Sample Midterm Questions

1. A **skip-bigram language model** assumes that (i) every sentence $x_1 \ldots x_m$ is generated left-to-right and terminated by generating a special `STOP` symbol (as with the usual n-gram models), and that (ii) the i-th word x_i depends only on the $(i-2)$-th word x_{i-2} at any position i (you may assume a special symbol `*` that indicates x_{-1} and x_0). Write the probability of sentence $x_1 \ldots x_m$ as a function of model parameters $q(x'|x)$ where x' is any element in the vocabulary or `STOP` and x is any element in the vocabulary or `*`.

Answer.

$$p(x_1 \ldots x_m) = \prod_{i=1}^{m} q(x_i|x_{i-2}) \times q(\texttt{STOP}|x_{m-1})$$

2. Consider the unigram language model $q(x)$ and a sentence $x_1 \ldots x_m$. Show that the model parameters that maximize the probability $p(x_1 \ldots x_m)$ under the model are the same as the model parameters that minimize the perplexity on $x_1 \ldots x_m$.

Answer.

$$\arg\max_q \prod_{i=1}^{m} q(x_i) = \arg\min_q -\frac{1}{m} \sum_{i=1}^{m} \log\ q(x_i) = \arg\min_q 2^{-\frac{1}{m}\sum_{i=1}^{m} \log\ q(x_i)}$$

3. Assume we have bigram HMM parameters with $|L|$ POS tag types. Now, for each word type x, suppose we have a dictionary of possible POS tags $C(x)$ where $|C(x)| \leq K \leq |L|$ for some small number K. For instance, $C(\texttt{the}) = \{\texttt{DT}\}$ and $C(\texttt{saw}) = \{\texttt{VBD}, \texttt{NN}\}$. You may assume that this dictionary is correct: that is, $C(x)$ provides all tags that x can ever take. Modify the Viterbi algorithm so that it finds the optimal tag sequence in runtime $O(mK^2)$ rather than $O(m\,|L|^2)$ where m is the length of the sentence.

Answer.

1. For each $y \in C(x_1)$: set $\pi(y, 1) = t(y|*) \times o(x_1|y)$.
2. For $i = 2 \ldots m$:
   (a) For each $y' \in C(x_i)$:

$$\pi(y', i) = \max_{y \in C(x_{i-1})} \pi(y, i-1) \times t(y'|y) \times o(x_i|y')$$

4. Modify the CKY algorithm to compute the probability of the least likely parse tree of a sentence.

Answer.

1. For $i = 1 \dots m$, for each $a \in N$:

$$\pi(a,i,i) = \begin{cases} q(a \to x_i) & \text{if } a \to x_i \in R \\ 0 & \text{otherwise} \end{cases}$$

2. For $l = 2 \dots m$, for $i = 1 \dots m - l$:

   (a) Let $j = i + l$.

   (b) For each $a \in N$,

$$\pi(a,i,j) = \min_{\substack{a \to b\ c \in R \\ i \leq k < j}} q(a \to b\ c) \times \pi(b,i,k) \times \pi(c,k+1,j)$$

4. Assume we have bigram HMM parameters with $|L|$ POS tag types. Given some number $K < |L|$, derive a beam search algorithm to approximate the optimal tag sequence in runtime $O(|L|\,K \log Km)$ where m is the length of the sentence. You may assume an implementation of the leaky priority queue discussed in class. The algorithm must be in terms of the model parameters $t(y'|y)$ and $o(x|y)$.

Answer.

1. $q \leftarrow \texttt{leaky\_priority\_queue}(K)$
2. $q.\texttt{push}([y_1, t(y_1|\texttt{*}) \times o(x_1|y_1)]) \qquad \forall y_1 \in L$
3. For $i = 2 \dots m$:

   (a) $\mathcal{B}_{i-1} \leftarrow q.\texttt{dump}()$

   (b) For $(y_1 \dots y_{i-1}, s) \in \mathcal{B}_{i-1}$:

      i. If $i < m$:

$$q.\texttt{push}([y_1 \dots y_{i-1}\ y_i, s \times t(y_i|y_{i-1}) \times o(x_i|y_i)]) \qquad \forall y_i \in L$$

      ii. Else:

$$q.\texttt{push}([y_1 \dots y_{i-1}\ y_i, s \times t(y_i|y_{i-1}) \times o(x_i|y_i) \times t(\texttt{STOP}|y_i)]) \quad \forall y_i \in L$$

4. Return $q.\texttt{dump}()$.