Activity: Manipulating Text

Due Friday, February 10th at 12:01pm

Below are two activities to help emphasize regular expressions.

Question 1 [4 pts]
For each question, use regular expressions and python to perform each operation.

a. Print every line to find every line with the mention of “Twitter”
b. Use Python substitute every mention of “Twitter” with “MyStartup”.
c. Create a regular expression to find an address. That is, a string that starts in a set of numbers (the street number), and ends in a zip code (five decimal digits).
d. Create a regular expression to print either the main header line or the 4th level indent.

Question 2 [4pts]
Below are several spellings of the holiday Hanukkah. Each spelling is used somewhere. Create a regular expression that can find each of these spellings.
Hanukkah
Chanukah
Hanukah
Hannukah
Chanuka
Chanukkah
Hanuka
Channukah
Chanukka
Hanukka
Hannuka
Hannukkah
Channuka
Xanuka
Hannukka
Channukkah
Channukka
Chanuqa

Prompt: For the following questions, extract the twitter privacy policy statement using the linux command below. This command will extract twitter text and remove some extraneous information. Assume the processed text is saved in a file called *twitter.txt*. The line below can help download the document and format it appropriately.

w3m -dump -o display_charset=UTF-8 https://dev.twitter.com/overview/terms/agreement-and-policy | sed -n '/^Developer Agreement/,/Solutions$/ p'> twitter.txt

Be careful to respect the character encoding of the document. The usable content starts with the line that says “Developer Agreement” and ends with the line that says “Solutions”. You can refer to the sample output and the template to get started.

Question 3. [5 pts]
Use Python and the NLTK library to create a CSV file where each line of the CSV file contains a "sentence number þ sentence þ avg word size". Notice instead of the traditional comma we use the þ (Thorn) separator. Ensure that all spurious new lines in each row is removed. Hint: it is suggested to utilize the *sent_tokenize* function. Quotes around terms are optional.

Question 4. [5 pts]
Use Python and the NLTK library to transform the file or the CSV file into an XML file format as sampled below. There should be a single <sentence/> tag for each sentence extracted. Also a <text/> tag for the text part of the sentence and the average size of each word in the sentence within the <avg> tag. The id attribute inside of a sentence should contain the number sentence of that document.

```
<document>
 <sentences>
 <sentence id="">
  <text></text>
  <avg></avg>
  </sentence>
 </sentences>
</document>
```

Question 5. [5 pts]
 # Use Python or NLTK to transform the information in the above xml document to a valid json document. Below is an example of what the output should look like.

```
{
   "documents": {
      "sentences": [
         {
            "avg": 5,
            "id": 1,
            "text": "This is the sentence"
         },
         {
            "avg": 3,
            "id": 2,
            "text": "This is the sentence"
         }
      ]
   }
}
```