Midterm, COMS 4705

Name:

15	20	20	20

Good luck!

Part #1 15 points

Question 1 (5 points) We define a PCFG where non-terminal symbols are $\{S, A, B\}$, the terminal symbols are $\{a, b\}$, and the start non-terminal (the non-terminal always at the root of the tree) is S. The PCFG has the following rules:

S → A A	0.6
S → A B	0.4
A → A B	0.7
A → a	0.2
A → b	0.1
B → A B	0.9
B → a	0.05
B → b	0.05

For the input string *aab* show two possible parse trees under this PCFG, and show how to calculate their probability.

Question 2 (5 points) Consider the following lexicalized context-free grammar:

$$
\begin{array}{|l|}
\hline
\text{S(saw)} \rightarrow_2 \text{NP(Mary) VP(saw)} \\
\text{VP(saw)} \rightarrow_1 \text{V(saw) NP(tool)} \\
\text{NP(tool)} \rightarrow_2 \text{DT(the) NBAR(tool)} \\
\text{NBAR(tool)} \rightarrow_2 \text{NBAR(car) NBAR(tool)} \\
\text{NBAR(tool)} \rightarrow_2 \text{NBAR(metal) NBAR(tool)} \\
\text{NBAR(car)} \rightarrow_2 \text{NBAR(metal) NBAR(car)} \\
\text{NP(Mary)} \rightarrow \text{Mary} \\
\text{V(saw)} \rightarrow \text{saw} \\
\text{DT(the)} \rightarrow \text{the} \\
\text{NBAR(metal)} \rightarrow \text{metal} \\
\text{NBAR(car)} \rightarrow \text{car} \\
\text{NBAR(tool)} \rightarrow \text{tool} \\
\hline
\end{array}
$$

Show one valid parse tree under the grammar for the sentence *Mary saw the metal car tool*.

(Note: you may need to use space on the following blank page.)

(Page intentionally left blank.)

Question 3 (5 points) Recall that a PCFG defines a distribution $p(t)$ over parse trees t. For any sentence s, if we define $\mathcal{T}(s)$ to be the set of valid parse trees for the sentence s, the probability of the sentence under the PCFG is

$$p(s) = \sum_{t \in \mathcal{T}(s)} p(t)$$

Now consider the following PCFG:

S → NP VP	1.0
VP → V NP	1.0
NP → John	0.6
NP → Mary	0.4
V → saw	1.0

In the space below, write down the rules and parameters of a **lexicalized** PCFG in Chomsky Normal Form that gives the same distribution $p(s)$ over sentences as the PCFG shown above.

Part #2 20 points

For the following two questions, write TRUE or FALSE below the question. **PLEASE GIVE JUSTIFICATION FOR YOUR ANSWERS: AT MOST 50% CREDIT WILL BE GIVEN FOR ANSWERS WITH NO JUSTIFICATION.**

For all questions in this section we assume as usual that a language model consists of a vocabulary $\mathcal{V}$, and a function $p(x_1 \ldots x_n)$ such that for all sentences $x_1 \ldots x_n \in \mathcal{V}^\dagger$, $p(x_1 \ldots x_n) \geq 0$, and in addition $\sum_{x_1 \ldots x_n \in \mathcal{V}^\dagger} p(x_1 \ldots x_n) = 1$. Here $\mathcal{V}^\dagger$ is the set of all sequences $x_1 \ldots x_n$ such that $n \geq 1$, $x_i \in \mathcal{V}$ for $i = 1 \ldots (n-1)$, and $x_n = \text{STOP}$.

We assume that we have a bigram language model, with

$$p(x_1 \ldots x_n) = \prod_{i=1}^{n} q(x_i | x_{i-1})$$

The parameters $q(x_i|x_{i-1})$ are estimated from a training corpus using a discounting method, with discounted counts

$$c^*(v, w) = c(v, w) - \beta$$

where $\beta = 0.5$.

We assume throughout this question that all words seen in any test corpus are in the vocabulary $\mathcal{V}$, and each word in any test corpus is seen at least once in the training corpus.

Question 4 (4 points) True or False? For any test corpus, the perplexity under the language model will be less than ∞.

Question 5 (4 points) True or False? (3 points): For any test corpus, the perplexity under the language model will be at most $N + 1$, where N is the number of words in the vocabulary $\mathcal{V}$.

Question 6 (4 points) Now consider a bigram language model where for every bigram (v, w) where $w \in \mathcal{V}$ or $w = \text{STOP}$,

$$q(w|v) = \frac{1}{N+1}$$

where N is the number of words in the vocabulary $\mathcal{V}$.

True or False? For any test corpus, the perplexity under the langauge model will be equal to $N + 1$.

Question 7 (4 points) Recall that an HMM defines a joint distribution over sentences $x_1 \dots x_n$ and tag sequences $y_1 \dots y_{n+1}$ where $y_{n+1} = \text{STOP}$,

$$p(x_1 \dots x_n, y_1 \dots y_{n+1}) = \prod_{i=1}^{n+1} q(y_i | y_{i-1}) \prod_{i=1}^{n} e(x_i | y_i)$$

It also defines a distribution over sentences as

$$p(x_1 \dots x_n) = \sum_{y_1 \cdots y_{n+1}} p(x_1 \dots x_n, y_1 \dots y_{n+1})$$

True or false? For any bigram language model defining a distribution over sentences $p(x_1 \dots x_n)$, there is a bigram HMM that defines exactly the same distribution over sentences.

Question 8 (4 points) Recall that an HMM defines a joint distribution over sentences $x_1 \dots x_n$ and tag sequences $y_1 \dots y_{n+1}$ where $y_{n+1} = \text{STOP}$,

$$p(x_1 \dots x_n, y_1 \dots y_{n+1}) = \prod_{i=1}^{n+1} q(y_i|y_{i-1}) \prod_{i=1}^{n} e(x_i|y_i)$$

It also defines a distribution over sentences as

$$p(x_1 \dots x_n) = \sum_{y_1 \cdots y_{n+1}} p(x_1 \dots x_n, y_1 \dots y_{n+1})$$

True or false? For any **trigram** language model defining a distribution over sentences $p(x_1 \dots x_n)$, there is a bigram HMM that defines exactly the same distribution over sentences.

Part #3 20 points

Question 9 (10 points) In the box below, complete a version of the CKY parsing algorithm that takes as input a sentence $x_1 \ldots x_n$, and returns the number of parse trees for $x_1 \ldots x_n$ that have probability greater than 0.

In the algorithm you may use the following definition of the function $h(\alpha \rightarrow \beta)$:

$$
\begin{aligned}
h(\alpha \rightarrow \beta) &= 1 \text{ if } \alpha \rightarrow \beta \text{ is in the set of rules in the PCFG, and } q(\alpha \rightarrow \beta) > 0 \\
&= 0 \text{ otherwise}
\end{aligned}
$$

Input: a sentence $s = x_1 \ldots x_n$, a PCFG $G = (N, \Sigma, S, R, q)$.

Initialization:
For all $i \in \{1 \ldots n\}$, for all $X \in N$,

$$\pi(i, i, X) =$$

Algorithm:

- For $l = 1 \ldots (n-1)$
  - For $i = 1 \ldots (n-l)$
    - Set $j = i + l$
    - For all $X \in N$, calculate

$$\pi(i, j, X) =$$

Output: Return $\pi(1, n, S)$

Question 10

(10 points) In the box below, complete a dynamic programming algorithm that takes as input an integer n, and a probabilistic context-free grammar G, and returns the maximum probability under the PCFG for any tree that has exactly n words.

Hint: the algorithm fills in values for

$$\pi(i, X)$$

for all $i \in \{1 \ldots n\}$, and for all non-terminals X. The value for $\pi(i, X)$ should be the maximum probability for any parse tree with X at the root with exactly i words.

Input: a sentence $s = x_1 \ldots x_n$, a PCFG $G = (N, \Sigma, S, R, q)$ in Chomsky normal form where N is a set of non-terminals, Σ is the set of words, S is the start symbol, R is the set of rules in the grammar, and q is the set of rule parameters.

Initialization:
For all $X \in N$,

$$\pi(1, X) \quad =$$

Algorithm:

- For $i = 2 \ldots n$
  - For all $X \in N$, calculate

$$\pi(i, X) =$$

Output: Return $\pi(n, S)$

Part #4 20 points

Consider a bigram HMM tagger, where the joint probability of an input sentence $x_1 \ldots x_n$ and a tag sequence $y_1 \ldots y_{n+1}$ where $y_{n+1} = \text{STOP}$ is

$$p(x_1 \ldots x_n, y_1 \ldots y_{n+1}) = \prod_{i=1}^{n+1} q(y_i | y_{i-1}) \prod_{i=1}^{n} e(x_i | y_i)$$

The bigram HMM tagger defines a function from sentences $x_1 \ldots x_n$ to tag sequences $y_1 \ldots y_{n+1} = f(x_1 \ldots x_n)$ as follows:

$$f(x_1 \ldots x_n) = \arg \max_{y_1 \cdots y_{n+1}} p(x_1 \ldots x_n, y_1 \ldots y_{n+1})$$

Question 11 (10 points) Assume that we have an HMM with vocabulary $\mathcal{V} = \{a, b\}$ and a set of possible tags $\mathcal{K} = \{A, B\}$. We would like to build an HMM such that

$$\begin{aligned}
f(a) &= \text{A STOP} \\
f(aa) &= \text{A A STOP} \\
f(aaa) &= \text{A A A STOP} \\
\ldots & \\
f(b) &= \text{B STOP} \\
f(bb) &= \text{B B STOP} \\
f(bbb) &= \text{B B B STOP} \\
\ldots &
\end{aligned}$$

In other words if the input sentence consists of one or more a's, the output of the tagger should be a sequence of all A's. If the input sentence consists of one or more b's, the output of the tagger should be all B's. For sentences that contain both symbols a and b you do not need to worry about the behaviour of the tagger.

In the space below, write down the parameters of the HMM such that it implements the function $f(\ldots)$ described above.

Question 12 (10 points) Again assume that we have an HMM with vocabulary $\mathcal{V} = \{a, b\}$ and a set of possible tags $\mathcal{K} = \{A, B\}$. Now assume that we would like to build an HMM such that $f(x_1 \ldots x_n) = y_1 \ldots y_{n+1}$ satisfies

$$\text{if } x_i = a \text{ for all } i = 1 \ldots n \text{ then } y_i = A \text{ for all } i = 1 \ldots n$$

$$\text{if there exists some } i \text{ such that } x_i = b, \text{ then } y_i = B \text{ for all } i = 1 \ldots n$$

For example we have

$$
\begin{aligned}
f(a) &= \text{A STOP} \\
f(aa) &= \text{A A STOP} \\
f(aaa) &= \text{A A A STOP} \\
\ldots & \\
f(b) &= \text{B STOP} \\
f(ab) &= \text{B B STOP} \\
f(ba) &= \text{B B B STOP} \\
f(baa) &= \text{B B B STOP} \\
\ldots &
\end{aligned}
$$

In other words if the input sentence consists of only a's, the output of the tagger should be a sequence of all A's. If the input sentence contains at least one b, the output of the tagger should be all B's.

In the space below, write down the parameters of the HMM such that it implements the function $f(\ldots)$ described above.

(Page intentionally left blank.)

(Page intentionally left blank.)