**1. Discussion of data format**

I downloaded the dataset from Kaggle on my personal laptop and unzipped the folder. The dataset had a large size so it took a long time to be extracted from the zip folder. The folder was saved on my desktop and I opened the jupyter notebook. From the JSON files, I thought that working with ‘body\_text’ would be much faster and accurate. I used the glob module and glob function to access all the JSON files from different folders. The total length of this before taking 10 percent of the files was 45,941. Then, I used random.choice function from the numpy module and passed the JSON files to obtain the 10 percent of the files. Then I wrote a function to obtain the text from ‘body\_text’, where I used the ‘load’ function from JSON module. After enumerating through the 10 percent file, I open each JSON file and searched for ‘text’ under ‘body\_text’. Then I appended these results to the list I had already defined. Then I joined the results into a variable called my\_str, and due to this, I got a list of one huge string.

Potential Improvement:

I just tried removing the my\_str ='\n'.join(my\_list) line of code and the result I obtained was a list that has multiple strings, and each string is a text value that was contained by the key body\_text.

If I could tokenize the sentences in this single string, then I would be able to maintain the each text from each text\_body. This would be passed to the tf-idf part for feature extraction and perhaps could make a difference in the outcome for my clustering.

The length of my\_str varies when I run it at different times, and I don’t understand why. If the length increased, there would be more texts meaning more string components being passed to tokenizer and clustering part.

2**. Discussion of tokenizer**

For tokenizing, I used the spaCy library and I did only sentence tokenization. By doing this, I was able to convert the big string I had from the previous data format part into sentences. After sentence tokenization, I wrote a function to remove numbers from those sentences. I used a sub method from regular expression library. After removing the numbers, I stored each sentences in a data frame from Pandas library using DataFrame object and I assigned a column by the name ‘Title’ for the sentences. I had been working with only strings previously, but I switched to the data frame structure hoping to make it easy while clustering, however, it did not make any difference. The number of rows for the data frame changes every time I run my code from the beginning.

After storing each sentence into the data frame, I used TfidVectorizer to extract features. I used the ravel function to avoid an error I was getting.

**3. Discussion of clustering method**

I used k-means clustering method. I chose this method because it looked like it is the most common clustering method used over text clustering. I had investigated mean shift clustering method, but I decided not to choose it as it was more common on images than texts. First, I built a k-means model for 2 to 20 clusters. Then I used Principal Components Analysis (PCA) to reduce the dimensions of the features array while maintaining 99 percent of variance. I used this new reduced matrix to in my k-means model and a Euclidean distance to obtain the optimal number of clusters based on the reduced features array. I showed my results on a graph (Elbow method), though the blue line would change every time I run my code because my original data would be changing. As of my latest run, the optimum number of clusters was somewhere between 10 and 12.5, so I chose number of clusters to be 11 to fix it on my data. My silhouette score was calculated at last and the result I got was 0.0600. Though this was one way of determining the number of clusters, 0.06 was very far from 1, which means that the best number of clusters wouldn’t be 11.

Then, I run my model and printed the clusters out. However, each cluster contained words instead of sentences. I understand that each cluster should be composed of sentences and I should summarize these sentences from the clusters. Therefore, there is something that I am doing wrong which is, most likely, combining the texts from ‘body\_text’ into a single. If I looked deeply into this issue, I would be able to obtain the feature of each sentences from a ‘body\_text’ and pass that onto to continue with clustering to obtain a cluster of sentences.

4. Text Summarization

As I didn’t succeed on getting a cluster of sentences, I had tried summarizing my tokenized data. I used network library that enabled me to get different functions such as pagerank and from\_scipy\_sparse\_matrix. I used the 2016 edition book from page 256 – 261. I was giving it just a try to see the result, but the result was an array of number, that I am not sure what they represent.

My next steps will be figuring out the clustering and summarizing part. I’m sorry for my uncompleted work!

Resources:

<https://medium.com/@jyotiyadav99111/selecting-optimal-number-of-clusters-in-kmeans-algorithm-silhouette-score-c0d9ebb11308>

<http://brandonrose.org/clustering#K-means-clustering>

<https://github.com/MaksimEkin/COVID19-Literature-Clustering/blob/master/COVID19_literature_clustering.ipynb>