Name: ________________

CS4442/9542: Artificial Intelligence II
Winter 2013: Quiz 2 Solution

Instructions:

Show all the work you do. Use the back of the page, if necessary. Calculators are allowed, laptops, cell phones, or any other communication devices are not allowed. This is an open notes exam. However, the sample quiz solution is not allowed.

1. (10%) Suppose our corpus is the following text:

   I liked Herman. Herman was a hippopotamus. And Herman thought about his friend the Mouse. I thought it made a lot of sense, for Herman the hippo to jump our fence. Now, Herman followed me to my house, where he stayed about the pool, and made no noise, like a mouse.

   Ignore punctuation, capitalization (i.e. "To" and "to" should be treated as the same word) and treat the corpus as one continuous sentence.

   (a) (5%) Using unigram model, which of the following sentences is more likely under ML estimation and why? You do not have to compute the exact probabilities of each sentence, but may do so, if you wish.

   - Herman liked to jump
   - I liked the pool

   Solution:

   C(Herman) = 5, C(liked) = 1, C(to) = 2, C(jump) = 1.
   C(I) = 2, C(liked) = 1, C(the) = 3, C(pool) = 1.

   The combined unigram counts for the first sentence are higher, so the probability of the first sentence is higher.

   (b) (5%) Now use the bigram model. Which of the following two sentences are more likely under the ML estimation and why? Again, you do not have to compute the exact probabilities.

   - Herman was a pool mouse
   - I thought Herman was our mouse

   Solution:

   The bigram "a pool" is missing in the training set, so probability of the first sentence is zero. The bigram "was our" is missing from the training set, so the probability of the second sentence is zero. So none of them is likely. I actually messed up this quesion, I meant to make only one of the sentences to be zero probability.

2. (30%) Our corpus is the following text:

   Herman was a mouse. Herman lives in a house with a mouse.

   Assume the vocabulary size $V = 10$. Use a bigram model and Good-Turing method for estimating the probability of the unseen bigrams. Ignore punctuation. For the observed bigrams, use probabilities based on ML estimation. You do not have to renormalize the numbers to get the true probabilities, but if you want to, you may do so. What is the probability of the following bigrams. In your answer, you can leave expressions not calculated out completely, that is something like $2/30 - 1/6$ is fine.

   - *mouse was*
   - *Herman lives*
   - *a mouse*

   Solution: Let us count the bigram first:

   C(Herman was) = 1, C(was a) = 1, C(a mouse) = 2, C(mouse Herman) = 1, C(Herman lives) = 1, C(lives in) = 1, C(in a) = 1, C(a house) = 1, C(house with) =1, C(with 1) = 1.

   From the above counts, $N_1 = 9$ and $N_2 = 1$. There are total of 10^2 bigrams possible, but only 10 are observed. So $N_0 = 10^2 - 10 = 90$. Remember that we take the training data set size to be the number of words, so $N = 12$.

   We estimate probability of unseen bigrams according to Good-Turing. The bigram "mouse was" is unobserved, so

$$P_G T(mouse\ \ was) = (0+1)\frac{N_1}{N \cdot N_0} = \frac{9}{12 \cdot 90} = \frac{1}{120}$$

   The other two bigrams are observed, so we use ML estimation for them.

$$P(Herman\ \ lives) = \frac{C(Herman\ \ lives)}{N} = \frac{1}{12}$$

$$P(a\ \ mouse) = \frac{C(a\ \ mouse)}{N} = \frac{2}{12} = \frac{1}{6}$$

3. (30%)

Suppose we have a sentence *To play pay*, and possible POS tags are as follows:

To: PREP CONJ	Play: NOUN VERB	pay: NOUN VERB

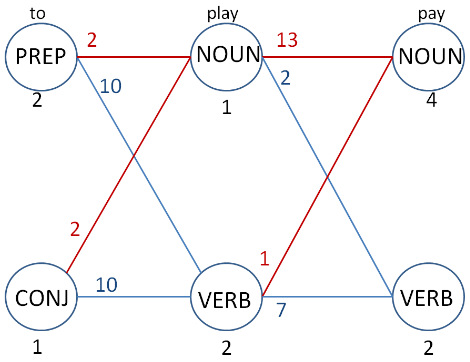
Also, the log-likelihoods are as follows:

L(to|PREP) = 2	L(play|NOUN) = 1	L(pay|NOUN)= 4
L(to|CONJ) = 1	L(play|VERB) = 2	L(pay|VERB) = 2

L(VERB|PREP) = 10	L(NOUN|NOUN) = 13
L(NOUN|PREP) = 2	L(VERB | NOUN) = 2
L(VERB|CONJ) = 10	L(NOUN|VERB) = 1
L(NOUN|CONJ) = 2	L(VERB|VERB) = 7

(a) (20%) For the given sentence, build the graph for solving the POS tagging problem via dynamic programming, like the one used in lecture notes. Show all the connections and the weight of all connections and nodes.

Solution:



(b) (10%) Tag the sentence. You can do it by observation or by running dynamic programming. If you do it by observation, justify with a couple of words. If you do dynamic programming, show all the details.

Solution: It can be easilty done by observation for this graph, but I will run the full DP algorithm:

Initialization:

C(to,prep) = 2, P(to,prep) = null; C(to,conj)= 1, P(to,conj) = null

Iteration 1:

C(play,noun) = 1+ min$\{C(to, prep) + 2, C(to, conj) + 2\}$= 3, P(play,noun) = (to,conj)
C(play,verb) = 2+ min$\{C(to, prep) + 10, C(to, conj) + 10\}$= 11, P(play,verb) = (to,conj)

Iteration 2:

C(pay,noun) = 4+ min$\{C(play, noun) + 13, C(play, verb) + 1\}$= 12, P(pay,noun) = (play,verb)
C(pay,verb) = 2+ min$\{C(play, noun) + 2, C(play, verb) + 7\}$= 5, P(pay,verb) = (play,noun)

The smallest cost path is 5, going through (pay,verb). Tracing back from (pay,verb) through the parent array "P" the tagging is "conj noun verb"

4. (30%)

(a) (20%)

Suppose inverted index has 3 terms and 3 documents, and the term/document counts are as follows:

$$doc_1 = \begin{bmatrix} term_1 & term_2 & term_3 \\ 3 & 1 & 2 \end{bmatrix}, doc_2 = \begin{bmatrix} term_1 & term_2 & term_3 \\ 4 & 3 & 1 \end{bmatrix},$$

$$doc_3 = \begin{bmatrix} term_1 & term_2 & term_3 \\ 1 & 2 & 1 \end{bmatrix}.$$

Use term counts for the weights. User query is "$term_1$ $term_2$", i.e. it has one occurrence of $term_1$, and one occurrence of $term_2$. Using the cosine similarity, rank all the documents according to their similarity to the query, where rank 1 means the highest similarity, and rank 3 means the lowest similarity. Show all work.

Solution:

Let us first normalize the documents:

$$d_1 = [\frac{3}{\sqrt{14}}; \frac{1}{\sqrt{14}}; \frac{2}{\sqrt{14}}]$$

$$d_2 = [\frac{4}{\sqrt{26}}; \frac{3}{\sqrt{26}}; \frac{1}{\sqrt{26}}]$$

$$d_3 = [\frac{1}{\sqrt{6}}; \frac{2}{\sqrt{6}}; \frac{1}{\sqrt{6}}]$$

The query is $q = [1 \ \ 1 \ \ 0]$. Normalizing it for length (it is actually not necessary for relative ranking, but still let's do it):

$$q = [\frac{1}{\sqrt{2}}; \frac{1}{\sqrt{2}}; 0]$$

Now, cosign similarity is:

$$cos(d_1, q) = d_1 \cdot q = \frac{3}{\sqrt{14}} \cdot \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{14}} \cdot \frac{1}{\sqrt{2}} = \frac{4}{\sqrt{28}} \approx 0.76$$

$$cos(d_2, q) = d_1 \cdot q = \frac{4}{\sqrt{26}} \cdot \frac{1}{\sqrt{2}} + \frac{3}{\sqrt{26}} \cdot \frac{1}{\sqrt{2}} = \frac{7}{\sqrt{52}} \approx 0.97$$

$$cos(d_3, q) = d_1 \cdot q = \frac{1}{\sqrt{6}} \cdot \frac{1}{\sqrt{2}} + \frac{2}{\sqrt{6}} \cdot \frac{1}{\sqrt{2}} = \frac{3}{\sqrt{12}} \approx 0.89$$

Therefore the ranking is d_2, d_3, d_1.

(b) (10%) Suppose inverted index has 3 terms and 3 documents, and the term/document counts are as follows:

$$doc_1 = \begin{bmatrix} term_1 & term_2 & term_3 \\ 100 & 10 & 1000 \end{bmatrix}, doc_2 = \begin{bmatrix} term_1 & term_2 & term_3 \\ 10 & 0 & 100 \end{bmatrix},$$

$$doc_3 = \begin{bmatrix} term_1 & term_2 & term_3 \\ 10000 & 10 & 0 \end{bmatrix}.$$

Suppose you decide to use tfxidf term weighting.

- What is the weight for term 2 in document 2?
- What is the weight for term 3 in document 2?

You don't have to work out the expressions completely, you can leave answer in the format $\frac{40}{3}log(4/5)$.

Solution:

In case $tf_{td} = 0$, the weight of term t in document d is defined as 0. Otherwise, it is defined as:

$$w_{td} = (1 + \log tf_{td} \times \log (N/df_t)$$

Since the count of term 2 in document 2 is 0, $w_{22} = 0$.

The count of term 3 in document 2 is 100. The document frequencyof term 3 is 2. The total number of documents is 3. Therefore,

$$w_{32} = (1 + \log 100 \times \log (3/2) = 3 \log (3/2)$$