CMPT 413 - Spring 2012 - Midterm #2

Please write down "Midterm #2" on the top of the answer booklet.

When you have finished, return your answer booklet along with this question booklet.

(1) (10pts) **Language Modeling**

Consider a language model over character sequences that computes the probability of a word based on the characters in that word, so if word $w = c_0, c_1, \ldots, c_n$ then $P(w) = P(c_0, \ldots, c_n)$. Let us assume that the language model is defined as a bigram character model $P(c_i \mid c_{i-1})$ where

$$P(c_0, \ldots, c_n) = \prod_{i=1,2,\ldots,n} P(c_i \mid c_{i-1}) \tag{1}$$

For convenience we assume that we have explicit word boundaries: $c_0 = \texttt{bos}$ and $c_n = \texttt{eos}$ where `bos` stands for *begin sentence marker* and `eos` stands for *end of sentence marker*.
Based on this model, for the English word *booking* the probability would be computed as:

$$P(booking) = P(b \mid \texttt{bos}) \times P(o \mid b) \times P(o \mid o) \times P(k \mid o) \times P(i \mid k) \times P(n \mid i) \times P(g \mid n) \times P(\texttt{eos} \mid g)$$

The inflection *ing* is a suffix and is generated after the stem *book* with probability

$$P(ing) = P(i \mid k) \times P(n \mid i) \times P(g \mid n) \times P(\texttt{eos} \mid g)$$

In Semitic languages, like Arabic and Hebrew, the process of inflection works a bit differently. In Arabic, for a word like *kitab* the stem would be *k-t-b* where the place-holders '-' for inflection characters have been added for convenience. We will assume that each word is made up of a sequence of consonant-vowel sequences CVCVCV… and the vowels always form the inflection.

a. (4pts) Provide the definition of an n-gram model that will compute the probability for the word *kitab* and *k-t-b* as follows:

$$P(kitab) = P(k \mid \texttt{bos}) \times P(t \mid k) \times P(b \mid t) \times P(i \mid b) \times P(a \mid i) \times P(\texttt{eos} \mid a)$$

$$P(k\text{-}t\text{-}b) = P(k \mid \texttt{bos}) \times P(t \mid k) \times P(b \mid t) \times P(\text{-} \mid b) \times P(\text{-} \mid \text{-}) \times P(\texttt{eos} \mid \text{-})$$

Write down the equation for this n-gram model in the same mathematical notation as equation (1).

Answer:

$$P(c_0, \ldots, c_n) = \begin{cases} \prod_{i=1}^{n} P(c_i \mid c_{i-1}) & \text{if } n \leq 3 \\ \left(P(c_1 \mid c_0) \times \prod_{i=3,5,\ldots}^{\ell} P(c_i \mid c_{i-2})\right) \times & \\ \left(P(c_2 \mid c_{\ell_o}) \times \prod_{i=4,6,\ldots}^{\ell} P(c_i \mid c_{i-2}) \times P(c_n \mid c_{\ell_e})\right) & \text{if } n > 3 \end{cases}$$

Define $\ell = n - (n \mod 2)$ and ℓ_o is the last odd number less than ℓ and ℓ_e is the last even number less than ℓ. As long as the boundary cases are right for the bigrams, we don't penalize off by one in the length, and we don't penalize for $n \leq 3$.

b. (2pts) Using your n-gram model show how $P(kitab) = P(ktb) \times P(ia)$.

Answer:

$$\begin{aligned}
P(kitab) &= P(c_0 = \texttt{bos}, c_1 = k, c_2 = i, c_3 = t, c_4 = a, c_5 = b, c_6 = \texttt{eos}) \\
&= P(ktb) \times P(ia, \texttt{eos}) \\
P(ktb) &= P(c_1 = k \mid c_0 = \texttt{bos}) \times P(c_3 = t \mid c_1 = k) \times P(c_5 = b \mid c_3 = t) \\
&\quad \text{this term corresponds to the first bracket in the eqn above} \\
P(ia) &= P(c_2 = i \mid c_{\ell_o} = c_5 = b) \times P(c_4 = a \mid c_2 = i) \times P(c_n = c_6 = \texttt{eos} \mid c_{\ell_e} = c_4 = a) \\
&\quad \text{corresponds to the second bracket in the eqn above}
\end{aligned}$$

c. (4pts) For bigram probabilities $P(c_i \mid c_{i-1})$, Katz backoff smoothing is defined as follows:

$$P_{katz}(c_i \mid c_{i-1}) = \begin{cases} \frac{r^*(c_{i-1}, c_i)}{r(c_{i-1})} & \text{if } r(c_{i-1}, c_i) > 0 \\ \alpha(c_{i-1}) P_{katz}(c_i) & \text{otherwise} \end{cases}$$

where $r(\cdot)$ provides the (unsmoothed) frequency from training data and $r^*(\cdot)$ is the Good-Turing estimate of the frequency r defined as follows:

$$r^*(c_{i-1}, c_i) = (r(c_{i-1}, c_i) + 1) \times \frac{n_{r(c_{i-1}, c_i)+1}}{n_{r(c_{i-1}, c_i)}}$$

where $n_{r(c_{i-1}, c_i)}$ is the number of different c_{i-1}, c_i types observed with count $r(c_{i-1}, c_i)$. We assume that linear interpolation has provided all missing $n_{r(\cdot)}$ values required.
$\alpha(c_{i-1})$ is chosen to make sure that $P_{katz}(c_i \mid c_{i-1})$ is a proper probability. Provide the equation for $\alpha(c_{i-1})$.

Answer:

Step by step derivation below. We are just looking for the end result.

$$\begin{aligned}
\sum_{c_i} \left(\frac{r^*(c_{i-1}, c_i)}{r(c_{i-1})} + \alpha(c_{i-1}) P_{katz}(c_i) \right) &= 1 \\
\sum_{c_i : r(c_{i-1}, c_i) > 0} \frac{r^*(c_{i-1}, c_i)}{r(c_{i-1})} + \alpha(c_{i-1}) \sum_{c_i : r(c_{i-1}, c_i) = 0} P_{katz}(c_i) &= 1 \\
\alpha(c_{i-1}) \sum_{c_i : r(c_{i-1}, c_i) = 0} P_{katz}(c_i) &= 1 - \left(\sum_{c_i : r(c_{i-1}, c_i) > 0} \frac{r^*(c_{i-1}, c_i)}{r(c_{i-1})} \right) \\
\alpha(c_{i-1}) &= \frac{1 - \left(\sum_{c_i : r(c_{i-1}, c_i) > 0} \frac{r^*(c_{i-1}, c_i)}{r(c_{i-1})} \right)}{\sum_{c_i : r(c_{i-1}, c_i) = 0} P_{katz}(c_i)} \\
\alpha(c_{i-1}) &= \frac{1 - \left(\sum_{c_i : r(c_{i-1}, c_i) > 0} \frac{r^*(c_{i-1}, c_i)}{r(c_{i-1})} \right)}{1 - \left(\sum_{c_i : r(c_{i-1}, c_i) > 0} P_{katz}(c_i) \right)}
\end{aligned}$$

Also acceptable is the somewhat less precise answer which assumes $\sum_{c_i} P_{katz}(c_i) = 1$:

$$\alpha(c_{i-1}) = 1 - \sum_{c_i} \frac{r^*(c_{i-1}, c_i)}{r(c_{i-1})}$$

(2) (10pts) **Context-free Grammars**:

Consider the following context-free grammar:

$$NP \rightarrow NP\ NP$$
$$NP \rightarrow natural \mid language \mid processing \mid course$$

a. (2pts) How many distinct parse trees does the above grammar derive for the input string: *natural language processing course*.

> *Answer:*
> ```
> answer = 5
>
> (NP (NP (NP natural) (NP language)) (NP (NP processing) (NP course)))
> (NP (NP natural) (NP (NP language) (NP (NP processing) (NP course))))
> (NP (NP (NP (NP natural) (NP language)) (NP processing)) (NP course))
> (NP (NP natural) (NP (NP (NP language) (NP processing)) (NP course)))
> (NP (NP (NP natural) (NP (NP language) (NP processing))) (NP course))
> ```

b. (2pts) From the various parse trees you've listed above, provide the tree that corresponds to the natural meaning of the phrase: a course that teaches the processing of natural language.

> *Answer:*
> ```
> (NP (NP (NP (NP natural) (NP language)) (NP processing)) (NP course))
> ```

c. (4pts) Show how you can predict the number of parse trees for the above input string using the notion of Catalan numbers.

> *Answer:* Assume there is a hidden `and` between each *NP*, $NP \rightarrow NP$ `and` *NP*, and so we can transform the input string to *natural* `and` *language* `and` *processing* `and` *course* and just as in the coordination grammar covered in the lecture notes, the number of parse trees is given by *Cat(number of* `and`*s)* = *Cat*(3) = 5.

d. (2pts) True or false: The above grammar is in Chomsky Normal Form.

> *Answer:* True!

(3) (10pts) **Probabilistic Context-free Grammars**

Consider a Treebank where the following set of trees are repeated several times as indicated:

- 2× (S (B a a) (C a a))
- 1× (S (C a a a))
- 7× (S (B a))

a. (4pts) What is the probabilistic CFG that can be extracted from this Treebank. (*Hint*: make sure you take into account the frequency of the trees shown above).

> *Answer:*
> ```
> S -> B C   2/10        B -> aa    2/9
> S -> C     1/10        B -> a     7/9
> S -> B     7/10        C -> aa    2/3
>                        C -> aaa   1/3
> ```

b. (2pts) Given this probabilistic CFG what is the most likely tree for the input: *aaaa*

Answer: (S (B a) (C aaa)) is the most likely tree for input aaaa

c. (4pts) Does the most likely tree for input *aaaa* appear in the Treebank? If not, why not?

Answer: The subtrees for *B* and *C* are chosen independently due to the independence assumptions made by PCFGs, so the most likely tree contains the most likely *B* subtree and the most likely *C* subtree despite that fact that the most likely *B* subtree may have never co-occured with the most likely *C* subtree in the Treebank.