The CYK algorithm

L645
Autumn 2009

An example grammar

Lexicon:	Syntactic rules:
Vt → *saw*	S → NP VP
Det → *the*	VP → Vt NP
Det → *a*	NP → Det N
N → *dragon*	N → Adj N
N → *boy*	
Adj → *young*	

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$_{to}$
- 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: $A \rightarrow a$
  – RHS with two non-terminals: $A \rightarrow BC$
  – no ϵ 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: $\cdot_0$ the $\cdot_1$ young $\cdot_2$ boy $\cdot_3$ saw $\cdot_4$ the $\cdot_5$ dragon $\cdot_6$

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) $= \{A \mid A \Rightarrow^* w_{i+1} \ldots w_j\}$
- The matrix is triangular since no constituent ends before it starts.

Coverage Represented in the Chart

An input sentence with 6 words:

$$\cdot_0\ w_1\ \cdot_1\ w_2\ \cdot_2\ w_3\ \cdot_3\ w_4\ \cdot_4\ w_5\ \cdot_5\ w_6\ \cdot_6$$

Coverage represented in the chart:

TO:

FROM:		1	2	3	4	5	6
	0	0–1	0–2	0–3	0–4	0–5	0–6
	1		1–2	1–3	1–4	1–5	1–6
	2			2–3	2–4	2–5	2–6
	3				3–4	3–5	3–6
	4					4–5	4–6
	5						5–6

Example for Coverage Represented in Chart

Example sentence:

$\cdot_0$ the $\cdot_1$ young $\cdot_2$ boy $\cdot_3$ saw $\cdot_4$ the $\cdot_5$ dragon $\cdot_6$

Coverage represented in chart:

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

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:
$\cdot_0$ the $\cdot_1$ young $\cdot_2$ boy $\cdot_3$ saw $\cdot_4$ the $\cdot_5$ dragon $\cdot_6$

Chart:

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

Filling in the Chart

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

	1	2	3	4	5	6
0	**1**	3	6	10	15	21
1		**2**	5	9	14	20
2			**4**	8	13	19
3				**7**	12	18
4					**11**	17
5						**16**

for j := 1 to length($string$)
    lexical_chart_fill$(j-1, j)$
    for $i := j-2$ down to 0
        syntactic_chart_fill(i, j)

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:

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

syntactic_chart_fill(i,j)

- Idea: Perform all reduction steps using syntactic rules such that the reduced symbol covers the string from i to j.
- Realized as: $chart(i,j) = \left\{ A \;\middle|\; \begin{array}{l} A \to BC \in P, \\ i < k < j, \\ B \in chart(i,k), \\ C \in chart(k,j) \end{array} \right\}$
- Explicit loops over every possible value of k and every context free rule:

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

The Complete CYK Algorithm

Input: start category S and input $string$

  $n :=$ length($string$)

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

Output: if $S \in chart(0,n)$ 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 $\to$ 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)

Chart for recovering parses

	1	2	3	4	5	6
0	Det		NP (D,0,1) (N,1,3)			S (NP,0,3) (VP,3,6)
1		Adj	N (A,1,2) (N,2,3)			
2			N			
3				V, N		VP (V,3,4) (NP,4,6)
4					Det	NP (D,4,5) (N,5,6)
5						N

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 $\to \alpha \in$ P:
   - check whether there are fields in the chart for which the symbols can be concatenated to α 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

	1	2	3	4	5
0	D	NP (D,0,1;N,1,2)			S (NP,0,2;VP,2,5)
1		N			
2			V		VP (V,2,3;NP,3,5) (V,2,3;NP,3,4;VP,4,5)
3				NP (N,3,4) D, N	NP (D,3,4;N,4,5)
4					VP (V,4,5) N, V