CMPT 413 - SPRING 2008 - QUIZ #2

Please write down “QUIZ #2” on the top of the answer booklet.

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

Mar 10, 2008

(1) (10pts) You are given the following input where all the vowels have been removed:

```
whl nrn stck clmbd nd wll strt ws stll prmtng t ,
a grp f 29 nrn xctvs nd drctrs bgn t sll thr shrs .
```

Each de-vowelized ‘word’ in this input is consistent with many alternative words:

```
stck: stack, stick, stock, stuck
t:    to, it, at, out, too, auto, eat, tie, tea, ate, toe, tee, oat, iota
```

Assume you are given the following resources:

1. A dictionary called *withVowels* where each de-vowelized ‘word’ is mapped to a list of possible words with the vowels inserted, e.g. *withVowels*[’`stck`’] provides four possible words in the example above.
2. A probability distribution $P(dvw \mid vw)$ where *dvw* is a de-vowelized ‘word’ and *vw* is a word with vowels inserted.
3. A bigram language model $P(vw_i \mid vw_{i-1})$ that has been trained on a large amount of training data of naturally occurring sentences (obviously consisting of words with vowels). The probability distribution has been suitably smoothed so that every pair of words vw_{i-1}, vw_i has some non-zero probability.

Using these resources, show how you can use an existing algorithm (that we have studied in this course) to convert a de-vowelized input sentence into the most likely output vowelized sentence. For the de-vowelized input shown above the output would look like:

```
while norian stock climbed and wall street was still promoting it ,
a group of 29 norian executives and directors began to sell their shares .
```

Your task is to convert this problem and phrase it in terms of a Hidden Markov Model (HMM) and use an existing HMM algorithm that we have studied in this course to solve this problem. Explain how that algorithm can be used to convert de-vowelized sentences into normal English sentences.

Answer: For an input sentence of de-vowelized words $\mathbf{dvw} = dvw_1, \ldots, dvw_n$, we wish to find the most likely sentence with vowels inserted, $\mathbf{vw} = vw_1, \ldots, w_n$.

$$\operatorname*{argmax}_{\mathbf{vw}} P(\mathbf{vw} \mid \mathbf{dvw})$$

We use Bayes rule:

$$P(\mathbf{vw} \mid \mathbf{dvw}) \approx P(\mathbf{dvw} \mid \mathbf{vw}) \cdot P(\mathbf{vw})$$

For each dvw_i construct an HMM with emission probability $P(dvw_i \mid vw_i)$ and transition probability $P(vw_i \mid vw_{i-1})$ for each vw_i such that vw_i is in the dictionary mapping *withVowels*$[dvw_i]$. Note that the states of this HMM are the words with vowels inserted, vw_i, and all possible vowelized word sequences, $vw_1, \ldots, vw_n$ can be scored using this HMM. Then,

$$\operatorname*{argmax}_{\mathbf{vw}} P(\mathbf{dvw} \mid \mathbf{vw}) \cdot P(\mathbf{vw}) = \operatorname*{argmax}_{\mathbf{vw}=vw_1,\ldots,vw_n} \prod_{i=1}^{n} P(dvw_i \mid vw_i) \cdot P(vw_i \mid vw_{i-1})$$

We assume we start with a special `None` state for $i = 0$. The mostly likely output vowelized word sequence $\mathbf{vw}$ can be computed using the Viterbi algorithm for this constructed HMM and with the input string $\mathbf{dvw}$. Let k be the maximum number of vw words returned from the dictionary lookup *withVowels* for ‘word’ dvw. The worst case time complexity is $O\left(k^2 \cdot n\right)$.

Two additional issues that you many wonder about (neither of which are required to solve this question):

- Notice that in the example with the vowels removed, not all the vowels were actually removed. The word `a` was kept as it is in the de-vowelized input sentence because removing the vowel would eliminate a word from the original sentence. In this case, the number of words in the desired output could be longer than the input de-vowelized sentence making the search procedure a lot more complicated. Can you design an algorithm that can handle insertion of missing words?
- $P(dvw \mid vw)$ is the probability of a de-vowelized ‘word’ *given* a vowelized word, but a more useful probability might be $P(vw \mid dvw)$. Note that it is not easy to see how $P(vw \mid dvw)$ could be combined with the language model, but $P(dvw \mid vw)$ can be easily seen as the emission probability if the language model is the transition probability. But using Bayes Rule,

  $$P(dvw \mid vw) = \frac{P(vw \mid dvw) \cdot P(dvw)}{P(vw)}$$

  so we could use the value of $P(vw \mid dvw) \approx P(dvw \mid vw) \cdot P(vw)$.

(2) (5pts) Let c^{i}_{i+k} represent a sequence of characters $c_i, c_{i+1}, \ldots, c_{i+k}$. Assume that you're given a 4-gram character model: $P(c_i \mid c^{i-3}_{i-1})$. Note that $\sum_{c_i} P(c_i \mid c^{i-3}_{i-1}) = 1$. Assume that all the characters have been observed at least once in the training data so that $P(c_i)$ is never zero in unseen data.

a. Write down the maximum likelihood estimate of $P(c_i \mid c^{i-3}_{i-1})$ in terms of the observed frequencies of the ngrams in the training data: $f(c^{i-3}_{i})$ and $f(c^{i-3}_{i-1})$

Answer:

$$P(c_i \mid c^{i-3}_{i-1}) = \frac{f(c^{i-3}_{i})}{f(c^{i-3}_{i-1})}$$

b. Consider a Jelinek-Mercer style interpolation smoothing model $\hat{P}$:

$$\hat{P}(c_i \mid c^{i-3}_{i-1}) = \lambda_1 \cdot P(c_i \mid c^{i-3}_{i-1}) + \lambda_2 \cdot P(c_i \mid c^{i-2}_{i-1}) + \lambda_3 \cdot P(c_i \mid c_{i-1}) + \lambda_4 \cdot P(c_i)$$

State the condition on values assigned to $\lambda_1, \ldots, \lambda_4$ for $\hat{P}$ to be a well-defined probability model.

Answer: The condition is:

$$\sum_{i=1}^{4} \lambda_i = 1$$

c. Assume you are given some additional training data (separate from your original training data). Let's say this data is your *held-out* data called T. T will contain ngrams that were unseen in our original training data, and we can exploit this fact to compute values for $\lambda_1, \ldots, \lambda_4$ in the interpolation smoothing model shown above.

Let W_T be the length of T in number of character tokens. For each c_i in T, where $1 \leq i \leq W_T$, let $g_1(c_i) = 1$ when, for any c^{i-3}_{i-1} observed in T, the 4-gram probability $P(c_i \mid c^{i-3}_{i-1})$ had a non-zero value (that is, whenever $f(c^{i-3}_{i}) > 0$). Similarly, let $g_2(c_i)$, $g_3(c_i)$ and $g_4(c_i)$ equal 1 when the trigram, bigram and unigram probability respectively had a non-zero value in T.

Show how you can compute the values for $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ for the equation in Question 2b by using g_1, g_2, g_3, g_4. State why the condition stated in your answer to Question 2b is satisfied by your answer.

Answer: We define each λ_i as follows:

$$\lambda_1 = \frac{\sum_{c_i} g_1(c_i)}{\sum_{c_i} g_4(c_i)} \qquad \lambda_3 = \frac{\sum_{c_i} g_3(c_i) - g_2(c_i)}{\sum_{c_i} g_4(c_i)}$$

$$\lambda_2 = \frac{\sum_{c_i} g_2(c_i) - g_1(c_i)}{\sum_{c_i} g_4(c_i)} \qquad \lambda_4 = \frac{\sum_{c_i} g_4(c_i) - g_3(c_i)}{\sum_{c_i} g_4(c_i)}$$

The condition in Question 2b was that $\sum_{i=1}^{4} \lambda_i = 1$ and this is satisfied because:

$$\begin{aligned} \text{Sum of numerators} &= \sum_{c_i} g_1(c_i) + g_2(c_i) - g_1(c_i) + g_3(c_i) - g_2(c_i) + g_4(c_i) - g_3(c_i) \\ &= \sum_{c_i} g_4(c_i) = \text{denominator} \end{aligned}$$