IST-718 Project Report

CoVID-19 Literature Analysis

# 

Group-2

Nikita Rao Sira

Naga Litin Kumar Behata

Hua Cui

Naga Meena Venkata

Contents

[Introduction 2](#_Toc39612551)

[Background 2](#_Toc39612552)

[Motivation 2](#_Toc39612553)

[Problem Statement 2](#_Toc39612554)

[Solution Approach 2](#_Toc39612555)

[Data cleaning and pre-processing 3](#_Toc39612556)

[Dataset creation 3](#_Toc39612557)

[Data Cleaning & pre-processing 4](#_Toc39612558)

[Stop Words removal 4](#_Toc39612559)

[Tokenization, TF – IDF 5](#_Toc39612560)

[Analysis 5](#_Toc39612561)

[Dimensionality Reduction with PCA 5](#_Toc39612562)

[K-means Clustering 6](#_Toc39612563)

[Latent Dirichlet Allocation 8](#_Toc39612564)

[Classification 9](#_Toc39612565)

[Conclusion and Roadmap for Future Work 9](#_Toc39612566)

[Concerns 10](#_Toc39612567)

[References 10](#_Toc39612568)

# Introduction

## Background

Coronavirus is a family of viruses that can cause illness, which can vary from common cold and cough to sometimes more severe disease. SARS-CoV-2 (n-coronavirus) is the new virus of the coronavirus family, which first discovered in 2019, which has not been identified in humans before. It is a contiguous virus which started from Wuhan in December 2019. Which later declared as Pandemic by WHO due to high rate spreads throughout the world. Currently (on date 27 April 2020), there have been almost 3M cases and a total of 200K+ Deaths across the globe. Pandemic is spreading all over the world; it becomes more important to understand about this spread. This report is an effort to analyze the cumulative data of confirmed, deaths, and recovered cases over time.

## Motivation

In response to the COVID-19 pandemic, the White House and a coalition of leading research groups have prepared the COVID-19 Open Research Dataset (CORD-19). CORD-19 is a resource of over 51,000 scholarly articles, including over 40,000 with full text, about COVID-19, SARS-CoV-2, and related coronaviruses. This freely available dataset is provided to the global research community to apply recent advances in natural language processing and other AI techniques to generate new insights in support of the ongoing fight against this infectious disease. There is a growing urgency for these approaches because of the rapid acceleration in new coronavirus literature, making it difficult for the medical research community to keep up. So, can we simplify the search for related publications by clustering similar research articles together? And how can the content of the clusters be qualified?

## Problem Statement

Given the large number of research papers, it would be hard for healthcare workers to keep up with new information on the virus. Our goal is to build a topic model to cluster similar research papers together which will in turn help research papers regarding the virus under similar topics easily.

# Solution Approach

1. Data pre-processing: to get a clean dataset by treating NAs, duplicates etc.
2. Text Preprocessing: the preprocessing step usually consists of the tasks such as tokenization and filtering out stop words and vectorization
3. TF-IDF: by using the TF-IDF weight, the document search can deliver results that are most relevant to what is searched for
4. Clustering: we use k-means clustering to partition n documents in the context of text data into k clusters. representative around which the clusters are built
5. Dimensionality reduction: this is done using PCA to obtain optimal number of components that capture the greatest amount of variance
6. Topic Modeling: using LDA (Latent Dirichlet Allocation) for mapping a specific article into the topic space, we can then find related articles
7. Classification: build three classification models that can be used for recommending similar literatures

# Data cleaning and pre-processing

### Dataset creation

Metadata table description :

|  |  |
| --- | --- |
| **Column name** | **Description** |
| sha | Unique ID that was used to join the json literature |
| source\_x | The source of the literature |
| title | Title of the article/ literature |
| doi | Digital Object Identifier is a unique identifier to help readers easily locate a document from your citation |
| Pmcid | ID on PMC |
| pubmed\_id | ID on pubmed |
| license | Creative Commons (CC) license |
| abstract | The abstract of the literature |
| publish\_time | Date on which it was published |
| authors | The authors of the article |
| journal | The journals they were published on |

The JSON files have a structure with the following details :

**paper\_id:**001b4a31684c8fc6e2cfbb70304354978317c429

**metadata:**{} 2 items

**abstract:**[] 1 item

**body\_text:**[] 105 items

**bib\_entries:**{} 26 items

**ref\_entries:**{} 16 items

**back\_matter:**[] 4 items

Using the “paper\_id” from the JSON file and “sha” from metadata, we join the 2 sources to create a final data that contails details from metadata with the body text for every literature.

### Data Cleaning & pre-processing

* Within the original large dataset, each literature in the above metadata dataset is stored in a JSON file and there were 51,000 of them. We took a sample of 10,000 articles for our analysis
* First, we handled null values in the dataset and removed duplicates from the dataset which could have occurred if the same article were present in different journals



Fig-1

### Stop Words removal

* Part of the preprocessing will be finding and removing stop words (common words that will act as noise in the clustering step). Apart from removing the common stop words, we also used the spacy library which removes stop words specific to scientific articles

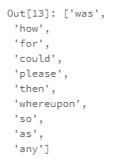
 

Fig-2 Fig-3

* Then let’s look at the body text word count. Most papers are about 680 words in length. The long tails are caused by outliers. In fact, ~98% of the papers are under 5,000 words in length while a select few are over 5,000

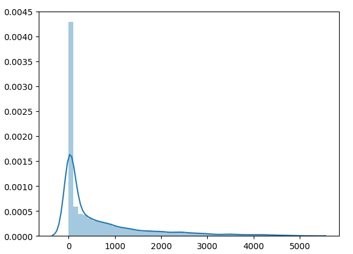


Fig-4

### Tokenization, TF – IDF

* By using tokenization and tf-idf, we converted our string formatted data into a measure of how important each word is to the instance out of the literature as a whole. Only the top 2 \*\* 12 features were used for vectorization because having more features means longer runtimes. All of these were a part of one spark pipeline that generated the data frame below.

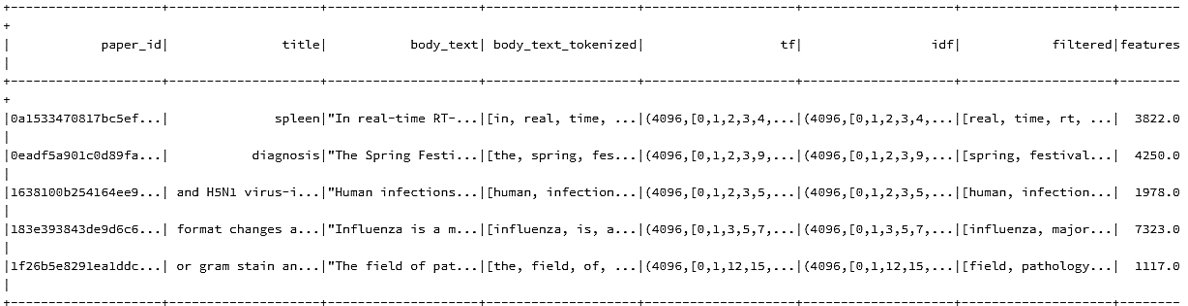


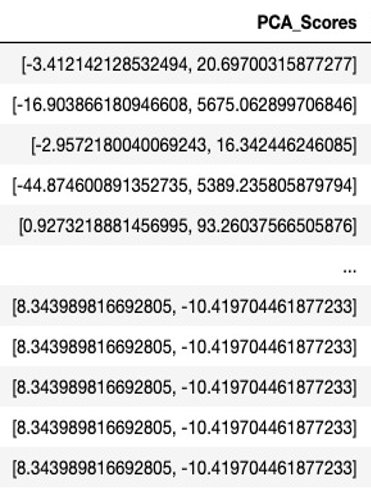
Fig-5

# Analysis

## Dimensionality Reduction with PCA

We used PCA to reduce the dimensionality and chose the top 2 components from PCA. This explained 92% of the variance​. In order to visualize the clusters it is easier if the dimensionality is reduced

We also chose the top 3 components in PCA. This explained 95% of the variance.

 A screenshot of a cell phone

Description automatically generated

Fig-6 (Top 2 components) Fig-7 (Top 3 components)

## K-means Clustering

* Before running k-means, we standardized our data
* By plotting the elbow plot we obtain the optimal number of clusters​
* We got 5 when we ran this on the sample dataset and 11 for the complete dataset

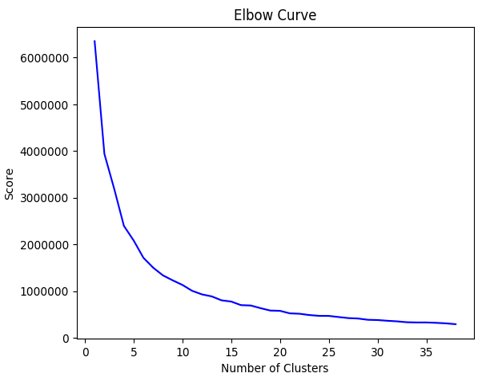


Fig-8

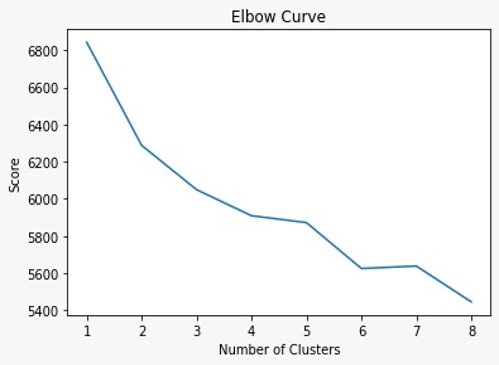


Fig-9

* We finalized the number as 6, and upon generating the k-means output we used the data that was originally transformed using top 2 components PCA in order to plot the clusters

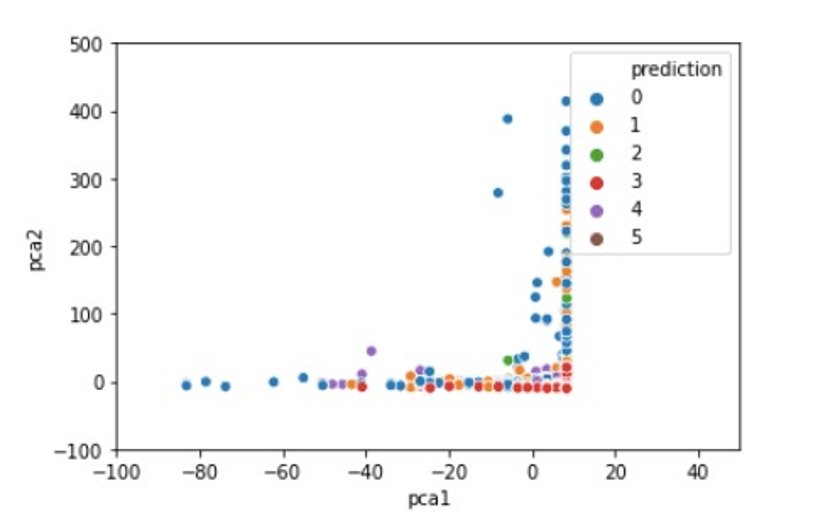


Fig-10

* The plot using the top 3 PCA components and k-means output within k=6

A picture containing text, map

Description automatically generated

Fig-11

## Latent Dirichlet Allocation

The topic modelling method LDA is an unsupervised, probabilistic modelling method which extracts topics from a collection of papers. A topic is defined as a distribution over a fixed vocabulary. For each topic from LDA, we specified a topic name on our own by analyzing the type of words within it.

The topics we finalized are shown below :

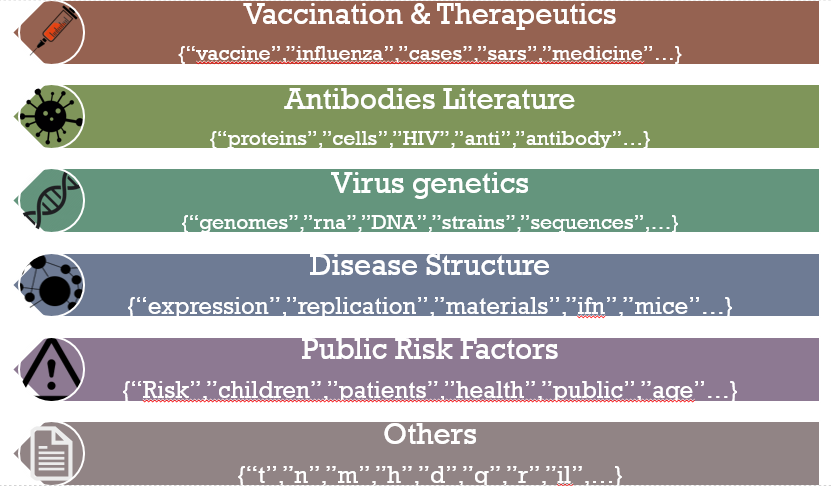


Fig-12

We can see that all the garbage words got classified into “Others” when we used 6 clusters instead of 5. When we used k=5 for clustering, the meaningless words were a part of other clusters.

## Classification

After running k-means, the data is now clustered and labelled. ​ Using this we can do supervised learning to see how accurately the literature can be classified into its clusters.​

This is just one way to evaluate the if k-means split the data into meaningful clusters​

We split our dataset into 80% test and 20% train for classification.

We used three classification techniques which are:​

* Logistic Regression (Multiclass)​
* Decision Tees​
* Random Forest​
* Multi-Layer Peceptron

The classification using 3 PCA components :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Dataset | Details | Logistic | Decision tree | Random Forest | Multi-layer perceptron |
| Train | Accuracy | 35.4% | 58.2% | 66.7% | 53.7% |
| Train | Weighted precision | 41.6% | 58.6% | 74.9% | 53.6% |
| Train | Weighted recall | 46.2% | 60.4% | 74.4% | 56.4% |
| Test | Accuracy | 32.8% | 54.4% | 60.4% | 51.8% |
| Test | Weighted precision | 38.2% | 54.4% | 60.5% | 52.4% |
| Test | Weighted recall | 41.9% | 56.4% | 61.6% | 54.3% |

# Conclusion and Roadmap for Future Work

Based on our findings we were able to cluster the sample of the literature articles. The samples closely match the tasks mentioned in the Kaggle challenge.

Grouping the literature in this way allows the professionals to quickly find material related to a specific topic. Instead of having to manually search for related work, every publication is connected to a larger topic cluster

Both the clusters and keywords are found through unsupervised learning models and can be useful in revealing patterns that humans may not have even thought about. In no part of this project did we have to manually organize the papers: the results are due to latent connections in the data

As this is an unsupervised learning problem, the validation of the results is difficult. First, we used k-means for clustering and labelling the data. After this, we used supervised learning to see how accurately the literature can be classified into clusters.

## Concerns

* Possible false positives, difficult to draw an exact line between subjects
* K-means might not necessarily group instances in a predictable way. Due to their unsupervised nature, there is no 'right answer' for how the papers should be clustered. This could be difficult to debug if problems arise

# References

* <https://towardsdatascience.com/apache-spark-mllib-tutorial-7aba8a1dce6e>
* <https://spark.apache.org/docs/1.5.2/ml-features.html#onehotencoder>
* <https://spark.apache.org/docs/2.1.0/ml-classification-regression.html>
* <http://apache-spark-user-list.1001560.n3.nabble.com/MLlib-Possible-to-use-SVM-with-Radial-Basis-Function-kernel-rather-than-Linear-Kernel-td14403.html>
* <https://www.analyticsvidhya.com/blog/2019/11/build-machine-learning-pipelines-pyspark/>
* https://spark.apache.org/docs/latest/ml-features
* https://github.com/MaksimEkin/COVID19-Literature-Clustering