Final for 6.864

Name:

30	30	40	20	45

Good luck!

Part #1 30 points

Question 1 (10 points)

This question concerns training IBM Model 2 for statistical machine translation. Assume that we have a bilingual corpus of English sentences $\mathbf{e}$ paired with sentences in a foreign language $\mathbf{f}$. Each English sentence is a string formed from the words $\{the, dog, ate, cat, a, banana\}$, while each foreign sentence is a string formed from the words $\{athe, adog, aate, acat, aa, abanana\}$. The set of training examples is as follows:

Training Example	English sentence $\mathbf{e}$	Foreign Sentence $\mathbf{f}$
1	*the dog ate*	*aate adog athe*
2	*the cat ate the banana*	*abanana athe aate acat athe*
3	*a dog ate a cat*	*acat aa aate adog aa*
4	*a cat ate*	*aate acat aa*

Recall that in IBM model 2 we have translation parameters of the form $P(f|e)$ where f is a foreign word, and e is an English word; and in addition we have alignment parameters of the form $P(a_i = j|l, m)$ where l is the length of the English sentence, m is the length of the foreign sentence, and this parameter denotes the probability of word i in the foreign string being aligned to word j in the English string.

Question: Specify parameter values for IBM model 2 that result in $P(\mathbf{f}|\mathbf{e}) = 1$ for all training examples shown in the table above.

Question 2 (10 points)

Now assume that the training set is as follows:

Training Example	English sentence **e**	Foreign Sentence **f**
1	*the dog ate*	*athe adog aate*
2	*the cat ate the banana*	*athe acat athe abanana aate*
3	*a dog ate a cat*	*aa adog aa acat aate*
4	*a cat ate*	*aa acat aate*

Can you define IBM Model 2 parameters such that $P(\mathbf{f}|\mathbf{e}) = 1$ for all examples in this training set? If your answer is yes, define the parameters of the model. If your answer is no, give a justification for your answer in 200 words or less.

Question 3 (10 points)

Now assume that the training set is as follows:

Training Example	English sentence **e**	Foreign Sentence **f**
1	*the dog ate*	*athe adog aate*
2	*the dog ate the dog*	*athe adog athe adog aate*
3	*a pink dog ate bananas*	*aa apink adog abananas aate*

Can you define IBM Model 2 parameters such that $P(\mathbf{f}|\mathbf{e}) = 1$ for all examples in this training set? If your answer is yes, define the parameters of the model. If your answer is no, give a justification for your answer in 200 words or less.

Part #2 30 points

Question 4 (30 points)

We will define a new model for statistical machine translation as follows:

$$P_{M3}(\mathbf{f}, \mathbf{a}|\mathbf{e}, m) = \prod_{j=1}^{m} T(f_j|e_{a_j})D(a_j|a_{j-1}, l, m)$$

Here $\mathbf{f}$ is a French sequence of words $f_1, f_2, \ldots f_m$, $\mathbf{a}$ is a sequence of alignment variables $a_1, a_2, \ldots a_m$, and $\mathbf{e}$ is an English sequence of words $e_1, e_2, \ldots e_l$. Note that the probability P_{M3} is conditioned on the identity of the English sentence, $\mathbf{e}$, as well as the length of the French sentence, m. The parameters of the model are translation parameters of the form $T(f|e)$ and alignment parameters of the form $D(a_j|a_{j-1}, l, m)$. We assume that a_0 is defined to be 0. Note that in contrast to IBM Model 2, the alignment parameters are now modified to be conditioned upon the previous alignment variable.

Assume we have $m = 6$, and $e =$ the dog ate the cat. The model then defines a distribution $P_{M3}(\mathbf{f}, \mathbf{a}|\text{the dog ate the cat}, 6)$, as well as a distribution over French strings,

$$P_{M3}(\mathbf{f}|\mathbf{e}, m) = \sum_{\mathbf{a}} P_{M3}(\mathbf{f}, \mathbf{a}|\mathbf{e}, m)$$

Define an HMM which defines a distribution $P_{HMM}(\mathbf{f})$ over French strings which is identical to the distribution $P_{M3}(\mathbf{f}|\mathbf{e}, m)$. Your HMM should have N states, an set of output symbols Σ that is the set of all possible French words, and parameters of the following types:

- π_j for $j = 1 \ldots N$ is the probability of choosing state j as the initial state.
- $a_{j,k}$ for $j = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, N$ is the probability of transitioning from state j to state k.
- $b_j(o)$ for $j = 1, 2, \ldots, N$ and $o \in \Sigma$ is the probability of emitting symbol o from state j.

This HMM will be slightly different from the HMMs seen in class, in that there is *no final state*, and the HMM is used to define a distribution over state/symbol sequences of length 6 only. For example, the state sequence $\langle 1, 2, 1, 3, 4, 2 \rangle$ paired with the output symbol sequence *le chat aime le chien bleu* would have probability

$$\pi_1 a_{1,2} a_{2,1} a_{1,3} a_{3,4} a_{4,2} b_1(le) b_2(chat) b_1(aime) b_3(le) b_4(chien) b_2(bleu)$$

You should describe how all of the parameters in the HMM model can be written in terms of the translation model parameters $T(f|e)$ and $D(a_j|a_{j-1}, l, m)$. Hint: your HMM should have 5 states, where the states correspond to the 5 positions in the English string *the dog ate the cat*.

Part #3

40 points

We define an HMM with 3 states $\{1, 2, 3\}$. We take state 3 to be the final state. The HMM has a set of two output symbols, $\Sigma = \{p, q\}$. The HMM has the following parameters:

- π_j for $j = 1 \ldots 3$ is the probability of choosing state j as the initial state.
- $a_{j,k}$ for $j = 1, 2$ and $k = 1, 2, 3$ is the probability of transitioning from state j to state k.
- $b_j(o)$ for $j = 1, 2$ and $o \in \Sigma$ is the probability of emitting symbol o from state j.

We are now going to use EM to estimate the parameters of the model. As training data, we have a single string, $x =$ pq. If $\mathcal{Y}$ is the set of possible state sequences for string x in the HMM, our goal is to maximize

$$L(\bar{\theta}) = \log P(x|\bar{\theta}) = \sum_{y \in \mathcal{Y}} \log P(x, y|\bar{\theta})$$

with respect to $\bar{\theta}$. Here $\bar{\theta}$ is a set of parameters in the model.

Question 5 (10 points)

For the input string pq, there are four possible state sequences under the HMM: $\langle 1\ 1\ 3\rangle$, $\langle 1\ 2\ 3\rangle$, $\langle 2\ 1\ 3\rangle$, $\langle 2\ 2\ 3\rangle$ (note that each of these state sequences ends in 3, the final state). The HMM model defines a joint distribution $P(x, y)$ over input strings x and state sequences y. Given that $x = \text{pq}$, write expressions for the probabilities $P(\text{pq}, y)$ for the four possible state sequences as a function of the parameters of the model.

Question 6 (30 points)

We are now going to derive the EM updates. Assume that we have a *current* set of parameters π_j, $a_{j,k}$, and $b_j(o)$ as described above. Assume again that we have $x = \text{pq}$ as the only string in the training data. The parameter values after the EM updates will be denoted as π'_j, $a'_{j,k}$ and $b'_j(o)$.

Write expressions for the new parameter values π'_1, $a'_{1,2}$, and $b'_2(p)$ as a function of the current values π_j, $a_{j,k}$ and $b_j(o)$.

Note: we do *not* expect or want you to use the forward-backward algorithm to derive the expressions for the updates. You should be able to use the expressions from the previous question (for the probabilities of the four possible state sequences) to derive the EM updates.

Part #4 — 20 points

The following algorithm is the pseudo-code from lecture for parsing with a PCFG. The algorithm finds the maximum probability for any tree when given as input a sequence of words $w_1, w_2, \ldots, w_n$. There are K non-terminals in the grammar, and we assume that the non-terminal 1 is the "start" symbol, which must always be a the root of the parse tree. The value stored in $\pi[1, n, 1]$ is then the highest probability for any tree spanning the words that is rooted in the start symbol.

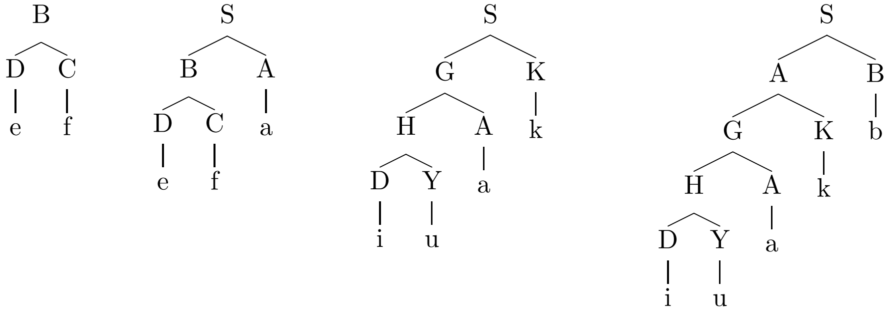
Initialization:

```
For i = 1 ... n, k = 1 ... K
    $\pi[i, i, k] = P(N_k \to w_i | N_k)$
```

Main Loop:

```
For $length = 1 \ldots (n-1)$, $i = 1 \ldots (n - 1ength)$, $k = 1 \ldots K$
    $j \leftarrow i + length$
    $max \leftarrow 0$
    For $s = i \ldots (j-1)$,
    For $N_l, N_m$ such that $N_k \to N_l N_m$ is in the grammar
        $prob \leftarrow P(N_k \to N_l N_m) \times \pi[i, s, l] \times \pi[s+1, j, m]$
        If $prob > max$
            $max \leftarrow prob$
    $\pi[i, j, k] = max$
```

Now assume that we want to find the maximum probability for any *left-branching* tree for a sentence. Here are some example left-branching trees:



It can be seen that in left-branching trees, whenever a rule of the form `X -> Y Z` is seen in the tree, then the non-terminal `Z` must directly dominate a terminal symbol.

Question 7 20 points Give new pseudo-code for a parsing algorithm that returns the maximum probability for any left-branching tree underlying a sentence $w_1, w_2, \ldots, w_n$.

Part #5 45 points

In this question, we will consider the task of building a single document summarizer. Given a text consisting of sentences $s_1, s_2, \ldots s_n$, we generate its summary by extracting a subset of salient sentences. We also have to ensure that extracted sentences form a coherent text. Assume that $sal(s_i)$ is a score indicating how good s_i is as a summary sentence, and $adj(s_i, s_j)$ is a score indicating how good it is to have sentence s_i immediately followed by sentence s_j in a summary.

Question 8 (25 points)

Assume that sentences in a summary follow the order of sentences in the original text. That is, a summary is a list of sentences $s_{i_1}, s_{i_2}, \ldots s_{i_k}$ such that $1 \leq i_1 < i_2 < \cdots < i_k \leq n$. The score of a summary $s_{i_1}, s_{i_2}, \ldots s_{i_k}$ is defined as

$$\sum_{j=1 \ldots k} sal(s_{i_j}) + \sum_{j=1 \ldots (k-1)} adj(s_{i_j}, s_{i_{j+1}})$$

The input to the algorithm is an input text and the desired summary length k (where $k < n$) measured by the number of extracted sentences. Describe a dynamic programming algorithm that selects a subset of k ($k < n$) input sentences which maximize the criterion shown above.

(Hint: The dynamic programming table is of the form $\pi[i, j]$, where $i = \{1 \ldots n\}$, and $j = \{1 \ldots k\}$.)

Question 9 (20 points)

Now, we will modify the way we are computing the overall salience score and modeling coherence. The salience score of a summary is the sum of salience scores ($sal(s_i)$) of the selected sentences and non-salience scores (1- $sal(s_i)$) of the remaining sentences. Instead of using $adj(s_i, s_j)$, we will use a score $rel(s_i, s_j)$ that captures the degree of relatedness between *any pair* of sentences s_i and s_j. Thus, the score of a summary $s_{i_1}, s_{i_2}, \ldots s_{i_k}$ is defined as

$$\sum_{j=1\ldots k} sal(s_{i_j}) + \sum_{s \in \{s_1 \ldots s_n\} - \{s_{i_1} \ldots s_{i_k}\}} (1 - sal(s)) + \sum_{j<r} rel(s_{i_j}, s_{i_r})$$

.

Our goal is to select a subset of k input sentences that simultaneously optimize salience and coherence scores (as defined above). Provide an ILP formulation that optimizes this criterion.

In your formulation, indicator variables a_i and a_{ij} equal 1 when a sentence or a pair of sentences are included in a summary. Assume that $\forall i, j \; a_i, a_{i,j} \in \{0, 1\}$. Your solution should contain the following parts:

a Formulate an optimization criterion:

b Formulate the length constraint that specifies that exactly k sentences are selected:

c Formulate constraints that establish relations between the individual and the pairwise indicator variables.