CS114B (Spring 2021) Written Assignment 2
Sequence Labeling

Due April 6, 2021

1 Hidden Markov Models

(You may find the discussion in Chapter A of the Jurafsky and Martin book helpful.)

You are given the following short sentences, tagged with parts of speech:

```
Alice/NN admired/VB Dorothy/NN
Dorothy/NN admired/VB every/DT dwarf/NN
Dorothy/NN cheered/VB
every/DT dwarf/NN cheered/VB
```

1. Train a hidden Markov model on the above data. Specifically, compute the initial probability distribution π:

y_1	NN	VB	DT
$P(y_1)$			

The transition matrix **A**:

$P(y_i \mid y_{i-1})$		y_i		
		NN	VB	DT
y_{i-1}	NN			
	VB			
	DT			

And the emission matrix **B**:

$P(x_i \mid y_i)$		y_i		
		NN	VB	DT
	Alice			
	admired			
	Dorothy			
x_i	every			
	dwarf			
	cheered			
	<UNK>			

Note that you should account for the unknown word <UNK>, but you don't need to account for the start symbol <S> or the stop symbol </S>. There are ways to train the probabilities of <UNK> from the training set, but for this assignment, you can simply let count(<UNK>, y) = 1 for all tags y (before smoothing). You should use add-1 smoothing on all three tables.

2. Use the forward algorithm to compute the probability of the following sentence:

   Alice cheered

   As part of your answer, you should fill in the forward trellis below:

	Alice	cheered
NN		
VB		
DT		

3. Use the Viterbi algorithm to compute the best tag sequence for the following sentence:

   Goldilocks cheered

   As part of your answer, you should fill in the Viterbi trellis below. You should also keep track of backpointers, either using arrows or in a separate table.

	Goldilocks	cheered
NN		
VB		
DT		

2 Structured Perceptrons

(You may find the discussion in Chapter 7.5.1 of the Eisenstein book helpful.)

Suppose we are given the following weight matrix $\mathbf{\Theta}$:

$\mathbf{\Theta}$	$y_i = \ldots$		
	NN	VB	DT
$y_{i-1} =$ <S>	-0.3	-0.7	0.3
$y_{i-1} =$ NN	-0.7	0.3	-0.3
$y_{i-1} =$ VB	-0.3	-0.7	0.3
$y_{i-1} =$ DT	0.3	-0.3	-0.7
$x_i =$ Alice	-0.3	-0.7	0.3
$x_i =$ admired	0.3	-0.3	-0.7
$x_i =$ Dorothy	-0.3	0.3	-0.7
$x_i =$ every	-0.7	-0.3	0.3
$x_i =$ dwarf	0.3	-0.7	-0.3
$x_i =$ cheered	-0.7	0.3	-0.3

(If you think this matrix looks like π, $\mathbf{A}$, and $\mathbf{B}$ stacked on top of each other, you are right! Note that we don't need to account for the unknown word `<UNK>`, and for simplicity, we will ignore the bias term.)

1. Suppose we are given the following training sentence:

   `Alice/NN admired/VB Dorothy/NN`

   (a) Use the Viterbi algorithm to compute the best tag sequence. As part of your answer, you should fill in the Viterbi trellis below. You should also keep track of backpointers, either using arrows or in a separate table.

	Alice	admired	Dorothy
NN			
VB			
DT			

   (b) Update the weight matrix. Use a constant learning rate $\eta = 1$.

2. Suppose we are given the following testing sentence:

   `Alice cheered`

   Use the Viterbi algorithm to compute the best tag sequence. Again, you should fill in the Viterbi trellis below, and keep track of backpointers.

	Alice	cheered
NN		
VB		
DT		

Submission Instructions

Please submit your solutions (in PDF format) to LATTE.