Natural Language Processing - 096260 - Homework Assignment 1
Due Date: 03.12.2015

Part 1 – Analytical Questions

Question 1

In the second tutorial we have seen a method for language modeling called *linear interpolation,* where the trigram estimate is defined as follows:

$$q(w_i \mid w_{i-2}, w_{i-1}) = \lambda_1 \times q_{ML}(w_i \mid w_{i-2}, w_{i-1}) + \lambda_2 \times q_{ML}(w_i \mid w_{i-1}) + \lambda_3 \times q_{ML}(w_i)$$

Here $\lambda_1, \lambda_2, \lambda_3$ are weights for the trigram, bigram, and unigram estimates, and q_{ML} stands for the maximum- likelihood estimate.

One way to optimize the λ values is to use a set of validation data,

Say the validation data consists of n sentences, $S_1, S_2, \ldots, S_n$.

Define $c'(w_1, w_2, w_3)$ to be the number of times the trigram is seen in the validation sentences. Then λ values are chosen to maximize the following function:

$$L(\lambda_1, \lambda_2, \lambda_3) = \sum_{w_1, w_2, w_3} c'(w_1, w_2, w_3) \log q(w_3, |w_1, w_2)$$

Show that choosing λ values that maximizes $L(\lambda_1, \lambda_2, \lambda_3)$ is equivalent to choosing λ values that minimize the perplexity of the language model on the validation data.

Question 2

In the second tutorial we saw an improved method for linear interpolation (allowing the λ's to vary). Here we use the same function form (as in the tutorial), which maps trigrams into "bins", depending on their count:

$$
\begin{array}{lll}
\Phi(w_{i-2}, w_{i-1}, w_i) = 1 & \text{If} & Count(w_{i-2}, w_{i-1}, w_i) = 0 \\
\Phi(w_{i-2}, w_{i-1}, w_i) = 2 & \text{If} & 1 \leq Count(w_{i-2}, w_{i-1}, w_i) \leq 2 \\
\Phi(w_{i-2}, w_{i-1}, w_i) = 3 & \text{If} & 3 \leq Count(w_{i-2}, w_{i-1}, w_i) \leq 5 \\
\Phi(w_{i-2}, w_{i-1}, w_i) = 4 & \text{If} & 6 \leq Count(w_{i-2}, w_{i-1}, w_i)
\end{array}
$$

The trigram estimate $q(w_i \mid w_{i-2}, w_{i-1})$ is then defined as

$$
\begin{aligned}
q(w_i \mid w_{i-2}, w_{i-1}) \quad = \quad & \lambda_1^{\Phi(w_{i-2},w_{i-1},w_i)} \times q_{ML}(w_i \mid w_{i-2}, w_{i-1}) \\
& + \lambda_2^{\Phi(w_{i-2},w_{i-1},w_i)} \times q_{ML}(w_i \mid w_{i-1}) \\
& + \lambda_3^{\Phi(w_{i-2},w_{i-1},w_i)} \times q_{ML}(w_i)
\end{aligned}
$$

Notice that we now have 12 smoothing parameters, i.e., λ_j^i for $i = 1 \ldots 4$ and $j = 1 \ldots 3$.

Unfortunately this estimation method has a serious problem: what is it?

Question 3

We are going to come up with a modified version of the Viterbi algorithm for trigram taggers. Assume that the input to the Viterbi algorithm is a word sequence $x_1 \ldots x_n$. For each word in the vocabulary, we have a *tag dictionary* $T(x_i)$ that lists the tags y such that $e(x_i \mid y) > 0$. Take K to be a constant such that: $\forall x_i \text{ i=1..n } |T(x_i)| \leq K$

Give pseudo-code for a version of the Viterbi algorithm that runs in $O(nK^3)$ time where n is the length of the input sentence.

Question 4

Suppose a trigram language model as follows:

$$p(\vec{w}) \stackrel{\text{def}}{=} p(w_1) \cdot p(w_2 \mid w_1) \cdot p(w_3 \mid w_1, w_2) \cdot p(w_4 \mid w_2, w_3) \cdots p(w_n \mid w_{n-2}, w_{n-1})$$

1.) Expand the above definition of the LM using naive estimates of the parameters such as:

$$p(w_4 \mid w_2, w_3) \stackrel{\text{def}}{=} \frac{c(w_2 w_3 w_4)}{c(w_2 w_3)}$$

2.) One could also define a kind of reversed trigram language model that instead the regular trigram LM, assumes the words were generated in a reverse order ("from right to left"):

$$p_{reversed}(\vec{w}) \stackrel{\text{def}}{=} p(w_n) \cdot p(w_{n-1} \mid w_n) \cdot p(w_{n-2} \mid w_{n-1}, w_n) \cdot p(w_{n-3} \mid w_{n-2}, w_{n-1}) \cdots p(w_2 \mid w_3, w_4) \cdot p(w_1 \mid w_2, w_3)$$

By manipulating the notation, show that the two models are identical $\left(i.e., \mathrm{p}\left(\vec{w}\right) = \mathrm{p}_{reversed}\left(\vec{w}\right) \forall \vec{w}\right)$ provided that both models use MLE parameters estimated from the same training data.

3.) Suppose your data contains sentences which are delimited by <s> at the start and </s> at the end. For example, the following data set consists of a sequence of 3 sentences:

```
<s> do you think so </s> <s> yes </s> <s> at least i thought so </s>
```

Given English training data, the probability of:

```
<s> do you think the </s>
```

should be extremely low under any good language model. Why? In the case of the trigram model, which parameter or parameters are responsible for making this probability low?