The CYK algorithm

L645 / B659
Fall 2015

Where we’re going

We want to talk about probabilistic parsing
• First step: see how symbolic parsing works & what data structures are needed
Today, we’ll assume some working knowledge of context-free grammars (CFGs)
• Next classes: more on CFGs, PCFGs, & working with PCFGs

An example grammar

Lexicon:
Vt → saw
Det → the
Det → a
N → dragon
N → boy
Adj → young

Syntactic rules:
S → NP VP
VP → Vt NP
NP → Det N
N → Adj N

Problem: Inefficiency of recomputing subresults

Two example sentences and their potential analysis:
(1) He [gave [the young cat] [to Bill]].
(2) He [gave [the young cat] [some milk]].

The corresponding grammar rules:
• VP → V_ditrans NP PP
• VP → V_ditrans NP NP

Regardless of the final sentence analysis, the ditransitive verb (gave) and its first object NP (the young cat) will have the same analysis
⇒ No need to analyze it twice

Solution: Chart Parsing (Memoization)

• Store intermediate results:
  a) completely analyzed constituents:
    well-formed substring table or (passive) chart (today=review of L545)
  b) partial and complete analyses:
    (active) chart (L545)
• Instead of recalculating that the young cat is an NP, we store that information
  – Dynamic programming: never go backwards
• All intermediate results need to be stored for completeness.
• All possible solutions are explored in parallel.

CFG Parsing: The Cocke Younger Kasami Algorithm

• Grammar has to be in Chomsky Normal Form (CNF), only
  – RHS with a single terminal: A → a
  – RHS with two non-terminals: A → BC
  – no ε rules (A → ε)
• A representation of the string showing positions and word indices:

  0 1 2 3 4 5 6
  the young cat saw the dragon

For example:
  0 1 young 2 cat 3 saw 4 the 5 dragon
The well-formed substring table (passive chart)

• The well-formed substring table, henceforth (passive) chart, for a string of length \( n \) is an \( n \times n \) matrix.
• The field \( (i,j) \) of the chart encodes the set of all categories of constituents that start at position \( i \) and end at position \( j \), i.e.
  \[
  \text{chart}(i,j) = \{ A \mid A \Rightarrow w_{i+1} \ldots w_j \}
  \]
• The matrix is triangular since no constituent ends before it starts.

Coverage Represented in the Chart

An input sentence with 6 words:
·0 w1 ·1 w2 ·2 w3 ·3 w4 ·4 w5 ·5 w6 ·6
Coverage represented in the chart:

Example for Coverage Represented in Chart

Example sentence:
·0 the ·1 young ·2 boy ·3 saw ·4 the ·5 dragon ·6
Coverage represented in chart:

Parsing with a Passive Chart

• The CKY algorithm is used, which:
  – explores all analyses in parallel,
  – in a bottom-up fashion, &
  – stores complete subresults
• This algorithm is used is to:
  – avoid left-recursion problem of top-down parsing
  – store partial analyses

Filling in the Chart

• We build all constituents that end at a certain point before we build constituents that end at a later point.

An Example for a Filled-in Chart

Input sentence:
·0 the ·1 young ·2 boy ·3 saw ·4 the ·5 dragon ·6

Chart:

0 \hspace{1cm} 1 \hspace{1cm} 2 \hspace{1cm} 3 \hspace{1cm} 4 \hspace{1cm} 5 \hspace{1cm} 6

0 \{Det\} {\} {\} {\} {\} {\} {\} {\} {\} {\} {\}\}
1 \{Adj\} {\} {\} {\} {\} {\} {\} {\} {\} {\} {\} {\}
2 \{\} {\} {\} {\} {\} {\} {\} {\} {\} {\} {\} {\}
3 \{V, N\} {\} {\} {\} {\} {\} {\} {\} {\} {\} {\} {\}
4 \{\} {\} {\} {\} {\} {\} {\} {\} {\} {\} {\} {\}
5 \{\} {\} {\} {\} {\} {\} {\} {\} {\} {\} {\} {\}

Filling in the Chart

for \( j := 1 \) to length(string)
\[ \text{lexical_chart_fill}(j-1, j) \]
for \( i := j - 2 \) down to 0
\[ \text{syntactic_chart_fill}(i, j) \]
The Complete CYK Algorithm

Input: start category $S$ and input string

$n := \text{length}(\text{string})$

for $j := 1$ to $n$
  $\text{chart}(j-1,j) := \{X \mid X \to \text{word}_j \in P\}$

for $i := j-2$ down to $0$
  $\text{chart}(i,j) := \{\}$
  for $k := i+1$ to $j-1$
    for every $A \to BC \in P$
      if $B \in \text{chart}(i,k)$ and $C \in \text{chart}(k,j)$ then
        $\text{chart}(i,j) := \text{chart}(i,j) \cup \{A\}$

Output: if $S \in \text{chart}(0,n)$ then accept else reject

How memoization helps

If we look back to the chart for the sentence the young boy saw the dragon:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>{Det}</td>
<td>{}</td>
<td>{NP}</td>
<td>{}</td>
<td>{}</td>
<td>{S}</td>
</tr>
<tr>
<td>1</td>
<td>{Adj}</td>
<td>{N}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
</tr>
<tr>
<td>2</td>
<td>{N}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
</tr>
<tr>
<td>3</td>
<td>{V, N}</td>
<td>{}</td>
<td>{VP}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
</tr>
<tr>
<td>4</td>
<td>{}</td>
<td>{Det}</td>
<td>{NP}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
</tr>
<tr>
<td>5</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
</tr>
</tbody>
</table>

- At cell (3,6), a VP is built by combining the V at (3,4) with the NP at (4,6), based on the rule $VP \to V NP$
- Regardless of further processing, that VP is never rebuilt

Chart for recovering parses

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Det</td>
<td>NP</td>
<td>(D,0.1)</td>
<td>(N,1.3)</td>
<td>S</td>
<td>(NP,0.3)</td>
</tr>
<tr>
<td>1</td>
<td>Adj</td>
<td>N</td>
<td>(A,1.2)</td>
<td>(N,2.3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>N</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
</tr>
<tr>
<td>3</td>
<td>V, N</td>
<td>{}</td>
<td>VP</td>
<td>(V,3.4)</td>
<td>(NP,4.6)</td>
<td>{}</td>
</tr>
<tr>
<td>4</td>
<td>{}</td>
<td>Det</td>
<td>NP</td>
<td>(D,4.5)</td>
<td>(N,5.6)</td>
<td>{}</td>
</tr>
<tr>
<td>5</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
<td>{}</td>
</tr>
</tbody>
</table>

From recognition to parsing

Extend chart to store in each field

- mother symbol (as before)
- daughters and their field numbers (i.e., backpointers to the structure)
Extending CYK to CFG

We can allow for rules of arbitrary RHS length by doing the following:

1. initialize each field $i$, $i+1$ with the categories from the terminal rules

2. for each rule $A \rightarrow \alpha \in P$:
   - check whether there are fields in the chart for which the symbols can be concatenated to $\alpha$ so that an uninterrupted sequence of words $i, j$ is covered
   - insert $A$ into field $i, j$

3. if $S$ (the start symbol) is in field $1, n$ ($n =$ number of words), then accept the sentence

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>D</td>
<td>NP</td>
<td></td>
<td></td>
<td>S</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(D,0,1,N,1,2)</td>
<td></td>
<td></td>
<td>(NP,0,2,VP,2,5)</td>
</tr>
<tr>
<td>1</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>V</td>
<td></td>
<td></td>
<td>VP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(V,2,3,NP,3,5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(V,2,3,NP,3,4;VP,4,5)</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>NP</td>
<td>N</td>
<td></td>
<td>NP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(N,3,4)</td>
<td></td>
<td></td>
<td>(D,3,4,N,4,5)</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>VP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(V,4,5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N, V</td>
</tr>
</tbody>
</table>