Final Report Data 301

Summary

I propose to research how the trend of public sentiment towards the economy changed in the U.S during the time of COVID-19 pandemic. My initial research question included comparing 1 weeks' worth of data for April 2020 to the data for April 2019 and finding the change in the trend towards the economy. Since I had trouble extracting the data for 2019, I decided to switch the month from April 2019 to January 2020. I think that comparing the months of January 2020 to April 2020 will be much more relevant as I could compare the trend in the economy before and after COVID-19 was declared a pandemic which was 11th March 2020. To get a more precise output I also decided to increase my data size and expanded it from 7 days to 26 days. Now I will be comparing the data from 26 days in January 2020 to 26 days in April 2020.

To do this I will use the GDELT 2.0 data set and with the help of the API summary record the URLs of the newspaper articles for each day to extract the interesting set of keywords. One of the main algorithms that I will use to back up my proposed research question would be A-priori along with the association rules such as confidence and interest.

The intended result should reflect the drastic rise in the negative public sentiments related to the economy and the housing market with keywords such as rent, mortgages, bills, payments, job-losses, unemployment, loans, etc due to the spread of coronavirus when compared to the time in January when it COVID-19 was not declared as a pandemic. I would also be interested to check out the change in the interest of the pairs that were common during both the months.

Introduction

Background

To research the above question, I will be using GDELT Summary API. GDELT summary is a powerful browser-based global multilingual and visual online news search platform that offers a simple nontechnical interface to search the textual and visual narratives of the world's news media. It provides the ability to enter simple keywords, select dropdown filters such as location, time, date, source country and generates a summarized global coverage of the news articles related to that keyword. This summary can be shared or used simply by using the query URL generated from the above process. I used this summary API to restrict my research and filter the news articles related to the economy by using keywords such as unemployment, banks, loans, jobs, and coronavirus and restricting my source country to be just the U.S.

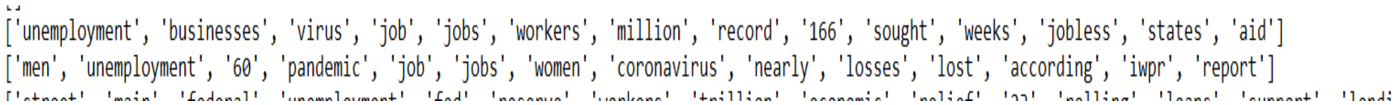
One of the algorithms I use is this project is the A-priori algorithm. This algorithm is used to find the frequent keyword pairs whose support is greater than the set threshold value. In order to understand this and make a bit more sense for the data, I also use other algorithms such as confidence, interest score, and cosine similarity. The frequent items set determined by the Apriori can be used to determine associations rules which highlight general trends in the database, the confidence score of a pair will reflect how often pairs occur together and the interest score will reflect the absolute value of the amount by which the confidence differs.

Research Question

Due to the spread of the pandemic COVID-19, there have been lots of travel restrictions, people have lost their jobs and unemployment has risen. I hypothesize that there has been an increase in the public sentiment towards the economy since COVID-19 was declared a pandemic. I will be looking into which economy-related keywords are most commonly associated together. By using the A-priori algorithm I will be able to find the most interesting keyword pairs that occur over the set threshold and further by applying the association rules such as confidence and interest score, I will directly be able to answer my question. The calculated value for the confidence and the interest scores would tell me which have been the top frequently occurring pairs and which ones have been the most interesting out of them.

Design and Method

1. Build the query using the GDELT API summary with the chosen economy-related keywords and US as the source country.
2. Download the data and combine it into one massive .csv containing all the URLs for the article in the specified date range in the query.
3. Read the downloaded.csv file, map each of the URLs in the file, and perform a keyword scrap operation using the python's newspaper3k library and write the scraped keywords into a text file. This process takes around 1.5 hr when running on a data range of 26 days.



```
['unemployment', 'businesses', 'virus', 'job', 'jobs', 'workers', 'million', 'record', '166', 'sought', 'weeks', 'jobless', 'states', 'aid']
['men', 'unemployment', '60', 'pandemic', 'job', 'jobs', 'women', 'coronavirus', 'nearly', 'losses', 'lost', 'according', 'iwpr', 'report']
['street', 'main', 'federal', 'unemployment', 'fed', 'reserve', 'workers', 'trillion', 'economic', 'relief', '23', 'rolling', 'loans', 'support', 'lendi
```

Figure: 1

4. Download the generated .txt files for selected months, as this will be used for the rest of the project to test the algorithms. The figure 1 reflects the format of the generated keyword file, where each list reflects the set of extracted keywords where each article represents a bucket.

5. Read the uploaded file into two different RDDs based on each month and remove the ones which just contained a single keyword.
6. Apply the two-step A-priori algorithm to get the pair count.
7. Calculate the confidence score for each pair followed by their interest scores.
8. Repeat the same steps for the same set of dates but for another month.
9. Make a list of interest scores for only the pairs that were common in both the months and compute the cosine similarity between them.

Libraries

Listed below are the libraries that I have used to get the required result:

Pyspark- A distributed framework that can handle big data analysis

GDELT Summary - To download the .csv file which contains summarized global news media coverage of a particular topic.

Newspaper, article - Python libraries for extracting and curating articles

Pandas, pytz, datetime- Date and time module in python

SQLContext - Used to create a data frame, register data frames, and run SQL queries on the generated tables.

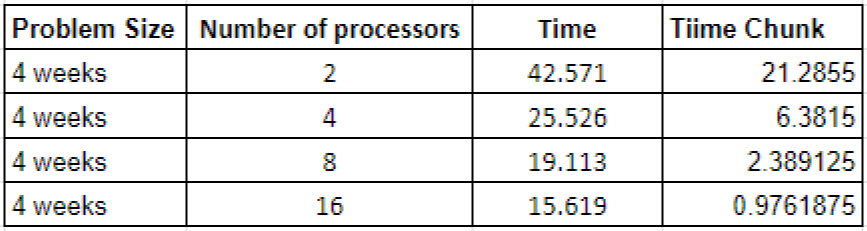
Methods

- **get_filename** - Returns the name of the filename that stores the downloaded GDELT data.
- **intofile** - call GDELT API and store data into files
- **combine_files-** combine all the individual files every single day and creates one massive CSV file.
- **generating_rdds**- uses the newspaper3K library to fetch the keywords from the .csv files and writes it to a text file. (The RDD's of keywords were written on to a text file to save time and speed up the testing.)
- **filtering_and_making_pairs** - generates the key-value pairs for from the keywords in each article
- **creating_broadcasted_pairs** - broadcast the list of key-value value pairs
- **check_pairs-** this function checks the broadcasted pairs against the key-value pairs
- **confidence -** Confidence the confidence of the key-value pairs
- **interest** - computes the interest scores of the key-value value pairs
- **cosine_similarity-** computes the cosine similarity between the interest scores

Result

Fixed problem size with increased number of cores

The table 1 reflects the time taken by each core to process the code for the same problem size. The problem size here was set to 4 weeks which nearly processes keywords that were extracted from around 7000 newspaper articles. The reflected time in the table is precisely for the code that runs the algorithm as the files were uploaded on google cloud instead of using cloud resources to download the data.



Problem Size	Number of processors	Time	Tiime Chunk
4 weeks	2	42.571	21.2855
4 weeks	4	25.526	6.3815
4 weeks	8	19.113	2.389125
4 weeks	16	15.619	0.9761875

Table 1

Figure 2 reflects the lower trend based on the data computed in table 1. For a fixed problem size, it can be noticed that the code is scaling well. As the number of cores is increased the time to process the data decreases. However, every time the cores have doubled the margin of decrease in the processing time reduces.

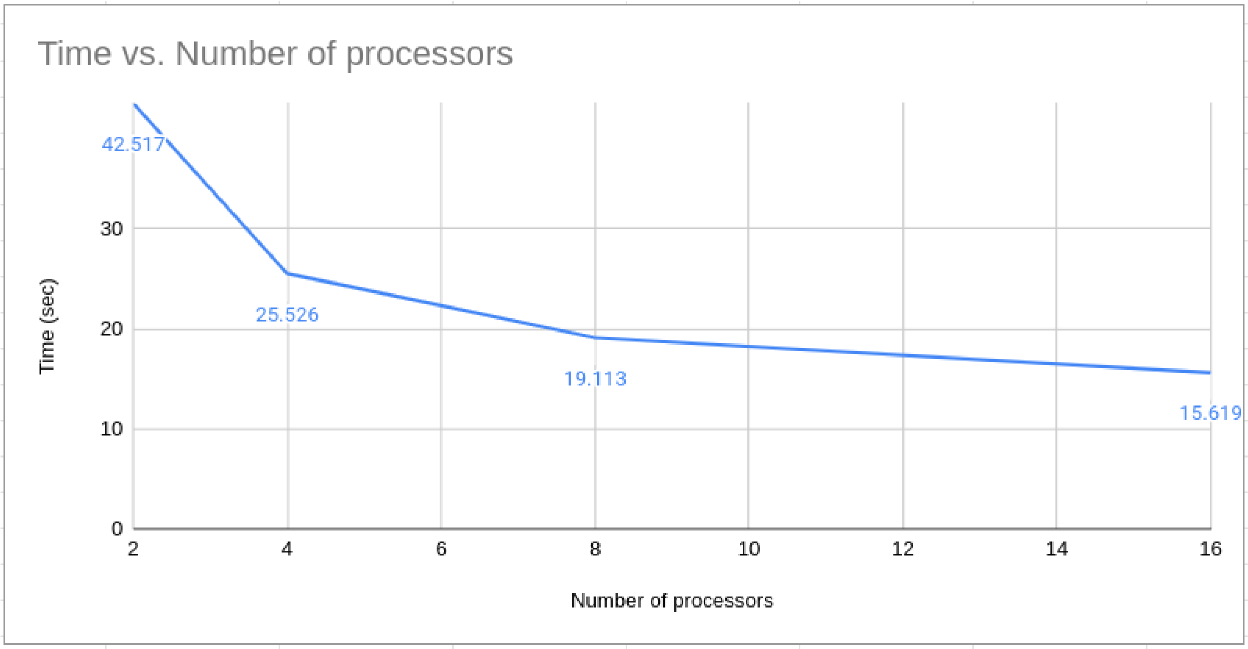
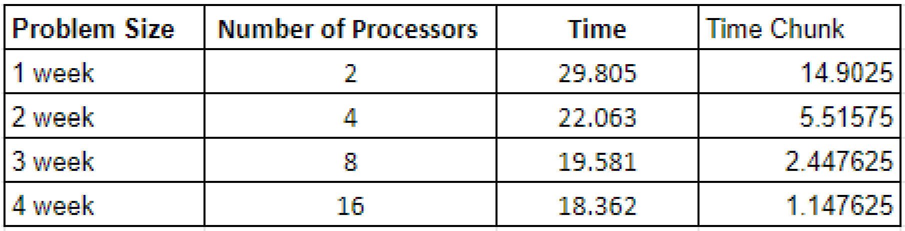


Figure 2

Increased number of cores with increased problem size

Table 2 reflects the time taken by each core to process the code depending upon the different problem sizes. The problem size in this scenario was increased at the same rate i.e was increased by 7 days each time the number of processors was doubled. The reflected time in the table is precisely for the code that runs the algorithm on the files that were downloaded from Collab and then uploaded to google cloud.



Problem Size	Number of Processors	Time	Time Chunk
1 week	2	29.805	14.9025
2 week	4	22.063	5.51575
3 week	8	19.581	2.447625
4 week	16	18.362	1.147625

Table: 2

Figure 3 reflects the lower trend based on the data computed in the above table. As the data gets bigger in size, it can be noticed that the code is scaling well. When the number of cores are increased to 16 the time taken by each core to process the data also reduces.

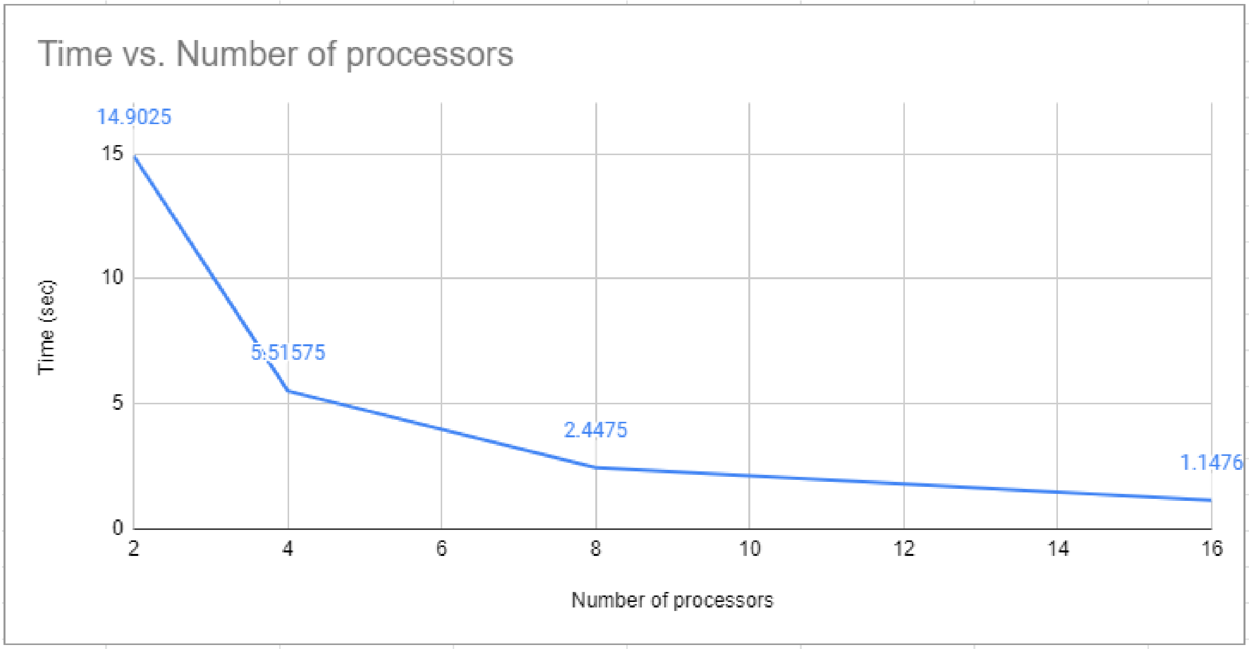


Figure: 3

Conclusion

I hypothesized to see the drastic rise in the public sentiment related to the economy especially after the COVID-19 was declared as a pandemic in the US. I planned to research my hypothesis by scraping the online news article by looking for special keywords related to the economy and then by applying two-step A-priori algorithms to compute the most frequent pairs, during that time. The results that were obtained from this analysis answer the proposed research question. Firstly, it confirms that since COVID-19 has been declared a pandemic and since lockdown and travel restrictions have been placed, the economy in the US has suffered to a great extent and so as the concern amongst the public related to the economy has increased.

During the initial steps of A-priori, I found that keyword pairs such as (‘ unemployment’, ’coronavirus’) were used 385 times in the month of January 2020 which increased to 985 during the same date range for the month of April. I also noticed that other keyword pairs such as (‘unemployment’, ‘workers’), (‘unemployment’,’ millions’) also showed a drastic increase in their pair count, which shows the direct negative impact of coronavirus on the economy. After applying the A-priori algorithm to find the set of topmost frequent pairs I also used association rules such as confidence and interest score to determine the context and popularity of different types of news articles. High confidence scores of keyword pairs such as (‘unemployment’, ‘filed’) of 0.989 and (‘jobless’, ‘sought’) of 0.979 when compared to the keywords in January 2020 such as (‘unemployment’, ‘benefit’) of 0.931 and (‘Wuhan’, ‘pneumonia’) of 0.839 shows the direct impact of coronavirus and the change in the context of the news articles. Though there was unemployment in January 2020 when compared to later months of a pandemic such as April this can be noticed to have a direct link with the outspread of the disease. The comparison of the interest scores between the common pairs between the two months also provides us with solid evidence of the negative impact of coronavirus on the economy. This can be seen as there was an extensive rise in the interest score of some of the important pairs such as (‘unemployment’, ‘benefit’) spiked up from 0.729 to 0.848, (‘unemployment’, ‘jobless’) spiked up from 0.683 to 0.802. These interest scores of common pairs between the two months were further used to calculate the change in the percentage between the two months.

In the future, I would like to analyze and compare the data around different major events such as the great recession, natural disasters, wars or other similar events between different countries that have also shown a drastic impact on the economy. Using multiple events to generate my data set and compare them would produce much more interesting pairs and would also increase the integrity of my result. I would also like to explore and introduce K - mean clustering to find the closely related events and perhaps group the events together based on their overall impact on the economy. When performed on multiple countries the result would provide me further understanding of the correlation between different countries and how the public sentiment towards the economy was similar.

Critique of Design and Project

As stated in the summary I had to modify the set of date ranges that were initially selected for the project due to the lack of relevant data. Also, I had to increase my set of dates to get a more significant output. Initially, I was going to process articles for only 2- 7 days which I decided to expand it up to 26 days to get a bit more concrete output. If I was going to work on the same project in future, I would like to extract the news article on an hourly basis instead of daily. The GDELT API summary provides a large-scale global coverage of the news article but has a constraint on the limits of articles that can be downloaded per day which is 250. In the future, I would like to extract the article on an hourly basis which would increase my scope of keywords and provide me with more data to construct the interested dictionary and would possibly produce more interesting keyword pairs. However, I still believe that using the A-priori algorithm and the process of market basket analysis to derive the interest vectors and compute the cosine similarity between the different time ranges is an effective approach to answer the question. That being said, I believe sectioning the date ranges based on other major events that have shown a similar impact on the economy as coronavirus and then applying the K-mean clustering algorithm on it will produce more solid results to back the research question.

For the current project, downloading the data from GDELT, scrapping newspaper articles and then writing that RDD to a file takes approximately around 1.5 hour. for 4 weeks of data and if the above approach of comparing different events for multiple countries is applied the processing time to retrieve the data will increase extensively.

Reflection

The project allowed me to understand the concept of market basket analysis and how association rules can be applied to the generated data to highlight changing trends in the economy. After implementing the algorithms on a real-world situation to produce a constructive result and to answer the proposed research question, I feel much more confident in working with PySpark operations such as reduceByKey, map, flatMap, and filter. The project has also helped me to explore and expand my knowledge on multiple python libraries such as newspapers, SQL context, and panda which I am sure would prove to be quite helpful in my career as a software engineer. As part of the course, I also learned how deploying the code on google cloud can help to check the scalability of the code when run with different numbers of processors. Overall, the project and the course provided me a solid foundation to write complex pyspark programs that can query big data and produce interesting results.

References

https://blog.gdeltproject.org/announcing-gdelt-summary/

https://api.gdeltproject.org/api/v2/summary/summary

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.date_range.html

https://spark.apache.org/docs/latest/api/python/pyspark.html?highlight=textfile