CSCI 455: Project 1

Brayden Faulkne and Christina Hintonr

February 24, 2019

1 Introduction

The main purpose of this program is to go through a set of html files, and find how frequently each term is used in the actual text of the page. The program also handles preprocessing of this text, including stemming and stop word removal.

2 Overview

The program works by reading in each individual file, getting the html free text, writing that text to a new file, then appending the text the a string containing all the text from all previous files. Fnmatch is used to ensure that all the files read in are HTML files. Once all the text is retrieved, all characters in the string all set to lower case, and punctation is translated to whitespace. Then the string is split into the individual words using the NLTK libraries tokenize function. Then every word is checked to see if it is a stop word or an integer. If it does not meet either of these criteria, the word is stemmed then added to a list of cleaned words. Then the frequency of each of the clean tokens is calculated by NLTK's FreqDist function. The frequencies and words are stored and sorted in a class, then printed out in order of least to greatest, so that the user would see the most common first.

3 Beautiful Soup

Beautiful Soup is a python library that is used to pull data from html files. It is used here to retrieve the cleaned text from the html files. This is accomplished by turning the files into a soup, and then using the get text function.

4 NLTK

NLTK, short for Natural Language ToolKit, is a python library used to help work with languages and strings. It is often used in data mining and information retrieval when working with word frequency and comprehension. The tokenize function is used to split the string containing all the text into the individual words. Then a Porter stemmer is used to stem each word in the list of words. We used a Porter stemmer as research revealed that the two most frequently used stemmer are the Porter stemmer and the Lancaster stemmer. We chose to go with the Porter stemmer as it is quicker than the Lancaster, with the trade off of occasionally improperly stemming a word. Lastly the FreqDist functions is used to calculate how often each word occurs in the filtered list of words.

5 Conclusion

Earlier versions of this program took as long as two-three hours to run. This was because a lot of the code was done using classes and contained many functions that involved reading over the same list multiple times, greatly increasing the run time of the program. After modifying the base code using functions and libraries I found, namely the NLTK library. We also discovered that operating over a list of all the words, rather than just the words in each file, at once saved a significant amount of time.

6 Results

Text files containing all the, the 200 least used, and the 200 most used, words can be found elsewhere in the directory. There were 51123 unique words were identified after stop word removal and stemming. The 200 most common words consist mainly of used in computer science, academia, and everyday talk, such as days of the week. Some of these words were stemmed in strange manners, as is expected of a Porter stemmer, but other than that it seems all the words were processed properly. The least used words consist of typos, names, and terms that have no general meaning, such as class numbers or room names. There are some words that we believe that may have been the result of an error in processing, but we were unable to identify or recreate the circumstances that caused them, so this might not be the case.

References

[1] Richardson, Leonard. "Beautiful Soup." *Beautiful Soup: We Called Him Tortoise Because He Taught Us.*, www.crummy.com/software/BeautifulSoup/. Accessed 24 Feb. 2019.

[2] "Natural Language Toolkit." *NLTK*, NLTK Project, 17 Nov. 2018, www.nltk.org/.

[3] "Stopwords." *Ranks.nl*, www.ranks.nl/stopwords.