NLP - Fall 2018 - Sample Midterm Exam

(1) TrueCasing is the process of taking text with missing or unreliable case information and producing the proper case for each word, e.g. if the input looks like:

```
as previously reported , target letters were issued last month to
michael milken , drexel 's chief of junk-bond operations ; mr. milken 's
brother lowell ; cary maultasch , a drexel trader ; james dahl , a
drexel bond salesman ; and bruce newberg , a former drexel trader .
```

Then the output of the TrueCasing program should be:

```
As previously reported , target letters were issued last month to
Michael Milken , Drexel 's chief of junk-bond operations ; Mr. Milken 's
brother Lowell ; Cary Maultasch , a Drexel trader ; James Dahl , a
Drexel bond salesman ; and Bruce Newberg , a former Drexel trader .
```

Assume **we can only use** the following two probability distributions:

- A *translation probability* $P(w \mid W)$ where w is the lowercase variant of the TrueCase word W (note that the TrueCase word might still be lowercase). The function `lower` can be used to lowercase a word, e.g. "*HAL9001*".`lower`() = "*hal9001*"
- A *bigram probability* $P(W \mid W')$. A language model $P(W_1, \ldots, W_n)$ is used to provide the probability of a sentence. A bigram language model approximates the probability of a sentence as follows:

$$\Pr(W_1, \ldots, W_n) \approx \prod_{i=1}^{n} P(W_i \mid W_{i-1})$$

  Assume that $W_{-1} = w_{-1} = none$ is a dummy word that begins each sentence.
- Assume that $c(\cdot)$ gives the frequency of unigrams, bigrams, etc.

a. Complete the following formula to provide a model of the TrueCasing task by using only the translation probability $P(w \mid W)$ and the bigram probability $P(W \mid W')$:

$$\begin{aligned} W_1^*, \ldots, W_n^* &= \arg\max_{W_1,\ldots,W_n} \Pr(W_1, \ldots, W_n \mid w_1, \ldots, w_n) \\ &= \textit{provide this formula} \end{aligned}$$

Answer:

$$\begin{aligned} W_1^*, \ldots, W_n^* &= \arg\max_{W_1,\ldots,W_n} P(W_1, \ldots, W_n \mid w_1, \ldots, w_n) \\ &= \frac{P(W_1, \ldots, W_n) \cdot P(w_1, \ldots, w_n \mid W_1, \ldots, W_n)}{P(w_1, \ldots, w_n)} \\ &\approx P(W_1, \ldots, W_n) \cdot P(w_1, \ldots, w_n \mid W_1, \ldots, W_n) \\ &= \prod_{i=1}^{n} \underbrace{P(w_i \mid W_i)}_{\textit{translation probability}} \cdot \underbrace{P(W_i \mid W_{i-1})}_{\textit{bigram language model}} \end{aligned}$$

b. Using maximum likelihood, provide a formula to estimate the the translation probability parameters $P(w \mid W)$ for lowercase words w and TrueCase words W. Assume you **only** have access to a sufficient amount of TrueCase text.

> *Answer:* For each TrueCase word W convert it to lowercase w using $w = W.\texttt{lower}()$ and count $f(w, W)$ and $f(W)$. Then,
> $$P(w \mid W) = \frac{f(w, W)}{\sum_{w'} f(w', W)}$$

c. Provide the equation that correctly computes add one smoothing for $P(w \mid W)$.

> *Answer:*
> $$P(w \mid W) = \frac{1 + f(w, W)}{|w| + \sum_{w'} f(w', W)}$$

d. Backoff smoothing for $P(W_i \mid W_{i-1})$ is defined as follows:

$$P_{bo}(W_i \mid W_{i-1}) = \begin{cases} \frac{c^*(W_{i-1}, W_i)}{c(W_{i-1})} & \text{if } c(W_{i-1}, W_i) > 0 \\ \alpha(W_{i-1}) P_{bo}(W_i) & \text{otherwise} \end{cases}$$

where $c^*(W_{i-1}, W_i) = c(W_{i-1}, W_i) - D$ for some $0 < D < 1$ and $\alpha(w_{i-1})$ is chosen to make sure that $P_{bo}(W_i \mid W_{i-1})$ is a proper probability. Provide the equation to compute $\alpha(W_{i-1})$. Assume that $\sum_{W_i} P_{bo}(W_i) = 1$.

> *Answer:*
> $$\alpha(W_{i-1}) = 1 - \sum_{W_i} \frac{c^*(W_{i-1}, W_i)}{c(W_{i-1})}$$

(2) **Language Models**

For the CFG G given below:

$$\begin{aligned} S &\rightarrow A \mid c \\ A &\rightarrow B\ a \\ B &\rightarrow b\ S \end{aligned}$$

a. What is the language $L(G)$?

> *Answer:* $b^n c a^n : n \geq 0$

b. Assign probabilities to each rule in the CFG above so that for each string $w \in L(G)$:

$$P(w) = exp\left(\frac{|w| - 1}{2} \times ln(0.3) + ln(0.7)\right)$$

where, $|w|$ is the length of string w, *exp* is exponentiation, and *ln* is *log* base e. Using an example, briefly explain *why* your PCFG provides the desired $P(w)$ for any w.

Answer:

```
0.3 S -> A
0.7 S -> c
1.0 A -> B a
1.0 B -> b S
```

Since $w \in \{b^n c a^n : n \geq 0\}$ then we always have one c in w which is derived using $S \rightarrow c$ with prob 0.7. Removing c from w we get length $|w| - 1$. Each $b \ldots a$ pair in w is derived by first using rule $S \rightarrow A$ with prob 0.3. The subsequent $A \rightarrow Ba$ and $B \rightarrow bS$ for each $b \ldots a$ pair each get prob 1.0 as there are no competing lhs A and B rules. There are $\frac{|w|-1}{2}$ matching $b \ldots a$ pairs in each w. Hence deriving all of them will take probability $(0.3)^{\frac{|w|-1}{2}}$. Since:

$$(0.3)^{\frac{|w|-1}{2}} \times 0.7 = exp\left(\frac{|w| - 1}{2} \times ln(0.3) + ln(0.7)\right)$$

we obtain the required definition for $P(w)$.