# homework #8

# Language Modeling

**EE 8993: Fundamentals of Speech Recognition**

December 6, 1998

*submitted to:*

Dr. Joseph Picone

*submitted by:*

Suresh Balakrishnama

Institute for Signal and Information Processing
Department of Electrical and Computer Engineering
Mississippi State University
MS 39762, USA
Email: balakris@isip.msstate.edu

## 1. INTRODUCTION

Language modeling involves determining the appropriate equivalence classification and a method of estimating the priori probability. The selection of a language model has a significant influence on the performance of a speech recognition system. Language model is an important subsystem of speech recognizer since it provides adequate information about the occurrence of words and constraints occurring for different sequence of words. Design of language model should be incorporated to accommodate all grammatical constraints, accent and style of the speaker, and also, the size of language model is very important for real time computational purposes.

In equation (1) the priori probability $P(W)$ can be computed using Bayes' formula for decomposition i.e., we need to compute the probability for every word string $W$. Mathematically, this can be expressed as:

$$P(W) = \left(\prod_{i=1}^{n} P((w_i/w_1)\ldots\ldots w_{i-1})\right) \qquad (1)$$

The recognizer thus determines estimates of the probabilities $P((w_i/w_1)\ldots\ldots w_{i-1})$. The choice of a word $w_i$ depends on the equivalence classes $\varnothing(w_1\ldots\ldots w_{i-1})$. Thus, the priori probability becomes:

$$P(W) = \left(\prod_{i=1}^{n} P((w_i/w_1)\ldots\ldots w_{i-1})\right) \qquad (2)$$

The language model comprises of determining the appropriate equivalence classification $\varnothing$ and a method of estimating the probabilities $P((w_i/w_1)\ldots\ldots w_{i-1})$.

## 2. Problem Description

Perform a simple language modeling experiment. For the large text corpus provided, do the following:

1. Generate a histogram of word unigrams, bigrams, and trigrams.Compute the entropy of the distribution and discuss the nature of the distributions. Plot the OOV rate as a function of the N most frequent words.
2. Select the most frequent 1000 words. Compute the trigram coverage using this vocabulary.
3. Partition the data into a kept set and a held-out set. Use 80% of the data for the kept set. Build a trigram LM for this data. Compute the coverage of this trigram LM for the held-out set. Repeat this for three more partitions of the data.

4. From the results of no. 3, suggest a reasonable interpolated LM.

## 3. Implementation and Results

This language modeling experiment was solved using Carnegie Mellon University - Cambridge Statistical Language Modeling toolkit v2. This software has small utilities to compute unigram, bigram and trigram in a large text corpus. The steps adopted to solve this problem are as follows:

1. Download the CMU LM toolkit so that we can use some of their nice routines (V2).
2. Obtain the train data from NIST (bn6252atr.text is enough for class project).
3. Use CMU LM toolkit to generate unigram, bigram, and trigram for the training data.
4. zcat data/bn625atr.text.Z | CMU-Cam_Toolkit_v2/bin/text2wngram -n 1 -temp temp > data/bn625atr.w1gram
5. zcat data/bn625atr.text.Z | CMU-Cam_Toolkit_v2/bin/text2wngram -n 2 -temp temp > data/bn625atr.w2gram
6. zcat data/bn625atr.text.Z | CMU-Cam_Toolkit_v2/bin/text2wngram -n 3 -temp temp > data/bn625atr.w3gram
7. Use util/calculate_entropy.pl to calculate the entropy of the distribution. The output is to be plotted with xmgr to see the distribution.
8. Sort the wngram files in descending order sort -r -k <field number> file
9. Use util/calculate_oov.pl to calculate the oov rate. The output is to be plotted as oov as function of n frequent words
10. Use the CMU LM toolkit to find the top 1000 words and generate the trigrams.
11. Use util/split_data1.pl to pick a held-out and a kept set.
12. Generate the trigrams for the kept set and held out sets.
13. Determine how many trigrams in the held out set can be found in the kept set.

## 4. CONCLUSIONS

Language model is an component of a speech recognizer. Language model used should depend on the use to which the recognizer will be put. The transcription of dictated radiological reports requires different language models than the writing of movie reviews. The construction of language model depends very much on the training data based on which models are built and text is produced.

## 5. SOFTWARE

Software utilities used for this project (CMU-CAM toolkit v2) is available for public from our website at *www.isip.msstate.edu* as well from *www.cs.cmu.edu/scs/scs.html.* The data and output files obtained for this experiment can as well be obtained from ISIP's ftp site.

## 6. REFERENCES

[1] J. Picone, *ECE 8993 Fundamentals of Speech Recognition*, Department of Electrical and Computer Engineering, Mississippi State University, Mississippi, 1998.