NLP - Fall 2017 - Midterm Exam

(1) **Language Models**

Mrs. Malaprop would like to build a spelling corrector focused on the particular problem of *there* vs *their*. The idea is to build a model that takes a sentence as input, for example:

1. He saw their football in the park
2. He saw their was a football in the park

For each instance of *their* or *there* Mrs. Malaprop wants to predict whether the true spelling should be *their* or *there*. So for sentence (1) the model should predict *their*, and for sentence (2) the model should predict *there*. Note that for the second example the model would correct the spelling mistake in the sentence. Mrs. Malaprop recently took some NLP classes so she wants to use a language model for this task. Given a language model $p(w_1, \ldots, w_n)$, return the spelling that gives the highest probability under the language model. So for example for the second sentence we would implement the rule: replace *there* with *their* and vice versa and compare the language model scores:

If p(*He saw there was a football in the park*) >
p(*He saw their was a football in the park*)
Then return(*there*)
Else return(*their*)

Mrs. Malaprop decides to use an unigram model: $p(w_1, \ldots, w_n) = \prod_{i=1}^{n} q(w_i)$ where $q(w_i) = \frac{\text{Count}(w_i)}{N}$ and $N = \sum_w \text{Count}(w)$. Count$(\cdot)$ returns the number of times a word was seen in the corpus and N is the sum of counts for all words in the corpus. Assume $N = 10,000$ and Count(*there*) = 110 and Count(*their*) = 50. Also assume that for every word w in the vocabulary Count$(w) > 0$.

a. What does the Mrs. Malaprop rule return for *He saw their was a football in the park*?

b. Is the Mrs. Malaprop rule a good solution to the *their* versus *there* problem? Say yes or no and give a short and precise one sentence justification for your answer.

(2) **Hidden Markov Models**

The probability model $P(t_i \mid t_{i-2}, t_{i-1})$ is provided below where each t_i is a part of speech tag, e.g. $P(\text{D} \mid \text{N}, \text{V}) = \frac{1}{3}$. Also provided is $P(w_i \mid t_i)$ that a word w_i has a part of speech tag t_i, e.g. $P(\text{flies} \mid \text{V}) = \frac{1}{2}$. The part of speech tag definitions are: `bos` (*begin sentence marker*), N (*noun*), V (*verb*), D (*determiner*), P (*preposition*), `eos` (*end of sentence marker*).

$P(t_i \mid t_{i-2}, t_{i-1})$	t_{i-2}	t_{i-1}	t_i
1	bos	bos	N
$\frac{1}{2}$	bos	N	N
$\frac{1}{2}$	bos	N	V
$\frac{1}{2}$	N	N	V
$\frac{1}{2}$	N	N	P
$\frac{1}{3}$	N	V	D
$\frac{1}{3}$	N	V	V
$\frac{1}{3}$	N	V	P
1	V	D	N
1	V	V	D
1	N	P	D
1	V	P	D
1	P	D	N
1	D	N	eos

$P(w_i \mid t_i)$	t_i	w_i
1	D	an
$\frac{2}{5}$	N	time
$\frac{2}{5}$	N	arrow
$\frac{1}{5}$	N	flies
1	P	like
$\frac{1}{2}$	V	like
$\frac{1}{2}$	V	flies
1	eos	eos
1	bos	bos

a. Consider a Jelinek-Mercer style interpolation smoothing scheme for $P(w_i \mid t_i)$:

$$P_{jm}(w_i \mid t_i) = \Lambda[t_i] \cdot P(w_i \mid t_i) + (1 - \Lambda[t_i]) \cdot P(w_i)$$

Λ is an array with a value $\Lambda[t_i]$ for each part of speech tag t_i, such that $0 \leq \Lambda[t_i] \leq 1$. Provide a condition on Λ that must be satisfied to ensure that P_{jm} is a well-defined probability model.

b. Provide a Hidden Markov Model (*hmm*) that uses the trigram part of speech probability $P(t_i \mid t_{i-2}, t_{i-1})$ as the transition probability $P_{hmm}(s_j \mid s_k)$ and the probability $P(w_i \mid t_i)$ as the emission probability $P_{hmm}(w_j \mid s_j)$.

Important: Provide the *hmm* in the form of two tables as shown below. The first table contains transitions between states in the *hmm* and the transition probabilities and the second table contains the words emitted at each state and the emission probabilities. Do not provide entries with zero probability.

from-state s_k	to-state s_j	$P(s_j \mid s_k)$

state s_j	emission w	$P(w \mid s_j)$

Hint: In your *hmm* the state $\langle N, \texttt{eos} \rangle$ will have emission of word `eos` with probability 1 and will not have transitions to any other states.

c. Based on your *hmm* constructed in 2b. what is the state sequence that would be provided by the Viterbi algorithm for the following input sentence:

`bos bos` time flies like an arrow `eos`