520.666
Information Extraction from Speech and Text

Homework # 5

Due April 19, 2024.

Review Chapter 5 from *Statistical Methods for Speech Recognition* by Frederick Jelinek.

1. *Two Best Paths*: Devise a two-pass algorithm to find the *two most likely paths* through the HMM trellis. Specifically, let the first pass be the Viterbi algorithm, and devise a second pass over the same trellis to find the second most likely path *knowing* the most likely path. Assume if needed that there are no cycles made up entirely of null arcs.

   (a) Describe your second pass in the same way the Viterbi algorithm is described in Chapter 2 (§2.4, page 22).

   (b) Redraw the trellis of Problem 1(e) in Homework #2, run your algorithm on it by hand, and color the two most likely paths.

   Hint: The answer expected here is different from the N-best algorithm described in Section 5.6: here, you assume in the second pass that you already know the best path, so there is no need to consider the two highest γ's in *every* state of the trellis. Why? Discuss.

2. *Back-off Bigram Decoding Graph*: In theory, the bigram decoding graph of Figure 5.2 has N *language-model* (LM) states along the rightmost column, and N^2 null arcs to represent $P(w \mid v)$ for every possible bigram $\langle v, w \rangle \in \mathcal{V} \times \mathcal{V}$, where $N = |\mathcal{V}|$. This makes the complexity of Viterbi decoding prohibitive in practice: $|\mathcal{V}| = $ 100k-400k words is not uncommon.

   In practice, we can reduce this complexity considerably. Consider a back-off bigram model

   $$P(w \mid v) \;=\; \begin{cases} \alpha_v f(w \mid v) & \text{if } C(v,w) > 0 \\ \beta_v P(w) & \text{otherwise,} \end{cases}$$

   where β_v is chosen using, say, the Good-Turing formula, and α_v ensures that $\sum_w P(w \mid v) = 1$ for each word v. For such a model, one need not draw N outgoing arcs from the LM state $e(v)$ for each v, but only draw

   - as many arcs $e(v) \to s(w)$ with probability $P(w \mid v)$ as there are w with $C(v,w) > 0$,
   - 1 null arc $e(v) \to e(\phi)$ with "probability" β_v to a new state $e(\phi)$ shared by all v, and
   - N null arcs $e(\phi) \to s(w)$ with probability $P(w)$, one arc for each $w \in \mathcal{V}$.

   We will analyze and understand this decoding graph in this problem.

   (a) Discuss how this construction assigns language model probabilities to unseen bigrams, e.g. when $\langle w_{i-1}, w_i \rangle$ has $C(w_{i-1}, w_i) = 0$.

(b) Discuss the size of this graph, i.e. describe how many null arcs with language model "probabilities" are needed here, and compare it with N^2 for Figure 5.2.

(c) Is there any shortcoming or incorrectness in this construction? If so, when can it impact the correctness of the most likely path?

Finally, determine the size N of the *word* vocabulary for the combined *Text A* and *Text B* of Project #1, treat the *Text A* as the *kept* data for training a back-off bigram LM, count the number of *seen* bigrams in *Text A*, and calculate and compare the size of the bigram decoding graph of Figure 5.2 and the graph suggested in this problem.