Assignment 7: *Write Up*

Sriramya Prayaga

March 14, 2022

Abstract

In this document, I will discuss what I learned from the Author Identification Assignment. I will discuss what happens when the number of noise words is increased and decreased, what happens when the size of the text changes, and the results of the different metrics, to be specific.

1 Noise Words

One important thing to observe with this program is what happens when increasing the number of noise words. Noise word texts are created by the Text data structure. In the function text_create(), the noise words (that is, commonly used words) are filtered out when creating the Text for a file.

The reason for filtering out noise words is due to them not being indicative of an author's unique diction, as according to the Assignment 7 document. Increasing the amount of noise words makes the identification program's results more accurate, to an extent. Filtering out more noise words, generally speaking, makes the distance values between the two words larger. This is because words that are commonly used are not being factored into the distance calculations. However, as the noise words text file also includes words that are not relevant at all to any particular text file, it may be unnecessary to add too input a high noise word filter to the identification program.

Decreasing the amount of noise words on the other hand, can decrease the accuracy of the identification program. The noise Text structure will factor in words like "the", "a", "be", etc., and this will make it appear like the a text file has a great deal in common with another text file (because these words are so common in most text files). This would artificially decrease the distance between the two files, which can lead to flawed conclusions about the similarities of text files.

2 Text Sizes

The size of the text also affects the results of the program. If the size of the text file is smaller, the size of the vector components with another text file tend to get larger. This is because, using the Manhattan distance for example, if there were only a few words in a text file (like four), and there are two repeating words, the size of the normalized frequency would be $2/4$. However, if there are 100 words in a text file instead, and there were two repeating words, the normalized frequency would be $2/100$. This reveals that, when compared to another text file with its own normalized frequency of the same particular word, the distance between the file with the smaller text file would be greater.

Given this insight, when I tried to run my program with a small passage of text that was somewhat similar to a text file in a data base, the results showed a larger, more inaccurate distance when compared to a larger text passage that had commonality with a text file in the data base.

Despite this problem of varying distances however, the program was able to identify the correct author for both large and small passages of texts (if the author's text was in the inputted data base.)

3 Results of Different Metrics

Another important factor to observe when looking at the output of the program, is the choice of distance measurement. This is handled/calculated in the text.c file.Something that can be seen from observing the results of the program was that using different metrics yielded (sometimes significantly) different results. Out of the three metrics that were used for the calculations –Manhattan, Cosine, and Euclidean– the Euclidean measurements yielded the smallest values, while the Manhattan calculations had larger distance values. The Cosine calculation was the most complicated form of calculating the distance estimation of the texts, so perhaps that's why using it as the measurement estimation for the size, yielded larger distance vector values. Manhattan tends fall between the other two distance measurements when it comes to accuracy.

4 What I Learned

From this assignment, I learned about the efficiency of using hash tables, and bloom filters – that is, I learned about how bloom filters can be a useful way to quickly discern if an item is not in a hash table. I learned about how bit vectors are useful in the implementation of bloom filters, and how they can compactly represent items present in a set. I learned about how the size of a text input into the identification input can vastly affect the accuracy of the result. I learned about the choice of distance measurement calculation, (and the noise words used) between the two texts also affects the accuracy of the calculation.