Demi Chu & Victoria Nguyen
Professor Li
MIS3640: Problem Solving with Software Design
Due: October 30th, 2019

Assignment 2:
Analyzing Programming Languages' Wikipedia Pages

Project Overview:

For this assignment, our team decided to explore the data source Wikipedia to analyze the text within the encyclopedia pages for the following three computer languages: Python, R, and C. In order to create these analyses, we will examine the word frequencies from the content section of the page and create a sentiment analysis using both the content and summary sections. As these are some of the most popular computer languages, we hoped to discover what similarities and differences the languages had in terms of word choice and biases.

Implementation:

In order to analyze the Wikipedia pages, we installed MediaWiki which allowed us to directly import the Wikipedia page content texts. We imported all three programming languages' Wikipedia Page and created a function to tokenize all the words into lists. However, tokenizing words into a new list is a new concept we have not learned in class so we did research on how to tokenize, remove punctuation, and put everything into lists.

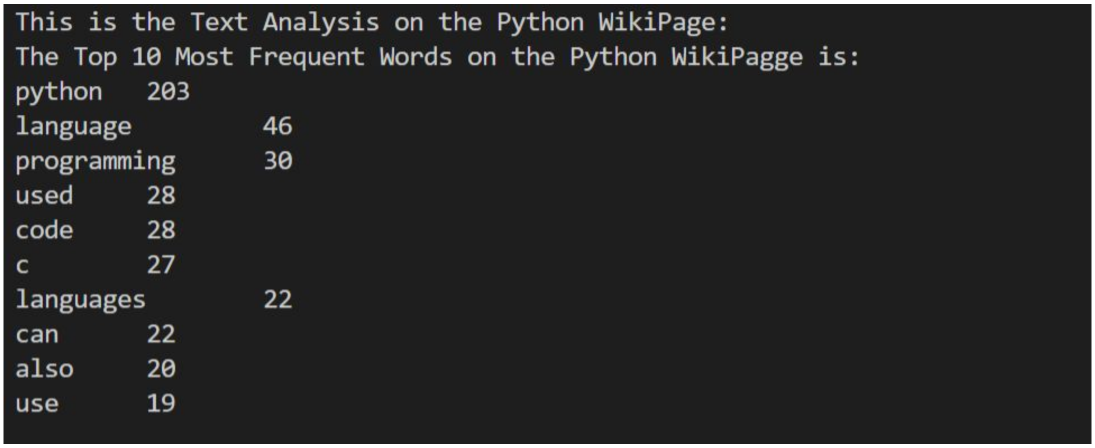
Two major components we focused on were using lists and dictionaries so that it would be easier for us to compare the most frequent words on each Wikipedia page. It was important for us to use functions that allowed us to strip stopwords (ex. the, is, and, a, etc.) from our lists, so our results for word frequencies would not be inaccurate. Furthermore, dictionaries helped us display the word and its number of frequencies side by side.

A design decision we had to make was how we wanted to print the dictionary consisting of all the words and its frequencies. Instead of just printing the dictionary with its original format, we were considering on creating our own format so the words and its frequencies would print on different rows for better visualization.

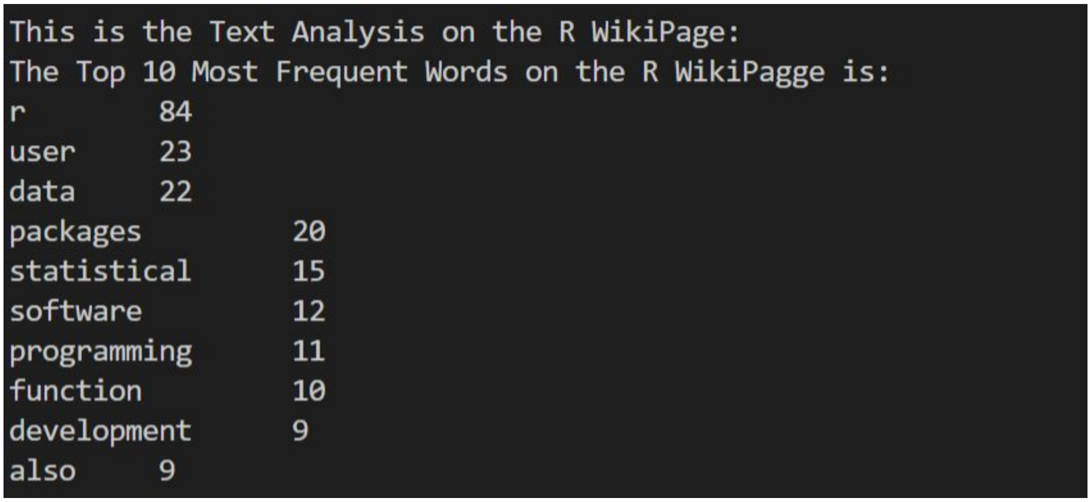
Results:

Text Analysis — Word Frequencies

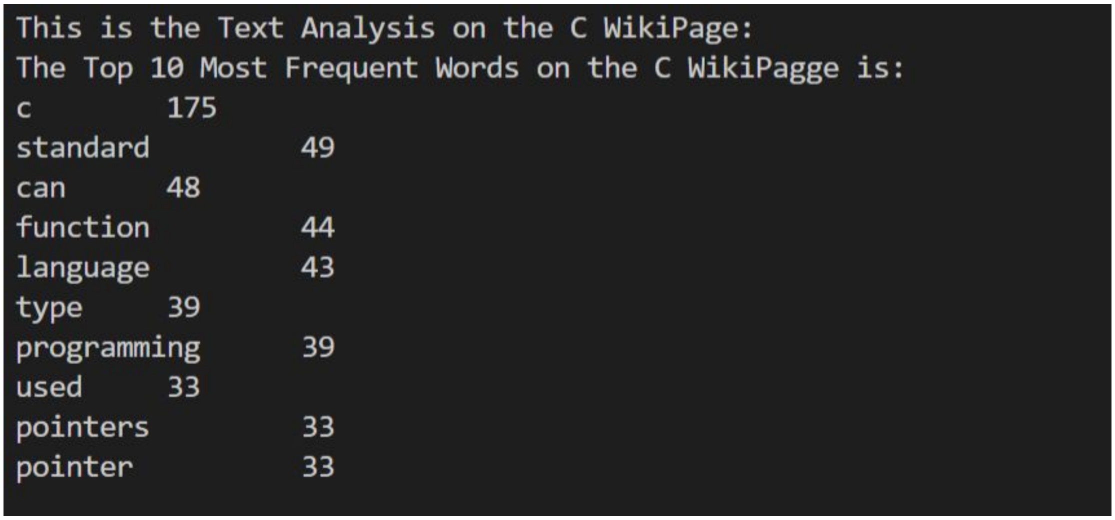
As expected, the most common word within an encyclopedia page should be the topic itself. Because of this, the top three words within the Top 10 Most Frequent Words list created were "python", "r", and "c". But, it was even more interesting to see the crossover between the top words, meaning we can see how the pages may intermingle with one another. For example, the list for Python ended up mentioning "c" as the sixth most frequently used word. As such, with all the three topics being computer languages, our team was surprised to find the "programming" as the only word that was featured across the lists.



```
This is the Text Analysis on the Python WikiPage:
The Top 10 Most Frequent Words on the Python WikiPagge is:
python   203
language         46
programming      30
used     28
code     28
c        27
languages        22
can      22
also     20
use      19
```



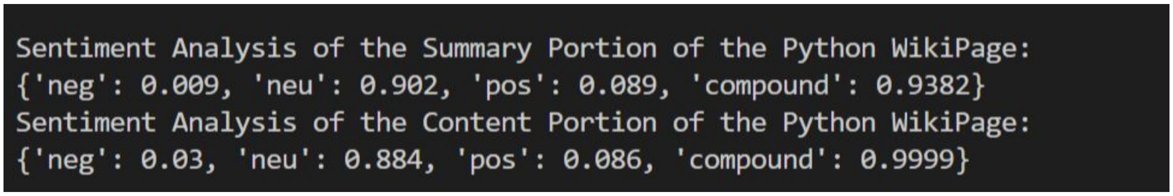
```
This is the Text Analysis on the R WikiPage:
The Top 10 Most Frequent Words on the R WikiPagge is:
r        84
user     23
data     22
packages         20
statistical      15
software         12
programming      11
function         10
development      9
also     9
```



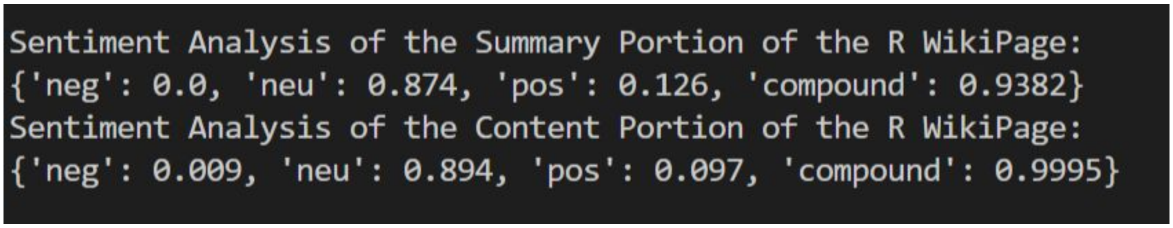
```
This is the Text Analysis on the C WikiPage:
The Top 10 Most Frequent Words on the C WikiPagge is:
c        175
standard         49
can      48
function         44
language         43
type     39
programming      39
used     33
pointers         33
pointer          33
```

Sentiment Analysis

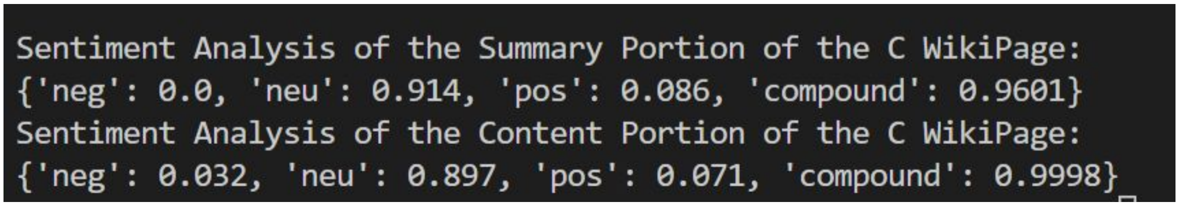
Holistically comparing the three languages, our team made the assumption that popularity would correlate to the overall impressions and biases within the articles. For example, as Python is the most popular computer language as of 2019, it would result in a greater positive sentiment analysis. As such, its text would include keywords that would describe characteristics such as the ease of its writing syntax, its vast number of libraries, etc. Whereas, C and more so, R, not being as popular in comparison to Python would have a higher neutral point in its overall sentiments. But, our results garnered a different output than expected as both the positive sentiments for R ended up with the highest number and its negative sentiments being the lowest across the three. The difference may seem marginal as the difference is not substantially more or less in terms of positivity and negativity, but is still a finding to take note of when looking past the neutral point being the greatest overall.



```
Sentiment Analysis of the Summary Portion of the Python WikiPage:
{'neg': 0.009, 'neu': 0.902, 'pos': 0.089, 'compound': 0.9382}
Sentiment Analysis of the Content Portion of the Python WikiPage:
{'neg': 0.03, 'neu': 0.884, 'pos': 0.086, 'compound': 0.9999}
```



```
Sentiment Analysis of the Summary Portion of the R WikiPage:
{'neg': 0.0, 'neu': 0.874, 'pos': 0.126, 'compound': 0.9382}
Sentiment Analysis of the Content Portion of the R WikiPage:
{'neg': 0.009, 'neu': 0.894, 'pos': 0.097, 'compound': 0.9995}
```



```
Sentiment Analysis of the Summary Portion of the C WikiPage:
{'neg': 0.0, 'neu': 0.914, 'pos': 0.086, 'compound': 0.9601}
Sentiment Analysis of the Content Portion of the C WikiPage:
{'neg': 0.032, 'neu': 0.897, 'pos': 0.071, 'compound': 0.9998}
```

Reflection:
In terms of the actual code itself, our team faced two major issues that both pertained to the set-up process to run the assignment. The package manager “pip” was unable to be installed on one of our computers, making the initial process to even begin the analysis difficult. Furthermore, with updates constantly implemented, our team did not realize that the initial “wiki” import to access Wikipedia’s information through an API key was outdated and changed to the new label of “mediawiki”. For these setbacks, we communicated a lot with the Professor but should try to initiate more research on our own to improve our debugging skills. Conversely, our team learned the very helpful function of tokenizing. This was a crucial skill learned that majorly increased the efficiency along with the simplicity of the code. It simplified the process of stripping the words down to insertable text to analyze on visual studio. All in all, our team’s biggest hurdle stemmed from understanding how to begin the assignment in terms of topic choices, installation, and choice of analysis. But after this point, we worked well to constantly communicate and meet with one another to utilize previous codes along with new ones to compile a concise analysis on Wikipedia pages. After choosing, we began our initial code to run the page “wikipedia” from Wikipedia. We later chose to narrow the topic down by comparing three computer languages as it aligned with the topic of the class. It was fun to use python to learn more about python!