Prob parsing assignment calculations

Jean Mark Gawron *

April 19, 2018

Let's assume that for any edge we have what you discovered in the last assignment, a category, a start point, and an endpoint. We will also assume a set of daughter records that captures all the ways in which that edge might have been built. So that will be:

E.cat	category of edge
E.start	starting index of edge
E.end	ending index of edge
E.dtrs	a set of daughter records (a set of pairs of edges)

Previously when considering an edge like *np(8,10)*, we thought of a daughter record as a triple like *(det,9,nbar)*, where *det* and *nbar* are the categories of the two daughter and *9* represents the index where the first daughter ends and the second begins. In this document, it will be useful to think of each daughter record as a pair of edges. The information captured in the triple is in the daughter edges: so for the *np(8,10)* example, the first daughter has the *cat* attribute *dt*, and the second the cat attribute *nbar*. The index 9 occurs as the end attribute of the first daaughter edge and also as the start attribute of the second. For lexical edges we will have:

E.cat	category of edge
E.start	starting index of edge
E.end	ending index of edge, always one more than e.start
E.dtrs	the empty set

*San Diego State University, Department of Linguistics and Oriental Languages, BAM 321, 5500 Campanile Drive, San Diego, CA 92182-7717, gawron@mail.sdsu.edu.

Then the recursion relation defining the Viterbi probability of an edge (Vit(E)):

$$\begin{array}{lcl} \text{Vit}(E) & = & \text{Max}_{\langle \text{D}_1, \text{D}_2 \rangle \in \text{E.dtrs}} \text{PProb}(\text{E}, \text{D}_1, \text{D}_2) \\ \text{PProb}(\text{E}, \text{D}_1, \text{D}_2) & = & \text{Vit}(\text{D}_1) * \text{Vit}(\text{D}_2) * \text{P}(\text{E.cat} \rightarrow \text{D}_1\text{.cat} \quad \text{D}_2\text{.cat}) \end{array}$$

Lexical edges are the degenerate case, sice they have no daughters. The Viterbi probability is simply:

$$\text{Vit}(E) \quad = \quad \text{P}(\text{E.cat} \rightarrow \text{input}[\text{E.start}]),$$

where *input* is the input list and *input[E.start]* is the word at index *E.start*.

The definitions make it clear that the the calculation must be performed bottom up. Construct all five complete parse trees for the given sentence and compute the Viterbi values for all the edges in a valid parse tree. In a Chomsky Normal form grammar, the number of edges used in a valid tree for a string of a given length is fixed, and it turns out that each of your five parse trees will have 19 nodes, and 5 * 19 = 95. But in fact you will have vastly fewer nodes to work on for this assignment, since a huge proportion of the edges will be shared among multiple parse trees. Many edges are shared by all five parse trees. So start with your lexical edges and work up.

Here's an example, which shows a useful format for presenting your work:

E	D_1	D_2	Vit(D_1)	Vit(D_2)	Rule	Vit(E)
np(8,10)	dt(8,9)	nbar(9,10)	.5	.01	.5	.0025
dt(8, 9)	Lexical					.5
nbar(9, 10)	Lexical					.01

When an edge is ambiguous and has multiple daughter records, the table rows should look like this:

E	D_1	D_2	Vit(D_1)	Vit(D_2)	Rule	Vit(E)
vp(2,10)	vbz(2,3)	np(3,10)				pprob value goes here
	X1(2,4)	pp(4,10)				pprob value goes here
	X2(2,7)	pp(7,10)				pprob value goes here
	X1(2,7)	pp(7,10)				pprob value goes here
				max		max of the four pprobs