CIS/STA 9665: Assignment 2

Applied Natural Language Processing

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\w*

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 use the regular expressions to 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) (use lower() and set()). Please use lower() to normalize the text to lowercase.

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 μw + 0.5 μs - 21.43. Compute the ARI score for each section of the Brown Corpus (i.e. News, Editorial,…, Humor). Please use two ways introduced in the class to calculate the average number of letters per word μw and the average number of words per sentences μs. You are supposed to get two different results.

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. Finally, please try Lemmatization by using WordNet Lemmatizer and describe any difference you observe compared to Porter Stemmer and Lancaster Stemmer.

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. Please try to retrieve some text from any web page in the form of HTML documents, tokenize the text, create an NLTK text and then output the most frequent 10 words. Please get rid of punctuation symbols, numbers, stopwords and any words that only differ in capitalization (lower())

8. Rewrite the following nested loop by using a list comprehension and regular expressions:

```
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 exactly 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 Hint:

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

10. Please first use sentence tokenization to print out the first 10 sentences in the "carroll-alice.txt" in the Gutenberg corpus. And then use basic corpus functionality sents() to return the first 10 sentences in this book. Please describe the difference between the two results. (hint: use sent_tokenzie (), pprint(), sents())

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.