Methods in Computational Linguistics 1
LING 78100
Fall 2010

Homework #3 - Processing files
Due at 11:50 on October 29

1. Short Problems -- Can be answered in the interpreter

A) [5] Rewrite the following loop as a list comprehension

```
>>> result = []
>>> for word in sent:
...     word_len = (word, len(word))
...     result.append(word_len)
```

B) [15] Write a function shorten(text, n) to process a text, omitting any words that occur *n* or more times. For example: shorten("the quick brown fox jumped over the lazy dog", 2) would produce "quick brown fox jumped over lazy dog". You may assume that the text is all lowercase, but it may contain punctuation, so you will want to use nltk.word_tokenize(text) to divide the text into words.

2. Longer Programing Assignment -- Distinguishing Mother speech from Child speech. A set of Childes files are available from the course website. You may use any files you want for processing and evaluating.

Based on the Practicum Assignment exploring a Childes file

This assignment should be answered using stored python files.

A) [10] Practicum Problem: Write a program to separate mother speech and child speech (without the prefix '*CHI:' or '*MOT:') Lines in the file that don't start with either prefix should not be written to either the mother or child file.
This program should read one Childes file, and produce two files, one for mother speech and one for child speech. These three files names, the childes filename, the mother speech file name and the child speech filename should be delivered to your program through the command line. Your program should be called as follows:

```
python process_childes.py <childes_file> <mother_file> <child_file>
```

B) [10] Practicum Problem: Write a program that represents the mother and child's vocabulary, counting the instances of each item, based on two input files. You may use FreqDist to do this. Your program should read the files that you created in part A.

```
python process_vocabulary.py <mother_file> <child_file>
```

C) [15] Extend the program you wrote in part B to identify those words that are more or less valuable in indicating that speech is produced by a child or a mother. There are many ways that you can identify these lists, but frequency of usage is a good place to start. Use this to generate two lists of words, one indicating child speech and one indicating mother speech. These lists can be as long or as short as you like.

D) [10] Using the two word lists, generated in part C, create a function that determines if a new sentence is more likely to be spoken by a mother or a child. You may copy the lists generated in part C by hand into a new python function.

E) [5] Process a new childes file as in part A, creating two new mother and child files.

F) [25] Write a program using the function you wrote in part D to evaluate the sentences contained in the new files created in part E. The program should output what percentage of mother sentences were correctly identified by your function and what percentage of child sentences? Additionally, print out any mistakes that the program makes

```
python evaluate_childes.py <mother_file> <child_file>
```

G) [5] Discuss the mistakes that your program makes. Are there any consistent errors? Try evaluating your program on a different childes file. Are the errors similar or different? How might these be remedied? Note: there may not be a clear consistent source of errors. If not, discuss specific errors and why the system made a mistake.