Assignment 2, CS562, Fall 2012
Contact: Prof. Kevin Knight (knight@isi.edu), TA Hui Zhang (zhangh1982@gmail.com)
Due at the beginning of class, October 2, 2012

Language Modeling

Today's language translation and speech recognition systems are powered, in large part, by **n-gram language models**. These models assign probabilities to proposed word sequences. The probability of an individual word token is based on the preceding words -- in the case of a 3-gram model, the two preceding words.

In this assignment, you will build n-grams models for scoring **character sequences**, not word sequences. We thus skirt some scaling issues, but the basic principles are the same. Our training and testing sequences consist of 27 distinct tokens -- the 26 letters of English, plus space.

The following character sequences are provided:

TRAIN (57,217 characters total) -- used for collecting n-gram statistics
HELDOUT (9,464 characters total) -- used for smoothing n-gram models
TEST (9,280 characters total) -- used for evaluating n-gram models

These corpora can be downloaded from the Assignment 2 folder on Blackboard.

A blind test corpus exists, but will not be provided in advance:

BLINDTEST -- based on September 2012 news articles, used for final evaluation

Part 1. Using TRAIN and/or HELDOUT, construct 1-gram, 2-gram, and 3-gram character language models. These models must not assign zero probability to any sequence. Store these models in Carmel's WFSA format. Evaluate your models on TEST. What probabilities do they assign? We will apply your models to BLINDTEST, and you will be graded in large part on how high a probability your models assign to that corpus.

To turn in:

- (USC Blackboard) Your three WFSAs. Please call the files: “uni.wfsa”, “bi.wfsa”, and “tri.wfsa”, put them in a directory called “<your-name>”, compress this directory into one zip file named “<your-name>.zip”, and submit the zip file to Blackboard. If you will need help with this, please contact the TA in advance.

- (on paper) The final sizes of your WFSAs in states & transitions. (Use: % carmel -c wfsa).

- (on paper) The corpus probabilities your WFSAs assign to TEST. (Use: % carmel -Sr wfsa TEST).

- (on paper) A description of your smoothing method. Your description should include the algorithms, whether/how you used HELDOUT, and what (if any) experiments you did before settling on your solution.

- (on paper) A sketch drawing of your 3-gram model in sufficient detail that someone could replicate it. Please consider this carefully -- examine your drawing after you have drawn it and evaluate whether someone (not you) could build the same WFSA you have built. We will consider it in the same light.

(over)

IMPORTANT NOTE #1: Every transition in your WFSA should have an *e* input symbol. So, your WFSA actually transduces the empty string into English (otherwise “carmel -Sr” will not work). **We have supplied a sample uni.wfsa** that demonstrates this.

IMPORTANT NOTE #2: To ensure that your WFSA represents a legitimate probability distribution, you should normalize it with "% carmel -HJmn your-wfsa >wfsa.norm". We will do this in any case.

IMPORTANT NOTE #3: To ensure that your WFSA represents a legitimate probability distribution, you should have a single final state *with no exiting transitions*. The sample uni.wfsa also demonstrates this.

Part 2. Vary the amount of TRAIN used to collect n-gram statistics. Plot the effects of TRAIN-size versus P(TEST). Do this for the 1-gram, 2-gram, and 3-gram models separately. Use the same smoothing strategy you used in Part 1.

To turn in:
- (on paper) Graphs of results.
- (on paper) A paragraph of your observations.

Part 3. Score these two sequences with your 3-gram model:
j o h n _ w e n t _ h o m e
j i h n _ u e n t _ h o m e

To turn in:
- (on paper) Which sequence is preferred, and by how much?