CYK Algorithm - Step 3

Gideon Maillette de Buy Wenniger
Statistical Language Processing and Learning Lab at the Institute for Logic Language and Computation (ILLC), University of Amsterdam, the Netherlands

Contact:
gemdbw - at - gmail - dot -com

Step 3 : Adding probabilities computation to parsing. Maximum likelihood / Viterbi Algorithm illustrated

- Last week we saw how the chart is filled and trees are build bottom up. We now extend this by showing how the probabilities are included.
- In the next few slides we briefly show how the chart is filled and probabilities are computed using our running example grammar
- We use the same structure as used in the Standford sheets: (http://nlp.stanford.edu/courses/lsa354/SLoSP-2007-2.pdf) but with slightly different indexing our *chart[begin-inclusive][end-inclusive]* corresponds to their *score[begin-inclusive][end-exclusive]* notation. (Our notation is closer to the implementation: it directly corresponds to array entries in a 2-D array implementing a chart)

Chart Structure

The empty chart before filling, for the sentence "time flies like an arrow"

time	flies	like	an	arrow
chart[0][0]	*chart*[0][1]	*chart*[0][2]	*chart*[0][3]	*chart*[0][4]
	chart[1][1]	*chart*[1][2]	*chart*[1][3]	*chart*[1][4]
		chart[2][2]	*chart*[2][3]	*chart*[2][4]
			chart[3][3]	*chart*[3][4]
				chart[4][4]

Initialization

Add unary inferences for length one entries (initialization step):
e.g. apply rule $NN \rightarrow time$ in chart entry [0][0]:
New probability $= P(NN \rightarrow time) = 2/4$

time	flies	like	an	arrow
$\{NN\,\|\,\{[time, 2/4]_{Max}\}\}$	$chart[0][1]$	$chart[0][2]$	$chart[0][3]$	$chart[0][4]$
	$NNS\,\|\,\{[flies, 1]_{Max}\}$ $VBZ\,\|\,\{[flies, 1]_{Max}\}$	$chart[1][2]$	$chart[1][3]$	$chart[1][4]$
		$VBP\,\|\,\{[like, 1]_{Max}\}$ $IN\,\|\,\{[like, 1]_{Max}\}$	$chart[2][3]$	$chart[2][4]$
			$DT\,\|\,\{[an, 1]_{Max}\}$	$chart[3][4]$
				$NN\,\|\,\{[arrow, 2/4]_{Max}\}$

(perform the same steps for all length 1 chart entries)

TOP → S : 1	PP → IN NP : 1
S → NP VP : 1	NN → time : 2/4
NP → NN : 1/4	NN → arrow : 2/4
NP → NN NNS : 1/4	NNS → flies : 1
NP → DT NN : 2/4	VBP → like : 1
VP → VBZ PP : 1/2	IN → like : 1
VP → VBP NP : 1/2	DT → an : 1
PP → IN NP : 1	VBZ → flies : 1

Initialization - continued

We must continue applying the unary rules a second time, to be able to generate the two-step unary productions that appear at the leaf nodes:

e.g. apply rule $NP \rightarrow NN$ in chart entry [0][0]:

New probability $= P(NP \rightarrow NN) \times P_{Max}(NN_{[0][0]})$

$= 1/4 \times 2/4 = 1/8$

time	flies	like	an	arrow
NN: $\{[time, 2/4]_{Max}\}$; NP: $\{[NN_{[0][1]}, 1/8]_{Max}\}$	$chart[0][1]$	$chart[0][2]$	$chart[0][3]$	$chart[0][4]$
	NNS: $\{[flies, 1]_{Max}\}$; VBZ: $\{[flies, 1]_{Max}\}$	$chart[1][2]$	$chart[1][3]$	$chart[1][4]$
		VBP: $\{[like, 1]_{Max}\}$; IN: $\{[like, 1]_{Max}\}$	$chart[2][3]$	$chart[2][4]$
			DT: $\{[an, 1]_{Max}\}$	$chart[3][4]$
				NN: $\{[arrow, 2/4]_{Max}\}$

TOP → S : 1	PP → IN NP : 1
S → NP VP : 1	NN → time : 2/4
NP → NN : 1/4	NN → arrow : 2/4
NP → NN NNS : 1/4	NNS → flies : 1
NP → DT NN : 2/4	VBP → like : 1
VP → VBZ PP : 1/2	IN → like : 1
VP → VBP NP : 1/2	DT → an : 1
PP → IN NP : 1	VBZ → flies : 1

(perform the same steps for all length 1 chart entries)

Main loop after initialization

We now continue adding inferences for increasingly large chart entries...

e.g. apply rule $NP \rightarrow NN\ NNS$ in chart entry [0][1]:

New probability =

$P(NP \rightarrow NN\ NNS) \times P_{Max}(NN_{[0][0]}) \times P_{Max}(NNS_{[1][1]})$
$= 1/4 \times 2/4 \times 1 = 1/8$

time	flies	like	an	arrow
NN | {[time, 2/4]$_{Max}$} NP | {[$NN_{[0][0]}$, 1/8]$_{Max}$}	NP | {[$NN_{[0][0]}$, $NNS_{[1][1]}$, 1/8]$_{Max}$}	*chart*[0][2]	*chart*[0][3]	*chart*[0][4]
	NNS | {[flies, 1]$_{Max}$} VBZ | {[flies, 1]$_{Max}$}	*chart*[1][2]	*chart*[1][3]	*chart*[1][4]
		VBP | {[like, 1]$_{Max}$} IN | {[like, 1]$_{Max}$}	*chart*[2][3]	*chart*[2][4]
			DT | {[an, 1]$_{Max}$}	*chart*[3][4]
				NN | {[arrow, 2/4]$_{Max}$}

(Continue with all other length 2 chart entries, length 3 ... length n chart entries)

TOP → S : 1	PP → IN NP : 1
S → NP VP : 1	NN → time : 2/4
NP → NN : 1/4	NN → arrow : 2/4
NP → NN NNS : 1/4	NNS → flies : 1
NP → DT NN : 2/4	VBP → like : 1
VP → VBZ PP : 1/2	IN → like : 1
VP → VBP NP : 1/2	DT → an : 1
PP → IN NP : 1	VBZ → flies : 1

Some more marks

- Dealing with the unaries can be a special extra step that is applied at every iteration. This is the approach taken in the standford slides. This is arguably a cleainer solution, so you can choose to follow that approach as well, instead of making a special case for the length 1 chart entries and length n entries.
- Notice that we keep in every chart entry a list of labels, and every label has a list of inferences. These inferences have a probability, and 1 or 2 pointers to the labels (with associated chart entries) from which they have been built.
- For a label's list of inference we keep track of the one with the maximum probability. Here we denote this one with $[RightSideLabel1, RightSideLabel2, Prob]_{\mathbf{Max}}$ or $[RightSideLabel1]_{\mathbf{Max}}$ for the case of unary inferences.
- An essential part of the Viterbi part is updating(replacing) the **Max** inference when a better inference for the same label in the same chart entry is found.