Data Extraction And Data Preprocessing

I extracted/scrapped the text data from an URL using beautifulsoup.

I had extract the text data from title separately and text data from the article differently

To store the extracted text data from the article in the ‘article’ variable I used for loop and append function.

I then added the list of text data from the title and the list of text data from the article and stored it into the variable ‘data’.

I joined and converted all the text data into lower case and then stored it into another variable ‘data2’ and ‘data2\_lower’.

I applied word tokenization and sentence tokenization to data2 using nltk and then stored the respective outputs in ‘data2\_lower\_tokenized\_word’ and ‘data2\_lower\_tokenized\_sentence’ respectively.

Then I removed all the punctuations from data2\_lower\_tokenized\_word and stored the output in

‘data2\_lower\_tokenized\_word\_no\_p’.

Then I removed all the stopwords from data2\_lower\_tokenized\_word\_no\_p which were mentioned in the stopwords.txt(I integrated all the stopword txt files into 1 txt file i.e. stopwords.txt for easier further implementations) and stored the output in ‘data2\_lower\_tokenized\_word\_no\_p\_no\_sw’.

(Since I don’t have that much space in my system I didn’t save the extracted text data into text files. Also I didn’t require to save the extracted text data into a text file for further implementation I skipped that part, but if you want to check if I know how to save a list to a text file, I used the following block of code to save the extracted text data from the URL\_ID = 150 into 150.txt:

with open("150.txt", "w") as output:

output.write(str(data2))

I will also be submitting 150.txt.)

Importing Positive Words and Negative Words

I imported the positive-words.txt and negative-words.txt files and converted them into respective lists, converted them to lower case and then stored those respective lists in ‘positivewords’ and ‘negativewords’ respectively.

Text Analysis

I defined functions to calculate positive score, negative score, polarity score, subjective score, average sentence length, percentage complex word, fog index, complex word count, average word length.

To which i gave data2\_lower\_tokenized\_word\_no\_p\_no\_sw, data2\_lower\_tokenized\_word\_no\_p\_no\_sw, PositiveScore and NegativeScore, PositiveScore and NegativeScore, data2\_lower\_tokenized\_word\_no\_p\_no\_sw and data2\_lower\_tokenized\_sentence, data2\_lower\_tokenized\_word\_no\_p\_no\_sw, AverageSentenceLength and PercentageComplexWord, data2\_lower\_tokenized\_word\_no\_p\_no\_sw, data2\_lower\_tokenized\_word\_no\_p respectively as the input for these functions and stored the outputs in ‘PositiveScore’, ‘NegativeScore’, ‘PolarityScore’, ‘SubjectiveScore’, ‘AverageSentenceLength’, ’PercentageComplexWord’, ‘FogIndex’, ‘ComplexWordCount’, ‘AverageWordLength’ respectively.

For calculating word count, I applied len on data2\_lower\_tokenized\_word\_no\_p\_no\_sw and stored the output in ‘WordCount’.

For calculating Average Number Of Words Per Sentence, I used len(data2\_lower\_tokenized\_word\_no\_p)/len(data2\_lower\_tokenized\_sentence)

And stored the output in ‘AverageNumberOfWordsPerSentence’

(here I didn't used the dataset with stopwords removed as in the Text Analyis.docx file, "total number of words"/total number of sentences is mentioned, but we can simply substitute the variable containing the dataset with stopwords removed by just replace the variable name to data2\_lower\_tokenized\_word\_no\_p\_no\_sw.)

For calculating Syllables Per Words I imported a library called ‘syllapy’ and used it with for loop on data2\_lower\_tokenized\_word\_no\_p\_no\_sw and stored the output into ‘SyllablePerWord’ using the append function.(This returns a list of the number of syllables per word, I was not sure if i need to submit an average number of syllables per word or syllables per word. I submitted syllables per word as it was written in the Text Analysis.docx. So I defined a variable ‘AverageSyllablePerWord’ where I store the output of sum(SyllablePerWord)/len(SyllablePerWord). This returns the average number of syllables per word.)

For calculating the count of personal pronouns I used regex and len(data2). I defined a variable ‘PronounCount’ to store the count of personal pronouns.

After this I substituted different URLs from the Input.xlsx one by one in the url variable and then copy pasted all the necessary outputs into the Output.xlsx file. The URLs which I highlighted with green are the URLs which return error 404 when accessed.