Information Extraction from Speech and Text

Homework # 4

Due March 5, 2013.

Review Chapter 4 from *Statistical Methods for Speech Recognition* by Frederick Jelinek.

1. We discussed linear interpolation for smoothing a bigram language model in class, namely

\[ P(w|v) = \lambda f(w|v) + (1 - \lambda)f(w), \]

where \( f(\cdot|\cdot) \) and \( f(\cdot) \) denote the appropriate relative frequency estimates, and \( \lambda \) is chosen so as to maximize the probability of some held-out data.

This problem considers a few alternative strategies for smoothing a bigram language model by directly modifying the counts observed in the training data. In particular, let \( C(v,w) \) denote the count of the bigram \( \langle v,w \rangle \) in the training text, and let \( C^*(v,w) \) be the modified count. For some constant \( \theta > 0 \), consider the three cases

(i) \( C^*(v,w) = C(v,w) + \theta \),
(ii) \( C^*(v,w) = C(v,w) + \theta C(v) \), and
(iii) \( C^*(v,w) = C(v,w) + \theta C(v)f(w) \).

In each case, the smoothed bigram probability is calculated as

\[ P^*(w|v) = \frac{C^*(v,w)}{\sum_{w' \in \mathcal{V}} C^*(v,w')} \]

Let \( N(v,w) \) denote the count of a bigram \( \langle v,w \rangle \) in the held-out text \( \mathcal{H} \).

(a) Derive an expression for the \( \theta \) that maximizes the (log-)probability

\[ P(\mathcal{H}) = \sum_{v \in \mathcal{V}} \sum_{w \in \mathcal{V}} N(v,w) \log P^*(w|v) \]

of the held-out text in each of the three cases (i), (ii) and (iii) above.

(b) Show that if \( N(v,w) = C(v,w) \) for all bigrams, then the optimal value is \( \theta = 0 \) in each case. Why is this an expected result?

(c) Show, in each case, that \( P^* \) may be written as the linear interpolation of a “simple” bigram language model and a lower order model. Identify the two models and \( \lambda \), and discuss the merits or drawbacks of each smoothing strategy.
2. Consider the back-off bigram language model

\[
P_{\text{BO}}(w|v) = \begin{cases} 
\frac{C(v,w) - \delta_1}{C(v)} & \text{if } C(v, w) > 0, \text{ and} \\
\alpha_1(v)f(w) & \text{otherwise} 
\end{cases}
\]

where \( \delta_1 \in (0, 1) \) is sometimes called a constant discount coefficient, \( f(\cdot) \) denotes unigram relative frequencies and \( \alpha_1(v) \), called the back-off weight, is chosen to make \( P_{\text{BO}}(\cdot|v) \) a bona fide probability.

(a) Develop an expression for \( \alpha_1(v) \) in terms of the discount coefficient \( \delta_1 \), unigram probabilities \( f(\cdot) \), and bigram counts \( C(\cdot, \cdot) \).

(b) Write a corresponding formula for replacing \( f(w) \) with a back-off unigram model \( P_{\text{BO}}(w) \) that uses a discount coefficient \( \delta_0 \) and backs off to a uniform distribution on the entire vocabulary. Develop an expression for the back-off weight \( \alpha_0 \).

(c) Does replacing \( f(w) \) with \( P_{\text{BO}}(w) \) necessitate recomputation of \( \alpha_1(v) \)? What does this say about the sequence in which back-off weights in an back-off \( N \)-gram model should be computed?

Start thinking about how to implement Project #2. Your chances of successful implementation will be higher if your code is well designed. Consider writing extensive pseudocode before writing any actual code.