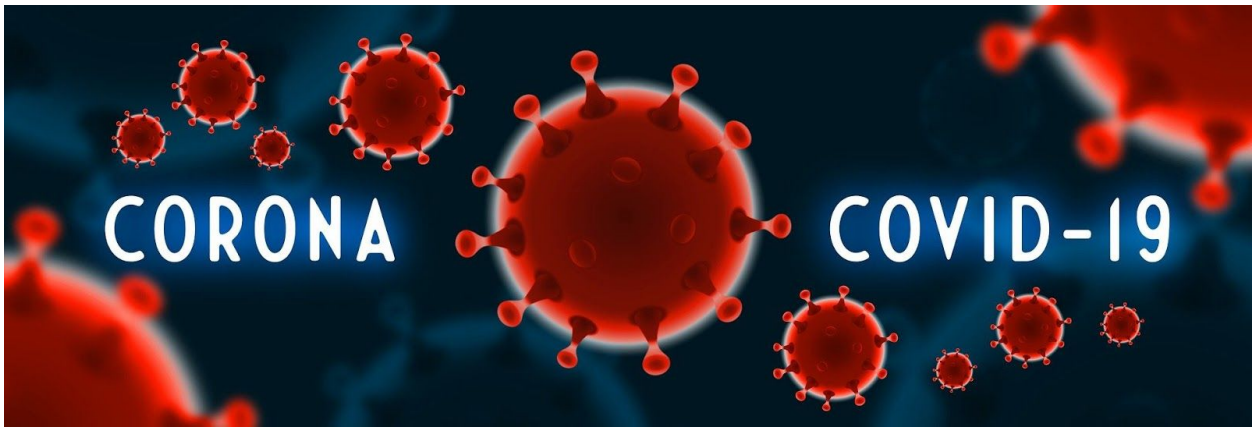

Project 4 : Clustering analysis on Open Research Dataset CORD 19

Overview

In response to the COVID-19 pandemic, the White House and a coalition of leading research groups have prepared the COVID-19 Open Research Dataset (CORD-19). CORD-19 is a resource of over 57,000 scholarly articles, including over 45,000 with full text, about COVID-19, SARS-CoV-2, and related coronaviruses. This freely available dataset is provided to the global research community. As a big data community, how can we help researchers to easily find the related research papers easily?

You can find the dataset and the main challenge on kaggle

https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge

Goals

Given the large number of literature and the rapid spread of COVID-19, it is difficult for health professionals to keep up with new information on the virus. Can clustering similar research articles together simplify the search for related publications? How can the content of the clusters be qualified? And over each cluster how can we recommend the most similar papers leveraging clustering?

Requirements

you are required to find out the best way to cluster the research papers using the research papers details in the JSON file with the metadata in the CSV file, then you should build a neighborhood recommender system to receive the title of research paper and recommend the most N similar papers to it based on its cluster. So you should find a way to represent the papers in vectors and cluster them then build a neighbourhood recommender system on the clusters.

Required Steps

1. **Read the dataset using spark**. The dataset is 8GB so we don't expect you can manage the whole entire dataset on your local machine. You can choose from three options:
    - Take a sample from the dataset on your local machine and proceed with the project.
    - Recommended: Use databricks.com as they already mounted the data and you can directly use it , the paths of the dataset

      https://databricks.com/

      `comm_use_subset_path = "/databricks-datasets/COVID/CORD-19/2020-03-13/comm_use_subset/comm_use_subset/"`

      `noncomm_use_subset_path = "/databricks-datasets/COVID/CORD-19/2020-03-13/noncomm_use_subset/noncomm_use_subset/"`

      `biorxiv_medrxiv_path = "/databricks-datasets/COVID/CORD-19/2020-03-13/biorxiv_medrxiv/biorxiv_medrxiv/"`

```
json_schema_path =
"/databricks-datasets/COVID/CORD-19/2020-03-13/json_schema.txt"
```

- Create a real cluster on digital ocean https://www.digitalocean.com you can use the starting free credit. You can watch this screencast to see how to setup the cluster: https://www.youtube.com/watch?v=gW7aDAAgka0&feature=youtu.be. You can also get a free 100$ account for 60 days: https://try.digitalocean.com/performance/

Remember to use the tricks covered in the course to enhance the performance like:

Repartition and saving the json files as parquet.

2. **Do exploratory data analysis:**

   Do the EDA to understand your data and extract insights help you in feature engineering ,Document your insights

3. **Preparation and Cleaning the data:**
   - Joining the json file with the metadata in the csv file
   - Handling Nulls.
   - Handling Duplications.
   - Keep Only the english documents (you can get help from any python libraries that can detect language to do that)

4. **Preprocessing:**

   Our main goal is to clean and preprocess the txt to prepare it to represent it in vectors. It is a mandatory step in NLP projects to preprocess the text. You can have a look in this article to explore some of well known preprocessing steps

   https://towardsdatascience.com/nlp-text-preprocessing-a-practical-guide-and-template-d80874676e79

   Required preprocessing and you can do more (bonus) (you can use spark ml):

   1. Remove stop words.

2. Remove custom stop words, Research papers will often frequently use words that don't actually contribute to the meaning and are not considered everyday stopwords and should be removed to enhance the accuracy.

   custom_stop_words = ['doi', 'preprint', 'copyright', 'peer', 'reviewed', 'org', 'https', 'et', 'al', 'author', 'figure','rights', 'reserved', 'permission', 'used', 'using', 'biorxiv', 'medrxiv', 'license', 'fig', 'fig.', 'al.', 'Elsevier', 'PMC', 'CZI', 'www']

3. Remove Punctuation, use this Regex `'!()-[]{};:'"\,<>./?@#$%^&*_~'` to remove it.
4. convert text to lower case

5. **Vectorization** :

   convert the data into a format that can be handled by our algorithms. For this purpose you can use.

   https://spark.apache.org/docs/latest/mllib-feature-extraction.html

   - TF-IDF. This will convert our string formatted data into a measure of how important each word is to the instance out of the literature as a whole.

     https://www.youtube.com/watch?v=hc3DCn8viWs

   - Or Word2vec

     https://www.youtube.com/watch?v=3eoX_waysy4

6. **Clustering**

   Apply clustering algorithm on the data and choose the best k you decide from the elbow method. You can use PCA to reduce the dimensions while still keeping 95% variance for better performance and hopefully remove some noise/outliers. Evaluate the algorithms you are going to use and justify which one you will use.

7. **Recommender system**

   Build a very basic recommender system:

   - Create a function with the signature recommendPaper(paper_title,N) where N is the number of recommended papers in the list and it returns the recommendation list.
   - Recommend top N recommendation list based on the most similar(cosine similarity) papers to it with respect to the cluster it belongs.

Deliverables

I. Jupyter notebook with your code labelled with our main seven requirements.

II. Detailed Documentation pdf file