CS 585 Natural Language Processing
Fall 2004
Homework 3: EM and Smoothing

Out: Tue, October 12, 2004
In: Thu, October 21, 2004

1 Word sense disambiguation

What is word sense disambiguation? Give two uses for it. Describe (in no more than two sentences each) two methods for word sense disambiguation.

2 Smoothing I

Given training data counts of $a = 3, b = 1, c = 1, d = 0, e = 0, f = 0, g = 0$, what is the Laplace estimate for $P(a)$ and $P(f)$? Given these same training data counts and also validation data counts $a = 2, b = 1, c = 0, d = 0, e = 1, f = 0, g = 0$, what is the held-out estimate of $P(a)$ and $P(f)$? Why might the held-out estimator be better?

3 Smoothing II

In the absolute discounting model of smoothing, all non-zero MLE frequencies are discounted by a constant amount δ:

Absolute discounting: If $C(w_n|w_1...w_{n-1}) = r$,

$$P_{abs}(w_n|w_1...w_{n-1}) = \begin{cases} (r-\delta)/N & \text{if } r > 0 \\ \frac{(V-N_0)\delta}{N_0 N} & \text{otherwise} \end{cases} \quad (1)$$

Recall that V is the size of the vocabulary, N is the total number of times $w_1...w_{n-1}$ has been seen, and N_0 is the number of distinct words that did not follow the context $w_1...w_{n-1}$.

In linear discounting, the estimated count of seen words is discounted by a certain percentage, for example $\alpha = 5\%$:

Linear discounting: If $C(w_n|w_1...w_{n-1}) = r$,

$$P_{lin}(w_n|w_1...w_{n-1}) = \begin{cases} (1-\alpha)r/N & \text{if } r > 0 \\ \alpha/N_0 & \text{otherwise} \end{cases} \quad (2)$$

Show that absolute discounting defines a probability model. That is, show that $\sum_{w_n \in V} P_{abs}(w_n|w_1...w_{n-1}) = 1$ and discuss what conditions are needed on δ. Next, show linear discounting also defines a probability model.

4 Smoothing and Cross-Entropy

This question requires you to use the same data (David Copperfield) as Question 4 in Homework 1, using the preprocessing procedure described there. After the preprocessing step, you may use Perl or any other language (including Unix commands) to answer this question, but you should mention what tools you used.

Find the first two chapters of David Copperfield), and calculate the probability distribution over words in the first chapter of text using Laplace smoothing (where the total vocabulary is the union of the vocabulary in both chapters). Now calculate the perplexity of the second chapter using this word distribution as the language model.

Now use absolute discounting with λ=0.75, and calculate the perplexity again. Which language model is better?

Extra credit: Try some other language models (for example, even using bi-grams) and larger sets of text.

5 Information gain

This question requires you to again use the David Copperfield data. You also need to download a copy of Tom Sawyer from `http://www.gutenberg.net/dirs/7/74/74.txt` and preprocess it the same way.

Let random variable W_{apple} indicate whether a word drawn at random from a book the word "apple." Let random variable B take on values indicating either *Tom Sawyer* or *David Copperfield*.

Using a programming language of your choice, find the 15 phrases with the highest information gain $I(W_{\text{foo}}, B)$ for all words "foo". Turn in a list that looks like

```
word_1  infogain_1
word_2  infogain_2
        ...
word_15 infogain_15
```

6 Expectation-maximization

Consider the following EM recipe for Naive Bayes:

Initialization: Set $P(c_i|d_k)$ and $P(c_i)$ to have uniform distributions, and set $P(w_j|c_i)$ to be random values (such that they are proper probability distributions).

E-step: Using the current parameters, set

$$P(c_i|d_k) = \frac{P(c_i)P(d_k|c_i)}{P(d_k)}$$

M-step: Using current estimates of $P(c_i|d_k)$, calculate the maximum-likelihood estimates for parameters $P(w_j|c_i)$:

$$P(w_j|c_i) = \frac{1+\sum_{d_k \in D} Count(w_j,d_k) \cdot P(c_i|d_k)}{|V|+\sum_{t=1}^{|V|} \sum_{d_k \in D} Count(w_t,d_k) \cdot P(c_i|d_k)}$$

Suppose there are 10 new articles $d_1 ... d_{10}$ over the alphabet {shot, money, bank, power, interest} and each is talking about either $c_1 = basketball$ or $c_2 = banks$. The document labels and the frequency of each word in the articles are as follows:

document	topic	count(shot)	count(money)	count(bank)	count(power)	count(inte
d_1	bank	0	3	5	2	1
d_2	bank	1	5	0	1	1
d_3	bank	0	1	7	0	5
d_4	bank	2	2	10	1	10
d_5	bank	1	0	2	4	7
d_6	bank	0	8	6	3	3
d_7	bank	2	1	5	2	2
d_8	basketball	10	0	3	1	0
d_9	basketball	3	0	1	0	2
d_{10}	basketball	5	2	0	2	0

Implement the NaiveBayes EM algorithm and run it for 5 iterations. Show the values of $P(c_i)$, $P(w_i|c_j)$, and $P(c_i|d_j)$ at each iteration. At the beginning of the algorithm, initialize each $P(c = basketball|d_j)$ and $P(c = bank|d_j)$ to be uniform.