CMSC409 Project 4 Questions Peter George/Joseph Longo

2b)

Tokenize: Easier to analyze words as opposed to phrases, sentences, or structures of groups of words. Also easier to parse.

Remove punc/special chars: They add little to know value, especially punctuation when analyzing by sentence, and further by word. A comma makes no sense with respect to individual words.

Remove Numbers: Increments, dates, labels all can be lost in removing numbers. Certain number could introduce bias for sentence clustering, especially if looked at as words where they're really conditions of other words such as miles(60) is about miles, not about 60.

Convert upper-case to lower-case: Start of sentences provides duplicate words. Removes noise. Can also cause loss in some data if words are capitalized for reason, such as Man One and Man Two as persons, or Autumn as a name becomes a season (likely).

Porter Stemming: Convert words to their basic or basal form, so that words like "coming, came, come" all register as "come" (Example taken from PP20, page 17) .

Linguistic information is lost but grouping is more accurate, often.

Combine Stemmed Words: Same as above, where coming, came, come become one column.

Extract Most Frequent Words: Important feature, especially for dimensionality. The original document has something like 150 unique words, and like 100 that were used only once. That is a lot of individual clusters and spacing around. Words under a threshold are trimmed. In our project, I set it to minimum of 3 occurances, and utilized 3/4 as minimum for runs. Some information is lost still, obviously, as words are literally lost.

3A)

With a word threshold of 3 (occurances), and an EDmax of 2.1, it has 23 clusters, with 15 individual clusters (1 sentence), and 8 clusters of groups, with a max cluster group size of 9.

With a word threshold of 4 and an EDmax of 1.85, it has 23 clusters, with 15 individual again, 8 group. Max cluster size is 10 now.

3B) The Dimensionality is driven by the number of "features" aka columns. To reduce it, we could remove more words such as the written verions of numbers, as well as combine words such as "little" and "tiny." Some data is lost in doing those things. The order of data does matter for clustering. A cluster in the center first will group more together than a cluster farthest away.

3C) Results output as print statement in code. Below is an example of the code run on threshold 4, EDmax 1.85. (Cluster topics not named as not required, just listed the clustering)

```
Cluster 1 has ['Sentence 1']

Cluster 2 has ['Sentence 2', 'Sentence 6', 'Sentence 16', 'Sentence 21', 'Sentence 24', 'Sentence 26']

Cluster 3 has ['Sentence 3']

Cluster 4 has ['Sentence 4']

Cluster 5 has ['Sentence 5', 'Sentence 8', 'Sentence 19']

Cluster 6 has ['Sentence 7']

Cluster 7 has ['Sentence 9']

Cluster 8 has ['Sentence 10']

Cluster 9 has ['Sentence 11', 'Sentence 12']

Cluster 10 has ['Sentence 13', 'Sentence 20', 'Sentence 25']

Cluster 11 has ['Sentence 14', 'Sentence 20', 'Sentence 28', 'Sentence 32', 'Sentence 39', 'Sentence 41', 'Sentence 43', 'Sentence 44', 'Sentence 45', 'Sentence 46']

Cluster 12 has ['Sentence 15', 'Sentence 22']

Cluster 13 has ['Sentence 17']

Cluster 14 has ['Sentence 18', 'Sentence 33', 'Sentence 35']

Cluster 15 has ['Sentence 23']

Cluster 16 has ['Sentence 27']

Cluster 17 has ['Sentence 29']

Cluster 18 has ['Sentence 30']

Cluster 19 has ['Sentence 31']

Cluster 20 has ['Sentence 34', 'Sentence 38', 'Sentence 42']
Cluster 21 has ['Sentence 36']
Cluster 22 has ['Sentence 37']
Cluster 23 has ['Sentence 40']
```