CIS/STA 9665: Assignment 2

Applied Natural Language Processing

Due Date: 11:59 pm Sep 25

Guidelines:

- Use Python as a programming language and finish this assignment in Jupyter Notebook
- Work is to be done individually for this assignment
- Students handing in similar work will both receive a grade of 0 and will face disciplinary actions.

Chapter 3. Processing Raw Text

1. Describe the class of strings matched by the following regular expressions. (No code is needed and just describe what the following regular expressions do/match).

a) [a-zA-Z]+

b) [A-Z][a-z]*

c) p[aeiou]{,2}t

d) \d+(\.\d+)?

e) ([^aeiou][aeiou][^aeiou])*

f) \w+|[^\w\s]+

2. Rewrite the following loop as a list comprehension:

```
sent = ['This', 'is', 'an', 'introduction', 'class']
result = []
for word in sent:
    word_len = (word, len(word))
    result.append(word_len)
result
```

3. Read in some text from your own document in your local disk, tokenize it, and print the list of all wh-word types that occur. (wh-words in English are used in questions, relative clauses and exclamations: who, which, what, and so on.) (hint: import nltk; from nltk import word_tokenize)

4. Create your own file consisting of words and (made up) frequencies, where each line consists of a word, the space character, and a positive integer, e.g. fuzzy 53. Read the file into a Python list using open(filename).readlines(). Next, break each line into its two fields using split(), and convert the number into an integer using int(). The result should be a list of the form: for example, [['fuzzy', 53], ...].

5. 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 μw to be the average number of letters per word, and μs 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: $4.71 \mu w + 0.5 \mu s - 21.43$. Compute the ARI score for each section of the Brown Corpus (i.e. News, Editorial,… Humor) (Hint: for category in brown.categories( )). 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. (Hint: from nltk.corpus import brown)

6. Use the **Porter Stemmer** to normalize some tokenized text (see below), calling the stemmer on each word. Do the same thing with the **Lancaster Stemmer** and describe any difference you observe by using these two stemmers.

text='Technologies based on NLP are becoming increasingly widespread. For example, phones and handheld computers support predictive text and handwriting recognition; web search engines give access to information locked up in unstructured text; machine translation allows us to retrieve texts written in Chinese and read them in Spanish; text analysis enables us to detect sentiment in tweets and blogs. By providing more natural human-machine interfaces, and more sophisticated access to stored information, language processing has come to play a central role in the multilingual information society'.

7. Obtain raw texts from two or more genres and compute their respective reading difficulty scores as in the earlier exercise on reading difficulty. Please compare the reading difficulties for ABC Rural News ("rural.txt") and ABC Science News("science.txt") (nltk.corpus.abc).(Hint: from nltk.corpus import abc)

8.Rewrite the following nested loop as a **nested list comprehension**. You need to use regular expression.

```
words = ['attribution', 'confabulation', 'elocution', 'sequoia', 'tenacious', 'unidirectional']
vsequences = set()
for word in words:
    vowels = []
    for char in word:
        if char in 'aeiou':
            vowels.append(char)
vsequences.add(".join(vowels))
sorted(vsequences)
```

9. Try to refer the following sample code to print the following sentences in a formatted way. (Hint: you should use str.format() method in print() and a for loop； for more information, please read the textbook section 3.9 in Chapter 3).

Output should look like:

```
The Tragedie of Hamlet was written by William Shakespeare in 1599
Leaves of Grass        was written by Walt Whiteman       in 1855
Emma                   was written by Jane Austen         in 1816
```

# Sample Code:

```
template = 'Lee wants a {} right now'
menu = ['sandwich', 'spam fritter', 'pancake']
for snack in menu:
    print(template.format(snack))
```

10. Define the variable quote to contain the list ['Action', 'speaks', 'louder', 'than', 'words']. Process this list using a for loop, and store the length of each word in a new list lengths. Hint: begin by assigning the empty list to lengths, using lengths = []. Then each time through the loop, use append() to add another length value to the list. Then do the same thing using a list comprehension.

What to Submit

a. Use Python as a programming language and finish this assignment in Jupyter Notebook

b. I have created an ipynb file with questions. Please add your code and answers in this ipynb file

c. After completion, please save your finalized ipynb file as a PDF file

d. Submit both PDF file and ipynb file to Blackboard

e. Please answers questions clearly, concisely, and completely. To answer some questions, the code is not sufficient. You should complement your answers in words by using comments (#)

f. The assignment will be graded on the correctness of the answers, comprehensiveness of the analysis, clarity of results' presentation and neatness of the report.