Darshay Blount

CAP6307 – Text Mining

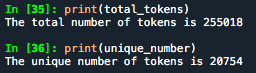
Assignment 1

Deliverability Report

Part I – Moby Dick

1. How many tokens and unique tokens in the text (words and punctuation symbols)?

The total number of tokens in the text is 255,018 and the number of unique tokens is 20,742. This was determined using the Natural Language Toolkit (nltk) in Python. Nltk computes the total number of tokens using the frequency distribution.



1. Apply lemmatizations on the verbs in the text, recalculate the number of tokens and unique tokens.

The total number of verbs tokens is 33,440. The number of unique verb tokens is 5,774. To compute the number of verbs, I used the part of speech tag (nltk.pos\_tag) to assign the tokens to their parts of speech. The tag for verbs are 'VB', 'VBD', 'VBG', 'VBN', 'VBP', and 'VBZ'. Then, I filtered the original list for just verbs to obtain the number of tokens with a verb tag. To determine the unique verb tokens, I used the nltk.FreqDist.



1. What percentage of tokens is ' HISTORY' or 'history'?

The percentage of tokens that is 'HISTORY' or 'history' is 0.00745%. I created two functions: one called ucFilter, to filter for occurrences of HISTORY, and lcFilter to filter for occurrences of ‘history.’ The function used a regular expression to filter words that start with either ‘history’ or ‘HISTORY.’ The sum of these was divided by the total number of tokens in the moby.txt file and multiplied by 100, to obtain the percentage.



1. What are the 10 most frequently occurring (unique) tokens in the text? What is their frequency?

[(',', 19204),

('the', 13715),

('.', 7308),

('of', 6513),

('and', 6010),

('a', 4545),

('to', 4515),

(';', 4173),

('in', 3908),

('that', 2978)]

The list above shows the 10 most frequently occurring tokens and their frequencies (token, frequency). To determine these values, I used the unique tokens list created in the first question and printed the most common 10.

Part II – Spelling Recommender

In this part of the assignment, the purpose was to create a spell checker using the nltk package in Python. To accomplish this, I created two functions, one to check the spelling and the other to return the five closest matches. After running the code, the user will need to type “BestFive(’string’)” and run that line, this will show the five closest spellings. I tested this program by using random word misspelling and ensuring the correct word was in the five shown.



Conclusion

In conclusion, the analysis of the Moby Dick text was an informative introduction into the world of text processing. Throughout the analysis of the text, I was able to see the different parts of speech and learn how to remove unwanted parts of speech from the corpus to make analysis easier. The spelling recommender portion of the assignment proved to be much more difficult. But, through the use of functions, the task was made more manageable. In the future, I would like to learn more about the Natural Language Toolkit before attempting this assignment or using a spell checker that is already installed within other Python packages.