CS114 (Spring 2020) Written Assignment 2
Language Modeling and Sequence Labeling

Due March 10, 2020

1 N-grams

You are given the following short sentences:

```
Alice admired Dorothy
Dorothy admired every dwarf
Dorothy cheered
every dwarf cheered
```

1. Train the following n-gram language models on the above data:

   (a) Unigram, unsmoothed

   (b) Bigram, unsmoothed

   (c) Bigram, add-1 smoothing

   (d) Bigram, interpolation ($\lambda_1 = \lambda_2 = 1/2$)

   Some notes:

   - As in HW1, it is recommended that you create (conditional) probability tables such as those shown below for a unigram model:

w_n	`Alice`	`admired`	`Dorothy`	`every`	`dwarf`	`cheered`	`</S>`	`<UNK>`
$P(w_n)$								

And for your bigram models:

$P(w_n \mid w_{n-1})$		w_n							
		Alice	admired	Dorothy	every	dwarf	cheered	</S>	<UNK>
w_{n-1}	<S>								
	Alice								
	admired								
	Dorothy								
	every								
	dwarf								
	cheered								
	<UNK>								

- Note that both unigram and bigram models must account for w_n being the stop symbol </S>. Additionally, bigram models must account for w_{n-1} being the start symbol <S>. Include <S> and </S> in your counts just like any other token.
- Also note that both unigram and bigram models must account for the unknown word <UNK>. There are ways to train the probabilities of <UNK> from the training set, but for this assignment (and PA3/PA4), you can simply set all the <UNK>-related counts equal to 1. In other words, if you make a table of word counts, you can fill the <UNK> column (and row, if applicable) with 1's.

2. For each of the above language models, compute the probability of the following sentences:

   `Alice cheered`

   `Goldilocks cheered`

2 Hidden Markov Models

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

You are given the same short sentences as before, this time 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:

t_1	`NN`	`VB`	`DT`
$P(t_1)$			

The transition matrix A:

$P(t_n \mid t_{n-1})$		t_n		
		`NN`	`VB`	`DT`
t_{n-1}	`NN`			
	`VB`			
	`DT`			

And the emission matrix B:

$P(w_n \mid t_n)$		w_n						
		`Alice`	`admired`	`Dorothy`	`every`	`dwarf`	`cheered`	`<UNK>`
t_n	`NN`							
	`VB`							
	`DT`							

Note that as before, you should account for the unknown word `<UNK>`, but you don't need to account for `<S>` or `</S>`. 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`

   In other words, 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`

   Again, 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		

Submission Instructions

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