CSC 248/448 Midterm

Closed Book **March 16, 2000**

1. Quick Questions (5 points each for a total of 40 points)

a. If the probability of the word "dog" occurring is .01, and the probability of the word "food" is also .01, and the probability of the sequence "dog food" is also .005, do the two words occur independently? Why? What is the probability of the word "food" given that the last word was "dog"?

b. If P(X=a) = .5, P(X=b) = .25, and P(X=c) = .25, what is the entropy of random variable X? What is the cross entropy of X with a corpus consisting of a b a b c?

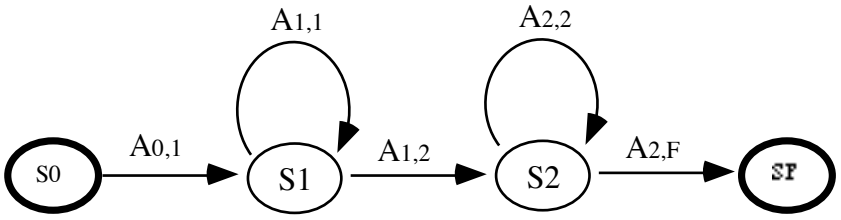
c. What is a probability distribution that would have a lower cross entropy based on the same corpus in part (b)? Is there be a distribution with a lower cross entropy yet? Why or why not?

d. Say we want to build a bigram model of a language with a vocabulary of 100 symbols from a corpus of 5000 elements. Would you use the MLE, Laplace or Lidstone estimator. Give a specific problem with the two approaches that you wouldn't use.

e. Briefly describe the Held-out estimator technique. Why do you get different estimates that simply just training on the combined data? Why would we expect it to be better?

f. What's a Markov Model? Define the Markov Property. What's the difference between a Markov Model and a Hidden Markov Model? Is a standard part-of-speech tagging model a Markov Model or an HMM.

g. Given the HMM;



A (transition matrix)	S1	S2	SF
S0	1.0	0	0
S1	.66	.33	0
S2	0	.84	.16

B (output prob)	U	C	V	S
S1	.75	.16	.08	.01
S2	.18	.27	.36	.18

What is the probability of being in state S2 after seeing the output UUU?

h. Define the forward and backward probabilites and describe what role they play in the Baum-Welch ("forward-backward") algorithm from training HMMs. (In other words, how are they used, and how does this help make the algorithm efficient).

2. (25 points total)

A recent expedition to Mars has discovered an intelligent life form that speaks a simple language. The scientists on the expedition discovered the following properties of the language:

there are two classes of words, one we call “nouns” and the other “verbs”.
all sentences begin with a noun and end with a verb
there are never two verbs in a row.

Unfortunately, on the way back, most of the data about the language is lost. So you end up knowing only one sentence in the language, namely

eep eep urp oop urp

Going on the assumption that this is a typical sentence in the language, you decide to try and explore the properties of the language further.

a) Draw an HMM model that captures the three known constraints on the language given above. Use one node to represent nouns and another to represent verbs.
b) Starting by using the uniform distribution for the transition and output probabilities, what would the first revised set of output probabilities using the forward-backward training algorithm. (You don’t necessarily need to hand-simulate the algorithm here as the number of possible paths is very small).
c) Do one more iteration of the algorithm yielding revised transition and output probabilities.
d) If we ran many iterations of the forward-backward algorithm on this model, what path through the HMM would become more and more likely? Why?
e) Based on your analysis, what would you predict is the most common 5 word sentence in the language?

3. (15 points total)

In applications such as information retrieval, it is often useful to extract out multi-word phrases that act like a unit. For instance, if we want to find articles about New York City on the web, searching for the pattern “new york city” will give much better results that searching for documents involving “new”, “york” and “city” individually and intersecting the answers. Such common phrases in a language are called **collocations**. This question concerns issues in finding useful collocations in text. We will restrict ourselves to two-word collocations such as “New York”, “ship captain”, “white house” and “digital computers”.

a. One approach to finding collocations would be to compute the bigram probabilities from a document and pick those bigrams with the highest probability of occurring. Why is this NOT a good measure. What type of answers would we expect to get using this technique.
b. Define a better measure for identifying collocations (be specific – i.e., give a formula), and argue why this avoids the pitfalls of the simple approach in question (a).
c. Say there are two words, A and B, that occur only once in a corpus, and they occur together in sequence A B. Would your measure in part (b) classify A B as a likely collocation? Is this justified? If you see a problem, how might you refine your measure to improve its performance on scarce data.