COMP550 Natural Language Processing
Assignment 2

Jonathan Guymont

October 14, 2018

Question 1

(a) **MAP estimate.** The set of possible tag is {N, C, V, J} and the lexicon is {that, is, not, it, good}. The add-1 smoothed initial probabilities are given by

$$\pi_i = \frac{count(Q_1 = i) + 1}{|sentence| + |corpus|}$$

where $|lexicon| = 5$, $|corpus| = 4$ because we only consider beginning of sentence, and $count(Q_1 = i)$ is the number of times a sentence start by the tag i.

$$\pi_N = \frac{0+1}{4+4} = 1/8, \quad \pi_C = \frac{2+1}{4+4} = 3/8, \quad \pi_V = \frac{2+1}{4+4} = 3/8, \quad \pi_J = \frac{0+1}{4+4} = 1/8$$

The add-1 smoothed transition probabilities are given by

$$a_{ij} = \frac{count(Q_{t+1} = j, Q_t = i) + 1}{count(Q_t = i) + 1 * |tags|}$$

where $|tags| = 4$ is the number of different tags.

	N	C	V	J	$count(Q_t = i)$
N	2	0	3	1	6
C	2	0	0	0	2
V	4	0	1	0	5
J	0	0	0	0	0

Table 1: Transition count

$$a_{NN} = \frac{2+1}{6+4} = 3/10, \quad a_{NC} = \frac{0+1}{6+4} = 1/10, \quad a_{NV} = \frac{3+1}{6+4} = 4/10, \quad a_{NJ} = \frac{1+1}{6+4} = 2/10,$$

$$a_{CN} = \frac{2+1}{2+4} = 3/6, \quad a_{CC} = \frac{0+1}{2+4} = 1/6, \quad a_{CV} = \frac{0+1}{2+4} = 1/6, \quad a_{CJ} = \frac{0+1}{2+4} = 1/6,$$

$$a_{VN} = \frac{4+1}{5+4} = 5/9, \quad a_{VC} = \frac{0+1}{5+4} = 1/9, \quad a_{VV} = \frac{1+1}{5+4} = 2/9, \quad a_{VJ} = \frac{0+1}{5+4} = 1/9,$$

$$a_{VN} = \frac{4+1}{5+4} = 5/9, \quad a_{VC} = \frac{0+1}{5+4} = 1/9, \quad a_{VV} = \frac{1+1}{5+4} = 2/9, \quad a_{VJ} = \frac{0+1}{5+4} = 1/9,$$

$$a_{JN} = \frac{0+1}{0+4} = 1/4, \quad a_{JC} = \frac{0+1}{0+4} = 1/4, \quad a_{JV} = \frac{0+1}{0+4} = 1/4, \quad a_{JJ} = \frac{0+1}{0+4} = 1/4,$$

$$A = \begin{pmatrix} 0.3 & 0.1 & 0.4 & 0.2 \\ 0.5 & 1/6 & 1/6 & 1/6 \\ 5/9 & 1/9 & 2/9 & 1/9 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$$

	that	is	not	it	good	$count(Q_t = i)$
N	4	0	2	2	0	8
C	2	0	0	0	0	2
V	0	6	0	0	0	6
J	0	0	0	0	1	1

Table 2: Emissions count ($count(O_t = k, Q_t = i)$)

The MLE of the emissions probability is given by $b_{i,k} = count(O_t = k, Q_t = i)/count(Q_t = i)$. The $count(\cdot, \cdot)$ are shown in table 2. To smooth the emissions probability with add-1, we add 1 to the numerator of the MLE and we add 5 to the numerator. The matrix of emissions probability is given by

$$B = \begin{pmatrix} 5/13 & 1/13 & 3/13 & 3/13 & 1/13 \\ 3/7 & 1/7 & 1/7 & 1/7 & 1/7 \\ 1/11 & 7/11 & 1/11 & 1/11 & 1/11 \\ 1/6 & 1/6 & 1/6 & 1/6 & 2/6 \end{pmatrix}$$

(b) **Viterbi.** The probability in the first column are given by $P(O_1, Q_1) = P(Q_1)P(O_1|Q_1) = \pi_{Q_1} = \pi_{Q_1} b_{O_1,Q_1}$. For example, the first entry is given by

$$\delta_N(1) = \pi_N b_N(that) = 1/8 \cdot 3/10 = 3/80$$

The value in the second column are given by $\max_i P(Q_{t-1} = i, O_{t-1})P(Q_t = j|Q_{t-1} = i)P(O_t|Q_t = j) = \max_i \delta_i(t-1)a_{ij}b_j(O_t)$. For example, the first entry of the second column is given by

$$\begin{aligned} \delta_N(2) =& \max_i \delta_i(1) a_{iN} b_N(is) \\ =& \frac{1}{13} \max\{\frac{5}{104} \cdot \frac{3}{10}, \frac{9}{56} \cdot \frac{1}{2}, \frac{3}{88} \cdot \frac{5}{9}, \frac{1}{48} \cdot \frac{1}{4}\} \\ =& \frac{5}{809} \end{aligned} \tag{1}$$

	that	is	good
N	5/104	9/1456	5/6864
C	9/56	3/784	1/3696
V	3/88	3/176	1/2904
J	1/48	1/224	1/1584

Table 3: Trellis of the Viterbi algorithm

Thus the most likely tags are:

$$(that, C), (is, V), (good, N)$$

(c) **EM.** First, we reestimate B with add-1 smoothing so the word *bad* do not have zero probability to occur.

$$B = \begin{pmatrix} 5/14 & 1/14 & 3/14 & 3/14 & 1/14 & 1/14 \\ 3/8 & 1/8 & 1/8 & 1/8 & 1/8 & 1/8 \\ 1/12 & 7/12 & 1/12 & 1/12 & 1/12 & 1/12 \\ 1/7 & 1/7 & 1/7 & 1/7 & 2/7 & 1/7 \end{pmatrix}$$

The parameters Π and A do not change, since the new examples are completely untaged. The next step is to use viterbi to make prediction on the tags of the new sentence. Then we should use those taged examples to retrain the model.

The prediction for the examples (following the same methods as in part b)) are respectively $[(bad, C), (is, V), (not, N), (good, J)$ and $[(is, V), (it, N), (bad, V)]$. Now we can use those two tagged examples, add them to training set, and recompute the parameters Π, A, and B just like in part (a).