COVID-19 Scientific Papers Analysis

DAT5550 Data Mining - Final Project Report

Asahi Cantu Moreno
University of Stavanger, Norway
a.cantumoreno@stud.uis.no

Aman Riaz
University of Stavanger, Norway
a.riaz@stud.uis.no

ABSTRACT

The abrupt appearance of COVID-19 in the world and its respective identification in December 2019 is a global phenomenon that highlighted the shortcomings and instability of society in various countries, and whose consequences will change the course of humanity in all its fields. Little could society know months before how devastating and fast this disease would be. By the time this report is written most of the countries have reached the so called "peak of the curve" and a further understanding of the disease, scientific collaboration and vaccine preparation is being observed.

On the technical side, this phenomenon is a peculiar milestone. It is astonishing how fast the scientific community collaborated and joined forces with doctors, virologists and researches from many different branches over the world to fight the virus. In particular, the amount of information generated regarding studies, discoveries, news and political social economical situations is massive, and still ongoing. Perhaps like no other time ever before, the amount of information generated flooded social media, web portals, research sites to the point it is hard to follow up and get accurate, distilled information that can lead to the progress on the understanding of this disease.

This report shows the application of Natural Language Processing tools with unsupervised learning Clustering techniques that help get a general overview over a big data set containing over 63,000 scholarly articles about COVID-19, SARS-CoV-2, and related coronaviruses provided by Kaggle[2], the employed techniques, relevant questions, statistics and results are presented with the possibility to use this research for future developments.

KEYWORDS

COVID-19, Data Mining, Clustering, Unsupervised Learning, Machine Learning, NLP, Scientific Paper, paper research,pandemic

1 INTRODUCTION

By the time this report is written (May, 2020) COVID-19 is not only a global disease, but a world term widely acquired. There is no place on earth that has not suffered the consequences of such pandemic: Lives have been taken, employments by millions have been lost, bankrupted industries, global confinement, scientists struggling to contain the infection, research for a vaccination, trying to better understand the virus. The information gathered and generated just in 5 months related to all these challenges have overwhelmed the scientific community, it is one of the biggest world problems nowadays and the main motivation for the development of this project. Different companies and organizations have managed to extract massive information and make it available to the general public to help them answer relevant questions and find insights that help fight and potentially find a cure to the virus. Kaggle made available in its web portal an important dataset that contains over 63,000 scholarly articles, including 51,000 with full text, about COVID-19, SARS-CoV-2, and related coronaviruses. The goal of this contribution is to apply recent advances in natural language processing and Artificial Intelligence algorithms to get new insights that help the medical research community to keep up. Due to the rapid acceleration of this pandemic, any research is vital to fight against this disease.

This report will show the workflow performed from data exploration, the questions raised and the development of the algorithms to try to answer them, to later on show the final results, explain the challenges during its development, and show relevant information that may help the lecturer use this report as the basis for future research over this or similar datasets. All algorithms and procedures developed in this project were performed using python scripting and relevant packages (Pandas, Numpy, Jupyter Notebook, Scikit-Learn, Spaci, Matplotlib, Bokeh). The full data repository is available in GitHub[6] ready to download and use in the URL https://github.com/asahicantu/DAT550.

1.1 Participation

Contributions and participation over the development of the project and report are given as follows:

- EDA and preprocessing for academic articles: Asahi Cantu
- EDA and preprocessing for authors: Aman Riaz
- NLP tools and techniques for text processing: Asahi Cantu
- ML model algorithm and training for Clustering: Asahi Cantu
- Report redaction: Aman Riaz, Asahi Cantu

1.2 Questions

(1) How many contributions have been done to the academic articles?
(2) Where do these academic papers come from?
(3) Which languages have the articles been written?
(4) What are the main topics of the articles?

2 EXPLORATORY DATA ANALYSIS

The dataset used for this report is publicly available in the Kaggle[2] website, and is continuously updated with new information and academic articles.

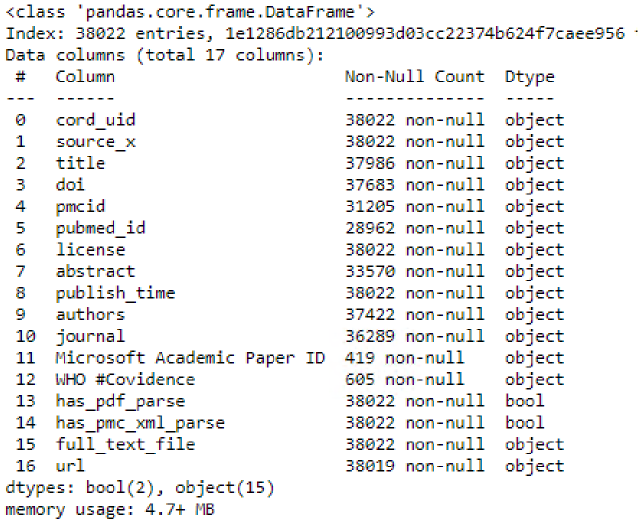
Supervised by Vinay Jayarama Setty, Magnus Særsten Book and Nguyen Khoa Le.

Project in Computer Science (DAT500), IDE, UiS
2020.

2.1 Dataset Specifications

Three metadata files containing the description and relevant information of the academic articles were used initially to explore and investigate on the given dataset.

(1) json_schema.txt - A text file containing the data structure description of each academical parsed file in JSON[1]. It was used to create a JSON parser to extract the information for each academic article.
(2) metadata.csv - A "Comma Separated Values" file containing metadata of each academic article such as authors, date of publication, abstract, paper title, etc. Was used as the main file to start indexing and processing the information for each academic article.
(3) metadata.readme - A text file containing relevant information about the changes to the dataset and the usage of metadata files.
(4) academic articles - Over 9GB of compressed data containing scientific papers from different websites, parsed and stored in JSON format.



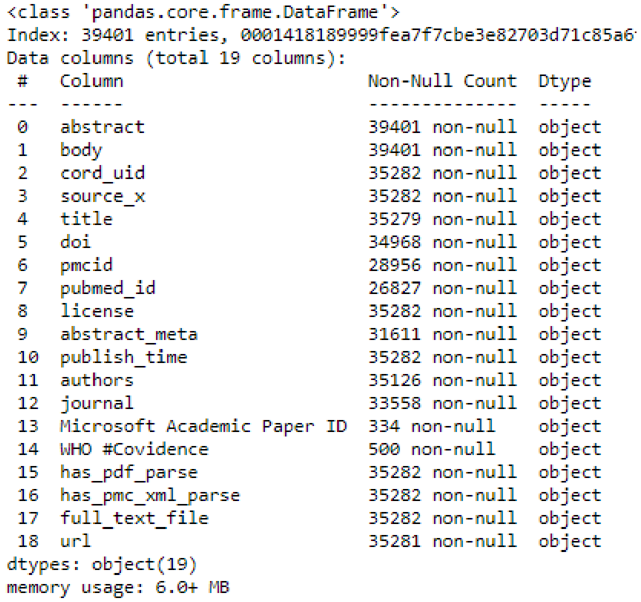
```
<class 'pandas.core.frame.DataFrame'>
Index: 38022 entries, 1e1286db212100993d03cc22374b624f7caee956
Data columns (total 17 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   cord_uid                    38022 non-null  object
 1   source_x                    38022 non-null  object
 2   title                       37986 non-null  object
 3   doi                         37683 non-null  object
 4   pmcid                       31205 non-null  object
 5   pubmed_id                   28962 non-null  object
 6   license                     38022 non-null  object
 7   abstract                    33570 non-null  object
 8   publish_time                38022 non-null  object
 9   authors                     37422 non-null  object
 10  journal                     36289 non-null  object
 11  Microsoft Academic Paper ID  419 non-null    object
 12  WHO #Covidence              605 non-null    object
 13  has_pdf_parse               38022 non-null  bool
 14  has_pmc_xml_parse           38022 non-null  bool
 15  full_text_file              38022 non-null  object
 16  url                         38019 non-null  object
dtypes: bool(2), object(15)
memory usage: 4.7+ MB
```

Figure 1: Statistics of the metadata.csv file, 38022 elements loaded

Initially each metadata file was read and interpreted to understand the relevant information and which features would be important to keep for future use.

Several JSON files were also chosen randomly to have a better understanding of its structure. From all these metadata files it was possible to observe that this set in some cases contained two articles or more on the same record with similar title. It was necessary to 'explode' such rows and atomize the dataset.

A set of functions and algorithms were built to parse each JSON article and reference it to its corresponding metadata row. The full set of articles was created and merged with its corresponding metadata file.



```
<class 'pandas.core.frame.DataFrame'>
Index: 39401 entries, 0001418189999fea7f7cbe3e82703d71c85a6
Data columns (total 19 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   abstract                    39401 non-null  object
 1   body                        39401 non-null  object
 2   cord_uid                    35282 non-null  object
 3   source_x                    35282 non-null  object
 4   title                       35279 non-null  object
 5   doi                         34968 non-null  object
 6   pmcid                       28956 non-null  object
 7   pubmed_id                   26827 non-null  object
 8   license                     35282 non-null  object
 9   abstract_meta               31611 non-null  object
 10  publish_time                35282 non-null  object
 11  authors                     35126 non-null  object
 12  journal                     33558 non-null  object
 13  Microsoft Academic Paper ID  334 non-null    object
 14  WHO #Covidence              500 non-null    object
 15  has_pdf_parse               35282 non-null  object
 16  has_pmc_xml_parse           35282 non-null  object
 17  full_text_file              35282 non-null  object
 18  url                         35281 non-null  object
dtypes: object(19)
memory usage: 6.0+ MB
```

Figure 2: Statistics of articles DataFrame, 39401 elements created

With this information it was possible then to build the relevant questions that will follow the development of the project.

With these questions and the available tools to explore the data the data preprocessing was performed.

3 DATA PRE-PROCESSING

The data was processed and cleaned according to the different observations and trials resulted from the developed workflows.

3.0.1 Articles text cleaning. It was observed that unknown characters or bad encoding were a problem, therefore the articles have to be filtered for unknown or wrongly formatted text.

3.1 Dropping duplicated articles

Some articles were submitted to different academic web sites and were extracted individually, generating therefore duplicates

3.2 Dropping empty articles (some articles were not given in the dataset)

Some articles contained in the metadata section were not actually available on the data section, they were discarded

3.3 Dropping unreadable articles (some articles were wrongly parsed)

Later on after a deeper analysis it was found that few articles contained unreadable language or encoding, they were also dropped.

3.4 Finding relevant information about the countries

Several typos and misspellings were fount in the country section of the papers, a time consuming and deep text cleaning was performed

[1] (JavaScript Object Notation). A lightweight data-interchange format easy for humans to understand and for machines to parse and generate.[3]

to properly identify the origin of the articles and authors. Several elements also came with empty countries, which for other analysis were discarded as well.

After the data cleaning process 38915 out of 39401 articles remained.

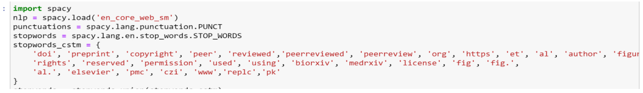
4 ALGORITHMS AND WORKFLOWS

Since the main target was to look for relevant topics and information that enabled scientists to easily identify academic papers and authors, it is considered that the dataset came unlabeled and required the development of algorithms to properly find or define the topics over the dataset.

4.1 Natural Language Processing

Text classification for machine learning algorithms required the dataset to be properly processed and parsed to make easy the training processes. Natural Language processing techniques were applied over the dataset using python package spaCy[5][2]. The NLP Tasks employed were:

4.1.1 Cleanup. Irrelevant text features, such as numeric data, stop words and word separators were identified and cleaned form the whole dataset. Common English vocabulary stop words were identified and removed, and also irrelevant or common words added to the papers such as "Figure", "Copyright" (See Figure 3). Each academic paper was split into tokens of size 1, a list of words.



```
: import spacy
  nlp = spacy.load('en_core_web_sm')
  punctuations = spacy.lang.punctuation.PUNCT
  stopwords = spacy.lang.en.stop_words.STOP_WORDS
  stopwords_cstm = {
      'doi', 'preprint', 'copyright', 'peer', 'reviewed','peerreviewed', 'peerreview', 'org', 'https', 'et', 'al', 'author', 'figur
      'rights', 'reserved', 'permission', 'used', 'using', 'biorxiv', 'medrxiv', 'license', 'fig', 'fig.',
      'al.', 'elsevier', 'pmc', 'czi', 'www','replc','pk'
  }
```

Figure 3: Code snipped showing the employed stop words after deep article examination

4.1.2 Tokenization.

4.1.3 Word stemming and lemmatization. Each word token was reduced to its root form for the whole article dataset

4.1.4 Language identification. Each word token was reduced to its root form for the whole article dataset

4.2 Unsupervised Learning classification

It was necessary to identify the best approach to perform an unsupervised learning classification algorithm, for this task the python package scikit-learn[4][3] was used.

Before applying any classification algorithm it was necessary to reduce the amount of features and find the proper way to classify all the tokens given previously by the NLP algorithms. Fortunately scikit-learn provides such tools to make the job easier.

4.2.1 Tf-Idf. TF-Idf[4] algorithm was used to find the measures of the impact of the words in the whole document set by looking the proportion of times the word appears in the document and the number of documents available. This way it was possible to give a suitable measure over the tokens and prepare the data numerically to be understood by the classification algorithm.

4.2.2 PCA. PCA[5] was applied to the given TF-Idf results. The intention was to preserve 95% of the relevant information with 23 clusters.

4.2.3 K-Means Clustering. K-Means[6] was then used and run 49 times to find the suitable number of clusters that at which the dataset could be properly partitioned. This is part of the process known as "Elbow Method technique" which consists in running multiple times the classification algorithm (in this case K-Means) each time with an increasing number of clusters. The premise is that the algorithm will find over the time the most suitable number of clusters that will maximize the classification and labeling over the elements. Once applied over the data set it was found that the suitable number of clusters to classify the dataset was **23**. The process was then run one more time for 23 clusters and each document was then properly labelled to its corresponding cluster.

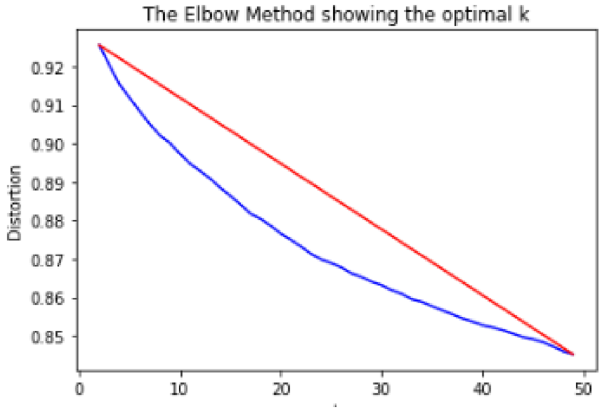


Figure 4: The elbow method show an optimum classification between 20-25 clusters.

4.2.4 T-SNE. T-SNE[7] Algorithm was finally employed over the trained dataset to plot and visualize the data.

[2] spaCy is a free open-source library for Natural Language Processing in Python.

[3] Scikit-learn is a free software machine learning library for Python. It features various classification, regression and clustering algorithms

[4] Tf-Idf stands for term frequency-inverse document frequency. Is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.

[5] Principal component analysis (PCA). Technique used to emphasize variation and bring out strong patterns in a dataset. Useful to train machine learning algorithms with reduced amount of variables.

[6] Unsupervised machine learning algorithm that identifies k number of centroids, and allocates every data point to the nearest cluster, while keeping the centroids as small as possible.

[7] t-distributed stochastic neighbor embedding is a technique for dimensionality reduction particularly well suited for the visualization of high-dimensional datasets.

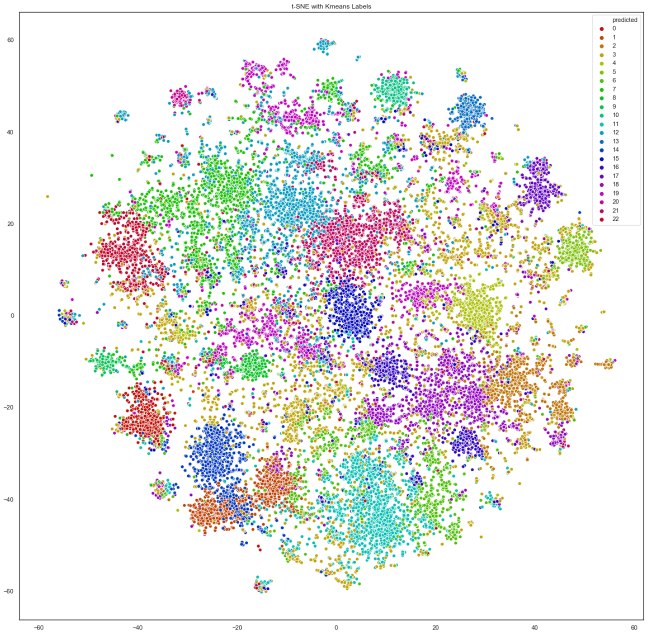


Figure 5: Cluster visualization via T-SNE Technique.

4.3 Plot

Finally the algorithm was properly classified and the dataset is well structured for future works. Data is ploted and deployed over an interactive chart using python Bokeh[1] package[8]

5 CHALLENGES AND LIMITATIONS

Several challenges and limitations were encountered during the project, some of them could be partially solved whereas others simply required more computational resources.

5.1 Environment resources

Writing the scripts to process the data and having such a big dataset like this in a single batch was not possible due to the considerable amount of RAM Memory it takes. It was necessary to load the dataset by batches and process it to later store it in binary files that could make the job easier. A system with 32GB of RAM or V-RAM can better handle this barrier.

The increasing amount of time it takes for the Machine Learning algorithms to run the classification was considerable. In some cases and scenarios the whole set can take up to 1 day for the final classification. If during that time there was some other program in use the system would show "out of memory" and the process had to start again. Fortunately many of these complications were solved by saving promptly the processed dataset into a binary file, and the trained model as well.

A code for memory allocation was implemented as well, although sometimes it does not work as expected since the python garbage collector takes some time to release unused variables.

5.2 Text inconsistencies and wrong data formats

Surprisingly many research documents contain errors in its metadata, besides being a standard for properly writing the documents several attempts had to be made to clean it and format in a more understandable way. This challenge was particularly visible in the processing of countries and author names.

5.3 Language barriers

More than 1000 documents have to be discarded for having either language other than English or not being properly formatted. It could be that relevant information is available in the documents and generates an individual cluster.

5.4 Dataset Continuous updates

Initially the whole project was intended to be done in Kaggle, but it was not possible to overcome RAM Memory size problems and the project had to be taken to a personal computer. It is important to mention that the dataset is 'alive', every week more information is inserted, it is foreseeable that such information will evolve in its structure and eventually the user will need to re-write the source code to adapt to the new formats.

5.5 Time and experience

The limited amount of time and experience in the development of this kind of projects is a key limitation. It is important to highlight that there will be many different approaches for this problem and maybe different classifiers could outperform the proposed one.

It was also expected to go deeper with NLP techniques to provide a text summarizing and give more information insights, but this requires as well more knowledge handling big text datasets.

6 RESULTS

These results are the answers to the questions proposed in section 2. Exploratory Data Analysis.

6.0.1 How many contributions have been done to the academic articles? 197643 different contributions were done to the whole academic data set. It implies that generally the research articles are written by more than 2 scientists and in several cases these scientist contributed to more than one article.

6.0.2 Where do these academic papers come from? Scientists from 132 different nationalities have contributed to the research. Most of them come from United States and China. More information provided in Figure 6.

6.0.3 Which languages have the articles been written?

[8] Bokeh is an interactive visualization library for modern web browsers. Provides elegant, concise construction graphics, and affords high-performance interactivity over large or streaming datasets

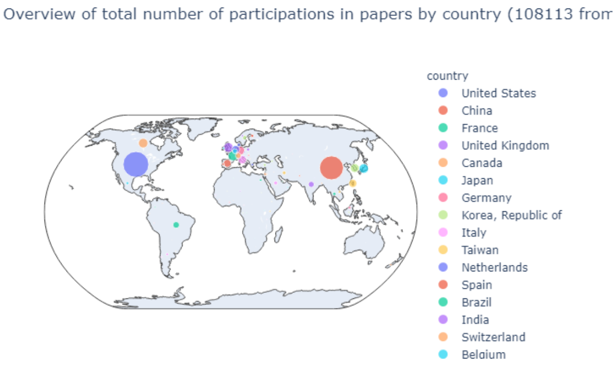


Figure 6: Academic participation by country. 132 different nationalities

Country List	
Language	Written articles
English	37957
French	311
Spanish	297
German	100
Dutch	52
Italian	15
Greek	11
Portuguese	9
Polish	2
Norwegian	1
Dari	1
Danish	1

6.0.4 What are the main topics of the articles? Figures 7 and 8 show the information and key insights derived from the applied algorithms.

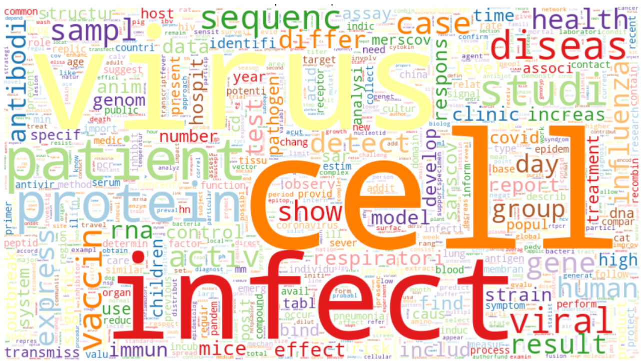


Figure 7: Topics and words for the whole document corpus

6.0.5 Other results. Important key points were taken in consideration:

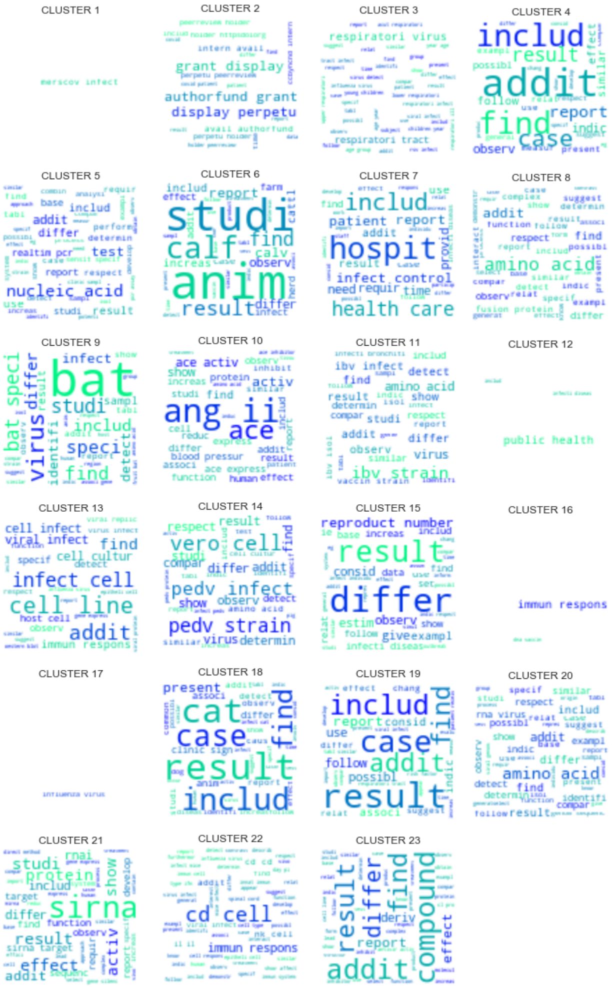


Figure 8: Topics and words per cluster

- 37957 Documents were processed (Only English-written articles)
- 23 Clusters were the optimum classification parameter
- 23.3 Hours took the Machine learning algorithm to be trained and applied.
- 1.2GB is the size of the final processed Data Frame (compressed format)
- 16GB in RAM is very restrictive to work with this dataset. 32MB is advised
-

7 FUTURE WORK

The key purpose on the development of this project was to successfully accomplish the classification of academic documents into different clusters and create a tool that helped the community to visualize it. Several improvements and approaches can be performed to the current development:

7.1 Classification algorithm improvement

It is probable that different Machine Learning classification algorithms can be applied to this problem to have a better classification on the clusters. from Figure 5 it is possible to visualize some articles located away from their corresponding clusters. Perhaps a higher dimensional visualization or different classification technique can improve the mechanism.

7.2 File and memory allocation

Dealing with such a big dataset bring problems when allocation computational resources to address the problem and develop the code. Improving the memory allocation and facilitating the data processing by batches can allow computers with less resources to run the algorithms. This is particularly important to submit the code to cloud systems such as Kaggle or Goggle Collab, where the amount of RAM Memory is very limited.

7.3 Improvements on NLP Techniques

Having a trained model for the whole scientific corpus and try to label it based on the current clustering development could lead to the development of knowledge based systems where specific questions can be asked to the system, which could then look at its information and provide accurate answers based on the feeding documents.

7.4 ML algorithms updates

Since the dataset is refreshed every month, it will be important to develop a script that synchronizes the data and retrains the model for better and more accurate predictions. It is estimated that more papers and discoveries will be done over the next months, and therefore it is important to keep the data updated.

7.5 Scientific community network dashboards

With the current trained models it would be possible to create a network dashboard that links all the scientists with its corresponding article and at the same time each article with its corresponding references, this would help the community to better visualize the information and find the relationships between different academic articles. This could mean that a document from a specific cluster is the basis or reference for the research of a different article which in the same way belongs to a different cluster

7.6 Cluster dashboard interactions

More improvements can be made to the cluster dashboard that would allow a better interaction for the users and understanding of the documents. Some of the future works and tasks that can be applied over this problems are:

- Test and try other classification techniques
- Perform Text summarizing and better NLP techniques
- Create a knowledge system and trained model capable of answering questions based on all the knowledge extracted from the documents
- Create a Dashboard showing the scientific community participation and relevant information to communicate with them
- Translate the articles written in different languages and include them in the model
- Improve the plotting tools to simplify the research workflows
- Improve the provided code to support bigger datasets
- Include this information with historic infection cases, disease symptoms and other datasets that can provide better insights of the virus.

8 CONCLUSION

In the development of the project it was possible to cluster the published literature on COVID-19 and properly apply the algorithms to reduce the number of features of the corpus, which in the end provided key visual insights of the topics. Data preprocessing, authors and articles identification made possible the visualization of documents by countries, language identification and document classification by its topic with a total of 23 clusters. By doing these tasks it was possible to create an interactive scatter plot of articles grouped by similar topics and highlighted in different colors. This creates a general dashboards which provides instant and useful information for all the English documents and facilitates the search of information. The data was clustered using K-Means over vector transformed datasets. The clusters and points represented for each topic show the relationships among all the articles. Those with similar topics are close to each other. This project can be useful for the scientific community and general public interested in researching on this subject or further develop it. It has been observed over the previous months that tools like this are not standardized yet and is of vital importance to contribute with different ideas and solutions to this worldwide problem. It is also evident the job of data scientists and experienced people in the field is vital for managing big amounts of information and collaborating with other sciences.

REFERENCES

[1] [n. d.]. Bokeh 2.0.2 Documentation. https://docs.bokeh.org/en/latest/index.html

[2] [n. d.]. COVID-19 Open Research Dataset Challenge (CORD-19) | Kaggle. https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge

[3] [n. d.]. JSON. https://www.json.org/json-en.html

[4] [n. d.]. scikit-learn: machine learning in Python — scikit-learn 0.23.0 documentation. https://scikit-learn.org/stable/

[5] [n. d.]. spaCy · Industrial-strength Natural Language Processing in Python. https://spacy.io/ Library Catalog: spacy.io.

[6] Asahi Cantu. [n. d.]. asahicantu/DAT550. https://github.com/asahicantu/DAT550 original-date: 2020-04-09T16:38:55Z.