Earley Parsing

Informatics 2A: Lecture 21

Shay Cohen

5 November 2018

A succinct representation of CKY

We have a Boolean table called Chart, such that $\mathrm{Chart}[A, i, j]$ is true if there is a sub-phrase according the grammar that dominates words i through words j

Build this chart recursively, similarly to the Viterbi algorithm:

For $j > i + 1$:

$$\mathrm{Chart}[A, i, j] = \bigvee_{k=i+1}^{j-1} \bigvee_{A \to B\ C} Chart[B, i, k] \wedge Chart[C, k, j]$$

Seed the chart, for $i + 1 = j$:
$\mathrm{Chart}[A, i, i + 1] = \mathrm{True}$ if there exists a rule $A \to w_{i+1}$ where w_{i+1} is the $(i + 1)$th word in the string

Computational complexity of CKY

We use big-O notation

We count how many "basic operations" it takes to fill-in the CYK chart as a function of the length of the sentence n

The CKY algorithm loops over: splitting points ($O(n)$), beginning point ($O(n)$) and end points in spans ($O(n)$) – as such, its complexity is $O(n^3)$

What would be the complexity as a function of the size of the grammar $|G|$?

Note about CYK

The CYK algorithm parses input strings in Chomsky normal form. Can you see how to change it to an algorithm with an arbitrary RHS length (of only nonterminals)?

Note about CYK

The CYK algorithm parses input strings in Chomsky normal form. Can you see how to change it to an algorithm with an arbitrary RHS length (of only nonterminals)?

We would have to split a given span into all possible subspans according to the length of the RHS. What is the complexity of such algorithm?

Note about CYK

The CYK algorithm parses input strings in Chomsky normal form. Can you see how to change it to an algorithm with an arbitrary RHS length (of only nonterminals)?

We would have to split a given span into all possible subspans according to the length of the RHS. What is the complexity of such algorithm?

Still $O(n^2)$ charts, but now it takes $O(n^{k-1})$ time to process each cell, where k is the maximal length of an RHS. Therefore: $O(n^{k+1})$. For CYK, $k = 2$.

Can we do better than that?

A Simple Grammar: The Problem with CKY

Consider this simple grammar in Chomsky normal form:

Binary rules	**Lexical rules**
S $\rightarrow$ B X	X $\rightarrow$ a
X $\rightarrow$ X X	B $\rightarrow$ b
S $\rightarrow$ C Y	Y $\rightarrow$ a
Y $\rightarrow$ Y Y	C $\rightarrow$ c

What is the language of this grammar?

A Simple Grammar: The Problem with CKY

Consider this simple grammar in Chomsky normal form:

Binary rules	**Lexical rules**
S $\rightarrow$ B X	X $\rightarrow$ a
X $\rightarrow$ X X	B $\rightarrow$ b
S $\rightarrow$ C Y	Y $\rightarrow$ a
Y $\rightarrow$ Y Y	C $\rightarrow$ c

What is the language of this grammar? $(b|c)a^{+}$

What will CKY do if we try to parse *baaaaa*?

A Simple Grammar: The Problem with CKY

Consider this simple grammar in Chomsky normal form:

Binary rules	**Lexical rules**
S $\rightarrow$ B X	X $\rightarrow$ a
X $\rightarrow$ X X	B $\rightarrow$ b
S $\rightarrow$ C Y	Y $\rightarrow$ a
Y $\rightarrow$ Y Y	C $\rightarrow$ c

What is the language of this grammar? $(b|c)a^+$

What will CKY do if we try to parse *baaaaa*?

What will CKY do if we try to parse *caaaaa*?

CYK Chart entries

The CYK algorithm avoids redundant work by storing in a chart all the constituents it finds.

But it populates the table with phantom constituents, that don't form part of any complete parse. Can be a significant problem in long sentences.

The idea of the *Earley algorithm* is to avoid this, by only building constituents that are compatible with the input read so far.

Earley Parsing

Key idea: as well as completed productions (ones whose entire RHS have been recognized), we also record incomplete productions (ones for which there may so far be only partial evidence).

- Incomplete productions (aka incomplete constituents) are effectively predictions about what might come next and what will be learned from finding it.
- Incomplete constituents can be represented using an extended form of production rule called a dotted rule, e.g. $VP \rightarrow V \bullet NP$.
- The dot indicates how much of the RHS has already been found. The rest is a prediction of what is to come.

Earley Parsing

- Allows arbitrary CFGs
- Top-down control
- Fills a table in a single sweep over the input
- Table entries represent:
  - Completed constituents and their locations
  - In-progress constituents
  - Predicted constituents

States

The table entries are called states and are represented with dotted-rules.

$S \rightarrow \bullet\ VP$ [0,0]	A *VP* is predicted at the start of the sentence
$NP \rightarrow Det \bullet Nominal$ [1,2]	An NP is in progress; seen *Det*, *Nominal* is expected
$VP \rightarrow V\ NP \bullet$[0,3]	A VP has been found starting at 0 and ending at 3

Once chart is populated there should be an S the final column that spans from 0 to N and is complete: $S \rightarrow \alpha \bullet [0, N]$. If that's the case you're done.

Sketch of Earley Algorithm

1. Predict all the states you can upfront, working top-down from S
2. For each word in the input:
   2.1 Scan in the word.
   2.2 Complete or extend existing states based on matches.
   2.3 Add new predictions.
3. When out of words, look at the chart to see if you have a winner.

The algorithm uses three basic operations to process states in the chart: PREDICTOR and COMPLETER add states to the chart entry being processed; SCANNER adds a state to the next chart entry.

Predictor

- Creates new states representing top-down expectations
- Applied to any state that has a non-terminal (other than a part-of-speech category) immediately to right of dot
- Application results in creation of one new state for each alternative expansion of that non-terminal
- New states placed into same chart entry as generating state

$$S \rightarrow \bullet VP, [0,0]$$

$$\begin{array}{lll} VP & \rightarrow \bullet & Verb, [0,0] \\ VP & \rightarrow \bullet & Verb\ NP, [0,0] \\ VP & \rightarrow \bullet & Verb\ NP\ PP, [0,0] \\ VP & \rightarrow \bullet & Verb\ PP, [0,0] \\ VP & \rightarrow \bullet & VP\ PP, [0,0] \end{array}$$

Scanner

- Applies to states with a part-of-speech category to right of dot
- Incorporates into chart a state corresponding to prediction of a word with particular part-of-speech
- Creates new state from input state with dot advanced over predicted input category
- Unlike CYK, only parts-of-speech of a word that are predicted by some existing state will enter the chart (top-down input)

$$\mathrm{VP} \rightarrow \bullet\ \mathit{Verb}\ \mathit{NP},\ [0,0]$$

$$\mathit{VP} \quad \rightarrow \quad \text{book} \bullet \mathit{NP},\ [0,1]$$

Completer

- Applied to state when its dot has reached right end of the rule
- This means that parser has successfully discovered a particular grammatical category over some span of the input
- Completer finds and advances all previously created states that were looking for this category at this position in input
- Creates states copying the older state, advancing dot over expected category, and installing new state in chart

$$NP \rightarrow Det\ Nominal\ \bullet,\ [1,3]$$

finds state $\quad VP \rightarrow Verb \bullet NP,\ [0,1]$

finds state $\quad VP \rightarrow Verb \bullet NP\ PP,\ [0,1]$

Completer

- Applied to state when its dot has reached right end of the rule
- This means that parser has successfully discovered a particular grammatical category over some span of the input
- Completer finds and advances all previously created states that were looking for this category at this position in input
- Creates states copying the older state, advancing dot over expected category, and installing new state in chart

$$NP \rightarrow Det\ Nominal\ \bullet,\ [1,3]$$

finds state	VP	$\rightarrow$	$Verb \bullet NP$, [0,1]
finds state	VP	$\rightarrow$	$Verb \bullet NP\ PP$, [0,1]
adds complete state	VP	$\rightarrow$	$Verb\ NP \bullet$, [0,3]
adds incomplete state	VP	$\rightarrow$	$Verb\ NP \bullet PP$, [0,3]

Earley parsing: example

We will use the grammar to parse the sentence "*Book that flight*".

Grammar Rules

$S \rightarrow NP\ VP$	$VP \rightarrow Verb$
$S \rightarrow Aux\ NP\ VP$	$VP \rightarrow Verb\ NP$
$S \rightarrow VP$	$VP \rightarrow Verb\ NP\ PP$
$NP \rightarrow Pronoun$	$VP \rightarrow Verb\ PP$
$NP \rightarrow Proper\text{-}Noun$	$VP \rightarrow VP\ PP$
$NP \rightarrow Det\ Nominal$	$PP \rightarrow Preposition\ NP$
$Nominal \rightarrow Noun$	$Verb \rightarrow book \mid include \mid prefer$
$Nominal \rightarrow Nominal\ Noun$	$Noun \rightarrow book \mid flight \mid meal$
$Nominal \rightarrow Nominal\ PP$	$Det \rightarrow that \mid this \mid these$

Earley parsing: example[0]

state	rule	start/end	reason
S1	$S \rightarrow \bullet\ NP\ VP$	[0,0]	Predictor
S2	$S \rightarrow \bullet\ Aux\ NP\ VP$	[0,0]	Predictor
S3	$S \rightarrow \bullet\ VP$	[0,0]	Predictor
S4	$NP \rightarrow \bullet\ Pronoun$	[0,0]	Predictor
S5	$NP \rightarrow \bullet\ Proper\text{-}Noun$	[0,0]	Predictor
S6	$NP \rightarrow \bullet\ Det\ Nominal$	[0,0]	Predictor
S7	$VP \rightarrow \bullet\ Verb$	[0,0]	Predictor
S8	$VP \rightarrow \bullet\ Verb\ NP$	[0,0]	Predictor
S9	$VP \rightarrow \bullet\ Verb\ NP\ PP$	[0,0]	Predictor
S10	$VP \rightarrow \bullet\ Verb\ PP$	[0,0]	Predictor
S11	$VP \rightarrow \bullet\ VP\ PP$	[0,0]	Predictor

Earley parsing: example[0]

state	rule	start/end	reason
S1	$S \rightarrow \bullet\ NP\ VP$	[0,0]	Predictor
S2	$S \rightarrow \bullet\ Aux\ NP\ VP$	[0,0]	Predictor
S3	$S \rightarrow \bullet\ VP$	[0,0]	Predictor
S4	$NP \rightarrow \bullet\ Pronoun$	[0,0]	Predictor
S5	$NP \rightarrow \bullet\ Proper\text{-}Noun$	[0,0]	Predictor
S6	$NP \rightarrow \bullet\ Det\ Nominal$	[0,0]	Predictor
S7	$VP \rightarrow \bullet\ Verb$	[0,0]	Predictor
S8	$VP \rightarrow \bullet\ Verb\ NP$	[0,0]	Predictor
S9	$VP \rightarrow \bullet\ Verb\ NP\ PP$	[0,0]	Predictor
S10	$VP \rightarrow \bullet\ Verb\ PP$	[0,0]	Predictor
S11	$VP \rightarrow \bullet\ VP\ PP$	[0,0]	Predictor

Earley parsing: example[0]

state	rule	start/end	reason
S1	$S \rightarrow \bullet\ NP\ VP$	[0,0]	Predictor
S2	$S \rightarrow \bullet\ Aux\ NP\ VP$	[0,0]	Predictor
S3	$S \rightarrow \bullet\ VP$	[0,0]	Predictor
S4	$NP \rightarrow \bullet\ Pronoun$	[0,0]	Predictor
S5	$NP \rightarrow \bullet\ Proper\text{-}Noun$	[0,0]	Predictor
S6	$NP \rightarrow \bullet\ Det\ Nominal$	[0,0]	Predictor
S7	$VP \rightarrow \bullet\ Verb$	[0,0]	Predictor
S8	$VP \rightarrow \bullet\ Verb\ NP$	[0,0]	Predictor
S9	$VP \rightarrow \bullet\ Verb\ NP\ PP$	[0,0]	Predictor
S10	$VP \rightarrow \bullet\ Verb\ PP$	[0,0]	Predictor
S11	$VP \rightarrow \bullet\ VP\ PP$	[0,0]	Predictor

Earley parsing: example[0]

state	rule	start/end	reason
S1	$S \rightarrow \bullet\ NP\ VP$	[0,0]	Predictor
S2	$S \rightarrow \bullet\ Aux\ NP\ VP$	[0,0]	Predictor
S3	$S \rightarrow \bullet\ VP$	[0,0]	Predictor
S4	$NP \rightarrow \bullet\ Pronoun$	[0,0]	Predictor
S5	$NP \rightarrow \bullet\ Proper\text{-}Noun$	[0,0]	Predictor
S6	$NP \rightarrow \bullet\ Det\ Nominal$	[0,0]	Predictor
S7	$VP \rightarrow \bullet\ Verb$	[0,0]	Predictor
S8	$VP \rightarrow \bullet\ Verb\ NP$	[0,0]	Predictor
S9	$VP \rightarrow \bullet\ Verb\ NP\ PP$	[0,0]	Predictor
S10	$VP \rightarrow \bullet\ Verb\ PP$	[0,0]	Predictor
S11	$VP \rightarrow \bullet\ VP\ PP$	[0,0]	Predictor

Earley parsing: example[0]

state	rule	start/end	reason
S1	$S \rightarrow \bullet\ NP\ VP$	[0,0]	Predictor
S2	$S \rightarrow \bullet\ Aux\ NP\ VP$	[0,0]	Predictor
S3	$S \rightarrow \bullet\ VP$	[0,0]	Predictor
S4	$NP \rightarrow \bullet\ Pronoun$	[0,0]	Predictor
S5	$NP \rightarrow \bullet\ Proper\text{-}Noun$	[0,0]	Predictor
S6	$NP \rightarrow \bullet\ Det\ Nominal$	[0,0]	Predictor
S7	$VP \rightarrow \bullet\ Verb$	[0,0]	Predictor
S8	$VP \rightarrow \bullet\ Verb\ NP$	[0,0]	Predictor
S9	$VP \rightarrow \bullet\ Verb\ NP\ PP$	[0,0]	Predictor
S10	$VP \rightarrow \bullet\ Verb\ PP$	[0,0]	Predictor
S11	$VP \rightarrow \bullet\ VP\ PP$	[0,0]	Predictor

Earley parsing: example[0]

state	rule	start/end	reason
S1	$S \rightarrow \bullet\ NP\ VP$	[0,0]	Predictor
S2	$S \rightarrow \bullet\ Aux\ NP\ VP$	[0,0]	Predictor
S3	$S \rightarrow \bullet\ VP$	[0,0]	Predictor
S4	$NP \rightarrow \bullet\ Pronoun$	[0,0]	Predictor
S5	$NP \rightarrow \bullet\ Proper\text{-}Noun$	[0,0]	Predictor
S6	$NP \rightarrow \bullet\ Det\ Nominal$	[0,0]	Predictor
S7	$VP \rightarrow \bullet\ Verb$	[0,0]	Predictor
S8	$VP \rightarrow \bullet\ Verb\ NP$	[0,0]	Predictor
S9	$VP \rightarrow \bullet\ Verb\ NP\ PP$	[0,0]	Predictor
S10	$VP \rightarrow \bullet\ Verb\ PP$	[0,0]	Predictor
S11	$VP \rightarrow \bullet\ VP\ PP$	[0,0]	Predictor

Earley parsing: example[1]

state	rule	start/end	reason
S12	$Verb \rightarrow book \bullet$	[0,1]	Scanner
S13	$VP \rightarrow Verb \bullet$	[0,1]	Completer
S14	$VP \rightarrow Verb \bullet NP$	[0,1]	Completer
S15	$VP \rightarrow Verb \bullet NP\ PP$	[0,1]	Completer
S16	$VP \rightarrow Verb \bullet PP$	[0,1]	Completer
S17	$S \rightarrow VP \bullet$	[0,1]	Completer
S18	$VP \rightarrow VP \bullet PP$	[1,1]	Completer
S19	$NP \rightarrow \bullet Pronoun$	[1,1]	Predictor
S20	$NP \rightarrow \bullet Proper\text{-}Noun$	[1,1]	Predictor
S21	$NP \rightarrow \bullet Det\ Nominal$	[1,1]	Predictor
S22	$PP \rightarrow \bullet Prep\ NP$	[1,1]	Predictor

Earley parsing: example[1]

state	rule	start/end	reason
S12	$Verb \rightarrow book \bullet$	[0,1]	Scanner
S13	$VP \rightarrow Verb \bullet$	[0,1]	Completer
S14	$VP \rightarrow Verb \bullet NP$	[0,1]	Completer
S15	$VP \rightarrow Verb \bullet NP\ PP$	[0,1]	Completer
S16	$VP \rightarrow Verb \bullet PP$	[0,1]	Completer
S17	$S \rightarrow VP \bullet$	[0,1]	Completer
S18	$VP \rightarrow VP \bullet PP$	[1,1]	Completer
S19	$NP \rightarrow \bullet Pronoun$	[1,1]	Predictor
S20	$NP \rightarrow \bullet Proper\text{-}Noun$	[1,1]	Predictor
S21	$NP \rightarrow \bullet Det\ Nominal$	[1,1]	Predictor
S22	$PP \rightarrow \bullet Prep\ NP$	[1,1]	Predictor

Earley parsing: example[1]

state	rule	start/end	reason
S12	$Verb \rightarrow book\ \bullet$	[0,1]	Scanner
S13	$VP \rightarrow Verb\ \bullet$	[0,1]	Completer
S14	$VP \rightarrow Verb\ \bullet\ NP$	[0,1]	Completer
S15	$VP \rightarrow Verb\ \bullet\ NP\ PP$	[0,1]	Completer
S16	$VP \rightarrow Verb\ \bullet\ PP$	[0,1]	Completer
S17	$S \rightarrow VP\ \bullet$	[0,1]	Completer
S18	$VP \rightarrow VP\ \bullet\ PP$	[1,1]	Completer
S19	$NP \rightarrow \bullet\ Pronoun$	[1,1]	Predictor
S20	$NP \rightarrow \bullet\ Proper\text{-}Noun$	[1,1]	Predictor
S21	$NP \rightarrow \bullet\ Det\ Nominal$	[1,1]	Predictor
S22	$PP \rightarrow \bullet\ Prep\ NP$	[1,1]	Predictor

Earley parsing: example[1]

state	rule	start/end	reason
S12	$Verb \rightarrow book \bullet$	[0,1]	Scanner
S13	$VP \rightarrow Verb \bullet$	[0,1]	Completer
S14	$VP \rightarrow Verb \bullet NP$	[0,1]	Completer
S15	$VP \rightarrow Verb \bullet NP\ PP$	[0,1]	Completer
S16	$VP \rightarrow Verb \bullet PP$	[0,1]	Completer
S17	$S \rightarrow VP \bullet$	[0,1]	Completer
S18	$VP \rightarrow VP \bullet PP$	[1,1]	Completer
S19	$NP \rightarrow \bullet Pronoun$	[1,1]	Predictor
S20	$NP \rightarrow \bullet Proper\text{-}Noun$	[1,1]	Predictor
S21	$NP \rightarrow \bullet Det\ Nominal$	[1,1]	Predictor
S22	$PP \rightarrow \bullet Prep\ NP$	[1,1]	Predictor

Earley parsing: example[1]

state	rule	start/end	reason
S12	$Verb \rightarrow book\ \bullet$	[0,1]	Scanner
S13	$VP \rightarrow Verb\ \bullet$	[0,1]	Completer
S14	$VP \rightarrow Verb\ \bullet\ NP$	[0,1]	Completer
S15	$VP \rightarrow Verb\ \bullet\ NP\ PP$	[0,1]	Completer
S16	$VP \rightarrow Verb\ \bullet\ PP$	[0,1]	Completer
S17	$S \rightarrow VP\ \bullet$	[0,1]	Completer
S18	$VP \rightarrow VP\ \bullet\ PP$	[1,1]	Completer
S19	$NP \rightarrow \bullet\ Pronoun$	[1,1]	Predictor
S20	$NP \rightarrow \bullet\ Proper\text{-}Noun$	[1,1]	Predictor
S21	$NP \rightarrow \bullet\ Det\ Nominal$	[1,1]	Predictor
S22	$PP \rightarrow \bullet\ Prep\ NP$	[1,1]	Predictor

Earley parsing: example[1]

state	rule	start/end	reason
S12	$Verb \rightarrow book \bullet$	[0,1]	Scanner
S13	$VP \rightarrow Verb \bullet$	[0,1]	Completer
S14	$VP \rightarrow Verb \bullet NP$	[0,1]	Completer
S15	$VP \rightarrow Verb \bullet NP\ PP$	[0,1]	Completer
S16	$VP \rightarrow Verb \bullet PP$	[0,1]	Completer
S17	$S \rightarrow VP \bullet$	[0,1]	Completer
S18	$VP \rightarrow VP \bullet PP$	[1,1]	Completer
S19	$NP \rightarrow \bullet Pronoun$	[1,1]	Predictor
S20	$NP \rightarrow \bullet Proper\text{-}Noun$	[1,1]	Predictor
S21	$NP \rightarrow \bullet Det\ Nominal$	[1,1]	Predictor
S22	$PP \rightarrow \bullet Prep\ NP$	[1,1]	Predictor

Earley parsing: example[1]

state	rule	start/end	reason
S12	$Verb \rightarrow book \bullet$	[0,1]	Scanner
S13	$VP \rightarrow Verb \bullet$	[0,1]	Completer
S14	$VP \rightarrow Verb \bullet NP$	[0,1]	Completer
S15	$VP \rightarrow Verb \bullet NP\ PP$	[0,1]	Completer
S16	$VP \rightarrow Verb \bullet PP$	[0,1]	Completer
S17	$S \rightarrow VP \bullet$	[0,1]	Completer
S18	$VP \rightarrow VP \bullet PP$	[1,1]	Completer
S19	$NP \rightarrow \bullet Pronoun$	[1,1]	Predictor
S20	$NP \rightarrow \bullet Proper\text{-}Noun$	[1,1]	Predictor
S21	$NP \rightarrow \bullet Det\ Nominal$	[1,1]	Predictor
S22	$PP \rightarrow \bullet Prep\ NP$	[1,1]	Predictor

Earley parsing: example[1]

state	rule	start/end	reason
S12	$Verb \rightarrow book \bullet$	[0,1]	Scanner
S13	$VP \rightarrow Verb \bullet$	[0,1]	Completer
S14	$VP \rightarrow Verb \bullet NP$	[0,1]	Completer
S15	$VP \rightarrow Verb \bullet NP\ PP$	[0,1]	Completer
S16	$VP \rightarrow Verb \bullet PP$	[0,1]	Completer
S17	$S \rightarrow VP \bullet$	[0,1]	Completer
S18	$VP \rightarrow VP \bullet PP$	[1,1]	Completer
S19	$NP \rightarrow \bullet Pronoun$	[1,1]	Predictor
S20	$NP \rightarrow \bullet Proper\text{-}Noun$	[1,1]	Predictor
S21	$NP \rightarrow \bullet Det\ Nominal$	[1,1]	Predictor
S22	$PP \rightarrow \bullet Prep\ NP$	[1,1]	Predictor

Earley parsing: example[2]

state	rule	start/end	reason
S23	$Det \rightarrow that \bullet$	[1,2]	Scanner
S24	$NP \rightarrow Det \bullet Nominal$	[1,2]	Completer
S25	$Nominal \rightarrow \bullet Noun$	[2,2]	Predictor
S26	$Nominal \rightarrow \bullet Nominal\ Noun$	[2,2]	Predictor
S27	$Nominal \rightarrow \bullet Nominal\ PP$	[2,2]	Predictor

Earley parsing: example[2]

state	rule	start/end	reason
S23	$Det \rightarrow that \bullet$	[1,2]	Scanner
S24	$NP \rightarrow Det \bullet Nominal$	[1,2]	Completer
S25	$Nominal \rightarrow \bullet Noun$	[2,2]	Predictor
S26	$Nominal \rightarrow \bullet Nominal\ Noun$	[2,2]	Predictor
S27	$Nominal \rightarrow \bullet Nominal\ PP$	[2,2]	Predictor

Earley parsing: example[2]

state	rule	start/end	reason
S23	$Det \rightarrow that \bullet$	[1,2]	Scanner
S24	$NP \rightarrow Det \bullet Nominal$	[1,2]	Completer
S25	$Nominal \rightarrow \bullet Noun$	[2,2]	Predictor
S26	$Nominal \rightarrow \bullet Nominal\ Noun$	[2,2]	Predictor
S27	$Nominal \rightarrow \bullet Nominal\ PP$	[2,2]	Predictor

Earley parsing: example[2]

state	rule	start/end	reason
S23	$Det \rightarrow that \bullet$	[1,2]	Scanner
S24	$NP \rightarrow Det \bullet Nominal$	[1,2]	Completer
S25	$Nominal \rightarrow \bullet Noun$	[2,2]	Predictor
S26	$Nominal \rightarrow \bullet Nominal\ Noun$	[2,2]	Predictor
S27	$Nominal \rightarrow \bullet Nominal\ PP$	[2,2]	Predictor

Earley parsing: example[3]

state	rule	start/end	reason
S28	$Noun \rightarrow \bullet\ flight$	[2,3]	Scanner
S29	$Nominal \rightarrow Noun\ \bullet$	[2,3]	Completer
S30	$NP \rightarrow Det\ Nominal\ \bullet$	[1,3]	Completer
S31	$Nominal \rightarrow Nominal\ \bullet\ Noun$	[2,3]	Completer
S32	$Nominal \rightarrow Nominal\ \bullet\ PP$	[2,3]	Completer
S33	$VP \rightarrow Verb\ NP\ \bullet$	[0,3]	Completer
S34	$VP \rightarrow Verb\ NP \bullet\ PP$	[0,3]	Completer
S35	$PP \rightarrow Prep\ \bullet\ NP$	[3,3]	Predictor
S36	$S \rightarrow VP\ \bullet$	[0,3]	Completer
S37	$VP \rightarrow VP\ \bullet\ PP$	[0,3]	Completer

Earley parsing: example[3]

state	rule	start/end	reason
S28	$Noun \rightarrow \bullet\ flight$	[2,3]	Scanner
S29	$Nominal \rightarrow Noun\ \bullet$	[2,3]	Completer
S30	$NP \rightarrow Det\ Nominal\ \bullet$	[1,3]	Completer
S31	$Nominal \rightarrow Nominal\ \bullet\ Noun$	[2,3]	Completer
S32	$Nominal \rightarrow Nominal\ \bullet\ PP$	[2,3]	Completer
S33	$VP \rightarrow Verb\ NP\ \bullet$	[0,3]	Completer
S34	$VP \rightarrow Verb\ NP \bullet\ PP$	[0,3]	Completer
S35	$PP \rightarrow Prep\ \bullet\ NP$	[3,3]	Predictor
S36	$S \rightarrow VP\ \bullet$	[0,3]	Completer
S37	$VP \rightarrow VP\ \bullet\ PP$	[0,3]	Completer

Earley parsing: example[3]

state	rule	start/end	reason
S28	*Noun* → • *flight*	[2,3]	Scanner
S29	*Nominal* → *Noun* •	[2,3]	Completer
S30	*NP* → *Det Nominal* •	[1,3]	Completer
S31	*Nominal* → *Nominal* • *Noun*	[2,3]	Completer
S32	*Nominal* → *Nominal* • *PP*	[2,3]	Completer
S33	*VP* → *Verb NP* •	[0,3]	Completer
S34	*VP* → *Verb NP* • *PP*	[0,3]	Completer
S35	*PP* → *Prep* • *NP*	[3,3]	Predictor
S36	*S* → *VP* •	[0,3]	Completer
S37	*VP* → *VP* • *PP*	[0,3]	Completer

Earley parsing: example[3]

state	rule	start/end	reason
S28	$Noun \rightarrow \bullet\ flight$	[2,3]	Scanner
S29	$Nominal \rightarrow Noun\ \bullet$	[2,3]	Completer
S30	$NP \rightarrow Det\ Nominal\ \bullet$	[1,3]	Completer
S31	$Nominal \rightarrow Nominal\ \bullet\ Noun$	[2,3]	Completer
S32	$Nominal \rightarrow Nominal\ \bullet\ PP$	[2,3]	Completer
S33	$VP \rightarrow Verb\ NP\ \bullet$	[0,3]	Completer
S34	$VP \rightarrow Verb\ NP \bullet\ PP$	[0,3]	Completer
S35	$PP \rightarrow Prep\ \bullet\ NP$	[3,3]	Predictor
S36	$S \rightarrow VP\ \bullet$	[0,3]	Completer
S37	$VP \rightarrow VP\ \bullet\ PP$	[0,3]	Completer

The Earley Algorithm

```
function EARLEY-PARSE(words, grammar) returns chart

  ENQUEUE((γ → • S, [0,0]), chart[0])
  for i ← from 0 to LENGTH(words) do
   for each state in chart[i] do
     if INCOMPLETE?(state) and
              NEXT-CAT(state) is not a part of speech then
        PREDICTOR(state)
     elseif INCOMPLETE?(state) and
              NEXT-CAT(state) is a part of speech then
         SCANNER(state)
     else
        COMPLETER(state)
   end
  end
  return(chart)
```

The Earley Algorithm

procedure PREDICTOR($(A \rightarrow \alpha \bullet B\, \beta,\ [i,j])$)
    for each $(B \rightarrow \gamma)$ **in** GRAMMAR-RULES-FOR(B, *grammar*) **do**
        ENQUEUE($(B \rightarrow \bullet\, \gamma,\ [j,j])$, *chart*[$j$])
    end

procedure SCANNER($(A \rightarrow \alpha \bullet B\, \beta,\ [i,j])$)
    if B $\subset$ PARTS-OF-SPEECH(*word[j]*) **then**
        ENQUEUE($(B \rightarrow word[j],\ [j, j+1])$, *chart[j+1]*)

procedure COMPLETER($(B \rightarrow \gamma\ \bullet,\ [j,k])$)
    for each $(A \rightarrow \alpha \bullet B\, \beta,\ [i,j])$ **in** *chart*[j] **do**
        ENQUEUE($(A \rightarrow \alpha\, B \bullet \beta,\ [i,k])$, *chart[k]*)
    end

Earley: Pseudo-code Simplified

To make things easier to define, we will assume all strings end in \$ and that there is a special additional top-level symbol S' with rule $S' \to S\$$.

Parsing an input $x = x1 \cdots x_n\$$. S_i will be a state of Earley chart items with an ending point i.

Start with $S_0 = \{[S' \to \bullet S\$, 0, 0]\}$. Then, for $0 \leq i \leq n$ do:

1. Process each item $s \in S_i$ in order by applying to it the *single* applicable operation among:
   - Predictor (adds new items to S_i)
   - Completer (adds new items to S_i)
   - Scanner (adds new items to S_{i+1})
2. If $S_{i+1} = \emptyset$ *Reject* the input
3. If $i = n$ and $S_{n+1} = \{[S' \to S\$\bullet, 0, n+1]\}$ *Accept* the input

Parsing the Input

As with CYK we have formulated a recognizer. We can change it to a parser by adding backpointers so that each state knows where it came from.

Chart[1]	S12	*Verb* $\rightarrow$ *book* $\bullet$	[0,1]	Scanner
Chart[2]	S23	*Det* $\rightarrow$ *that* $\bullet$	[1,2]	Scanner
Chart[3]	S28	*Noun* $\rightarrow$ *flight* $\bullet$	[2,3]	Scanner
	S29	*Nominal* $\rightarrow$ *Noun* $\bullet$	[2,3]	(S28)
	S30	*NP* $\rightarrow$ *Det Nominal* $\bullet$	[1,3]	(S23, S29)
	S33	*VP* $\rightarrow$ *Verb NP* $\bullet$	[0,3]	(S12, S30)
	S36	*S* $\rightarrow$ *VP* $\bullet$	[0,3]	(S33)

Comparing Earley and CYK

- For such a simple example, there seems to be a lot of useless stuff in the chart.
- We are predicting phrases that aren't there at all!
- That's the flipside to the CYK problem.

Comparing Earley and CYK

- For such a simple example, there seems to be a lot of useless stuff in the chart.
- We are predicting phrases that aren't there at all!
- That's the flipside to the CYK problem.

Did we solve ambiguity?

Comparing Earley and CYK

- For such a simple example, there seems to be a lot of useless stuff in the chart.
- We are predicting phrases that aren't there at all!
- That's the flipside to the CYK problem.

Did we solve ambiguity? Both CYK and Earley may result in multiple S structures for the $[0, N]$ table entry. Of course, neither can tell us which one is 'right'.

The Asymptotic Complexity of Earley and CKY

- Both algorithms are cubic in n (length of string)
- CKY needs to construct $O(n^2)$ elements in the chart (in the worst-case), and processing each element to create it is $O(n)$, so it is $O(n^3)$ in total
- Earley also needs to construct $O(n^2)$ elements, and the COMPLETER operation takes $O(n)$ time. It could potentially run on $O(n^2)$ elements, so the complexity is again $O(n^3)$

More about Asymptotic Complexity of Earley

- The COMPLETER operation really takes $O(i^2)$ at iteration i
- For unambiguous grammars, Earley shows that the COMPLETER operation can take at most $O(i)$ time
- This means that the complexity for unambiguous grammars is $O(n^2)$
- There are also some specialised grammars for which the Earley algorithm takes $O(n)$ time

Connection between the Earley Algorithm and CKY

What happens if we run the Earley algorithm on a grammar in Chomsky normal form?

- This is essentially CKY with top-down filtering
- It will only create (completed) elements in the chart, if there is a left-most derivation that leads to that constituent

Summary

- The Earley algorithm uses dynamic programming to implement a top-down search strategy.
- Single left to right pass that fills chart with entries.
- Dotted rule represents progress in recognizing RHS of rule.
- Algorithm always moves forward, never backtracks to previous chart entry, once it has moved on.
- States are processed using PREDICTOR, COMPLETER, SCANNER operations.

Reading: Same as for Lecture 20
Next lecture: Resolving ambiguity using statistical parsing.