# ITCS 4111/5111 Introduction to Natural Language Processing

**Assignment 1** **Due: September 15th, 2017 11:59 pm**

In this assignment, we will lay the foundations of all programming assignments going forward.

Before starting working on the problems in this assignment, you should download and install Python and NLTK on your machine. (http://www.nltk.org/)
In addition, you should download and install the NLTK Data (http://www.nltk.org/data.html).

If you are not familiar with Python and/or NLTK, this tutorial is a good place to start: http://www.nltk.org/book/

In addition, download and install R Studio and R Markdown on your machine.

Your submissions should be in a single .py file with the following naming convention: lastname_assignment1.py and the written/output part should be in a file named lastname_assignment1.docx (or .pdf). All submissions should be uploaded to Canvas. Email submissions will not be considered for grading.

The problems marked with ** are optional for students enrolled in the ITCS 4111 section of this course, all other problems are mandatory. ALL problems are mandatory for the students enrolled in ITCS 5111.

**Problem 1 (Python/NLTK warmup) (5 points).** Define a variable `funny` to contain the string "colorless green ideas sleep furiously" (from Chomsky's famous phrase). Now write code to perform the following tasks:

a. Write a function to split `funny` into a list of strings, one per word, using Python's split() operation.

b. Write a function to extract the second letter of each word in silly and join them into a string and output that string.

c. Write a function to build a list called `phrases` consisting of all the words up to (but not including) *sleep* in `funny`.
Hint: use the index() function in combination with list slicing.

d. Write a function to combine the words in `phrases` back into a single string, using join(). Make sure the words in the resulting string are separated with whitespace.

e. Write a function to print the words of `funny` in alphabetical order, one per line

**Problem 2 (Python/NLTK warmup) (3 points).** Write a function that takes a sentence expressed as a single string, splits it and counts up the words. Get it to print out each word and the word's frequency, one per line, in alphabetical order.

**Problem 3 (Regex warmup/no code) (2 points).** Write regular expressions to match the following classes of strings:
a. A single determiner (assume that a, an, and the are the only determiners).
b. An arithmetic expression using integers, addition, and multiplication, such as 5*6+1.

**Problem 4 (Python/NLTK/Regex) (5 points).** Using re.findall(), write a regular expression which will extract pairs of values of the form login name@email domain from the following string:

```
>>> str = """
... austen-emma.txt:hart@vmd.cso.uiuc.edu (internet) hart@uiucvmd (bitnet)
... austen-emma.txt:Internet (72600.2026@compuserve.com); TEL: (212-254-5093) .
.. austen-persuasion.txt:Editing by Martin Ward (Martin.Ward@uk.ac.durham)
... blake-songs.txt:Prepared by David Price, email ccx074@coventry.ac.uk
... """
```

**Problem 5 (Python/NLTK/Regex) (10 points).** Write a function to eliminate duplicate lines from an input file. Assume the input file contains strings with alphanumeric characters.

**Problem 6** (Python/NLTK/Regex) (10 points).** What does the following Python program do? What would be an example input file? What would the output be?

```
import sys
import re

try: infile = open(sys.argv[1])
except: infile = sys.stdin
for line in infile:
      line = line.strip()
      if not line or line[0] == '#': break
      parens=re.sub('[^()]','.',
                  line.replace('(','(.').replace(')', '.)'))
      try: matchparens = re.compile(parens)
      except:
            print >> sys.stderr, "Error in input"
            continue
      for t in matchparens.match(line).groups(): print t
```

The next problem gives you hands-on practice with morphological analysis. We are going to use morphological analysis to analyze the first presidential debate from the 2012 election. The debate is part of the homework files (debate.txt).

**Problem 7**. **(Morphological analysis/Regex/NLTK)** **(65 points).**

a. **(15 points)** Write a function to load the debate file and assign statements to speakers. There are three main speakers: Obama, Romney and Lehrer but not statements made by the speakers begin with the speaker tag. Your function should have a rule to concatenate statements made by the same speaker. Your code should ignore crosstalk and notes about audience behavior.

b. **(15 points)** Use the porter, snowball and lancaster stemmers from the nltk package to stem each of the speaker statements.

   Note: You will need to do the following preprocessing steps on the statements BEFORE you apply the stemmers:
   - Discard punctuation
   - Remove capitalization
   - Remove stop words (use NLTK stopwords)
   - Tokenize the words
   - Apply each of the stemmers

c. **(5 points)** Using the results, output the list of 10 most frequent words (stemmed) used by each speaker separately based on each of the stems.

   Note: Your output may be different from another student's due to any differences in the way that speaker statements are merged in part a. above. Any such differences are ok.

The above three sub-problems allow us determine which words are most often used by each speaker. What differences do you observe between the different stemmer outputs?

Next, we want to know which speaker might use positive words most often.

d. **(10 points)** Write a function to load the dictionary of positive words provided with this homework (positive.txt). Use the porter stemmer to stem the words in this dictionary.

e. **(10 points)** Using the stemmed statements of each speaker from part b. above (only the porter stemmer output), determine which speaker uses the positive words listed in the positive word dictionary most often. Print the 10 most frequent positive words used by each speaker.

f. ****(10 points)** Create a visualization using R that compares the overall positive word rate for Obama, Romney, and Lehrer. What patterns do you notice? There is no one right answer, be creative! Output for this problem should be in an R markdown file and the corresponding html should be submitted (lastname_assignment1.html).

Note: ITCS 5111 students will be graded out of a 100 points total for all problems in this assignment. ITCS 4111 students will be graded out of 80 points total for all problems in this assignment, except for those marked with **.