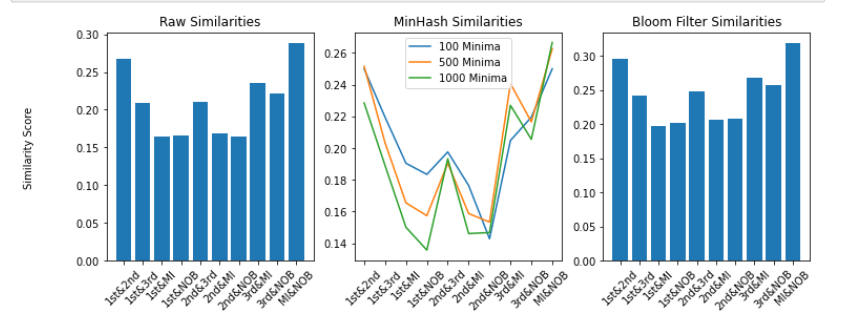
Sketch

Process Pipeline

1. Wanted to find some text that MAY have some similarities, so I found Emily Dickinson who has written 3 series of poems. I also wanted to compare it to another 2 poems to see if there was any difference or similarities between two different author’s poems, so I used two poems by Robert Frost.
2. Parsed through the poems and separated each word for comparison. I complied all of this into one set called word\_sets.
3. I then used the functions in part one to compare the poems. As you can see from the graph below, the first bar shows the comparison between the first and second poem etc.

Visualization using Python:

This visualization helped me decide which K and M values to use.



**K Value reasoning:**

Data points:

K = 100, 500, 1000

{100: [0.25, 0.21951219512195122, 0.19047619047619047, 0.1834319526627219, 0.19760479041916168, 0.17647058823529413, 0.14285714285714285, 0.20481927710843373, 0.21951219512195122, 0.25], 500: [0.25156445556946183, 0.20336943441636582, 0.1655011655011655, 0.1574074074074074, 0.19047619047619047, 0.15874855156431056, 0.15340253748558247, 0.24069478908188585, 0.21654501216545013, 0.26262626262626265], 1000: [0.2285012285012285, 0.18906064209274673, 0.15008625646923518, 0.1357183418512209, 0.19331742243436753, 0.14613180515759314, 0.14678899082568808, 0.22699386503067484, 0.2055455093429777, 0.26662444585180495]}

K = 100, 500, 10,000

{100: [0.25, 0.21951219512195122, 0.19047619047619047, 0.1834319526627219, 0.19760479041916168, 0.17647058823529413, 0.14285714285714285, 0.20481927710843373, 0.21951219512195122, 0.25], 500: [0.25156445556946183, 0.20336943441636582, 0.1655011655011655, 0.1574074074074074, 0.19047619047619047, 0.15874855156431056, 0.15340253748558247, 0.24069478908188585, 0.21654501216545013, 0.26262626262626265], 10000: [0.2667989793025234, 0.20834597875569044, 0.16462320574162678, 0.1660767764769943, 0.20996860282574567, 0.16825437693099898, 0.16385115180153573, 0.23543763342482465, 0.22182468694096602, 0.2884843102427472]}

Higher K value seems to make little to no difference as you can see when I increase the K value from 1,000 to 10,000. In fact, there is not much difference between 500 and 1000. Therefore, to reduce time and space complexity, I would keep the K value around 500 – 1,000. You can also see that the 1,000 minima is very close the raw similarity so I would consider that as sufficiently good. \*The K value decision also depends on how accurate you would like the numbers to be.

**M value reasoning:**

Data points:

m = 2000

[0.9739609414121182, 0.9634634634634635, 0.9589794897448725]

m = 10,000

[0.5544698270170342, 0.49965953969767124, 0.47162162162162163]

m = 100,000

[0.29572753535046375, 0.24104392932403956, 0.1974894059326777]

\*Note that I put only the first three comparisons above so that the page does not get cluttered.

Increasing the value of m does seem to make a difference on the Bloom filter similarities. The similarity between the Bloom filters decreases from 0.97 to 0.55 when m is increased from 2000 to 10,000. I went ahead and tried a larger size of 100,000 and I got the result of : [0.29572753535046375, 0.24104392932403956, 0.1974894059326777]. This is more similar to what we got with the raw similarities so I believe we are getting closer to what we want. I also tried 1,000,000, but it took too long to process, so I think that a number around 100,000 would be sufficient for our accuracy.