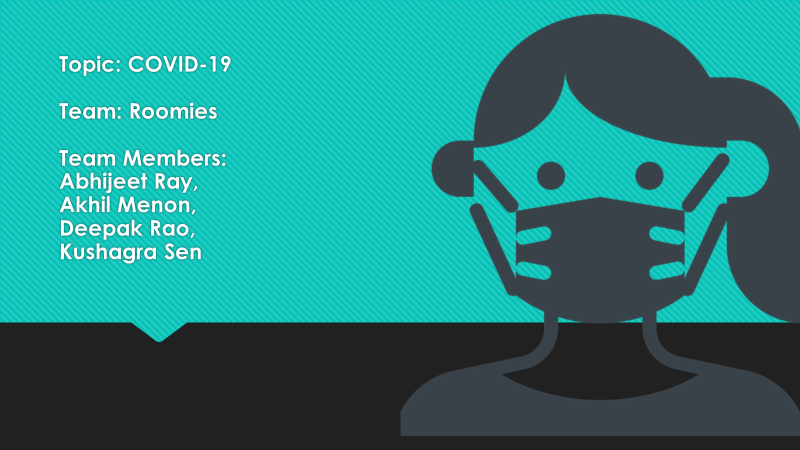
**COVID – 19 PAPER PRESENTATION**



**CONTENTS:**

|  |  |  |
| --- | --- | --- |
| S.No. | Title | Pg.no |
| 1 | Introduction | 3 |
| 2 | Objective | 3 |
|  | 2.1 Business Problem | 3 |
|  | 2.2 Business Solution | 4 |
| 3 | Architecture | 5 |
| 4 | Data Sourcing | 6 |
| 5 | Data Cleaning | 8 |
| 6 | Data Extraction | 9 |
| 7 | Data Transformation | 10 |
| 8 | Data Loading | 14 |
| 9 | Data Storage | 16 |
|  | 9.1 High-Level Architecture | 17 |
|  | 9.2 Storage | 17 |
| 10 | Data Visualization | 18 |
|  | 10.1 Symptoms | 18 |
|  | 10.2 Current Situation Worldwide | 19 |
|  | 10.2.1 Virus Spread in Continents & most affected Countries | 19 |
|  | 10.2.2 Cases as of 17th April 2020 | 20 |
|  | 10.3 Sentiment and polarity (To check for Biases) | 20 |
|  | 10.4 Exploring Global Data | 21 |
|  | 10.5 Parts of world generating the bulk of coronavirus research? | 21 |
|  | 10.6 How to contain Covid-19 | 22 |
| 11 | Finding Possible cures for COVID-19 | 23 |
|  | 11.1 Using Word2Vec to find cures | 24 |
|  | 11.2 PCA | 29 |
|  | 11.3 Takeaways | 35 |
| 12 | Anserini+BERT-SQuAD for Semantic Corpus Search | 35 |
|  | 12.1 BERT-Serini Architecture | 35 |
|  | 12.2 Question Answer session with BERT | 36 |
| 13 | References | 38 |

**1. INTRODUCTION [1]**

Coronavirus is a family of viruses that can cause illness, which can vary from common cold and cough to sometimes more severe disease. Middle East Respiratory Syndrome (MERS-CoV) and Severe Acute Respiratory Syndrome (SARS-CoV) were such severe cases with the world already has faced.

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 14 April 2020), this leads to a total of 120K+ Deaths across the globe, including 83+ deaths alone in Europe.

Pandemic is spreading all over the world; it becomes more important to understand about this spread. This is an effort to analyze the cumulative data of confirmed, deaths, and recovered cases over time. In this notebook, the main focus is to analyze the spread trend of this virus all over the world.

**2.OBJECTIVE [2]**

The objective of this project is to perform analytics to understand the pandemic COVID-19 and further find cures and ways to contain it. The secondary task is to integrate the state-of-the-art Natural Language Processing (NLP) techniques to create a method to ask a database a question and have it return sensible results that answer that question. We are applying modern document analysis methods to extract tabular, figure, and numeric data when possible.

**2.1 BUSINESS PROBLEM**

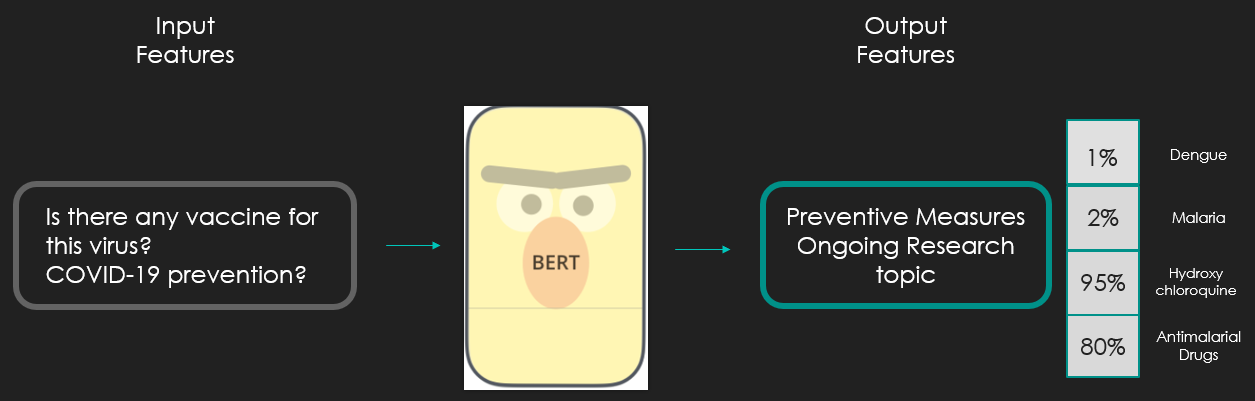
It’s difficult to find out what people are trying to say. It’s even tricky to try to search through a corpus of text to find something relevant in a short time. People across the globe should be able to search effectively, without getting bogged down in unnecessary or irrelevant messages. With so many differences in meaning and phrasing, how are we supposed to find what we want? This is where the scientific search system comes into the picture.

Did you find an interesting article and want to easily find related research without browsing all existing publications yourself? Do you have a specific research question and want to discover relevant articles?

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. We need to leverage the world's latest artificial intelligence techniques to develop text and data mining tools that can help the medical community develop answers to high priority scientific questions.

Also, in this world of digital marketing, people bid for a very relevant keyword but show irrelevant content which increases their website traffic but reduces the importance of organic results. This is a problem for researchers who are trying to find relevant articles on search engines but only find a few!

* 1. **BUSINESS SOLUTION [3]**



We will be using BERT model to achieve our objective. BERT applies the bidirectional training of Transformer, a popular attention model, to language modelling. This is in contrast to previous efforts which looked at a text sequence either from left to right or combined left-to-right and right-to-left training. The results from many research paper’s show that a language model which is bidirectionally trained can have a deeper sense of language context and flow than single-direction language models.

As shown in the above diagram input features which are the questions related to COVID-19 in our case will be fed to the BERT model. The BERT model will process the data and then return the valid answers based on the research articles used to train it.

In general, the best procedure can be:

* Develop a model to search relevant and reliable information quickly (Using Bert)
* Perform analytics to understand the pandemic COVID-19 and further find cures and ways to contain it.
* Insights regarding which parts of the world generate the bulk of coronavirus research.
* Sentiment of Abstracts in paper articles
* Finding cures for COVID-19
* Finding ways to contain COVID-19

Once ready, this tool would be generally useful for all of the scientific R&D and would like to build this into a publically available web-based tool that would be free to use.

**3.ARCHITECTURE [4]**

A close up of a map

Description automatically generated

**Data Extraction**

We are extracting the date from multiple sources:

COVID-19 Open Research Dataset, a free resource of over 47,000 scholarly articles about COVID-19.

Each paper is represented as a single JSON object and the dataset contains all COVID-19 and coronavirus-related from the following sources: PubMed's PMC, WHO, bioRxiv and medRxiv pre-prints. This dataset is intended for applying recent advances in NLP to generate new insights in support of the fight against this infectious disease.

The data is extracted using a selenium web driver with Python to overcome the manual work of downloading data every time it is updated. We are extracting the data in JSON format and is needed to get converted into CSV format for our future analysis and modeling.

**Data Transformation**

We have written a Python function to convert JSON into CSV.

**Data Loading**

The converted CSV data is then loaded into Google Big Query which is a fully-managed data warehouse on RESTful web service. It enables scalable, cost-effective and fast analysis in conjunction with Cloud Storage.

The data will be pulled from the cloud storage as per the requirement for Predictive analytics, BERT, and data visualization.

**4.Data Sourcing [5]**

We have collected data from COVID-19 Open Research Dataset, a free resource of over 52,000 scholarly articles, including over 41,000 with full text, about COVID-19 and the coronavirus family of viruses for use by the global research community. The corpus is updated weekly as new research is published in peer-reviewed publications and archival services like bioRxiv, medRxiv, and others.

The dataset contains all COVID-19 and coronavirus-related research (e.g. SARS, MERS, etc.) from the following sources:

·      **PubMed's PMC** open access corpus using this query (COVID-19 and coronavirus research)

· Additional COVID-19 research articles from a corpus maintained by the **WHO**

·      **bioRxiv and medRxiv** pre-prints using the same query as PMC (COVID-19 and coronavirus research)

·      **Latest Corona Cases (Source: Kaggle)**

Each paper is represented as a single JSON object and the dataset contains all COVID-19 and coronavirus-related information.

Simplified format of the above-mentioned datasets:

{

"paper\_id": <str>, # 40-character sha1 of the PDF

"metadata": {

"title": <str>,

"authors": [ # list of author dicts, in order

{

"first": <str>,

"middle": <list of str>,

"last": <str>,

"suffix": <str>,

"affiliation": <dict>,

"email": <str>

},

...

],

"abstract": [ # list of paragraphs in the abstract

{

"text": <str>,

"cite\_spans": [

# list of character indices of inline citations

# e.g. citation "[7]" occurs at positions 151-154 in "text"

# linked to bibliography entry BIBREF3

{

"start": 151,

"end": 154,

"text": "[7]",

"ref\_id": "BIBREF3"

},

...

],

"ref\_spans": <list of dicts similar to cite\_spans>, # e.g. inline reference to "Table 1"

"section": "Abstract"

},

...

],

"body\_text": [ # list of paragraphs in full body

# paragraph dicts look the same as above

{

"text": <str>,

"cite\_spans": [],

"ref\_spans": [],

"eq\_spans": [],

"section": "Introduction"

},

...

{

...,

"section": "Conclusion"

}

],

"bib\_entries": {

"BIBREF0": {

"ref\_id": <str>,

"title": <str>,

"authors": <list of dict> # same structure as earlier,

# but without `affiliation` or `email`

"year": <int>,

"venue": <str>,

"volume": <str>,

"issn": <str>,

"pages": <str>,

"other\_ids": {

"DOI": [

<str>

]

}

},

"BIBREF1": {},

...

"BIBREF25": {}

},

"ref\_entries":

"FIGREF0": {

"text": <str>, # figure caption text

"type": "figure"

},

...

"TABREF13": {

"text": <str>, # table caption text

"type": "table"

}

},

"back\_matter": <list of dict> # same structure as body\_text

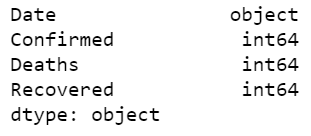
}

}

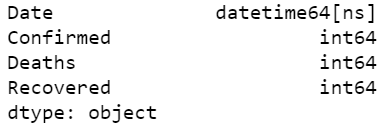
**5.Data Cleaning**

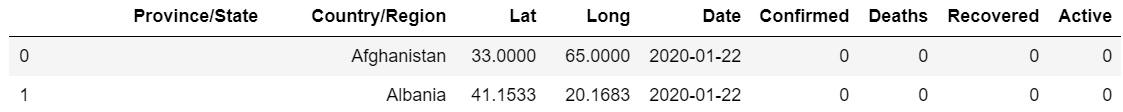
1. Dataset on which the cleaning task was performed: Latest Corona Cases.
2. The following activities were performed on the above dataset:

* Meaningful names were used for the column names which were too long and were non-descriptive. For example, the “Recovered/from+cases” column was converted to “Recovered”.
* The data type of Date column was changed to DateTime data type from object datatype by invoking the “to\_datetime” function of the pandas library on the Date column.

Datatype of “Date” column before conversion:

The data type of “Date” column after conversion:



* The total number of null values in each column was identified and then they were replaced with blank spaces.
* Country/Region column had samples like “Mainland china” which had to be shortened for efficient analysis, so we converted those names to descriptive & shorter sample names like “China”
* Created a new column “Active” to have a clear picture of active cases globally. The column values were derived using Active = Confirmed cases – Deaths – Recovered.
* There were odd data samples in the Province/State and Country/Region column, these odd samples were cases that were identified on the ships and not in any country or state. To handle these odd samples, we created a new variable “ships” and inserted such samples ex. “Grand Princess” and “Cruise Ship”

**6.Data Extraction**

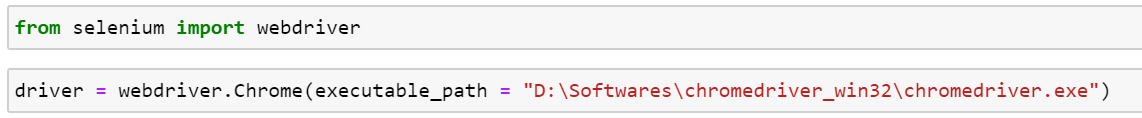
The data had to be fetched from multiple sources and COVID-19 being progressing issue articles are being added daily. To avoid manual intervention and automate the entire processing of extracting the datasets we used the Selenium Webdriver framework.

Selenium:

It’s a web driver used to automate the data extraction for real-time processing.

**Steps used in the extraction of datasets:**

* Create a WebDriver instance.



* Navigate to a Web page.



* Locate an HTML element on the Web page.



* Perform an action on an HTML element.
* Anticipate the browser response to the action.

**7.DATA TRANSFORMATION**

The JSON data extracted from the above step have to be transformed into an easily analyzable format. We chose to convert it to a .CSV file so that we can import this easily into Google BigQuery. We use python code to do the transformation.

JSON is an acronym standing for JavaScript Object Notation. The JSON library in python can parse JSON from strings or files. The library parses JSON into a Python dictionary or list. We came across various circumstances where we receive data in JSON format and we need to send or store it in CSV format. Since the CSV files can easily be opened using **LibreOffice Calc** in ubuntu or **Microsoft Excel** in windows the need for JSON to CSV conversion usually increases.

We quickly explore the biorxiv subset of the papers. Since it is stored in JSON format, the structure is likely too complex to directly perform analysis. Thus, we not only explore the structure of those files, but we also provide the following helper functions for us to easily format inner dictionaries from each file:

For eg.

format\_name(author)

format\_affiliation(affiliation)

format\_authors(authors, with\_affiliation=False)

format\_body(body\_text)

format\_bib(bibs)

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generated

Parsing JSON: [6]

The parsing of the JSON file can be done using the functions defined below:

load\_files(dirname)

generate\_clean\_df(all\_files)

**Code Snippet:**

**A screenshot of a cell phone

Description automatically generated**

**For Example:**

Now json.loads() function parses the JSON string data and it can be used as a normal dictionary in python. And we can access the values using keys. For example, here we can access email Id using key ‘email’ as shown below:

print json\_parsed[‘email’]

‘[**smithjack@gmail.com**](mailto:smithjack@gmail.com)**’**

This is a simple example of parsing JSON, but we can have a JSON file where data is much more complex. For example JSON data where values are lists and are accessed using list index as shown below :

Data = '{"employee\_name": "James", "email": "james@gmail.com", "job\_profile": ["Team Lead", "Sr. Developer"]}'

It can be parsed like this :

Parsed\_data = json.loads(Data)

print Parsed\_data[‘job\_profile’][0]

**"Team Lead"**

The complexity increases further for JSON data where values are lists having dictionaries within. Again, values will be accessed using a list index and key-value pair as shown below:

Data = '{"employee\_name": "James", "email": "james@gmail.com", "job\_profile": [{"title1":"Team Lead", "title2":"Sr. Developer"}]}'

It can be parsed like this:

Parsed\_data = json.loads(Data)

print Parsed\_data[‘job\_profile’][0][‘title1’]

**"Team Lead"**

Similarly, other fields can be accessed from the parsed data.

**Converting JSON to CSV using Python:**

CSV (Comma Separated Values) format is the most common import and export format for spreadsheets and databases. CSV file format separates values using commas as delimiters. For simple JSON data, keys will be headers for the CSV file and values the descriptive data.

In this section, we show you how to manually generate the CSV files. As you can see, it's now super simple because of the format\_ helper functions. In the next sections, we show you how to generate them in 3 lines using the load\_files and generate\_clean\_dr helper functions.

This can be parsed and converted to CSV using python as shown below:

A screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated

The python program written above will open a CSV file in the tmp folder and write the content of the JSON file into it and close it at the end. Make sure to close the file at the end to save the contents.

JSON files can have much more complex structures than CSV files, so a direct conversion is not always possible. The above-mentioned examples will lay the foundation for converting JSON data with high complexity to CSV.

**8.DATA LOADING [7]**

Big query is a really good, cheap, quick (insert superlative) platform to query and visualize data. It has a simple connector into DataStudio which is great to use in reporting as it’s very sharable.

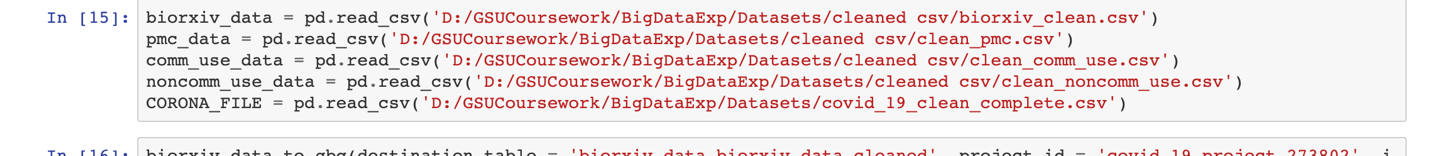
Big Query has its limits and one of them is getting data into it. To upload directly within the interface there is a 10mb limit. You can import from Cloud Storage, but you might find issues with the data format. E.g. commas in fields being read incorrectly by Big Query.

In this case, we need to follow the below steps to achieve successful loading:

**Step 1) Read the CSV**

df = pd.read\_csv('path\_to\_csv', encoding='utf-8')

**Code Snippet:**



Replace the path to CSV with the relative or absolute path to your file

Set the projectid of your project you are going to import into, you can get this from the URL of the big query dashboard.

**Step 2) Load the CSV in the BigQuery using to\_gbq function**

Finally set the dataset and the table name in the to\_gbq function.

This will put the entire dataset into the big query with the specified table

to\_gbq(df, 'datasetname.tablename', projectid ,chunksize=10000, if\_exists='append')

**Code Snippet:**

A screenshot of a social media post

Description automatically generated

    There you have it a simple way to import data into Big Query.

**Step 3) Read data from bigquery using Python**

Next step is to read dataset from bigquery using read\_gbq in jupyter notebook. We can achieve this by using gbq connector. The above details can be explained by the code snippet below:

**Code Snippet:**

A screenshot of a social media post

Description automatically generated

**9.DATA STORAGE: BigQuery [8]**

In this project, we have decided to use bigquery for our storage purpose. Google’s BigQuery is an enterprise-grade cloud-native data warehouse. BigQuery was first launched as a service in 2010 with general availability in November 2011. Since inception, BigQuery has evolved into a more economical and fully-managed data warehouse that can run blazing fast interactive and ad-hoc queries on datasets of petabyte-scale. Also, BigQuery now integrates with a variety of Google Cloud Platform (GCP) services and third-party tools which makes it more useful. BigQuery is serverless, or more precisely data warehouse as a service. There are no servers to manage or database software to install. BigQuery service manages underlying software as well as infrastructure including scalability and high-availability. The pricing model is quite simple - for every 1 TB of data processed you pay $5. BigQuery exposes a simple client interface which enables users to run interactive queries. Overall, you don’t need to know much about underlying BigQuery architecture or how this service operates under the hood. That’s the whole idea of BigQuery - you don’t need to worry about architecture and operation.

**9.1 High-level architecture**

BigQuery is built on top of Dremel technology which has been in production internally in Google since 2006. Dremel is Google’s interactive ad-hoc query system for analysis of read-only nested data. Original Dremel papers were published in 2010 and at the time of publication, Google was running multiple instances of Dremel ranging from tens to thousands of nodes.

BigQuery and Dremel share the same underlying architecture. By incorporating columnar storage and tree architecture of Dremel, BigQuery offers unprecedented performance. But, BigQuery is much more than Dremel. Dremel is just an execution engine for the BigQuery. BigQuery service leverages Google’s innovative technologies like Borg, Colossus, Capacitor, and Jupiter.

As illustrated below, a BigQuery client (typically BigQuery Web UI or bg command-line tool or REST APIs) interacts with the Dremel engine via a client interface. Borg - Google’s large-scale cluster management system - allocates the compute capacity for the Dremel jobs. Dremel jobs read data from Google’s Colossus file systems using the Jupiter network, perform various SQL operations and return results to the client. Dremel implements a multi-level serving tree to execute queries that are covered in more detail in the following sections.

A screenshot of a cell phone

Description automatically generated

Figure-1: A high-level architecture for BigQuery service.

It is important to note, BigQuery architecture separates the concepts of storage (Colossus) and compute (Borg) and allows them to scale independently - a key requirement for an elastic data warehouse. This makes BigQuery more economical and scalable compared to its counterparts.

**9.2 Storage:**

The most expensive part of any Big Data analytics platform is almost always disk I/O. BigQuery stores data in a columnar format known as Capacitor. As you may expect, each field of BigQuery table i.e. column is stored in a separate Capacitor file which enables BigQuery to achieve a very high compression ratio and scan throughput. In 2016, Capacitor replaced ColumnIO - the previous generation optimized columnar storage format. Unlike ColumnIO, Capacitor enabled BigQuery to directly operate on compressed data, without decompressing the data on the fly.

You can import your data into BigQuery storage via Batch loads or Streaming. During the import process, BigQuery encodes every column separately into Capacitor format. Once all column data is encoded, it’s written back to Colossus. During encoding, various statistics about the data are collected which is later used for query planning.

BigQuery leverages Capacitor to store data in Colossus. Colossus is Google’s latest generation distributed file system and successor to GFS (Google File Systems). Colossus handles cluster-wide replication, recovery and distributed management. It provides client-driven replication and encoding. When writing data to Colossus, BigQuery makes some decisions about the initial sharding strategy which evolves based on the query and access patterns. Once data is written, to enable the highest availability BigQuery initiates geo-replication of data across different data centers.

In a nutshell, Capacitor and Colossus are key ingredients of industry-leading performance characteristics offered by BigQuery. Colossus allows splitting of the data into multiple partitions to enable blazing-fast parallel read whereas Capacitor reduces requires scan throughput. Together they make possible to process terabyte data per second.

**10. Data Visualizations to find insights**

For visualizations, we are using **Plotly** Python Open Source Graphing Library. Plotly is another great Python visualization tool that’s capable of handling geographical, scientific, statistical, and financial data. Plotly has several advantages over matplotlib. One of the main advantages is that only a few lines of codes are necessary to create aesthetically pleasing, interactive plots. The interactivity also offers several advantages over static matplotlib plots such as:

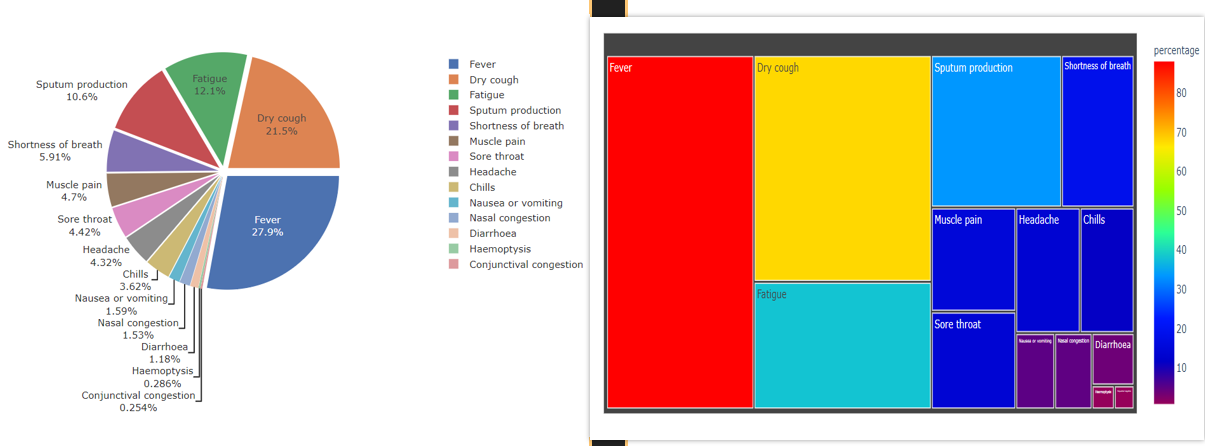
1) Saves time when initially exploring the dataset

2) Makes it easy to modify and export the plot

3) Offers a more ornate visualization, which is well-suited for conveying the important insights hidden within the dataset

**10.1 Symptoms [9]**

Understanding symptoms are important in order to fight this virus. Signs and symptoms of coronavirus disease 2019 (COVID-19) may appear two to 14 days after exposure. This time after exposure and before having symptoms is called the incubation period. We collected the data from Source: <https://en.wikipedia.org/wiki/Coronavirus_disease_2019>. After collecting the data, we visualized tree plot and pie plot as shown below.



**Observation:** When the virus does cause symptoms, common ones include fever, dry cough, fatigue, sputum production, sore throat, muscle pain, chills, headache.

Fever constitutes 30% of the symptoms indicating having a fever is a strong indication of Corona, but there is a chance that the person might be suffering fever because of various reasons. As can be seen from the pie plot, fever, dry cough and fatigue contributes to around 65% to symptoms informing that if a person is suffering from these 3 combinations, he must consult doctor immediately.

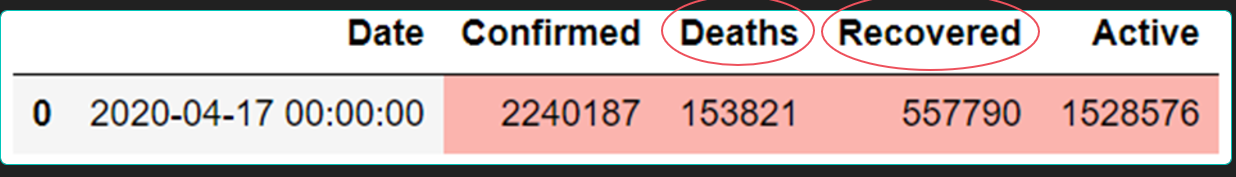
**10.2 Current Situation Worldwide**

**10.2.1 Spread Trends in the World, Continents and most affected Countries**



**Observation:** US, China, Germany, Italy, Spain, France and UK are badly hit by virus

**10.2.2 Cases as of April 17th 2020**



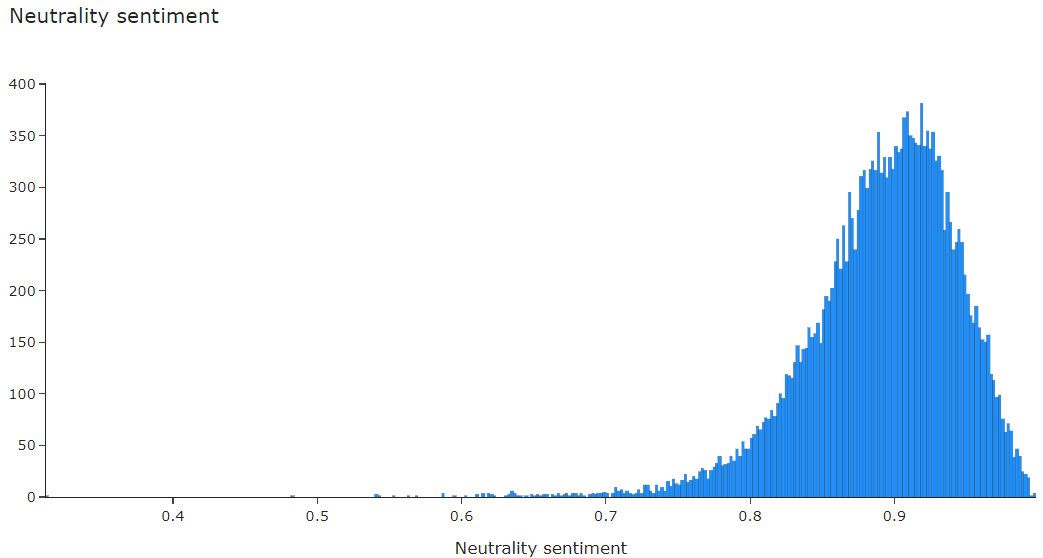
#### Observation: There are more recovered cases than deaths at this point of time

### **10.3 Sentiment and polarity (To check for Biases)**

Sentiment and polarity are quantities that reflect the emotion and intention behind a sentence. Now, we will look at the sentiment of the paper abstracts using the NLTK library. The main purpose of sentiment analysis is to make sure the abstracts of the papers don’t have any biases. To find this, we performed neutrality check of the Research Papers

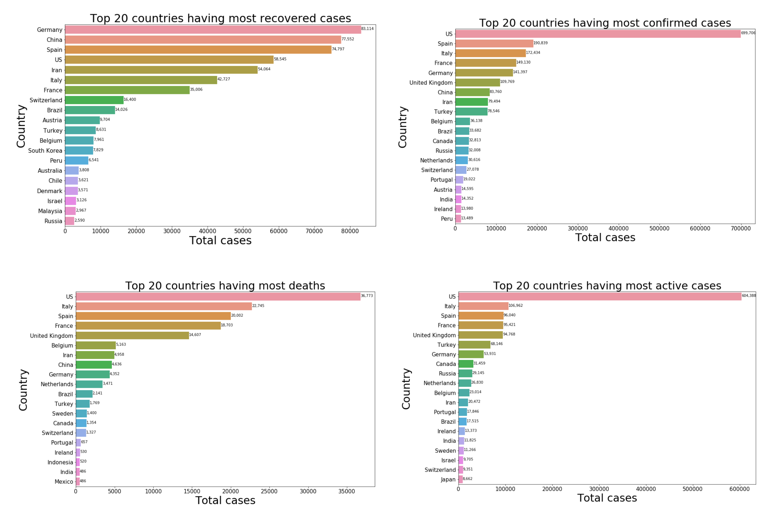
### **Neutrality sentiment**

Neutrality sentiment refers to the level of bias or opinion in the text. It is a score between 0 and 1; the greater the score, the more neutral/unbiased the abstract is.



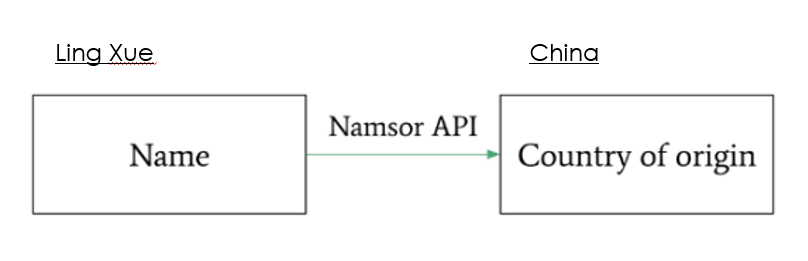
**Observation:** From the above plot, we can see that the neutrality sentiment distribution has a strong leftward (negative) skew. There is also a significant peak at 1 (the maximum value). This suggests that the abstracts tend to be very neutral and unbiased in general, which is great news. After all, research papers are meant to spread facts and not opinion.

* 1. **Let’s explore the global data:**

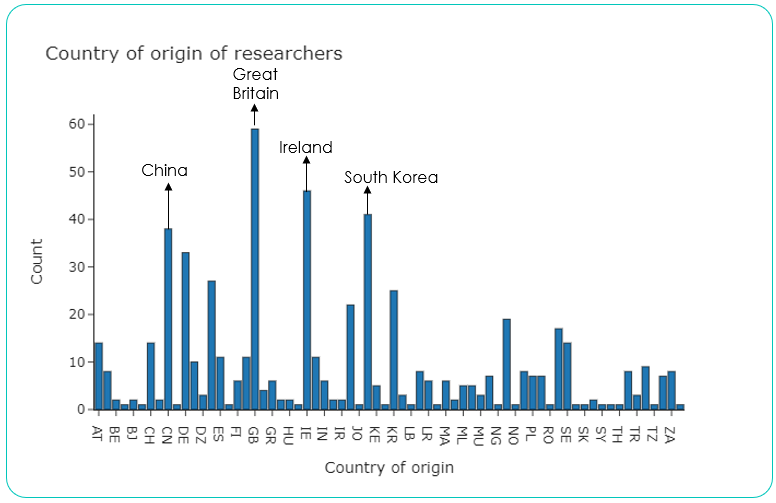


**Observation:** The above four charts depict the recovered, confirmed, deaths, and active cases of the top 20 countries globally. A common trend was observed in all the graphs except the one which shows the number of recovered cases with Germany surpassing china in terms of most recovered cases. No wonder Germany has the best healthcare facilities.

* 1. **Which parts of the world generate the bulk of coronavirus research?**



* Every research paper in the corpus is written by one or more authors, and the names of these authors could provide some insights regarding which parts of the world generate the bulk of coronavirus research.
* To find out a country that is actively contributing towards the cause we had to fetch country names of authors using **Namsor API** because our dataset didn’t have country names.
* As shown in the above figure Namsor API takes in the author names, for example, Ling Xue and returns the country of origin.



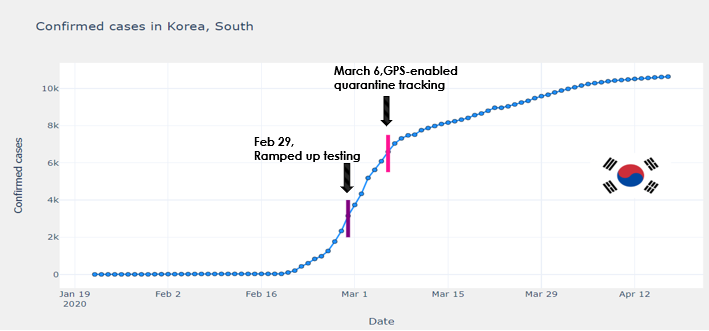
* Using the fetched names from Namsor API we went ahead with our analysis to find the country with the most number of research papers.
* Observation: As evident from the above histogram Great Britain tops the chart followed by Ireland, China, and South Korea.
  1. **How to contain Covid-19?**
* Now the major question remains unanswered. How do we contain Covid-19? The best way is to learn from countries which were able to do so.
* Let’s take a look at top 5 countries:

 A screenshot of a social media post

Description automatically generated A screenshot of a cell phone

Description automatically generated

* The number of confirmed cases in the above three countries (US, Italy, Iran) is growing exponentially which means they are still not able to flatten the curve.
* Let’s go further and take a look at China and South Korea:

**Observations:**

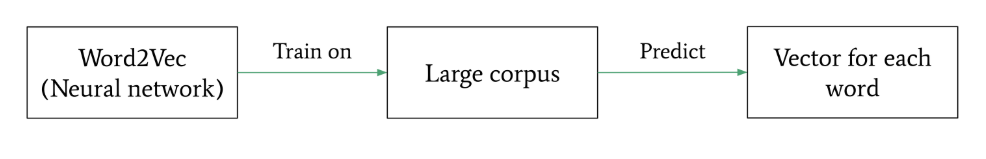
* Both China and South Korea were able to flatten the curve (number of confirmed cases over the period), unlike the above three countries.
* COVID-19 originated in China, despite which they were able to flatten the curve using three strict measures at specific times accompanied by constant monitoring of curve.
* The orange represents when Wuhan was locked down (Jan 23), the blue represents when factories were closed across China (Feb 13), and the green represents when the complete (total) lockdown was imposed across China (Feb 19). Notice how the curve starts to flatten after the complete lockdown is imposed. Complete lockdown helps reduce community transmission and mitigate the virus.
* Coming to South Korea, unlike china they couldn’t afford a complete shutdown because it would have shackled their economy. The purple represents when South Korea ramped up testing (Feb 29), and the pink represents when a new GPS-enabled quarantine tracking app was deployed by the South Korean government (March 6). These two measures have together worked to reduce community transmission and flatten curve towards the end of the first week of March.

# **11. Finding Possible cures for COVID-19**

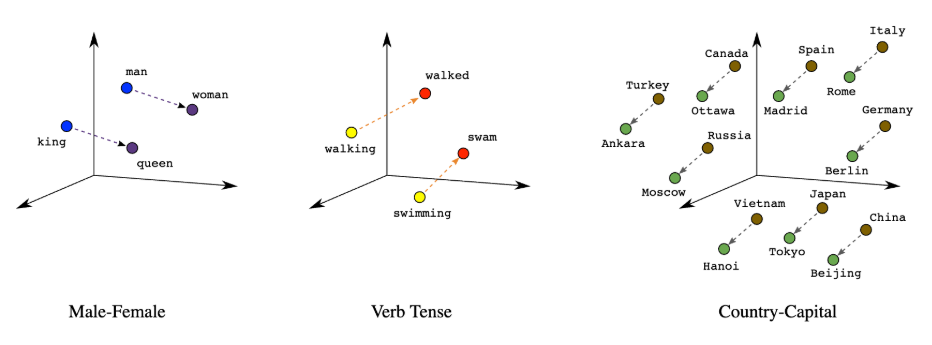
# Now, we will leverage the power of unsupervised machine learning to try and find possible cures (medicines and drugs) to COVID-19

### **Unsupervised NLP and Word2Vec**

### Unsupervised NLP involves the analysis of unlabeled language data. Certain techniques can be used to derive insights from a large corpus of text. One such method is called Word2Vec. Word2Vec is a neural network architecture trained on thousands of sentences of text. After training, the neural network finds the optimal vector representation of each word in the corpus. These vectors are meant to reflect the meaning of the word. Words with similar meanings have similar vectors.



As stated earlier, each word is associated with a vector. Amazingly, these vectors can also encode relationships and analogies between words. The diagram below illustrates some examples of linear vector relationships representing the relationships between words.



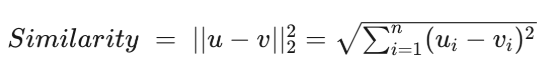
In the above image, we can see that word vectors can reflect relationships such as "King is to Queen as Man is to Woman" or "Italy is to Rome" as "Germany is to Berlin". These vectors can be also be used to find unknown relationships between words. These unknown relationships may help us find latent knowledge in research papers and find drugs that can possibly cure COVID-19.

## **11.1 Using Word2Vec to find cures**

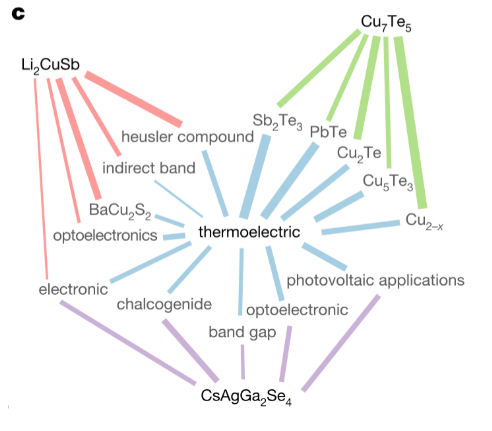
We can take advantage of these intricate relationships between word vectors to find cures for COVID-19. The steps are as follows:

1. Find common words related to the study of COVID-19, such as "infection", "CoV", "viral", etc.
2. Find the words with lowest Euclidean distance to these words (most similar words).
3. Finally, find the words most similar to the words (second order similarity) found in step two. These words will hopefully contain potential COVID-19 cures.

Note that the similarity between two Word2Vec vectors is calculated using the formula below (where u and v are the word vectors).



The three steps mentioned above is actually inspired by a research paper called ["Unsupervised word embeddings capture latent knowledge from materials science literature"](https://www.nature.com/articles/s41586-019-1335-8), where the authors find new materials with desirable properties (such as thermoelectricity) solely based on a large corpus materials science literature. These materials were never used for these purposes before, but they outperform old materials by a large margin. I hope to emulate the same method to look for COVID-19 cures. The diagram below illustrates what the authors did in their research.



In the diagram above, we can see that the authors found two levels of words similar to "thermoelectric" in a heirarchical manner. The second order similar words contained compounds like Li2CuSb, Cu7Te5, and CsAgGa2Se4, which turned out to be very good thermoelectric materials in real life.

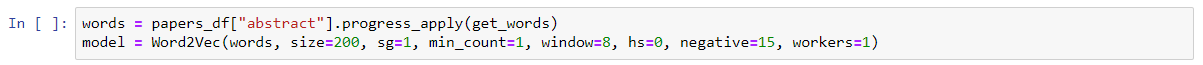
#### Now, coming back to our problem, first, we need to find the most common words in the corpus to continue our analysis. To find the most common words, we can create a word cloud

### **Word cloud of abstracts**

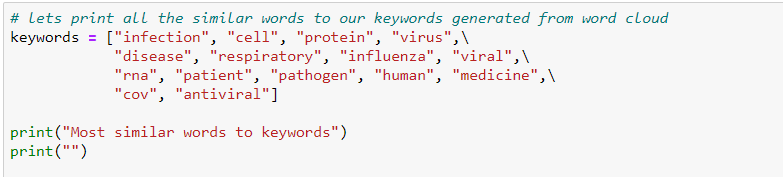
#### 

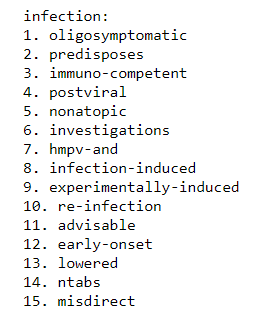
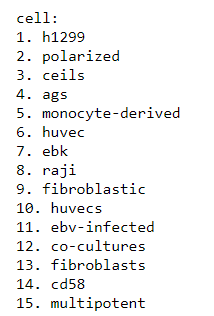
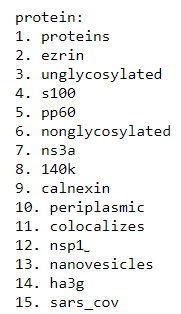
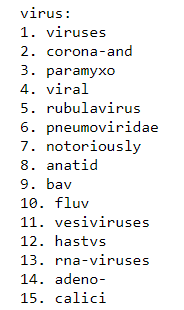
#### Observation: From the above word cloud, we can see that "infection", "cell", "virus", and "protein" are among the most common words in COVID-19 research paper abstracts. These words will form our "keyword" list.

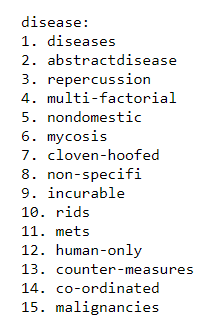
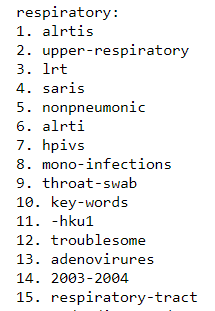
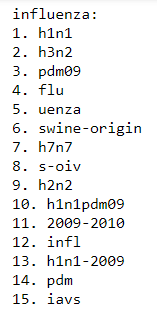
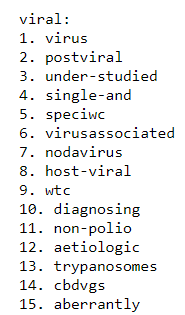
#### Train Word2Vec Model

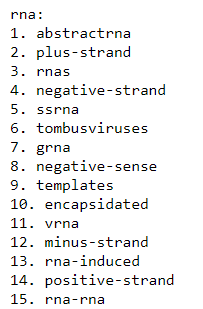
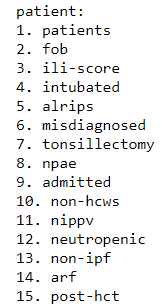
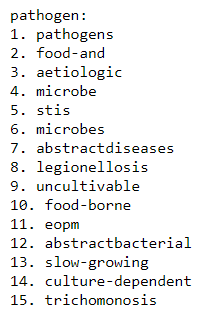
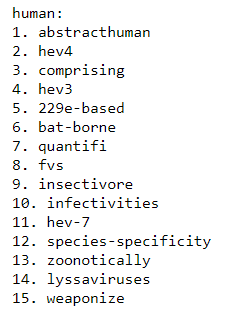


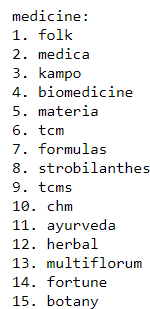
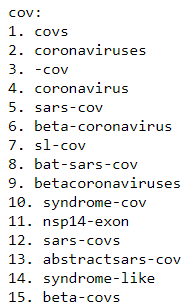
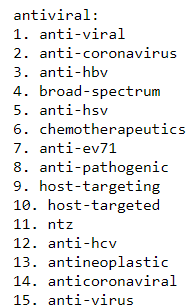
## **Visualize most similar words to the keywords**



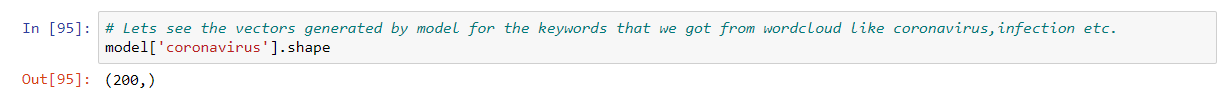
   

We have printed the most similar words to the 15 keywords (based on Euclidean distance). These words will form the next batch of words, which we will analyze to find cures to COVID19.

Now, we had 200D vectors which is not possible to visualize, to overcome this challenge, we used a dimensionality reduction technique called **PCA**



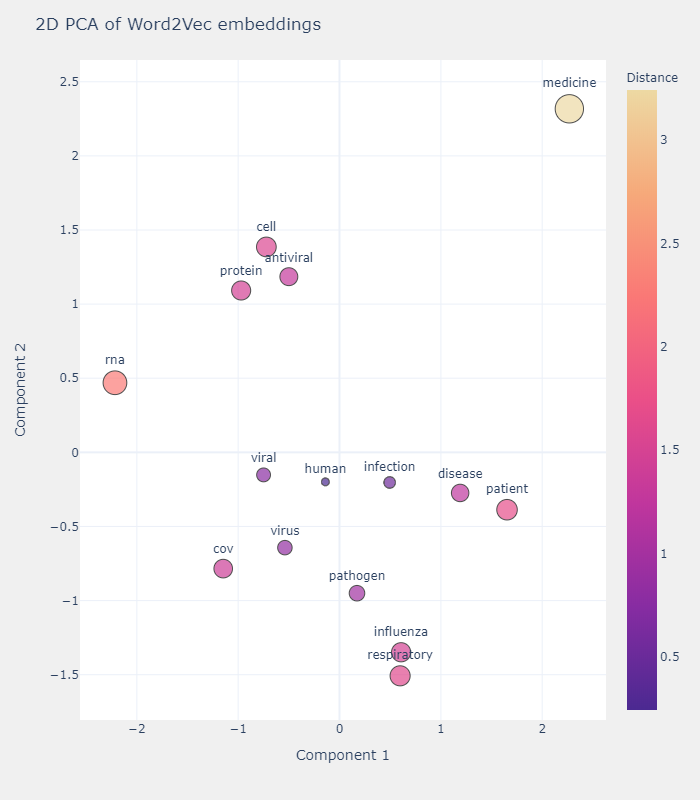
#### Observation: We can see, the above screen shot displays for the keyword 'coronavirus', we had 200D vectors, similarly we have 200D for all the keywords

#### 11.2 PCA

PCA is a dimensionality reduction method which takes vectors with several dimensions and compresses it into a smaller vector (with 2 or 3 dimensions) while preserving most of the information in the original vector (using some linear algebra). PCA makes visualization easier while dealing with high-dimensional data, such as Word2Vec vectors.



### **2D PCA of keyword vectors**



**Observation:** In the above plot, we can see the 2D PCA of the keywords' vectors.

1)The words "virus", "viral", and "CoV" form a cluster in the bottom-right part of the plot, indicating that they have similar meanings. This makes sense because CoV is a virus.

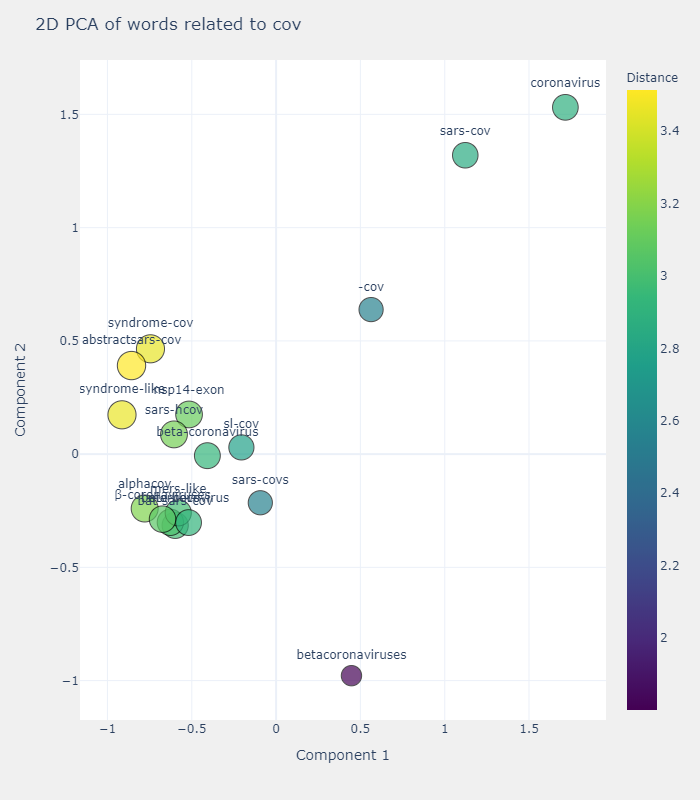
2)The words "medicine" and "patient" are both on the far-left end of the image because these words are used together very frequently.

3)The "pathogen", "influenza", and "respiratory" form a cluster in the bottom-left part of the plot, indicating that they have similar meanings. This makes sense because influenza is a respiratory disease.

These abstract linguistic relationships are successfully represented by word vectors.

### **For testing purpose:** Lets visualize the similar words to couple of my keywords. Visualizing below 2D PCA of words similar to Cov and disease

### **Keyword: Cov**

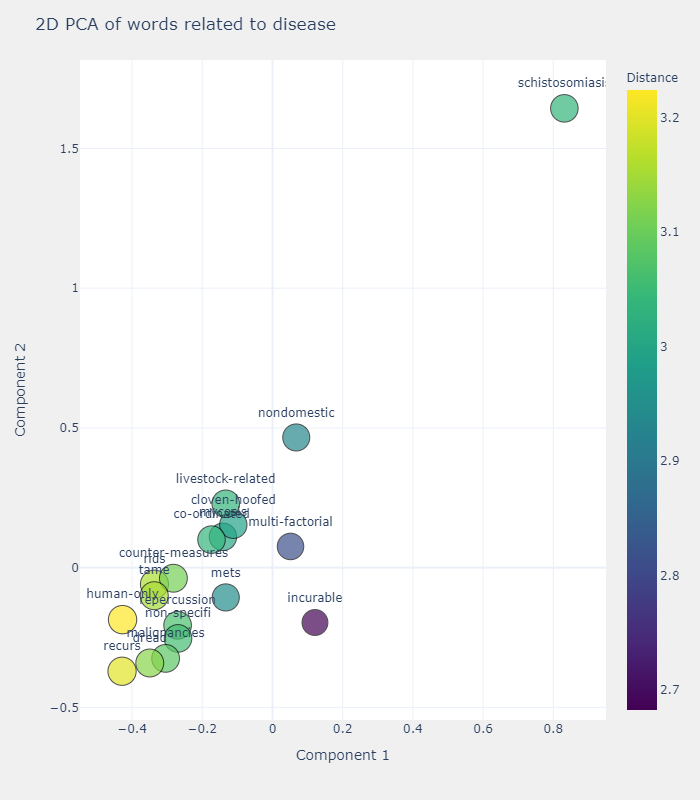


**Observation:** We have plotted the 2D PCA of the words most similar to CoV (stands for CoronaVirus) above.

We can see few words like "coronavirus", "SARS-CoV", and "coronaviral" which are almost synonymal with CoV. These words are surprisingly very close to "CoV" in the vector space.

We can also see a clear cluster in the bottom-left corner of the plot, and these words are also closely linked with the word "CoV".

### **Keyword: disease**



**Observation:** We can see few words like "echinococcus", "mets", and "echinococcus", these words are closely linked with the our keyword "disease"

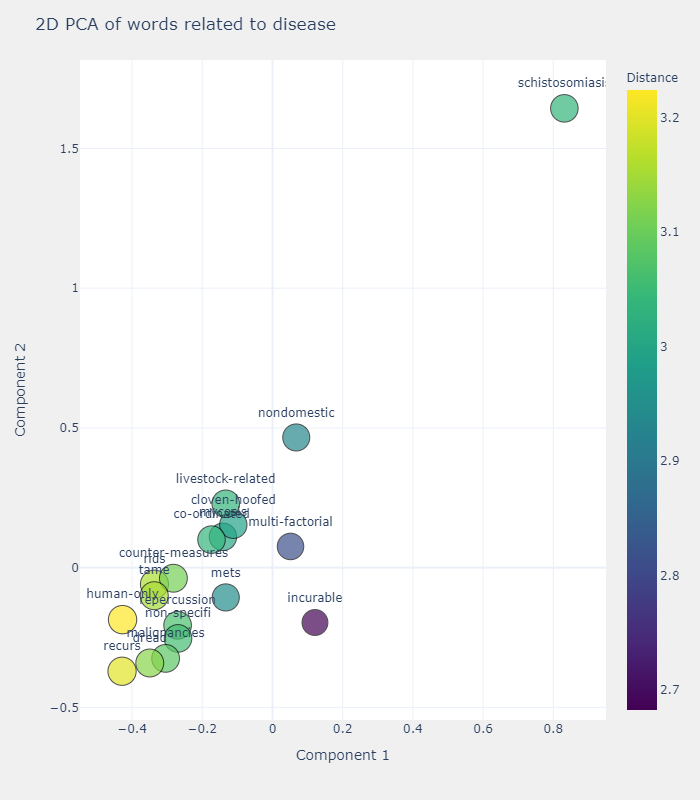
echinococcus is a granulosus disease, it is a tapeworm infection that affects the liver, lungs, brain, and other organs.

mets stands for Metastatic is a bone disease that results in spinal tumor

Ailment disease is a particular abnormal condition that negatively affects the structure or function of all or part of any living organism.

### **Lets find the cure now**

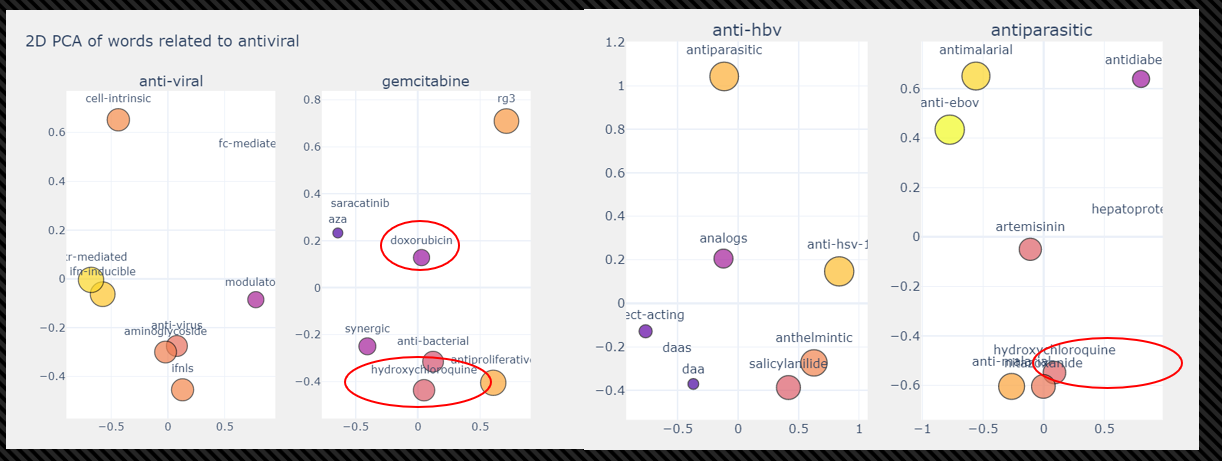
Second-order word similarities Now, we will look at the words similar to the words found above (second order similarity) to hopefully, find potential cures for COVID-19.

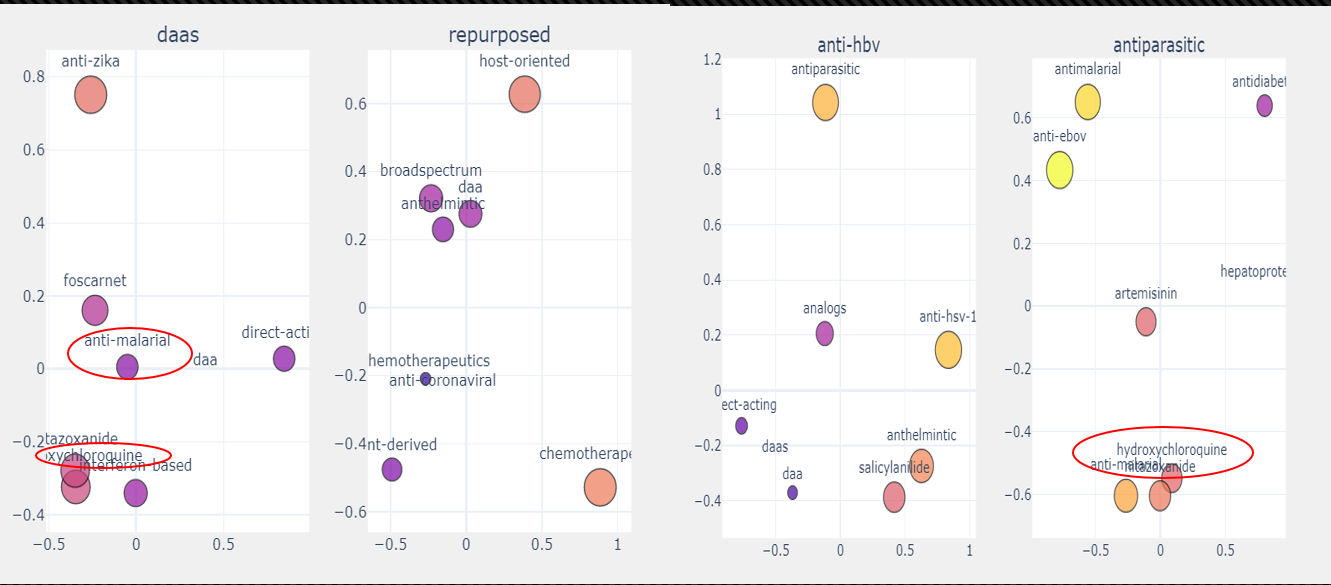


**Observation:** We have plotted the 2D PCA of the words most similar to antiviral above. We can see a lot of different types of antivirals and other drugs in the plot, such as "saracatinib", an anti-malarial and anti-HIV drug. The list also includes "antiparasitic", "ant-HBV", and "anti-EV71".

### **Second-order word similarities**

### Now, we will look at the words similar to the words found above (second order similarity) to hopefully, find potential cures for COVID-19.





#### Observation: We can see some amazing patterns in the plots above. We see certain drugs and chemicals that keep repeating, including "anti-malarial", "hydroxychloroquine", and "doxorubicin". It is amazing that these drugs have actually been successfully applied on COVID-19 patients across the world. There are cases of anti-malarial drugs working for COVID-19!

### **11.3 Takeaways**

### Several antimalarial drugs such as hydroxychloroquine might be potential drugs to cure COVID-19. Antimalarial drugs have been successfully tested on COVID-19 patients in certain countries.

### The best ways to control the virus is **mass testing, partial or complete lockdown, and use of technology** (good examples are China and South Korea).

#### 12. Anserini+BERT-SQuAD for Semantic Corpus Search[10]

The objective of this project is to integrate the state-of-the-art Natural Language Processing (NLP) techniques to return sensible results that answers any challenging question. In addition, applied modern document analysis methods to extract tabular, figure, and numeric data when possible. Eventually, this could be built into a scientific search system that can reduce a body of knowledge to a small number of records and then extract data into machine readable formats to support further modeling and data analysis.

A screenshot of a cell phone

Description automatically generated

**12.1 BERT-SERINI Architecture**

In this work, we combined Anserini and Bert-SQuAD to produce a single "best" answer. [Anserini](https://github.com/castorini/anserini) is a search engine which is built on top of Apache Lucene which can go and search literature and return ranked records. On the other hand, Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowd workers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable. We used an NL question to drive Anserini and return a ranked number of segments from all of wikipidia. Following this, we ran the same question over all the segments using BERT-SQuAD to produce set of spans. The spans were then ranked by a linear combination of the span and segment score and the top was then taken as the answer. We believe this is a great way to go because everything is analyzed by semantics of the question being posed and the number of records that need to be searched by BERT-SQuAD is minimal which is good because that is quite intense.

**12.2 Question Answer session with BERT**

Now let’s make first query of the CORD-19 database. Also, let’s look at the top 10 results based on our query

Let’s break the query up into two parts:

1) a natural language questions

2) a set of keywords that can help drive the Anserini search engine towards the most interesting results for the question we want to ask

**Examples with sample output:**

**Query 1:** How long is the incubation period for the virus

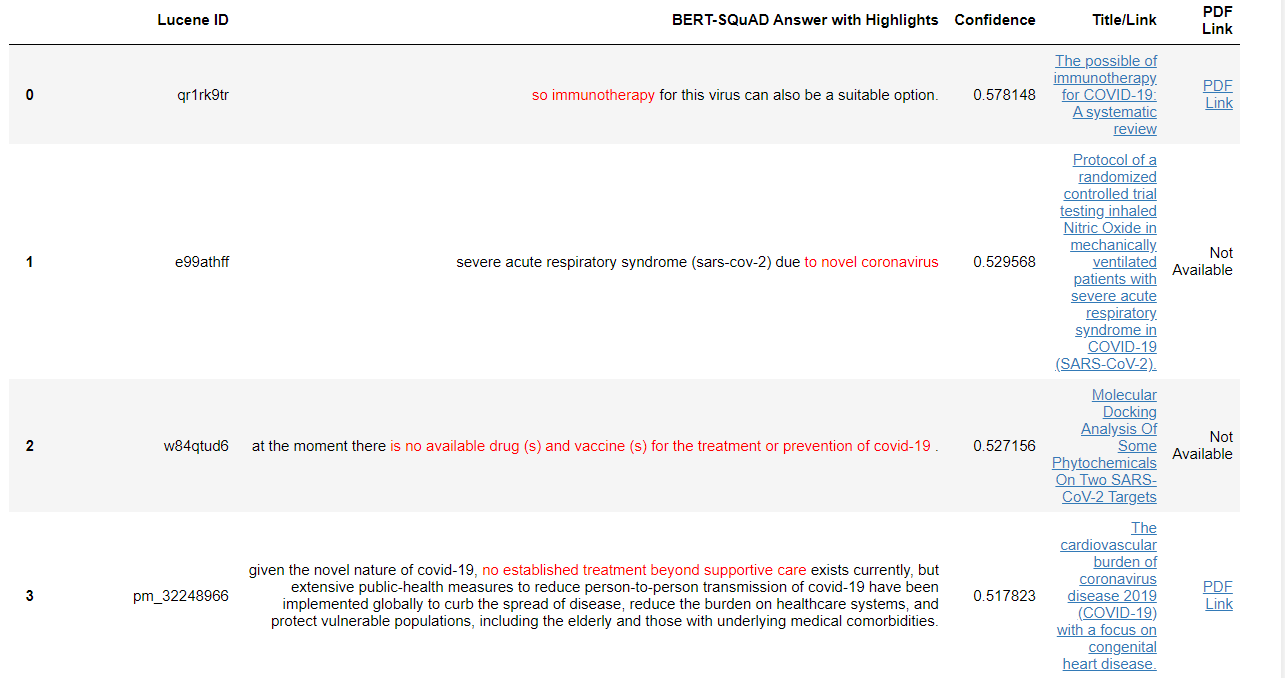
**BERT Abstractive Summary:** The incubation period for covid-19 is between 2-10 days, according to the world health organization (who). source of infection including the patients, asymptomatic carrier and patients in the incubation period are contagious. to avoid the risk of virus spread, all potentially exposed subjects are required to be isolated for 14 days, which is the longest predicted incubation time.

A screenshot of a cell phone

Description automatically generated

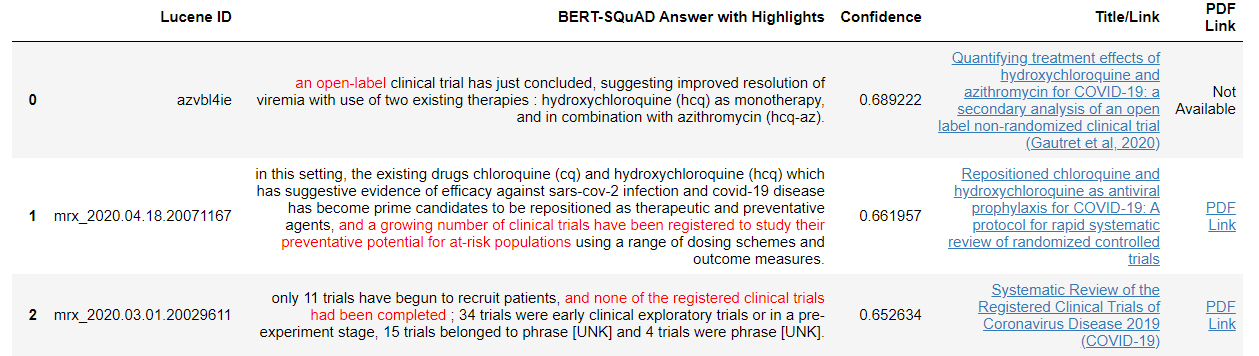
**Query 2:** What is possible treatment for 2019-nCoV

**BERT Abstractive Summary:** No established treatment beyond supportive care exists currently, but extensive public-health measures to reduce person-to-person transmission of covid-19 have been implemented globally. immunotherapy for this virus can also be a suitable option



**Query 3:** What clinical trials for hydroxychloroquine have been completed

**BERT Abstractive Summary:** An open-label clinical trial has just concluded, suggesting improved resolution of viremia with use of two existing therapies. hydroxychloroquine (hcq) as monotherapy, and in combination with azithromycin (hcq-az) only 11 trials have begun to recruit patients, and none of the registered clinical trials had been completed. 34 trials were early clinical exploratory trials or in a pre-experiment stage.



**Observation:**

As per the previous results from Word2vec algorithm, hydroxychloroquine was the potential cure for COVID-19. And in order to prove this further, we wanted to understand the clinical treatments carried out with hydroxychloroquine using BERT-SERINI. The result as per the last question indicated that countries are running clinical treatments with hydroxychloroquine that could lead to it being a potential vaccine for COVID-19.

**13. REFERENCES**

[1] <https://www.kaggle.com/dirktheeng/anserini-bert-squad-for-semantic-corpus-search>

[2] <https://www.kaggle.com/dirktheeng/anserini-bert-squad-for-semantic-corpus-search>

[3] <https://www.kaggle.com/dirktheeng/anserini-bert-squad-for-semantic-corpus-search>

[4] <https://www.kaggle.com/dirktheeng/anserini-bert-squad-for-semantic-corpus-search>

[5] <https://pages.semanticscholar.org/coronavirus-research>

[6] <http://blog.appliedinformaticsinc.com/how-to-parse-and-convert-json-to-csv-using-python/>

[7] <https://willcecil.co.uk/upload-a-big-csv-file-into-big-query/>

[8] <https://panoply.io/data-warehouse-guide/bigquery-architecture/>

[9]<https://www.yalemedicine.org/stories/2019-novel-coronavirus/>

[10] <https://deeplearn.org/arxiv/62668/end-to-end-open-domain-question-answering-with-bertserini>