The CYK algorithm

L645
Autumn 2009

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
  b) partial and complete analyses: (active) chart
- In other words, instead of recalculating that the young cat is an NP, we’ll 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 $\epsilon$ rules ($A \rightarrow \epsilon$)
- A representation of the string showing positions and word indices:
  - $\cdot_0 w_1 \cdot_1 w_2 \cdot_2 w_3 \cdot_3 w_4 \cdot_4 w_5 \cdot_5 w_6 \cdot_6$
  - For example: the _1 young _2 boy _3 saw _4 the _5 dragon _6

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.
  - chart$(i,j) = \{A | A \Rightarrow^* w_{i+1} \ldots w_j\}$
- The matrix is triangular since no constituent ends before it starts.
Coverage Represented in the Chart

An input sentence with 6 words:

\[ w_0 \ w_1 \ w_2 \ w_3 \ w_4 \ w_5 \ w_6 \]

Coverage represented in the chart:

<table>
<thead>
<tr>
<th>FROM:</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>0–1</td>
<td>0–2</td>
<td>0–3</td>
<td>0–4</td>
<td>0–5</td>
<td>0–6</td>
</tr>
<tr>
<td>1</td>
<td>1–2</td>
<td>1–3</td>
<td>1–4</td>
<td>1–5</td>
<td>1–6</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2–3</td>
<td>2–4</td>
<td>2–5</td>
<td>2–6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3–4</td>
<td>3–5</td>
<td>3–6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4–5</td>
<td>4–6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5–6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example sentence:

\[ w_0 \ w_1 \ w_2 \ w_3 \ w_4 \ w_5 \ w_6 \]

Coverage represented in chart:

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

Parsing with a Passive Chart

- The CKY algorithm is used, which:
  - explores all analyses in parallel,
  - in a bottom-up fashion, &
  - stores complete subresults
- The reason this algorithm is used is to:
  - add top-down guidance (to only use rules derivable from start-symbol), but avoid left-recursion problem of top-down parsing
  - store partial analyses

An Example for a Filled-in Chart

Input sentence:

\[ w_0 \ w_1 \ w_2 \ w_3 \ w_4 \ w_5 \ w_6 \]

Coverage represented in chart:

<table>
<thead>
<tr>
<th>FROM:</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>{Det}</td>
<td>{NP}</td>
<td>{}</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>

Filling in the Chart

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

<table>
<thead>
<tr>
<th>FROM:</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>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

lexical_chart_fill(j-1, j)

- Idea: Lexical lookup. Fill the field \( j−1, j \) in the chart with the preterminal category dominating word \( j \).
- Realized as:

\[
\text{chart}(j−1, j) := \{ X | X \rightarrow \text{word}_j \in P \}
\]
The Complete CYK Algorithm

Input: start category $S$ and input string

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

for $j := 1$ to $n$

$$\text{chart}(j−1,j) := \{X \mid X \rightarrow \text{word}_j \in \text{P}\}$$

for $i := j−2$ down to 0

$$\text{chart}(i,j) := \{\}$$

for $k := i + 1$ to $j−1$

for every $A \rightarrow BC \in \text{P}$

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:

1 2 3 4 5 6
0 [Det] {} [NP] {} {} {} [S]
1 [Adj] [N] {} {} {} {}
2 [N] {} {} {} {}
3 [V, N] {} [VP]
4 [Det] [NP] {}
5 [N]

- 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 → V NP
- Regardless of further processing, that VP is never rebuilt

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 $A \rightarrow \alpha \in \text{P}$:
   - 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
## Ambiguous parses

*the boy saw her duck*

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