Assignment III

Natural Language Models and Interfaces 2012

4 April 2012

1 A smoothed bigram language model

Finish before the computer lab of April 20th.

For this exercise, we again use the OVIS corpus of the previous exercises, but this time we use the first 90% of the sentences of the corpus for training and the rest (10%) we reserve for testing. You can find these two parts of the corpus on the course's website.

1. Construct a bigram language model based on the training corpus. For this, use the program of the previous exercise.

2. Implement Good-Turing smoothing and apply it to the bigram model of the first part. Implement two versions of this algorithm:

   (a) Use smoothing for all bigrams (that is, for all counts c). For this, use the Good-Turing formula as given in class (formula 4.26 in the book Jurafsky and Martin, p135).

   (b) Use smoothing only for bigrams which appeared at most k times in the training corpus. Use the standard Good-Turing formula (4.26) for $c = 0$ and the following formula (4.31 in the book) for $1 \leq c \leq k$:

   $$c^* = \frac{(c+1)\frac{N_{c+1}}{N_c} - c\frac{(k+1)N_{k+1}}{N_1}}{1 - \frac{(k+1)N_{k+1}}{N_1}}$$

   Use no smoothing for $c > k$, that is, $c^* = c$.

   Remarks:

   (a) To calculate $P(w_i|w_{i-1})$, you need to use the formula $P(w_i|w_{i-1}) = \frac{Count(w_{i-1}w_i)}{\sum_{w \in V \cup \{STOP\}} Count(w_{i-1},w)}$ (where $Count$ is the frequency after

smoothing). It is no longer possible to use the formula $P(w_i|w_{i-1}) = \frac{Count(w_{i-1}w_i)}{Count(w_{i-1})}$. Make sure you calculate the sum $\sum_{w \in V \cup \{STOP\}} Count(w_{i-1}, w)$ in an efficient way.

(b) Assume that $|V|$ (the number of words in the vocabulary) is equal to the number of word types in the training corpus.

3. Test the model without smoothing and with both forms of smoothing (with $k = 5$ in the second smoothing model) on the test corpus, as follows:

   (a) Report the percentage of sentences which are assigned probability zero by each of the models.

   (b) Report the first 5 sentence in the test corpus which are assigned a probability of zero by each of the models.

   (c) Explain the differences between the models.

2 Paper: Language modelling for speech recognition

This is an individual assignment; hand in a first complete draft of the paper on April 23d via BlackBoard.

For the paper that you submit on April 23d, you investigate whether a bigram or a trigram language model is the better choice for disambiguating the output of a speech recognition system. The OVIS corpus contains a training set and a test set with word graphs (to be discussed in class). You train your language models on strings extracted from the train set word graphs; this is similar to the training data we have used so-far, but also contains symbols for pauses etc.

You then test the model on the test set word graphs, by selecting the path through the graphs with highest probability according to the language model. You calculate the accuracy of your output by comparing it to the gold standard test sentences.

To carry out this little research project, you need minimally:

1. a program to calculate the probability of candidate sentences at test time based on counts derived from a training corpus; for this you use the program from assignment 3.

2. a way to extract candidate sentences from word graphs; you can use some sed/grep/awk hacks (which we will distribute later) if necessary

to make the deadline of April 23d, but the nicer solution (which will be necessary for assignment 4 after the tussentoets) is to integrate this with the program for calculating ngram probabilities (suing dynamic programming - to be discussed in class).

3. a way to select the highest probability sentence among several candidate sentences and to compare it to the gold standard. We will provide help with this later as well.