SI 618 Project 1: COVID-19 Related Datasets Exploration

by MRJob and PySpark in Python

hyfrankl@umich.edu

I. Motivation

The COVID-19 pandemic has influenced the world for three years as a long-lasting hot topic attracting the public's attention. Many research papers on COVID-19 have been published to provide new insights into disease prevention and treatment. Its clinical symptoms may vary from one person to the other, so many research topics are focusing on other coronaviruses and respiratory diseases to find similarities. These research papers may reveal keywords in different years combined with the trend of newly confirmed COVID-19 cases in the world.

As a student at UMSI, I feel motivated to investigate how COVID-19 has promoted advances in medical research and how keywords in research papers match each stage of the pandemic. The former question investigates the advances in coronavirus research, while the latter one unveils how research papers lead or lag behind the development of COVID-19. They will cast a light on the interaction between the pandemic and society (especially, medical workers). We can gain lessons and observe patterns from COVID-19.

To narrow down the scope of this project, I explored the following three main questions.

1) How many researchers investigate the pandemic during each quarter of the year? How does this number correlate with COVID-19 confirmed cases?
2) Who are the productive researchers? And how does their contribution (number of published papers) change with COVID-19 confirmed cases?
3) How do some important keywords change along with the newly confirmed cases?

The first two questions handle the advances in medical research and can identify outstanding scholars and hot topics. The last one unveils the latency between pandemic and medical research.

II. Dataset

In this project, I use two datasets related to COVID-19. I downloaded two datasets on 2022/10/15. The total size of these two datasets is over 1 GB and most data are text, so it will be meaningful to implement data manipulation and analysis on cluster.

A. *metadata.csv from CORD-19 [1]*

1) **Source**: the research paper of **CORD-19** (COVID-19 Open Research Dataset) is stored on [2004.10706] CORD-19: The COVID-19 Open Research Dataset (arxiv.org). One can access the dataset by visiting historical releases [all versions]. For this project, I only use metadata.csv updated on 2022-06-02. For more details, one can also visit this github repository.
2) **Format**: metadata.csv (1.6 GB) is a csv file with 19 columns storing the metadata of papers in CORD-19 datasets including authors, title, publish time, and abstract.
3) **Important Variables**: I use the following columns in the analysis and prefer the abstract to the title because the abstract may contain more detailed information.
   a) authors: string
   b) publish_time: string
   c) journal: string
   d) abstract: string
4) **Records and Time Periods**: To focus on coronavirus analysis, I only include papers published between 2002 and 2022. There are around 9 million records between 2002 and 2022.

B. *COVID-19 Data Repository by Johns Hopkins University [2]*

1) **Source**: Retrieved from CSSEGISandData/COVID-19.
2) **Format, Records, and Periods**: time_series_covid19_confirmed_global.csv (1.5 MB) is a csv file with 290 rows (including header) whose columns are dates ranging from 2020-01-22 to 2022-10-15. Each row records the time series of accumulated confirmed cases in this country/region.
3) **Important Variables**: since the csv file stores the time series, the important variable is the number of confirmed cases in each entry.

III. Data Manipulation Methods

Summary: This section will show the work flow in the project. Since I use one large dataset (1.6 GB) with a small dataset (1.5 MB), it is convenient to extract information from the large dataset into a small file and then join the two datasets in the time series.

Code: the related code is in authorContrib.py with comments and si618prj.ipynb with corresponding section to each part mentioned below.

A. *Data Manipulation with CORD-19 Dataset*

There are some pre-task (**Task 0**) to finish before data join. It can be summarized in two steps and detailed instructions are shown below.

1) Analyze the number of published papers with MRJob
2) Identify frequent unigrams and bigrams in the abstract with PySpark

Pre-process with SparkSQL: as indicated in the description of the dataset, the dataset is large with many missing values. It is important to first preload and prune the dataset which is coded in 'authorContrib.py:preload'. Its main function can be summarized as

1) Include rows where publish year is between 2002 and 2022 because the SARS outbreak happened in 2002.
2) Preload and extract 'authors', 'publish_time' from metadata.csv.
3) Drop duplicates and missing rows that may influence the results.

Map-Reduce with MRJob: I will obtain my first intermediate dataset which describes the number of published papers per person per month. This intermediate product can well delineate how each author contributes to the COVID-19 research. I implement it with MRJob in python. First, use the map function to return each author per paper. Then combine with the columns journal and publish_time to return the (author, publish_time) pair. Finally, use thecombine and reduce function to count the number of published papers per person per month. The code is written in authorContrib.py with comments.

Data Exploration with SparkSQL: I first analyze the basic information of each author's contribution from the results in **Map-Reduce** part. I use SparkSQL to get the distribution of the number of published papers per person between 2002 and 2022. Then I identify the researchers who frequently appear in the top 5 authors with the most published paper (at least 10 in that year) between 2002 and 2022. The analysis identifies important researchers who will be used in further analysis.

Extract Abstract from Paper between 2019 and 2022: since we are analyzing the COVID-19, this project will only focus on the keywords appearing during the pandemic. So, I only include research from 2019 to 2022. Most importantly, I drop the missing rows lack of abstract.

Tokenization and Stop Words Removing: I use the nltk package to tokenize the abstract part of each paper by first sentence tokenization and then word tokenization. Besides, I adopt the English corpus in nltk package to remove stop words and regex expressions to pop up non-sense punctuations in the abstract part, which helps prune text.

Lemmatization and POS Tagging: I import the nltk package and lemmatize each word to its original form. With part-of-speech (POS) tagging, I can identify nouns in the text because nouns present the most contextual information on pandemic development and research directions. This process is accomplished by WordNetLemmatizer and nltk.pos_tag. I have tried to use PorterStemmer but the results are stem words with low readability. So, I chose lemmatization and part-of-speech tagging and return important unigram and bigram nouns here.

Count the Frequency of Unigrams and Bigrams with PySpark: since the data pipeline involves some basic natural language processing methods with a text string, it is fit for PySpark to deal with manipulation in memory without frequent disk I/O in MapReduce. Plus, the dataset is not large enough, so it is unnecessary to use an advanced storage plan like memory and disk option.

First, use flatMapValues to return sentence tokenization and word tokenization from the text. Then, prune the lowercase of the word with lemmatization and POS tagging and return the nouns. Finally, reduce the frequency of each unigram and bigram noun. The frequency of each unigram/bigram can reflect the trend of research topics as well as the pandemic. This result can serve for further analysis on frequent unigrams/bigrams vs. newly confirmed cases.

B. Data Manipulation with COVID-19 Data Repository

Since its size is small, I will manipulate data with pandas first then transfer to **pandas on Spark** to combine data with the previous results.

1) **Data Cleaning**: check whether missing data exists in the dataset and drop invalid rows from data
2) **Data Pre-process**: sum up confirmed cases in all countries/regions to avoid any bias on one specific country/region.
3) **Data Analysis**: gather newly confirmed cases in each quarter in each year.
4) **Task 1**: join and analyze confirmed cases in each quarter with the number of researchers who published papers in time series and examine the correlation.
5) **Task 2**: join and analyze confirmed cases in each quarter with the number of published papers by important researchers in time series.
6) **Task 3**: join and analyze confirmed cases in each quarter with the frequency of keywords(unigram/bigram) in time series and examine the correlation.

With the analysis, we can find each author's contribution and the trend of topics in the pandemic periods. It unveils the full picture of the fight between the pandemic and human beings.

C. Data Visualization

Visualization is in each part implemented by matplotlib. The project includes figures about

1) The distribution of published papers per author.
2) Word clouds for keywords between 2019 and 2021.
3) The number of researchers who published papers in each quarter.
4) Important researchers' contributions in each quarter.
5) Frequency of some keywords in each quarter.

Notice: the latter three figures will compare with the newly confirmed cases in each quarter to make conclusions.

IV. Analysis and Visualization

Before diving into the three main questions, I need to first get two intermediate results from CORD-19 to prepare for data join.

Then, I will analyze the results from each question. In each part, **I mark the exact part in the jupyter notebook**. Notice that the sampled data of metadata.csv may cause different results and lead to the failure of Task 3 as some ngrams may not appear in the sampled data.

Task 0: Data Manipulation

Operations with CORD-19 Dataset: in the map reduce part (in authorContrib.py), I first extract authors, publish_time from the dataset with data cleaning. Then, use flatMap function to obtain (author, publish_year, publish_month, 1) with the count 1 that represents the author published one paper in the given month. Then, group the map results by (author, publish_year, publish_month) that reflects the authors' contribution. Finally, I save the intermediate results to task1.txt file In the PySpark part, I first use map function to extract (abstract, publish_time) from the dataset with proper cleaning. Then, use flatMapValues function to obtain unigrams and bigrams from abstract with stopwords removing, lemmatization, and POS tagging (detailed in the previous section). Next, I count the frequency of each ngram in each month and save the results to task2.txt file.

Operations with COVID-19 Dataset: I sum up all the confirmed cases in all countries and regions and record newly confirmed cases in each quarter by load_data function in authorContrib.py.

Statistics of Published Papers: I intend to analyze how actively researchers involve in the pandemic. Therefore, I use the intermediate results from MapReduce part to get the distribution of the number of published papers per person shown in Fig. 1. Besides, I also analyze the important researchers by ranking the contributions of each researcher in each year and count their appearance in the top five position in each year. It can reflect the importance of the researcher. I obtain some important researchers including Wei Wang and Nicola Decaro.

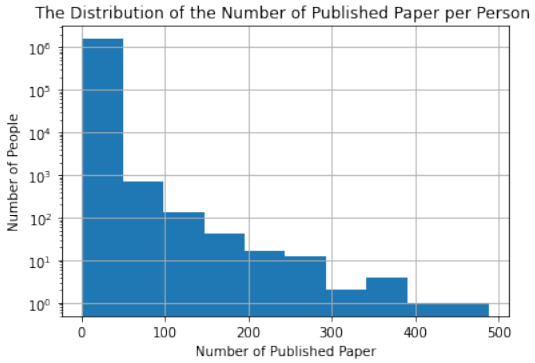

Fig. 1. The Distribution of Number of Published Papers per Person between 2002 and 2022

Statistics of Abstract: I use the concept of ngrams to define the frequent itemset/keywords in the text and obtain meaningful nouns from the text with remove_stopwords function because they contain the important information of research field of focus. Based on the word frequency in each year, I draw the word clouds with the WordCloud package (shown in Fig. 2).

Fig. 2. Word Clouds from Published Papers between 2019 and 2022

Findings:

1) The distribution of the number of published papers per person is exponentially decaying. It shows that few researchers can make great achievements.
2) The analysis identifies some important researchers including Wei Wang and Nicola Decaro.
3) The word cloud shows that the trend of topics changing. Especially, "health", "infection", and "respiratory" change a lot. They may be useful for further analysis.

A. Task 1: Newly Confirmed Cases vs. Number of Published Researchers

Operations: I first use PySpark to obtain summary that how many researchers have published at least one paper in each period. I first use map function with cnt_mapper which yields

(time, set_of_author) pair and then use reduce with cnt_reducer function to gather the (quarter_of_year, set_of_author) pair. Finally, I join this result with the newly confirmed cases in each period. For convenience, I choose the period as the quarter of year and got an interesting figure (shown in Fig. 3). In Fig. 3, the x-axis represents the quarter of a year. For example, 2021Q2 represents the second quarter of the year 2021.

Fig. 3. Newly Confirmed Cases vs. Number of Published Researchers with Pearson coefficient 0.09

Findings: the trend shown in Fig. 3 indicates that before the first quarter of 2022 (2022Q1), the trend of the number of published researchers is similar to the trend of newly confirmed cases. It lags almost two quarters than the newly confirmed cases. It makes sense because the fruits of research take time from the beginning to the end. However, there are fewer papers published after 2022Q1. To some extent, one possible reason can be the dataset only including published papers before 2022-06-02 leading Pearson coefficient between them is 0.09. So, it is fair to ignore the discrepancy here.

B. Task 2: Newly Confirmed Cases vs. Number of Published Papers by Important Researchers

Operations: to identify important researchers, I first find the researchers who frequently rank at top five in each year. I accomplish it by SparkSQL language. I first group by the data by researcher's each year published papers and use windows function "row_number" with "order by" to rank in each grouped data. Then I select the top five by filtering their rank number. After that, I count the times of their appearance in the list and filter those who have appeared three times in the top five. It includes "Buonavoglia, Canio; Decaro, Nicola; Drosten, Christian; Jiang, Shibo; Wang, Wei; Yuen, Kwok-Yung; Zhang, Wei".

After obtaining the candidates of important researchers, I further examine the trend of their contributions (number of published papers). I use the sub-query to choose the lines of which the author is a candidate of important researchers. Following that, the query selects their number of publications in each quarter and join the results with newly confirmed cases.

Visualization: I draw the number of published papers by each candidate in one sub-figure, while the newly confirmed cases in the other figure. By comparing the trend, we may find some similarities.

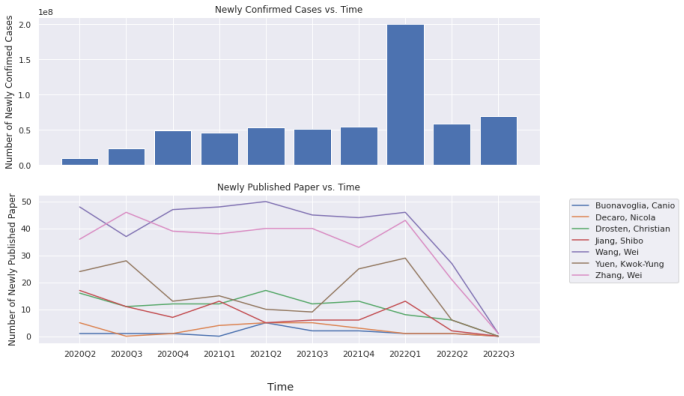

Fig. 4. Newly Confirmed Cases vs. Number of Published Papers by Important Researchers

Findings: First, one can find that Wei Wang and Wei Zhang are active researchers that contributes many published papers per quarter. Second, the trend of published papers by each candidate is similar except Kwok-Yung Yuen. From 2020Q4 to 2022Q1, the number of publications per candidate is stable with small oscillation. But all their contributions drop in 2022Q2 which might be the issue of the dataset. This trend is closely related to the stable newly confirmed cases from 2021Q1 to 2022Q1. If we estimated that the publications recorded in 2022Q2 is two-thirds of the actual publication since the dataset is updated in 2022-06-02. Then, we may observe the discrepancy between published papers and newly confirmed cases. There aren't much research papers in 2022Q2 or later.

C. Task 3: Newly Confirmed Cases vs. Keyword Frequency

Operations: I first obtained the intermediate result mentioned in the **Count the Frequency of Unigrams and Bigrams with PySpark** part in the previous section. Then I use the quarter_freq function to calculate the rate/probability of the keyword to all the words. The operation can well solve the issue that the data collected in 2022Q2 (and later) is incomplete, which is inspired by TF-IDF. Then, I join the data with the newly confirmed cases. Here I choose some important words like health, treatment, vaccine, and respiratory because these words seem more meaningful.

Visualization: use the bar plot to compare the similarities between newly confirmed cases and keywords with draw_freq function.

Findings: in the previous part, frequency is equal to count, while in this part frequency refers the probability of the word occurrence.

1) The word frequency almost lags two quarters behind the confirmed cases similar to the **task 1: published**

TABLE I
CORRELATION BETWEEN KEYWORD AND NEWLY CONFIRMED CASES

Keyword	Pearson Coefficient
health	0.964
infection	0.909
disease	0.899
protein	0.956
rna	0.955
drug	0.867

researchers part. Pearson coefficients are shown in Tbl. I. The frequency of health/rna/protein is most likely to correlate to the newly confirmed cases.

2) It indicates that the word frequency may reflect facts and topic trends in the pandemic.
   a) **Factual Reflection**: For the word "health", "infection", or "disease" in Fig. 5, its frequency is amazingly similar to the newly confirmed cases. It reflects that the trend of newly confirmed cases will arouse researchers' attention to these words. So, we can cluster those words as the factual reflection.
   b) **Research Focus**: "Vaccine", "protein", and "respiratory" in Fig. 5 reflects the focus of medical research. For example, "vaccine" increases all the time until 2021Q4. After 2021Q4, its occurrence is stable. Meanwhile, "respiratory" decreases all the time almost exponentially. It shows that with more confirmed cases, researchers pay less attention to "respiratory" disease, while they resort to the "vaccine" and "protein". It indicates that "respiratory" can't fully describe the essence of COVID-19 and vaccine (with antibody) become one of the mainstream treatment.
   c) **Methods**: "Rna", "antibody", and "drug" in Fig. 5 represent the normal treatment to virus. However, as confirmed cases steadily increase before 2022Q1, the word occurrence is stable. While the newly confirmed cases increase dramatically in 2022Q1, it leads to the rise of rna and drug in 2022Q3 (two quarters later). It reflects the change of treatments to COVID-19 after a dramatic increase in newly confirmed cases.

V. SUMMARY AND CONCLUSIONS

In this project, I want to investigate the relationship between medical research and the development of COVID-19. I propose three main questions

1) How many researchers investigate the pandemic during each quarter of a year? How does this number correlate with COVID-19 confirmed cases?
2) Who are the productive researchers? And how does their contribution (number of published papers) change with COVID-19 confirmed cases?
3) How do some important keywords change along with the newly confirmed cases?

For these questions, I use MRJob, PySpark, and SparkSQL to solve these problems and visualize the results. There are some detailed findings in the previous part. Here I only summarize the responses to the three main questions.

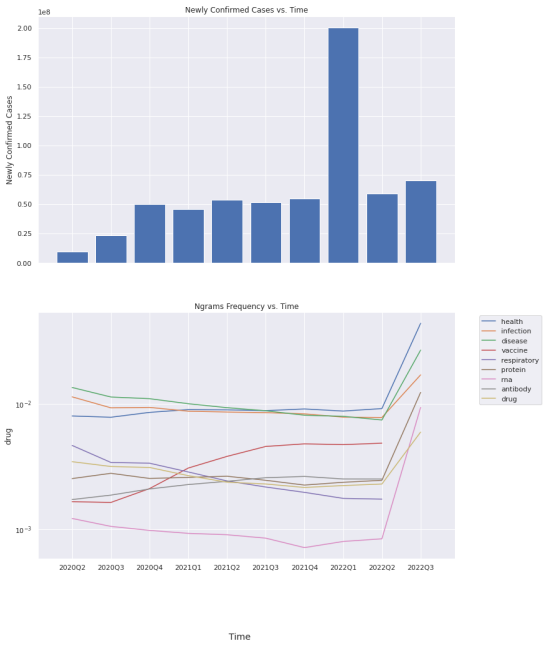

Fig. 5. Newly Confirmed Cases vs. Keywords Frequency

1) The result is visualized in Fig. 3. The figure shows that before 2022Q1 the number of published researchers lags half a year behind the number of newly confirmed cases.
2) Wei Wang, Wei Zhang, and Kwok-Yung Yuen are the most productive researchers. Their contributions are stable before 2022Q2. It is less correlated with the confirmed cases.
3) After comparing the word frequency and the trend of confirmed cases, I divide important words into three groups: factual reflection, research fields, and methods. The trend of factual reflection is the most similar to the change of newly confirmed cases. Words in research fields go to the right track with more confirmed cases appearing. Words in methods change after years. The frequency of "vaccine" changes after two quarters to three quarters, while that of "drugs" changes after almost two years. If we can accelerate the process faster, fewer people will be infected by COVID-19.
4) The frequency of health/rna/protein is most likely to correlate with the newly confirmed cases.

From this project, I have learned how to extract important words from text data and familiarize with big data tools like PySpark and SparkSQL. Task results are stored in task1.csv, task2.csv, task0-2.txt.

VI. CHALLENGE

I have encountered several challenges with this project.

First, CORD-19 is a relatively large research paper dataset (over 1 GB) which increases the task complexity. The project content is very close to text mining that I am not very familiar with. So, I have researched related papers about sentiment analysis of tweets and alike on arXiv. I learned lemmatization, POS tagging, Ngrams, and TF-IDF. These methods inspire me to

work on the project. And I implement lemmatization, POS tagging, and Ngrams(unigram/bigram) in the project. Since it's my first time using them, it cost me lots of time to figure out how to implement them with PySpark framework. So, I have to debug line by line to reach the final results.

Besides, the size of two datasets is quite different, so I have to extract information from CORD-19 with high compression rate. So, I need to accomplish many steps before I dive into the three main questions. These steps are labor-intensive involving MRJob, PySpark, and SparkSQL. But when I join two datasets, the conclusions can't be easily fetched. So, I have to try many times to get reasonable results from the joined data. It makes me realize that extracting patterns from the real-life dataset is rather difficult than thinking.

Additionally, the project content is more related to text mining. It is difficult to analyze over 1 GB of text data and obtain reasonable results. For example, there are many medical terminologies with punctuations like "covid-19" which I need to preserve "-" to maintain its contextual meaning. Therefore, after searching websites and many times of trials, I finally found a way to solve this issue with regex expression.

REFERENCES

[1] L. L. Wang, K. Lo, Y. Chandrasekhar, R. Reas, J. Yang, D. Burdick, D. Eide, K. Funk, Y. Katsis, R. Kinney, Y. Li, Z. Liu, W. Merrill, P. Mooney, D. Murdick, D. Rishi, J. Sheehan, Z. Shen, B. Stilson, A. Wade, K. Wang, N. X. R. Wang, C. Wilhelm, B. Xie, D. Raymond, D. S. Weld, O. Etzioni, and S. Kohlmeier, "Cord-19: The covid-19 open research dataset," 2020. [Online]. Available: https://arxiv.org/abs/2004.10706

[2] E. Dong, H. Du, and L. Gardner, "An interactive web-based dashboard to track covid-19 in real time," *The Lancet Infectious Diseases*, vol. 20, no. 5, pp. 533–534, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1473309920301201