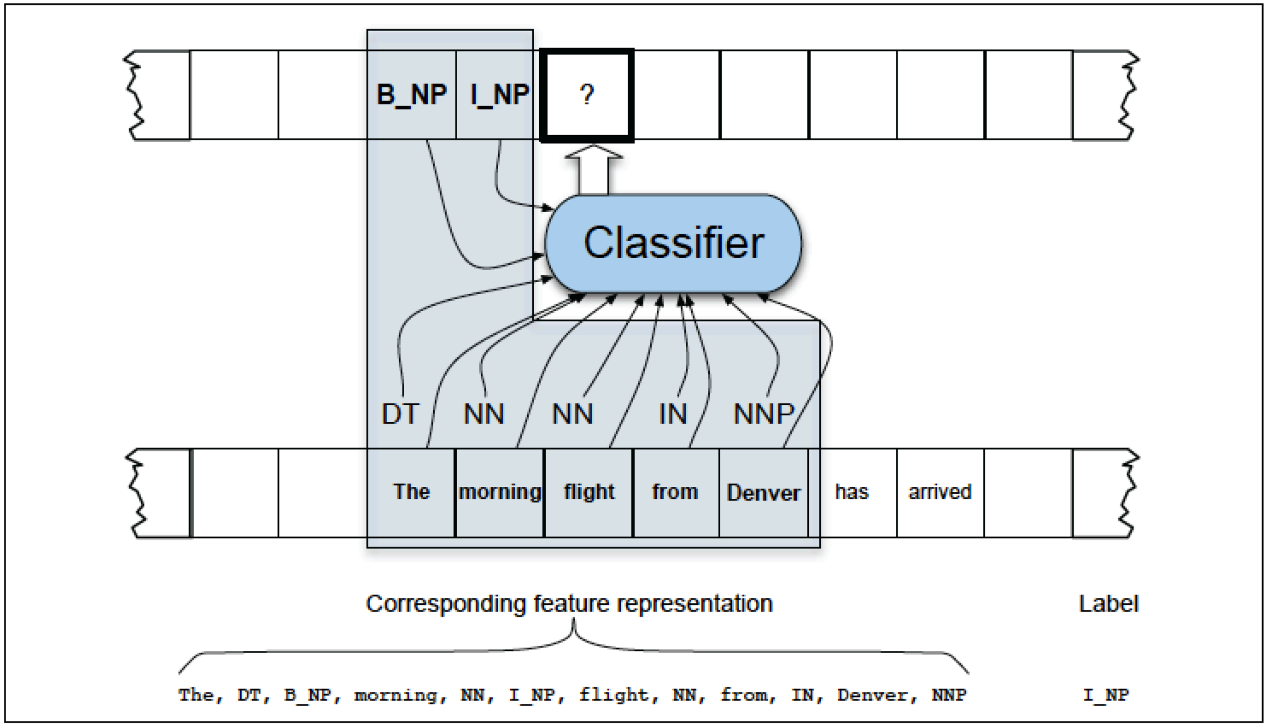
- Hand-labeling data is time-consuming and expensive

- Best source is existing treebanks such as Penn Treebank
  - contains complete parses
  - search for nonterminals corresponding to desired chunk base types
  - find the head of each such nonterminal, and include the pre-head material in the constituent
  - ignore post-head material

- This is somewhat error-prone, since depends on accuracy of head-finding algorithm
  - So, generally have humans review the data that is automatically extracted

# Features for Chunking

- Good performance generally obtained using
  - words and tags within 2 of current index
  - chunk tags for preceding 2 tokens




12 features
+ class label

**Figure 12.8** A sequential-classifier-based approach to chunking. The chunker slides a context window over the sentence, classifying words as it proceeds. At this point, the classifier is attempting to label *flight*. Features derived from the context typically include the words, part-of-speech tags as well as the previously assigned chunk tags.

*source: J&M (3d Ed. draft)*

# Chunking System Evaluation

- Use basic measures from information retrieval

$$Precision = \frac{\#\ correct\ chunks\ returned\ by\ system}{Total\ \#\ chunks\ returned\ by\ system}$$

$$Recall = \frac{\#\ correct\ chunks\ returned\ by\ system}{Total\ \#\ actual\ chunks\ in\ the\ text}$$

$$F_1 = \frac{2PR}{P + R}$$

- In particular contexts, there may be partial credit for chunks that overlap with but whose boundaries do not exactly match the gold standard output