Homework 6: NLTK & Dictionaries

L435/L555

Due Tuesday, November 1

1. NLTK, ch. 3, #25, slightly modified (`http://www.nltk.org/book/ch03.html`)

   Pig Latin is a simple transformation of English text. Each word of the text is converted as follows: move any consonant (or consonant cluster) that appears at the start of the word to the end, then append *ay*, e.g. *string* → *ingstray*, *idle* → *idleay*. (`http://en.wikipedia.org/wiki/Pig_Latin`)

   (a) Write code to convert a word to Pig Latin.

   (b) Write code that converts text, instead of individual words. Use the NLTK tokenizer for this.

   (c) L555 only (extra credit for L435): Extend it further to preserve capitalization, to keep *qu* together (i.e. so that *quiet* becomes *ietquay*), and to detect when *y* is used as a consonant (e.g. *yellow*) vs a vowel (e.g. *style*).

2. NLTK, ch. 3, #29: Readability measures are used to score the reading difficulty of a text, for the purposes of selecting texts of appropriate difficulty for language learners. Let us define μ_w to be the average number of letters per word, and μ_s to be the average number of words per sentence, in a given text. The Automated Readability Index (ARI) of the text is defined to be: 4.71 μ_w + 0.5 μ_s - 21.43. Compute the ARI score for various sections of the Brown Corpus, including section `f` (popular lore) and `j` (learned). Make use of the fact that `nltk.corpus.brown.words()` produces a sequence of words, while `nltk.corpus.brown.sents()` produces a sequence of sentences.

3. Write a program that reads in the POS tagged text from file *vm.pos* (available from canvas). Using NLTK's FreqDist() utility, store every POS tag and its frequency.

4. Adapted from *Think Python*, exercise 11.1 (`http://www.greenteapress.com/thinkpython/html/thinkpython012.html`:

   Write a function that reads in a text from Project Gutenberg, divides the text into words using the NLTK tokenizer, and stores each word into a dictionary. It doesn't matter what the values are.