Text Mining and Analysis Mini Project Writeup:
Claire Kincaid
February 24, 2016

Overview

For my project, I created a keyword search engine that takes in a .txt file and a list of keywords and returns a dictionary of the number of occurences of each of those keywords within the file. I designed it for research papers, so that a researcher could determine the applicability of a given paper based on keywords in the paper. It can also be compared to a qualitative analysis of the paper, it's writing, subject matter, and effectiveness. For my example data I used a .txt file of a terrible research paper and compared my qualitative analysis of the paper to the keywords IF.

Implementation

My code consists of four primary functions and a load data loop. The data loop loads a .txt file and reads through it line by line, concatenating all of the content to a long string. The first function, make_data(data), turns that string into a dataset readable by my other functions. To do this it makes all letters lowercase, removes all punctuation, and separates each of the words into a list. My second function, word_count(data), takes in a string, uses make_data to turn it into an analyzable list, and returns a histogram counting the occurences of each word in the list. My third function, keyword_search(data, keyword), takes in the dictionary created by word_count(data) and a keyword and returns the value for the keyword in the dictionary. My final function, keywords_search(data, keywords) takes in the data and a list of keywords and searches for the occurrence of each keyword within the dataset, returning a dictionary with each keyword as a key and the number of occurrences of that keyword the corresponding value.

I could have just left my project at keyword_search, but I wanted to make it more useful, and unique from the word search capabilities of a typical word processor. Therefore, I created keywords_search, which can look for multiple keywords at a time and return the occurrences of each. I feel this makes my project more applicable and better suited to the purpose for which I designed it.

Results

For my example text analysis, I converted a pdf of a research paper into a .txt file and mined for keywords that were the purported "subject" of the research paper. I had previously read this paper and judged it to be terrible and ineffectual writing because it does not address its stated "subject" hardly at all within the paper. I wanted to compare this qualitative analysis to the number of times the subject of the paper, "computer assisted collaborative learning", was mentioned within the paper itself. I ran several keyword searches, looking for instances of "computer", "computational", "collaborative", "interdisciplinary", "assisted", "learning", and "environment" within the paper.

As I suspected, computer and computational, in addition to other related words, were mentioned very few times, somewhere on the order of 0 – 4 times each. Assisted was mentioned once, collaborative, environment, and interdisciplinary did not occur at all within the paper, and the exception, "learning", occurred 31 times within the body of text. I found that this mostly matched my qualitative analysis of the paper, and was pleased that my project worked. I checked the accuracy by going through the paper without a computer an highlighting each occurrence of "learning". My functions were correct.

Reflection

I designed my project so that researchers can use it to easily identify relevant or effective research papers from the occurrences of specified keywords. From a process point of view, I find the

code readable, and my doctests did their job very well. I picked doctests that would encompass as many situations as possible while remaining simple and not taking up a lot of space or becoming cumbersome and difficult to read or understand. My code runs smoothly and serves its intended purpose well. I could stand to reduce redundancy in my code or start to practice optimizing my code for shortest length. For its intended purpose, my project was very well scoped, 1 paper at a time. For a future project, I would love to create a higher order program that could determine relevancy of the paper based on user specified parameters of keyword occurrence, without the user having to decide on their own after viewing the final keyword histogram. Additionally, I'd like to increase the scope of the program so that it can take in a multitude of research papers and determine/rate relevancy of the research papers with respect to each other and user specified parameters. I'd also like to experiment more with TF/IDF for research papers, maybe with something that could traverse a research paper and spit out the most common key word occurences to give the user a general idea of the subject of the research paper.