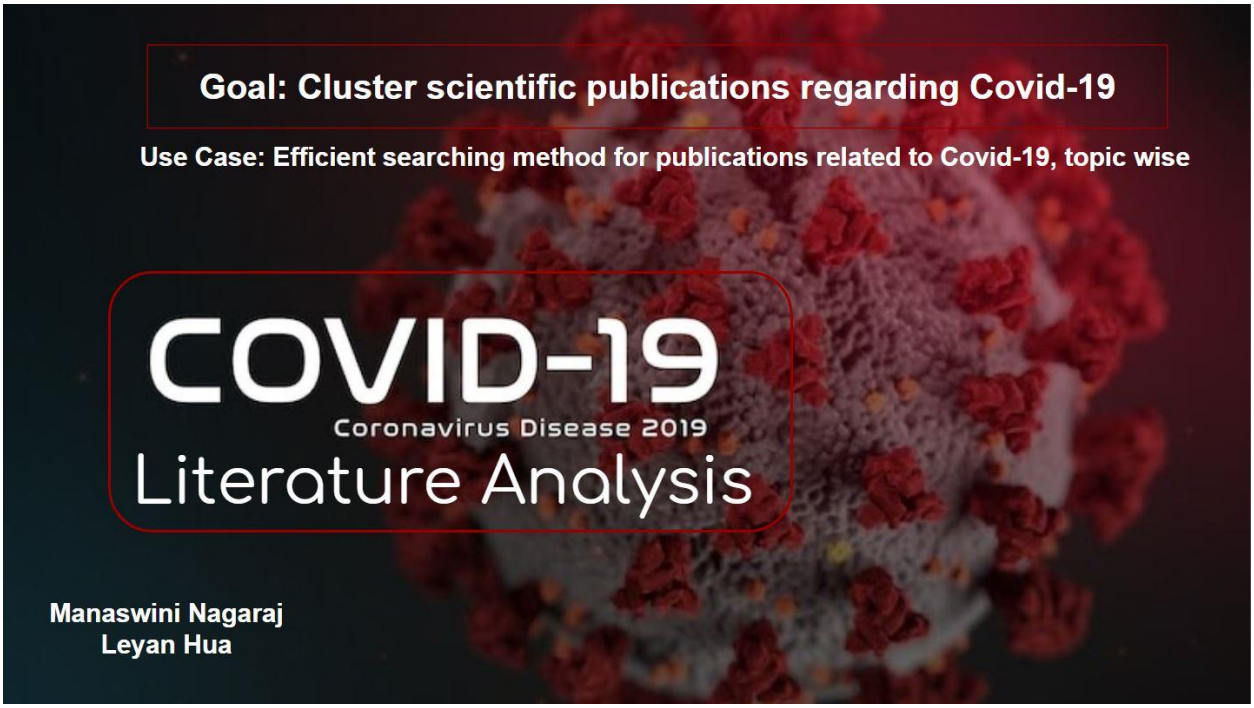
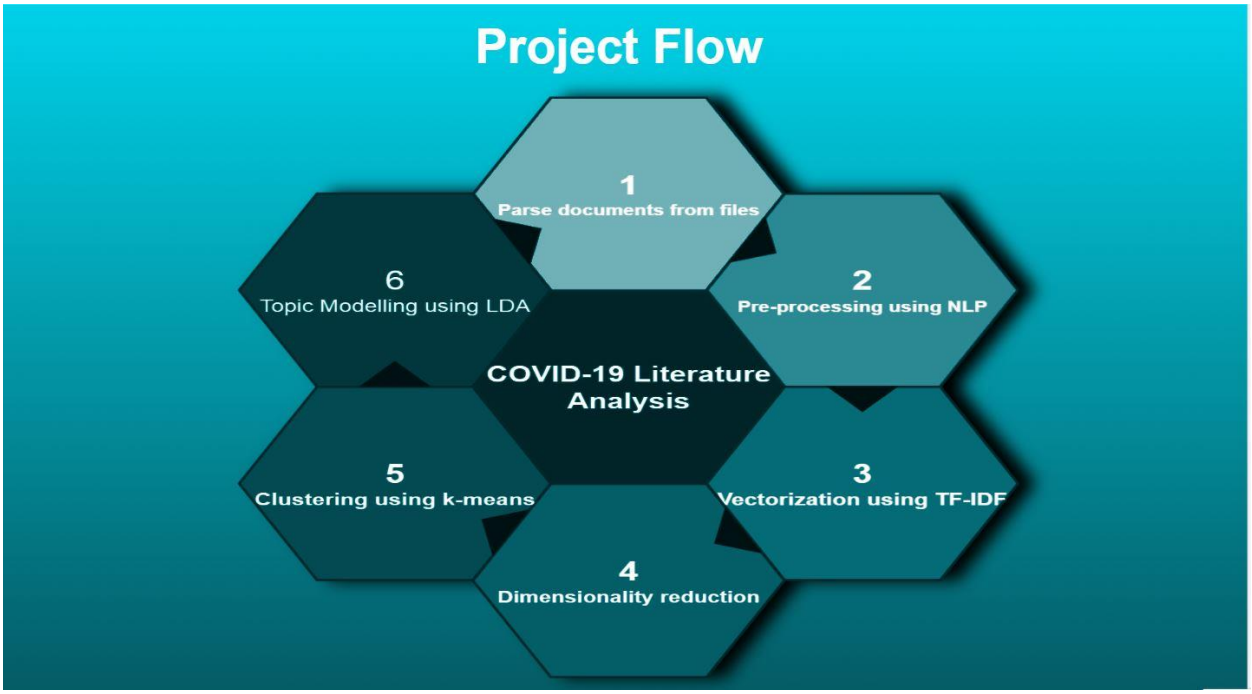
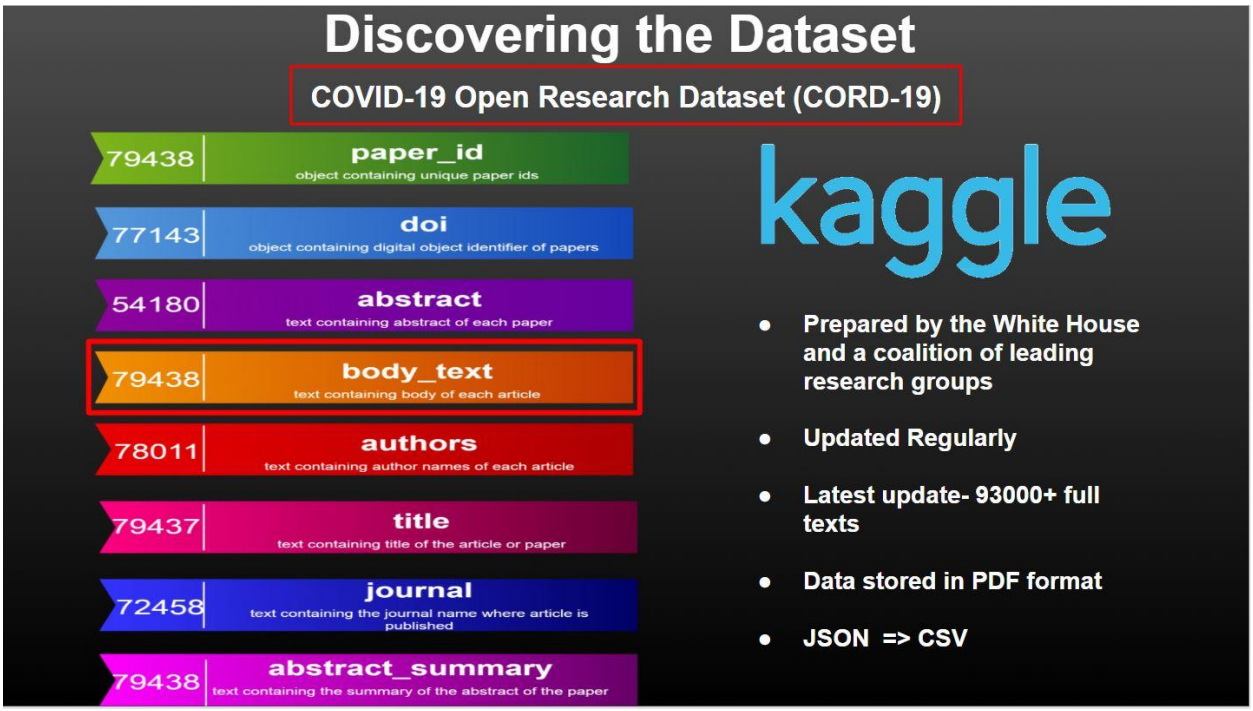


COVID-19 is a respiratory tract disease caused by SARS-CoV-2 coronavirus. What first started as a cluster of pneumonia cases of unknown causes in Wuhan city, China quickly spread to the rest of the world at an alarming rate, affecting more than 20 million people worldwide. Doctors and researchers are still working on introducing a clinically proven vaccine to the market.

In this project, COVID-19 Literature Analysis, we aim to analyze multiple scholarly articles about COVID-19 and other related coronaviruses to extract insights from and assist in searching for a vaccine. The key idea here is that it would be more efficient for researchers to find articles from among the labeled group than search for them in a sea of unlabelled articles.



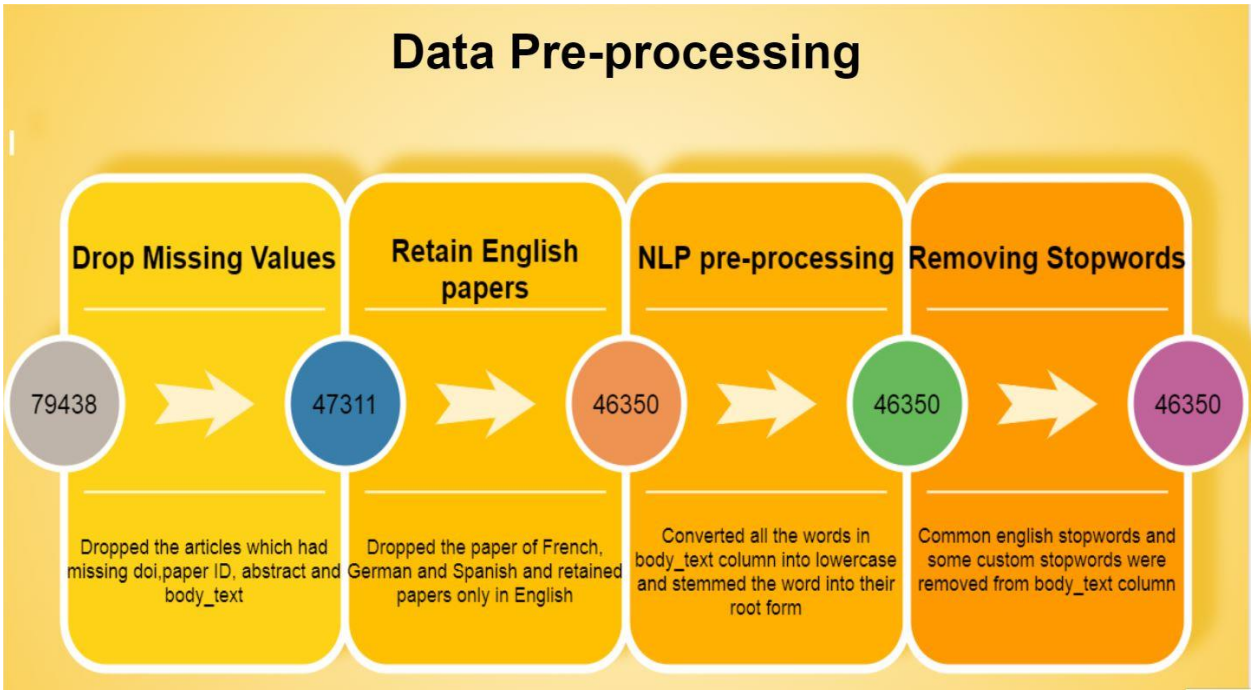
This project consists mainly of 6 modules. First, we parse the documents containing the text dataset using Natural Language Processing techniques and convert them into an appropriate format. Next, we pre-process the loaded dataset, after obtaining the cleaned and processed data, we vectorize the dataset. Once we have the feature vectors, we apply dimensionality reduction to project the data into a lower dimension for ease of use. After the dimensionality is reduced, we apply the clustering algorithm to cluster the data and label the clusters using the Topic Modelling technique.



COVID-19 Open Research Dataset (CORD-19) is an open dataset on Kaggle containing more than 192,000 articles about COVID-19, SARS-CoV-2, and other related coronaviruses, among which there are about 93,000 full-text articles and publications. The full-text articles are in PDF format, and the rest of the materials are in PMC format stored in json files. The dataset is prepared by the White House and a coalition of leading research groups in the world. The articles are all obtained from multiple sources and are updated regularly.

Each paper has a paper id and digital object identifier as unique identifiers, a title, a journal name, author names, abstract, body, and abstract summary. At the time of downloading, the dataset had about approximately 79000 full texts in it, but now it has 93000+ full texts.

The papers in PDF format were parsed using Natural Language Processing to obtain column-wise data and storing it in csv format. We loaded the json files and used the "dict[]" function to parse through it and stored the attribute wise data with a column for each of the paper attribute mentioned above in a data frame. This data frame was then exported into CSV format for easy access. We use the highlighted "body_text" column for clustering and topic modeling.



After we obtained the data frame, the pre-processing was initiated by dropping the rows that had missing abstract, body, title and paper id, doing this reduced the dataset from 79438 to 47311, reducing it by almost 50%. Next, since the data is collected from multiple sources, the data might have duplicates, which can act as unnecessary noise when clustering, hence we checked for duplicate articles and found none.

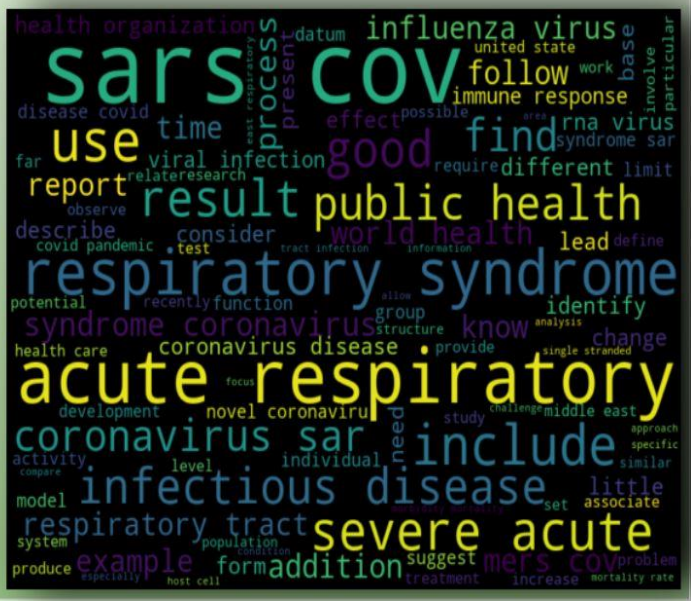
After this, we detected the language of each paper using the "langdetect" package in Python. The data frame was appended with a new column indicating the language of each paper. We found about 961 papers that were in French, German, and Spanish. These papers were dropped from the dataset to make the data processing easier. After dropping those, we retained 46350 papers.

After the initial phase of pre-processing, the attribute body_text of the dataset, which has the main content of each of the papers, was processed using Natural Language Processing techniques. The spaCy package was used to convert all the text into lowercase, remove punctuations, and eliminate stop words by applying the en_core_sci_lg parser, which is useful for scientific, bio-medical, or clinical texts. On further exploration of the retained words, there existed many more words that would act as noise while clustering. Hence a list of custom stop words was created, which included words like 'author,' 'figure,' 'table,' 'et,' 'al,' and others, and was eliminated from the text. After obtaining the cleaned text, the statistics generated for the word count are as follows the average count of body text is 2315 words, and the median is 2002 words.

Vectorization

Term Frequency- Inverse Document Frequency(TF-IDF)

- **Term Frequency = no. of times term occurs in document/ total no. of words in document**
- **Inverse Document Frequency = log(total no. of documents/ no. of documents containing term)**



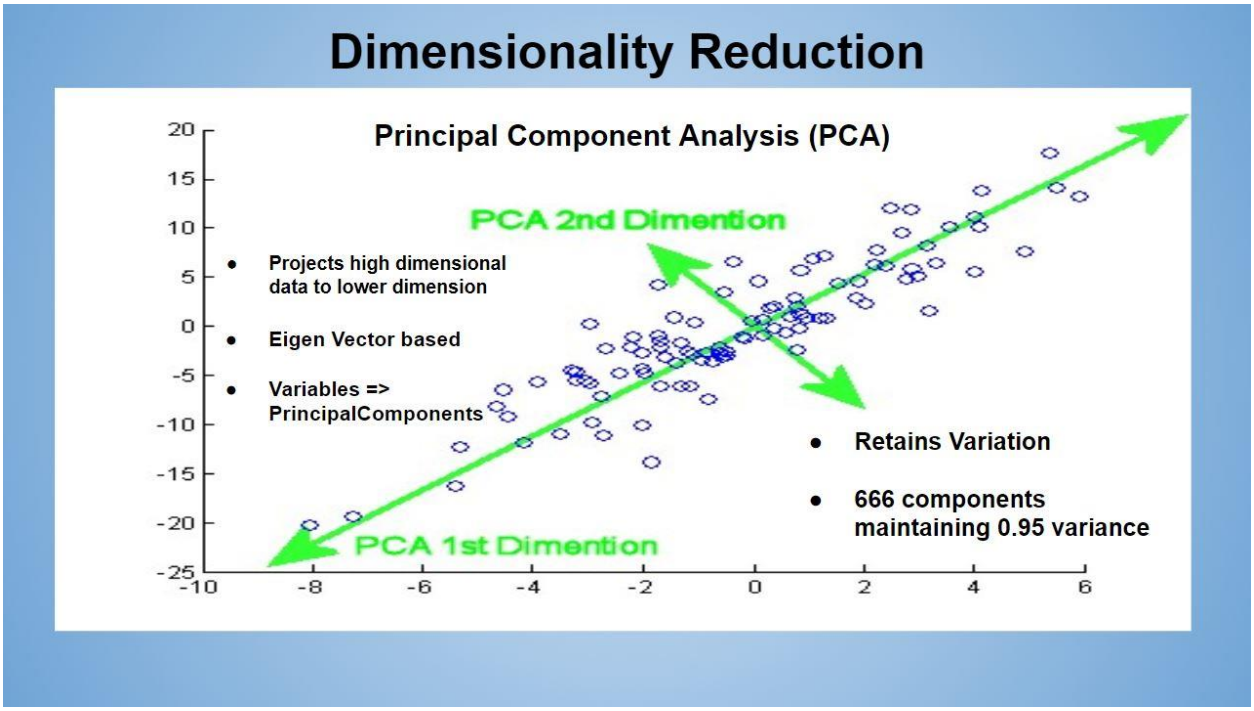
After pre-processing the data, in order to run the project within the available time and resources, we randomly sampled 10000 papers from the dataset and converted it into a format that the algorithms will be able to handle using TF-IDF. Term Frequency- Inverse Document Frequency converts each document into a measure reflecting how important each word is to the whole instance of the document.

Term Frequency (TF): Count of how frequently that term occurs in the document.

Inverse Document Frequency (IDF): It is used to calculate the weight of rare words across all documents in the corpus. This value is obtained by dividing the total number of documents by the number of documents containing the term and then taking the logarithm of that quotient.

Words that frequently occur within a document but not frequently within the corpus receive a higher weighting as they are assumed to contain more meaning. Now each document in the corpus is represented in a vector form that contains the product of TF and IDF values for each word in the document.

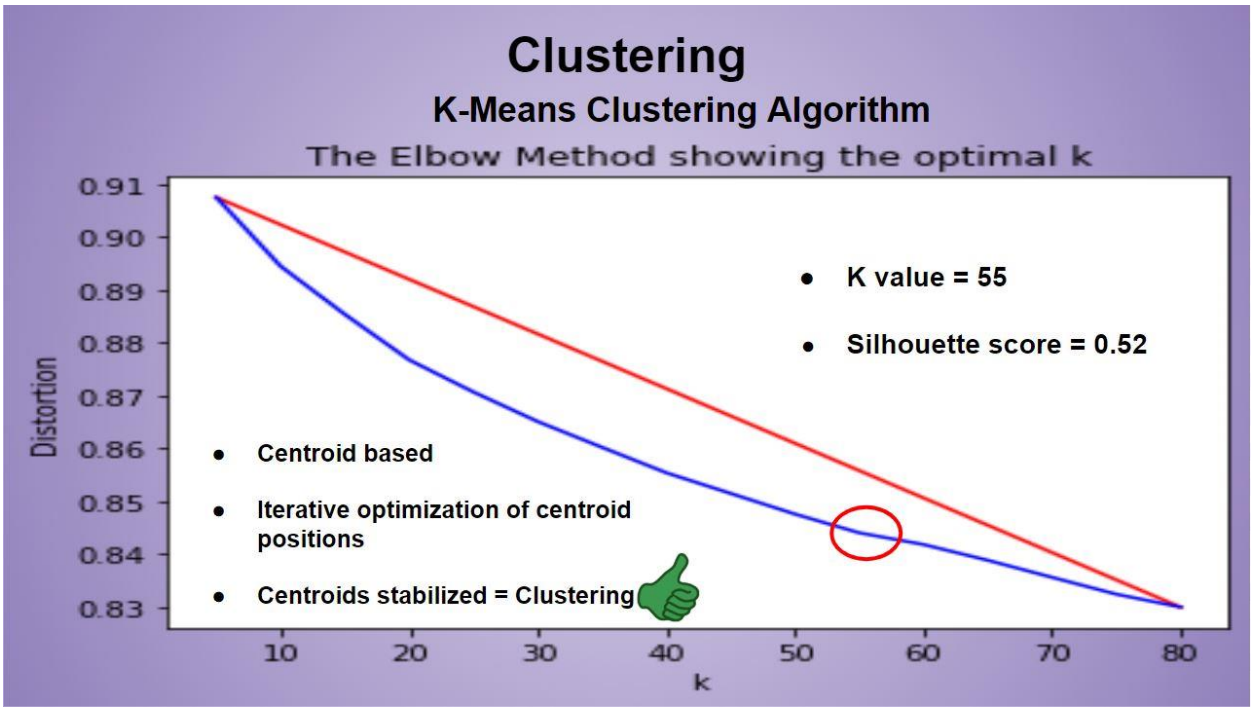
After vectorizing our dataset, we retained only the top 2^10 features for each of the papers to simplify the running time and to filter out any unwanted noise. To get an idea about the top features, we plotted a word cloud, as shown in the above slide, for the top 10 co-occurring features of 10 randomly sampled papers.



After vectorization, Principal Component Analysis (PCA) was applied as a dimensionality reduction technique to the vectorized data. PCA is an eigenvector based technique. It aims to find the directions of maximum variance in high-dimensional data and projects it onto a new subspace with fewer dimensions than the original. The new subspace's principal components can be interpreted as the directions of maximum variance given the new feature axes are orthogonal to each other.

In the above figure, the x and y axes are the original feature axes, and the horizontal green line-PC1 and the vertical green line-PC2 are the principal components. In using PCA for dimensionality reduction, we construct a d x k- dimensional transformation matrix that allows us to map a sample vector to any new k-dimensional feature subspace with lesser dimensions than the original d-dimensional feature space. Usually, the first principle component explains the most variance of the original data

In our dataset, 666 components explained 95% of the variance in the data. Hence only those 666 components were retained for further processes.



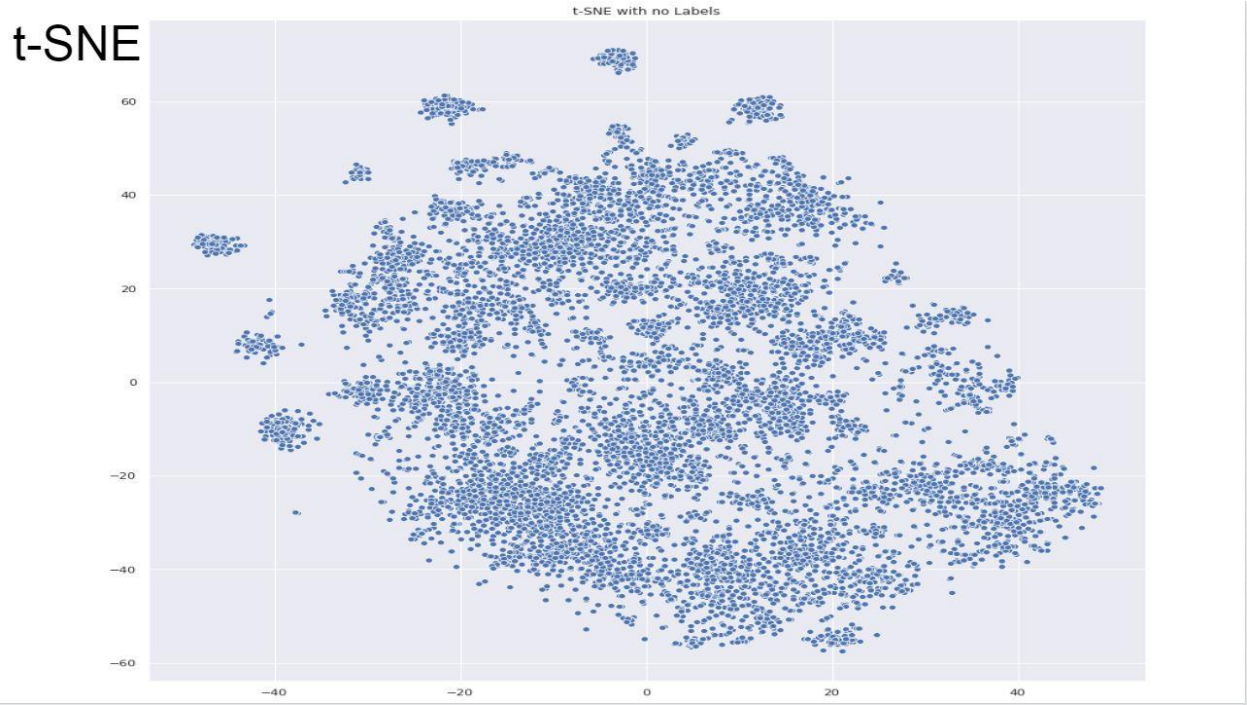
K-means is one of the popular clustering algorithms, the objective of k-means is to group similar data points and discover underlying patterns, to achieve this, k-means looks for a fixed number (k) of clusters in the data.

K-means is a centroid based algorithm, i.e., the number k entered is the number of centroids the algorithm will identify in the dataset. After identifying the centroids, the algorithm then allocates every data point to the nearest cluster while keeping the centroid as small as possible.

The algorithm first starts with k random centroids and using them as a beginning point, iteratively optimizes the position of the centroids until the centroids have been stabilized, i.e., no change in their values at which point the clustering is successful, or a certain indicated number of iterations has been achieved.

To find the best k value for k-means, we looked at the distortion at different k values. Distortion computes the sum of squared distances from each point to its assigned center. When distortion is plotted against k, there will be a k value, after which decreases in distortion are linear, and this is the desired number of clusters.

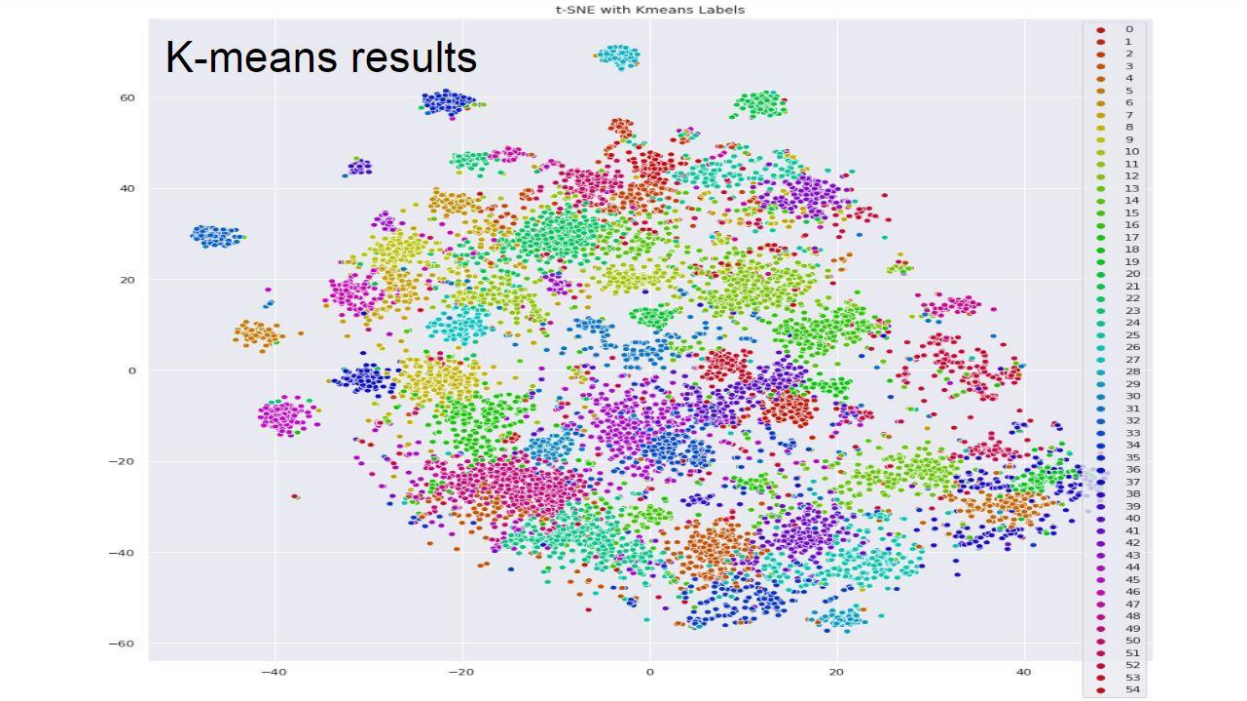
As observed in the distortion plot above, although we see a slight elbow at k=55, it is not enough to determine the k accurately, hence we did a silhouette analysis and found the 2 of the highest silhouette scores was 0.52 and 0.54 for k=55 and k= 75.



t-Distributed Stochastic Neighbor Embedding (t-SNE) reduces dimensionality while trying to keep similar instances close and dissimilar instances apart. It is mostly used for visualization. In particular, to visualize clusters of instances in high-dimensional space.

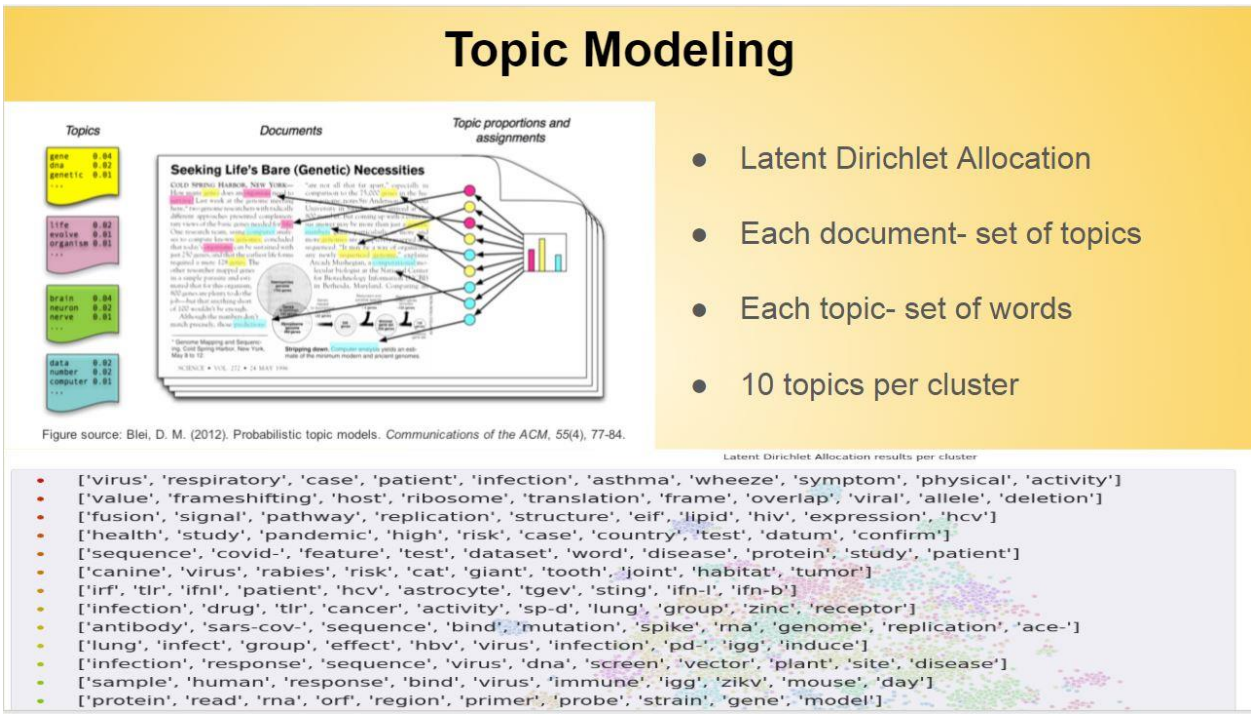
t-SNE works in 2 phases, first it constructs a probability distribution over pairs of high-dimensional objects in such a way that similar objects are assigned a higher probability while different points are assigned low probability. Next, it defines a similar probability distribution over the points in a low dimensional map and minimizes the Kullback-Leibler divergence (KL divergence) between the two distributions with respect to the location of points in the map.

As we can see in the above slide, we can immediately detect some clusters, but instances at the center are harder to distinguish. To label and distinguish the clusters clearly, we used the k-means generated clusters as labels; this helps us visually separate the different clusters.



After the silhouette analysis gave us two highest values close to each other, we tried clustering with both the k values, we obtained better results at k=55 than at k=75, as the clusters at k=55 were more homogeneous than the clusters at k=75.

The labeled plot gives better insight into how the papers are grouped. Here the location of each paper on the plot is determined by t-SNE, and k-means determine the label (color). The clusters we obtained here are mostly homogeneous. Looking at the set of tight set clusters towards the left side of the slide, we see that the t-SNE and k-means agree on the clustering as those clusters are mostly the same color, this shows us there exists a particular structure to the literature which can be observed and measured to some extent.

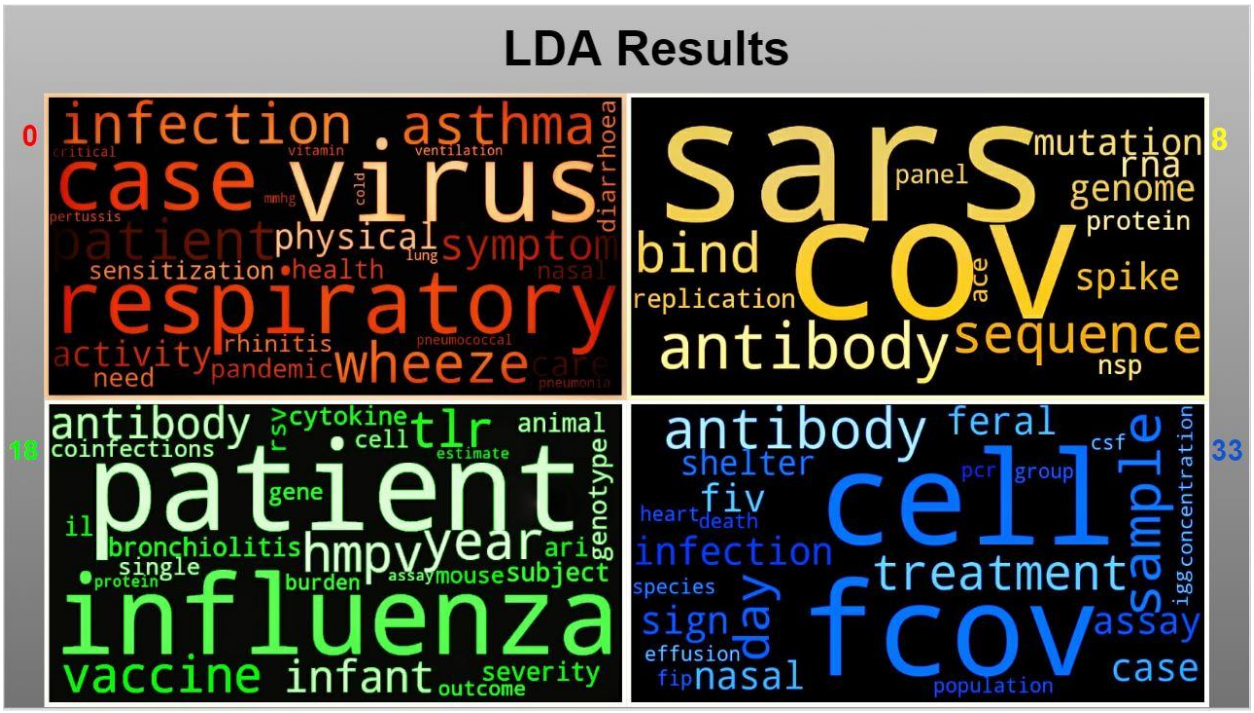


After obtaining the clusters and their color labels, we proceeded to obtain the topics that those clusters are based on, and we applied Topic modeling using Latent Dirichlet Allocation to do this.

Topic modeling is a statistical modeling technique that is applied to discover the abstract topics that occur in a collection of documents, Intuitively it means that given a document is about a specific topic, it is common for that word to appear more frequently than the other words. Topic models can help to organize and offer insights to understand the extensive collection of text documents.

Latent Dirichlet Allocation is an example of a topic model used to classify text in a particular document. It builds a topic per document model and words per document model, modeled as Dirichlet distributions. LDA is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the documents are similar.

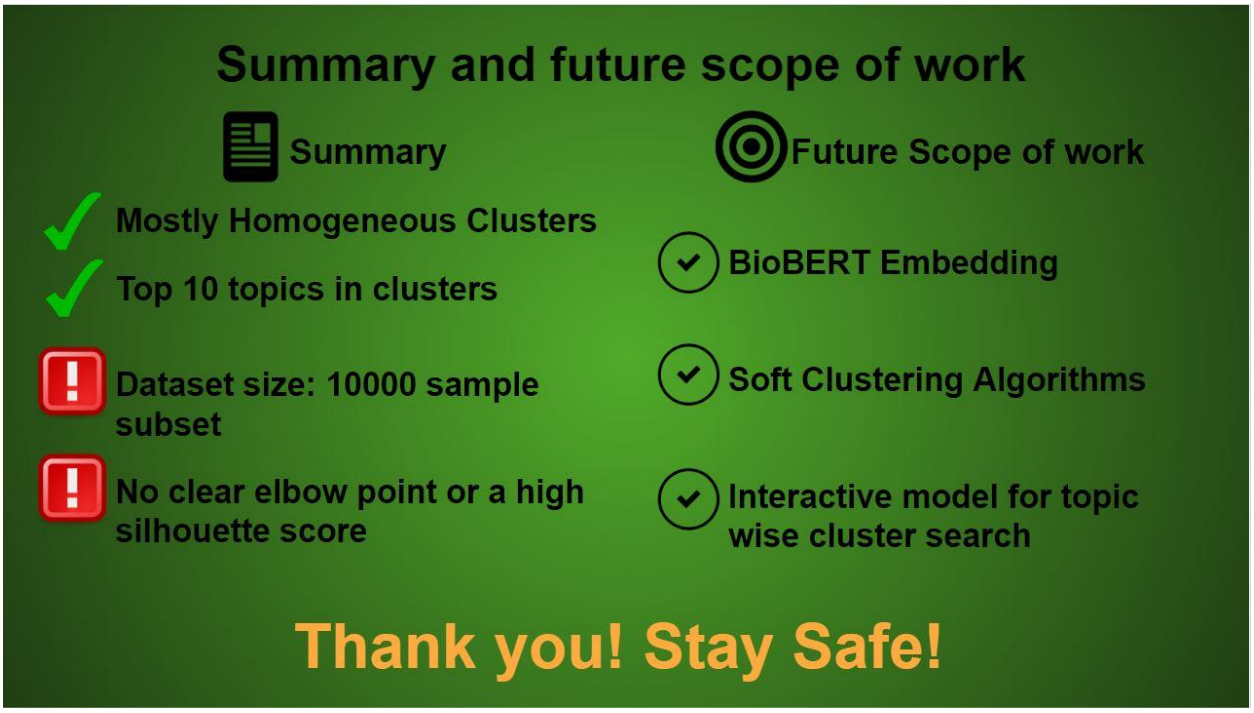
It allows us to view each document or set of documents in our case as a mixture of these topics.



The topic modeling results are displayed in two ways, one where we join the list of topics to the original data frame using the cluster number column, which was appended to the data frame after the clustering. And also as word clouds for visualization purposes.

Here we see four randomly selected cluster topics displayed, the color of the word cloud is matched with the color of the cluster displayed in the k-means plot.

We see that the first cluster has words like respiratory, asthma, wheeze, symptoms, etc. It is most likely that the cluster has papers that talk about the symptoms of the virus. Similarly, the 7th cluster has papers that are mostly about the structure or nature of virus replication.



Summary of the work done

The Covid-19 Open Research Dataset was downloaded from Kaggle, and the data was pre-processed and vectorized using TF-IDF. Since the data was massive, we randomly sampled 10,000 papers from among the 46,000 papers to reduce the dimensionality using PCA and then applied k-means over the reduced data to obtain the clusters. Although the clusters seem homogeneous and well clustered, the elbow curve used to select the optimal k-value did not show a very significant elbow point, which led to analyzing the silhouette scores, which again were not as high as we expected it to be. However, seeing how the data has an inherent structure to it makes it possible that the clustering could be better using other methods. After clustering, topic modeling using LDA was applied to obtain the topics that each cluster represents, and the results were displayed as word clouds.

Future scope for work

As stated above, other methods such as word embeddings using BioBERT can be used since BioBERT is trained explicitly on the medical text; it might improve the overall results. Also, applying soft clustering methods instead of hard clustering might give better or more exciting results as these papers and articles tend to have more than one major topic addressed in them.