Ind Study: Machine Learning **(Due: December 10, 2020)**

Project #3

Instructor: Bill Michael *Name:* Alex Allenspach

Problem

(a) Use ten(10) different cluster sizes(k) and evaluate the quality of the clustering results using the Davies-Bouldin index (DBI) and Silhouette Index (SI) for the three algorithms. Present your result with a plot. **(20 point)**

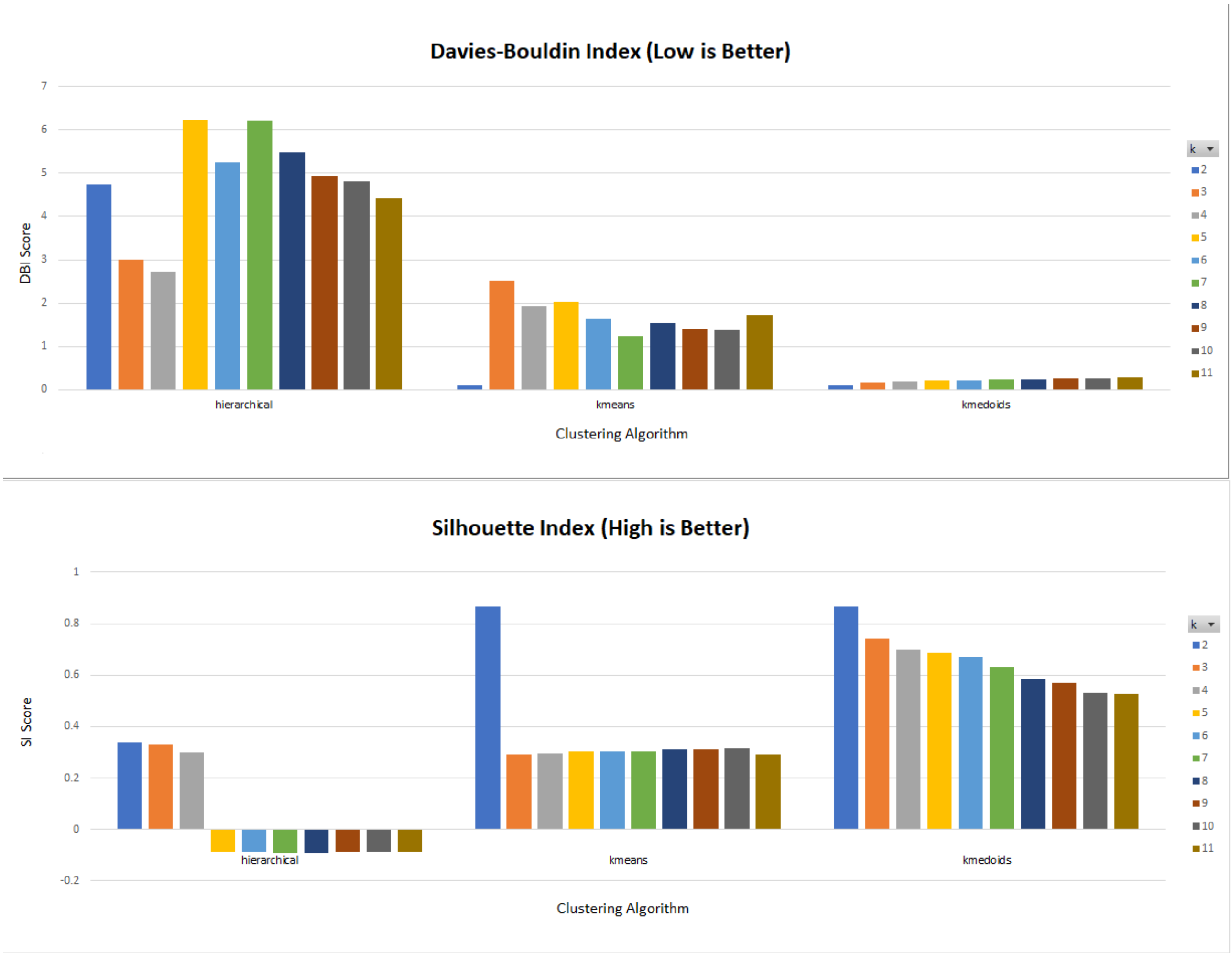

Figure 1: Quality of Clustering Algorithms.

(b) Comment on the performance of the algorithms on the different cluster sizes ? **(10 point)**

From Fig. 1 you can see that the optimal cluster number for all three algorithms when using the Silhouette and Davies-Bouldin Index is almost always 2. The only index that disagrees with two clusters is the DBI Index obtained from hierarchical clustering (Optimal is four). In the pre-processing stage I used different methods for clustering the provided DTM. Keep in mind that all three algorithms are using euclidean distance. These are the following methods I explored:

1. The DTM that was given is just word frequency (Bag of Words) which makes it discrete count data;

therefore scale is not a problem. Therefore I left the DTM as it was and Fig. 2 shows the results.

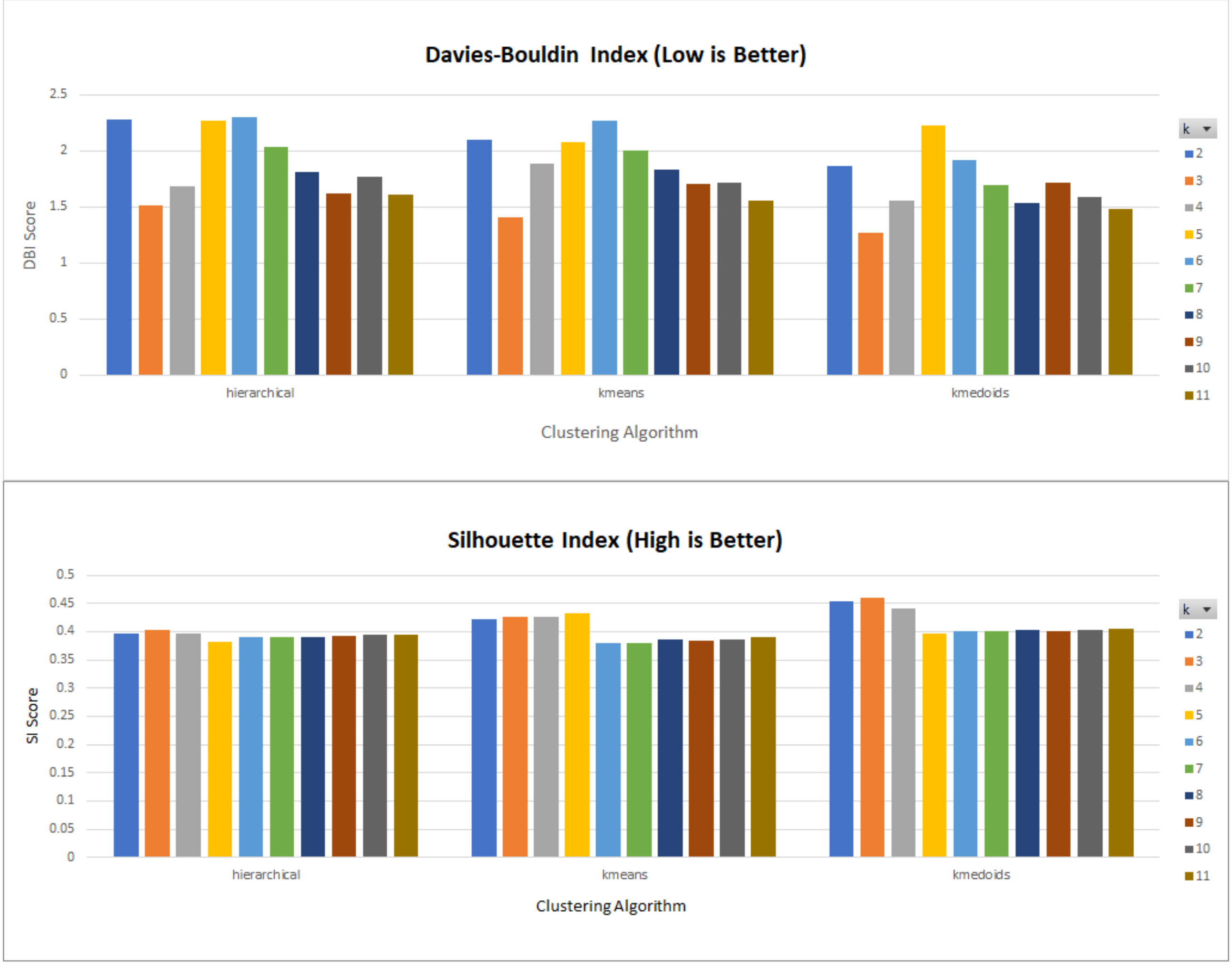

Figure 2: Quality of Clustering Algorithms.

Looking at Fig. 2 you can see that the optimal cluster number is almost always 3. The only index that disagrees with three clusters is the SI Index obtained from k-means clustering (Optimal is five).

2. I then realized I need to take into account the fact that some document's lengths are longer than others so at first I normalized the DTM by dividing the frequency of a term in a document with the euclidean norm of that document's vector. This method did lead to some errors and extra pre-processing would be required (replacing the NaN's with zeros) to fix. In the end I decided to normalize by the tf-idf weights as it was invented for document searching and will help in delivering results that are the most relevant to what we are searching for. Those results can be seen by looking at Fig. 1.

Overall the different cluster sizes from normalizing and not normalizing looks to be quite different by just glancing at the two figures, but it turns out that the optimal cluster value does not range significantly at all (2 or 3).

(c) At what k value was the best clustering quality achieved, and what is the running times of the three algorithms for this k ? **(5 point)**

The best k-value achieved from Fig. 1 is 2.

Table 1: Run Times of Clustering Algorithms

Algorithm	Run Time (sec)
K-Means	72.6
K-Medoids	73.72
Hierarchical	70.33

(d) Which of the algorithms would you recommend for solving text clustering? **(5 points)**

If I was to have big data then I would use k-means for clustering as it is computationally faster than both k-medoids and hierarchical clustering (if K is small). In general I would use hierarchical clustering as it outputs a hierarchy which is much more informative then the flat clusters returned form the other two algorithms. Hierarchical also doesn't require prior knowledge of the number of clusters you want to divide the data into as it displays a nice dendogram that can be used to decide the optimal cluster number.