NLP Sheet 5 Sol

1. Given 3 POS tags (Noun: N, Modal: M, Verb: V) and the following dataset

Sequence & Labels
Mary(N) Jane(N) can(M) see(V) Will(N)
Spot(N) will(M) see(V) Mary(N)
Will(M) Jane(N) spot(V) Mary(N)?
Mary(N) will(M) pat (V) Spot(N)

Train an HMM and use it to predict the tags for "Will can sport Mary"

Solution

// Required is to also use a <s> and </s>, the latter especially going beyond the lecture

1. Training an HMM means to find the Emission (B) and Transition (A) Probability Matrices for the dataset:

- Recall
- Transition probability Matrix is $(N+1) \times (N+1)$ for N tags
- Emission Probability Matrix is $N \times V$ for N tags and V words
- In either these matrices and including the Bigram matrix from lecture 2 we build them so that each element contains $P(Col \mid Row)$
  - Count how many times the row and column happen in the dataset then divide by the sum of the row in the end.

Transition Probability Matrix:

	Noun	Modal	Verb	</s>
<s>	$3/4$	$1/4$	0	0
Noun	$1/9$	$3/9$	$1/9$	$4/9$
Modal	$1/4$	0	$3/4$	0
Verb	$4/4$	0	0	0

- Modal occurred after Noun 3 times
- Noun occurred after <s> (i.e., at the beginning of the sentence) 3 times
- </s> occurred after Noun (i.e., noun at the end of the sentence) 4 times
- Observe that the sum of the column counts should be the number of times this POS appeared in the dataset (to check)

Emission Probability Matrix:

	Mary	Jane	Can	See	Will	Spot	Pat
Noun	$4/9$	$2/9$	0	0	$1/9$	$2/9$	0
Modal	0	0	$1/4$	0	$3/4$	0	0
Verb	0	0	0	$2/4$	0	$1/4$	$1/4$

- Will occurred 1 time as a Noun and 3 times as a Modal
- Observe that the sum of the row counts should be the number of times this POS appeared in the dataset (to check)

2. To predict the tags for "Will can spot Mary" we need to setup the Viterbi matrix

1. Setup the Viterbi Matrix
   - Using B matrix as initial values means we don't have to worry about multiplying emission later and we can pretend 0 cells don't exist.
   - Multiply the first column by the $P(tag| < s >)$ probability

	Will	Can	Spot	Mary
Noun	$\frac{1}{9} \times \frac{3}{4} = \frac{1}{12}$	0	$2/9$	$4/9$
Modal	$\frac{3}{4} \times \frac{1}{4} = 3/16$	$1/4$	0	0
Verb	0	0	$1/4$	0

2. Finish the Viterbi Matrix
   - Update all the cells starting from the second column with their Viterbi probabilities
   - Multiply the cell's value in the t_{th} column and j_{th} row by

$$Max(V_{t-1}[1]t * P(t_j|t_0), V_{t-1}[2]t * P(t_j|t_0), .., V_{t-1}[N]t * P(t_j|t_0))$$

	Will	Can	Spot	Mary
Noun	$\frac{1}{9} \times \frac{3}{4} = \frac{1}{12}$		$\frac{2}{9} * \frac{1}{144} * \frac{1}{4} = \frac{1}{2592}$	$\frac{4}{9} * Max\left(\frac{1}{2592} * \frac{1}{9}, \frac{1}{768} * 1\right) = \frac{1}{1728}$
Modal	$\frac{3}{4} \times \frac{1}{4} = 3/16$	$\frac{1}{4} * Max\left(\frac{1}{12} * \frac{3}{9}, \frac{3}{16} * 0\right) = 1/144$		
Verb			$\frac{1}{4} * \frac{1}{144} * \frac{3}{4} = \frac{1}{768}$	

- For example, to find the value for $t = 2$ and $j = 2$ we considered multiplying $P(Modal \mid Noun) = 3/9$ and $P(Modal \mid Modal) = 0$ by the corresponding Viterbi probabilities form the last column.
- In the second column, we dropped Max as the previous one had one value only

3. Find the Sequence of Tags
   - Consider the maximum from the last column and then backtrack from that maximum to find the optimal tag sequence

- Here there is one element and the decisions that lead to it are
  - The 4th column chose Verb for the previous
  - Verb in the 3rd column chose the only option, Modal for the previous
  - Modal in the 2nd column chose Noun for the previous
- The tag sequence is thus N, M, V, N which is clearly correct (English-wise)

Sidenote:

Suppose there were more than one element in the last column, then before taking their Max, we have to multiply each by its $P(</s> \mid pos)$ (you can add one more column with one cell for that if you wish, as you will see below)

2. Given the following A and B matrices and the partially computed Viterbi matrix shown on the right that use negative log probabilities

	PL	PN	PP	VB	EOS
BOS	11	2	3	4	19
PL	17	3	2	5	7
PN	5	4	3	1	8
PP	12	4	6	7	9
VB	3	2	3	3	7

	hen	vilar	ut
PL	17	17	4
PN	3	19	19
PP	19	19	3
VB	19	8	19

		hen	vilar	ut	
BOS	0				
PL		A	27	21	
PN		5	B	35	
PP		22	27	20	
VB		23	14	36	
EOS					C

Find A, B, C and the most probable tag Sequence

- Recall that taking the log of a product of probabilities means we will add the logs instead and the log doesn't change the max value
- Using negative logs changes the max into min (we still add the negative logs instead of multiplying probabilities). This means we will need to take the min over the previous column rather than max

1. To find A
   - ⇒ We would usually multiply $P(hen|PL) * P(PL| < s >)$
   - ⇒ Instead, we will add their given negative logs: $A \ = \ 17 + 11 = 28$

2. To find B
   - ⇒ We would usually $P(vilar|PN) * \max\{previous\ cell * P(PN|previous\ POS)\}$ over all previous cells
   - ⇒ Instead, we will: $B \ = \ 19 + \min(28 + 3, 5 + 4, 22 + 4, 23 + 2) = 28$ where 3, 4, 4, 2 are the NLogs for $P(PN|PL), P(PN|PN), P(PN|PP), P(PN|VB)$ respectively.

3. To find C
   - ⇒ We would usually $max\ \{previous\ cell * P(</s>|previous\ POS)\}$ over all final cells
   - ⇒ Instead, we will: $C \ = \ \min(21 + 7, 35 + 8, 20 + 9, 36 + 7) = 28$ where 7, 8, 9, 7 are the NLogs for $P(</s> |PL), P(</s> |PN), P </s> |PP), P(</s> |VB)$ respectively.

4. To find the optimal tag sequence

   Since we didn't carry out the computations in each cell ourselves, we may need to recompute to know which POS from the previous column was chosen for max

   - ⇒ The 4^{th} column chose PL for the previous (by our formula for C, PL yielded the min)
   - ⇒ PL in the 3^{rd} column chose must have chosen VB as $21 = 4 + min(27 + 17, 28 + 5, 27 + 12, 14 + 3)$
   - ⇒ VB in the 2^{nd} column must have chosen PN as $14 \ = \ 8 + min(28 + 5, 5 + 1, 22 + 7, 23 + 3)$
   - ⇒ Thus, the predicted tag sequence is <s> PN VB PL </s>

3. Show the (IO/BIO/BIOES) NER tags for *"John Alex is going to New York after having an appointment at the Artificial Intelligence Corporate in Rome"*

Recall, BIOES=Begin Tag, Inside Tag, Outside Tag, End Tag, Span Tag

Start by highlighting the entities in the sentence and write O for any non entity and I for any entity

John	Alex	is	going	to	New	York	after	having	an	Appoint.	at
I	I	O	O	O	I	I	O	O	O	O	O
the	Artif.	Intel.	Cop.	in	Rome	IO Tagging					
O	I	I	I	O	I						

To generalize this into BIO, let the 1ST word of any entity be B

John	Alex	is	going	to	New	York	after	having	an	Appoint.	at
B	I	O	O	O	B	I	O	O	O	O	O
the	Artif.	Intel.	Cop.	in	Rome	BIO Tagging					
O	B	I	I	O	B						

To generalize this into BIOES, let the last word of any entity be E and any entity of one word only be S

John	Alex	is	going	to	New	York	after	having	an	Appoint.	at
B	E	O	O	O	B	E	O	O	O	O	O
the	Artif.	Intel.	Cop.	in	Rome	BIO Tagging					
O	B	I	E	O	S						

4. Which of the following feature templates can be used in linear chain CRF

Recall, linear chain CRF requires that each local feature for a given index depends on at most

⇒The index itself

⇒Any terms of the input sequence x

⇒Only $y_i$and y_{i-1}

Templates	Can be used?
$< \boldsymbol{y_i, x_i, y_{i-1}} >$	Yes
$< \boldsymbol{y_{i-2}, y_{i-1}, y_i} >$	No, it uses y_{i-2}
$< \boldsymbol{x_{i-2}, y_i, x_{i+2}} >$	Yes, recall x_{i+k} for any k works since its accessing terms from the input sequence
$< \boldsymbol{y_{i-1}, x_{i-1}, x_{i-2}, x_{i+1}} >$	Yes, same note as above.

Thank you <3