CS2302 - Data Structures

Spring 2020

Lab 5

Due Monday, April 13, 2020

For this lab, you will implement some functions to perform elementary analysis of text documents efficiently using hash tables. More specifically, your task is to write a program to read a text file and find the *content word* that appears the most times in the file. For our purposes, a content word is a word that is likely to be related to the document's topic. To find the content words in a document you will first extract all the words in the file to a list and then you will remove from that list all the words that appear in a list of *stop words*, which are the most-commonly used words in the English language and thus are assumed to provide little content information. Examples of stop words include *the, and, a, this, that, for* and *in*. Finally, you will count the occurrences of all the words in that list and return the one with the most occurrences.

To test your program, we will use a set of abstracts obtained from the scientific literature in neuroscience. Hopefully, the words that your program will find are neuroscience-related. The abstracts are in the file *abstracts.zip* and the stop words are in file *stop_words.txt*. Code also provided in program *read_files.py* to read text files, convert them to lowercase, and extract the individual words to a list of strings.

Your program should do the following:

1. Extract all the words in the stop word file into a list of strings
2. Store the words in a hash table
3. Display statistics describing the hash table you created, including number of buckets, number of keys, number of empty buckets, the number of long buckets (a bucket is considered long if it has more than one record) and the length of the longest bucket. Notice that a good hash function should yield few empty buckets, few long buckets, and a short longest bucket.
4. For every text file in *abs_00.txt,...,abs_49.txt* do the following:
   (a) Extract all the words into a list of strings
   (b) Remove from the list of string all stop words
   (c) Find the most common word using a hash table
   (d) Display the number of words in the document before and after stop word removal
   (e) Display statistics describing the hash table you created, as you did for the stop word hash table.

As usual, write a report describing your work.

Appendix: Sample output

The following shows the first lines of the output of your program. For all hash tables, we used a table size equal to the length of the list of words provided, thus the load factor is at most 1.

```
Analysis of stop word hash table
Total buckets: 429, total records: 423, load factor 0.986
Empty bucket fraction in table: 0.38
Long bucket fraction in table: 0.263
Length of longest bucket in table: 4

File: abs_00.txt
Total words: 233, total non-stop-words: 134
```

```
Analysis of abs_00.txt hash table
Total buckets: 134, total records: 70, load factor 0.522
Empty bucket fraction in table: 0.604
Long bucket fraction in table: 0.104
Length of longest bucket in table: 3
Most common word: rats - occurs 11 times

File: abs_01.txt
Total words: 377, total non-stop-words: 203
Analysis of abs_01.txt hash table
Total buckets: 203, total records: 106, load factor 0.522
Empty bucket fraction in table: 0.655
Long bucket fraction in table: 0.128
Length of longest bucket in table: 4
Most common word: injection - occurs 13 times

File: abs_02.txt
Total words: 217, total non-stop-words: 141
Analysis of abs_02.txt hash table
Total buckets: 141, total records: 86, load factor 0.61
Empty bucket fraction in table: 0.546
Long bucket fraction in table: 0.135
Length of longest bucket in table: 4
Most common word: eeg - occurs 7 times

File: abs_03.txt
Total words: 275, total non-stop-words: 170
Analysis of abs_03.txt hash table
Total buckets: 170, total records: 130, load factor 0.765
Empty bucket fraction in table: 0.441
Long bucket fraction in table: 0.171
Length of longest bucket in table: 4
Most common word: autism - occurs 6 times
```