CSCI 3907/6907
Intro to Statistical NLP
Homework Assignment #1

Due: Sept 19th 2017, 11:59pm EST

Instructor: Yassine Benajiba, yassine@gwu.edu
TA: Ali Seyfi, seyfi@gwmail.gwu.edu

I. General Instructions

1. Please submit the assignment through Blackboard
2. If you have questions about the assignment, please post them to Piazza or Blackboard. Please submit questions way in advance of due date.
3. *No grace period is allowed for this first assignment*
4. *In preparation for this assignment you would attend the offered tutorials* on Unix Shell programming and/or Python for NLP
5. Please include a README file that has the following aspects:
    a. Your name and email address
    b. Homework number
    c. A description of every file in your solution, the programming language used, supporting files, any additional resources used, etc.
    d. How your system operates, in detail.
    e. A description of special features (or limitations) of your system.
    f. How to run your system, with a sample command line.
    g. Within Code Documentation:
        i. All environmental variables should be set appropriately within the program.
        ii. Methods/functions/procedures should be documented in a meaningful way. This includes informative method/ procedure/ function/ variable names as well as explicit documentation.
        iii. Efficient and correct implementation
        iv. Don't hardcode things that should be variables, etc.
        v. Choose meaningful names for your variables

6. For programming assignments,
    a. The code should be written in Perl/Python/JAVA/C++
    b. The code should compile on the SEAS machines
    c. We will not debug the code
    d. You must write the code yourself. Do not use publicly available code or copy code from any other source and do not let anyone else look at or use your code (please refer to the Academic Integrity policy below and on the course syllabus or ask the instructor or TA if you have questions in this regard). Please check this explicitly before submitting your assignment.

        Academic Integrity
        Copying or paraphrasing someone's work (code included), or permitting your own work to be copied or paraphrased, even if only in part, is not allowed, and will result in an automatic grade of 0 for

the entire assignment or exam in which the copying or paraphrasing was done. Your grade should reflect your own work. If you believe you are going to have trouble completing an assignment, please talk to the instructor or TA in advance of the due date.

7. If you wish to use any additional tools please check with the TA first by posting a question to Blackboard or Piazza.
8. This assignment counts towards 7% of your overall grade

II. Before Starting

The United Nations Corpus

In this assignment, you will make use of the United Nations (UNCorpus), a corpus on the UN general assembly resolutions. The UNCorpus is a six-language parallel text in Arabic, Chinese, English, French, Russian and Spanish. The following paper describes the corpus:

> Alexandre Rafalovitch and Robert Dale. 2009. United Nations General Assembly Resolutions: A Six-Language Parallel Corpus. In Proceedings of the MT Summit XII, pages 292---299, Ottawa, Canada.
> URL: http://www.uncorpora.org/Rafalovitch_Dale_MT_Summit_2009.pdf

You can download the text of the UNCorpus from:

http://www.uncorpora.org/files/uncorpora_plain_20090831.zip

Unzipping the file produces a text file named **uncorpora_plain_20090831.tmx**. This file will be referred to as the UNCorpus in the rest of this document.

Unix Tools

Revise the usage of the following Unix commands (and some of their specific options), which you will need in this assignment: *cat, wc, sort (sort –nr), uniq (uniq – c), grep (grep –e; grep –a),* and *more*. You can use the *man* command to check the usage from any Unix terminal (eg. *man cat*). Other Unix commands you may want to consider checking are: *less, tr and sed*.

Additionally, revise the use of the pipeline and I/O redirections (| and >, specifically). For a quick introduction, see
http://www.westwind.com/reference/os-x/commandline/pipes.html

Finally, we recommend using the PERL interpreter in a Unix command pipeline mode to apply regex substitutions: *perl –pe '<substitute---regex>;'*. It is more powerful than *sed* or *tr* commands.

Regular Expressions

Revise the regular expression definitions in Chapter 2 in J+M Book. There is a cheat sheet in the inside cover of the book. Here is another link to a different cheat sheet also:
http://web.mit.edu/hackl/www/lab/turkshop/slides/regex-cheatsheet.pdf

Another useful resource to play with is http://www.regexr.com/.

QUESTIONS

Q1. (10 points) Write regular expressions that match the following patterns – This doesn't have to be a program, submit this as a theoretical answer (create a pdf file with the answers):

a) An odd digit followed by an even digit
(eg. 14 or 38)

b) A letter followed by a non-letter followed by a number
(eg. A#1)

c) A word that starts with an upper case letter and ends with some punctuation mark (`!', `?', `,', `;', `:', etc.)
(eg. Microphone!)

d) The string "ping" in any combination of upper and lower cases letters
(eg. In a sentence such as "Map*ping* the address was complex, so I *ping*ed and *ping*ed him time and again, gri*ping*. *Ping*ing him constantly seemed to annoy him")

e) A date in the form of one or two digits, a dot, one or two digits, a dot, two digits
(eg. 2.16.13, 03.15.70, 3.5.78)

Q2. (15 points) Write a program that reads a string from the standard input, and uses a regular expression to test whether the string is a valid IP address. (eg. 192.62.0.255 is a valid IP address, but 189.9,32 is not)

Q3. (15 points) Write a program that reads a number of variable length (eg. 73618, 829, 1, 980), adds up all the digits (would yield 25, 19, 1, 17, respectively), and displays the result.

Q4. (20 points) Write a program that simply tokenizes the following text, i.e., by separating the punctuation from the words (,.?!:.....). The program should display the output tokenized. You should try to address special cases such as abbreviations (do not separate the punctuation in an abbreviation, A.B.C., should **not** yield A . B . C .), apostrophe (should be attached to the letter that follows, e.g., tokenize Peter's as Peter 's), etc.

"Predictions suggesting that large changes in weight will accumulate indefinitely in response to small sustained lifestyle modifications rely on the half-century-old 3,500 calorie rule, which equates a weight alteration of 2.2 lb to a 3,500 calories cumulative deficit or increment," write the study authors Dr. Jampolis, Dr. Chaudry, and Prof. Harlen, from N.P.C Clinic in OH. The 3,500-

calorie rule "predicts that a person who increases daily energy expenditure by 100 calories by walking 1 mile per day" will lose 50 pounds over five years, the authors say. But the true weight loss is only about 10 pounds if calorie intake doesn't increase, "because changes in mass ... alter the energy requirements of the body's make-up." "This is a myth, strictly speaking, but the smaller amount of weight loss achieved with small changes is clinically significant and should not be discounted," says Dr. Melina Jampolis, CNN diet and fitness expert.

Q5. (20 points) Write a program that reads a series of numbers and it displays the digits in ascending order, together with their frequency. The program will also display the number of numbers in the series (Tokens) and the number of unique numbers (Types) in the series

E.g., for the input: 4 9 2 7 2 0 2 0 13 23 13. The program will display:

```
0 2
2 3
4 1
7 1
9 1
13 2
23 1
Tokens: 11
Types: 7
```

Q6. (20 points) Repeat Q5 above for the text in Q4 after tokenizing it. I.e. write a program to count the number of strings in the text and sort them alphabetically in ascending order, and also produce the number of tokens and types in the text. (Don't forget to count the punctuation marks).

Q7. (5 points) Ignoring letter case, how many lines of text in the English UNCorpus mention the term Human Rights?

Q8. (10 points) The Full UNCorpus

Answer the following questions using Unix commands and regex only. Each question should be answered with one command line (possibly consisting of multiple piped Unix commands)

a. How many lines does the UNCorpus file have?
b. How many segments <seg>?
c. How many non-segments? As in tags that are not <seg> like <tuv>?
d. How many English segments does the text have?
e. How many segments exist for each languages (Chinese, Arabic,...)? (again, done in one command)

Q9. (20 points) The English UNCorpus

Answer the following questions using Unix commands and regex only. Each question should be answered with one command line (possibly consisting of multiple piped Unix commands)

a. Extract the text without XML for only the English segments and put in a file called “uncorpus.eng.txt” (Hint, use “grep –a1”). The rest of the questions are about this file. How would you verify that you did not miss any lines?

b. Count the total number of words (tokens).
c. Count the total number of unique words (types).
d. Count the total number of unique words ignoring capitalization
e. Count the total number of pure digits tokens.
f. Count the total number of digits with non-word characters with them (e.g. 8,000.00).
g. Count the total number of words starting with capital letters.
h. What are the top 15 most common first words of sentences
i. What are the top most common capitalized words (that are not sentence initial).
j. Count all occurrences of Roman numerals

Q10. (10 points) This question uses the file “uncorpus.eng.txt” as the corpus. Compare the four sets of top 10 and the four sets of bottom 10 words. What words are similar, or different? Are you surprised (or not surprised) by the results? (why?)

Q11. (10 points) Back to the Original Corpus

a. Get the top 20 (most frequent) words in English, Arabic, Spanish and Russian of the UNCorpus. You will need four separate commands. Show the lists of words in your answer. For this task, consider a word to simply be white-space delimited (i.e. keep all punctuation and digits and separate on white space).

b. Use Google Translate to compare the meanings of these words to the English top and bottom words. What words are similar, what are different? Show your work including the results of Google Translate. You can list them in two tables: one for the top words and one for the bottom words.