Computer Science 204

Assignment #5

Document Analyzer

Due Date : Wednesday April 22, 2009

Due Time : 11:59:59 p.m.

40 Points

Objective

The project has two parts – a solution design document as well as a program to solve the program at hand.

Assignment Summary

Many times when people try to translate books and manuscripts it is necessary to analyze the words in the document and look at the words that are used the most and words that are used the least. By simplifying the most or least used words, dramatic differences in readabililty can be obtained. This problem grows more complex when one has to consider if the words are proper nouns. Many times proper nouns are used only once or twice in a text, so it throws off the count of the words that are used the least. In a similar manner, conjunctions, the definite article, and indefinite articles will also tend to throw off the words that are used the most. Numbers, like 234, not two three four, also tend to cause document analysis programs difficulty.

Your task is to write a document analysis program that will produce an alphabetized list of the 50 least used words in a document and the 50 most used words in the document as well as the number of times that they appear. While you need to process ALL words in your analysis code, the list of the most used words that you print out should not contain the following:

- the definite article "the"
- the indefinite articles "a" and "an"
- the conjunctions "and", "but', "or"
- the word "chapter"
- numeric values represented as cardinal numbers

You should, however, print out at the end of the most used words, a percentage of words found in the text that are in the itemized list above. Call this the "connective word percentage." Your final list of the least used words cannot contain proper nouns. Proper nouns in this context are those words in the text that are **always** capitalized.

Another concern in translating manuscripts is determining whether or not the work was produced by one or many authors. A factor that can be used in helping to deduce this is a comparison of where words

are used in the document. For example, if a particular author tends to use certain words and phrases, it would be expected that those would be found uniformly througout the document. In, however, you have two authors that are very different in writing styles, and one wrote one chapter and another wrote a different chapter, then you would expect a bimodal distribution of words with an average that is skewed away from the middle of the document.

You will need to compute an index for each word that demonstrates how it it dispersed through the text. To accomplish this you will need to keep up with the location in the text where the word was found (the easiest thing to do is keep with the occurences γ and line numbers α of those occurrences), and the total lines in the document (N). The word distribution funtion (W) is then,

$$W(\gamma, \alpha, N) = \frac{\frac{\sum_{i=0}^{\gamma} \alpha_i}{\gamma}}{N}$$

This will return a value between zero and one that should be multiplied by 100 to come up with a percentage. If the word distribution function is below 32% or above 68%, it is likely that the work was authored by more than one author. For your most used words, you should also print out the word distribution function.

Notes

1. By Tuesday at Midnight (April 14th) you should submit to me a document via e-mail describing how you are goint to organize your data and how you are going to solve the problem at hand. This can just be a detailed e-mail message. I will comment on it and send it back to you.
2. The title of your Java program should be `TextAnaysis.java`. Any class files you use should also be appropriately named. When making your final submission, please send along all Java code so I can build you class files.
3. Use proper indentation and documentation throughout. Provide comments for all variables and important statements.
4. There are going to be problems executing your programs in a timely manner if you do not use sorting and searching techniques. You will be expected to make heavy use of these in your code.
5. You are STRONGLY encouraged to use ArrayLists of objects in writing your code.

Sample Run

I will provide this for you later in the week.

Extra Credit

You will receive **50 bonus points** if you submit a second program that, in addition to doing the things described above, also computes the *Flesch Readability Index* (described on page 246-247 of your text) and prints it along with the grade level for the text you are analyzing. To receive any credit, both programs must be turned in before the due date and time.

Late Penalties, and Revision Policy

Get started now and let me know when you are having problem. **Late projects will be penalized 20% per day.** All assignments handed in on or before the due date that you do not receive full credit for in accuracy are candidates for revision.