Text mining for exploration of COVID-19 severity factors

LAI Khang Duy
Mariia KLIMINA

Université Paris Cité
UFR des Sciences Fondamentales et Biomédicales

Université Paris Cité

03-05-2022

Contents

1 Abstract

COVID-19 is the disease that caused by the Sar-COV-2 virus originated in China at the end of the year 2019. Over the time, studies have shown that there is some form of background diseases and risk factors that can hugely affect the severity cases rate of COVID-19. This project will apply NLP and text mining methods in order to explore CORD-19 dataset and extract background diseases and risk factors.

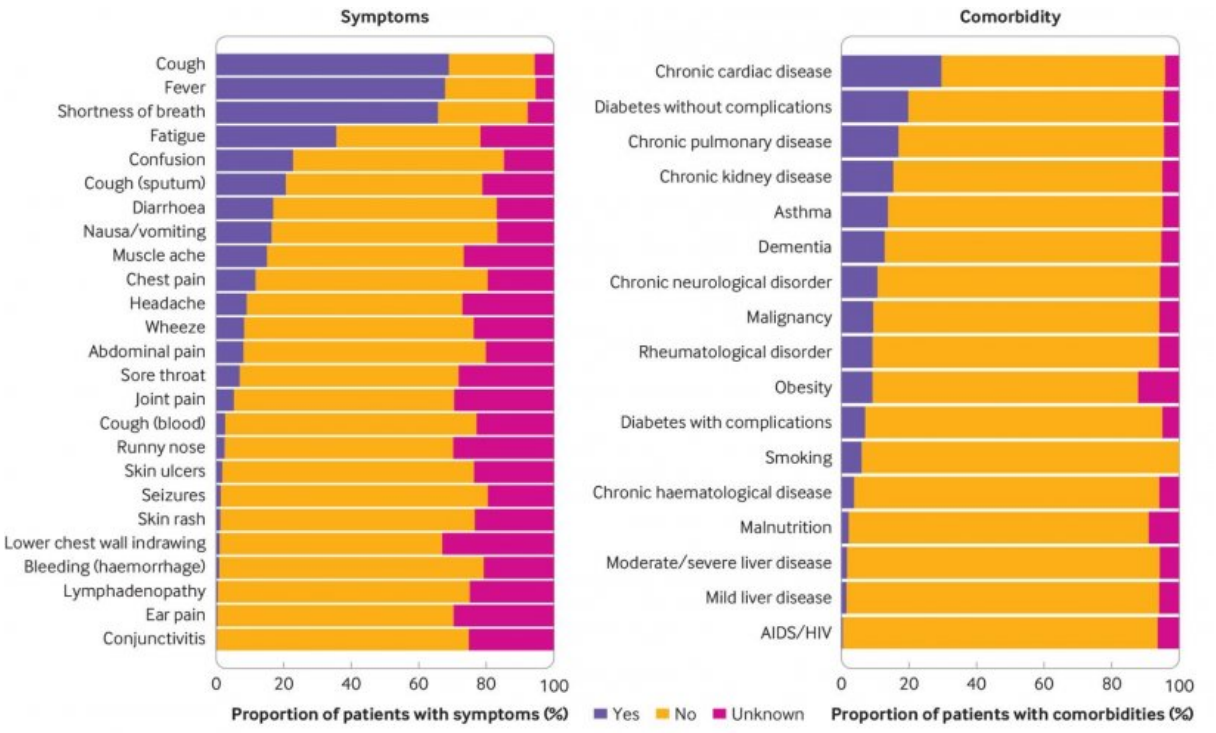

Figure 1: Cormobilities and symtoms of COVID-19 cases

2 Introduction

While working on this project, we applied text-processing methods on CORD-19 dataset. CORD-19 is a data collection of over one million scholarly articles, including over 350,000 with full text, about COVID-19, SARS-CoV-2, and related coronaviruses. The amount of data collected in CORD-19 provided an opportunity for a deep and various analysis, and allowed us to apply different NLP techniques such as LDA(Latent Dirichlet Allocation) and NER(Named-entity recognition). In this block, the structure of our project will be explained.

The coding process consisted of 4 parts: Data Exploration, Preprocessing, Data selection, Named-entity recognition application.

- Data Exploration
- Preprocessing
  - Reformating the json data to csv dataframe.
  - Removing all non-english paper.
  - Tokenizing.
  - Removing stopwords.
  - Stemming.
  - Lemmatisation.
- Data selection
  - Selecting articles with risk factors and severity key-words.
  - Clustering using Latent Dirichlet Allocation.
- Applying NER (Named-entity recognition).

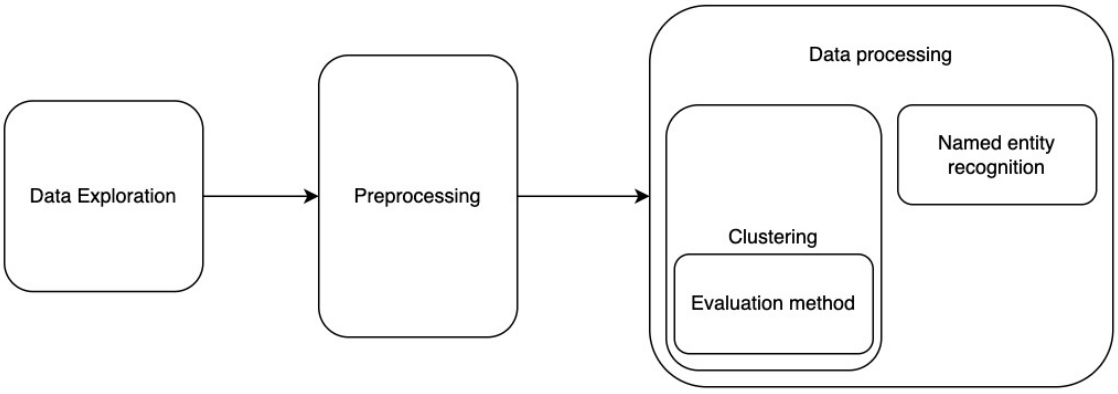

Figure 2: Data processing flow

3 Data exploration

In this part we will cover the main features that we discovered during the data exploration. The successful outcome of this block helped us to apply preprocessing and understood the data we were working with. It is important to mention, that in this part we used only metadata dataset which contained all useful information.

3.1 Dataset information

This block is divided by two parts: the general information of a dataset and a language specificity.

- As we can see on a picture, at our disposal are more than one milion papers.

Figure 3: Paper's number

- The metadata data collection consists of following columns.

```
['cord_uid', 'sha', 'source_x', 'title', 'doi', 'pmcid', 'pubmed_id',
'license', 'abstract', 'publish_time', 'authors', 'journal', 'mag_id',
'who_covidence_id', 'arxiv_id', 'pdf_json_files', 'pmc_json_files',
'url', 's2_id']
```

- `cord_uid`: A `str`-valued field that assigns a unique identifier to each CORD-19 paper.
- `sha`: A `List[str]`-valued field that is the SHA1 of all PDFs associated with the CORD-19 paper.
- `source_x`: A `List[str]`-valued field that is the names of sources from which we received this paper.
- `title`: A `str`-valued field for the paper title
- `doi`: A `str`-valued field for the paper DOI
- `pmcid`: A `str`-valued field for the paper's ID on PubMed Central.
- `pubmed_id`: An **`int`**-valued field for the paper's ID on PubMed.
- `license`: A `str`-valued field with the most permissive license we've found associated with this paper.
- **`abstract`**: A `str`-valued field for the paper's abstract
- `publish_time`: A `str`-valued field for the published date of the paper. This is in `yyyy-mm-dd` format.
- `authors`: A `List[str]`-valued field for the authors of the paper.

- `journal`: A `str`-valued field for the paper journal.
- `who_covidence_id`: A `str`-valued field for the ID assigned by the WHO for this paper.
- `arxiv_id`: A `str`-valued field for the arXiv ID of this paper.
- `pdf_json_files`: A `List[str]`-valued field containing paths from the root of the current data dump version to the parses of the paper PDFs into JSON format.

- `pmc_json_files`: A `List[str]`-valued field. Same as above, but corresponding to the full text XML files downloaded from PMC, parsed into the same JSON format as above.
- `url`: A `List[str]`-valued field containing all URLs associated with this paper.
- `s2_id`: A `str`-valued field containing the Semantic Scholar ID for this paper.

	cord_uid	sha	source_x	title	doi	pmcid	pubmed_id	license	abstract	publish_time	author
0	ug7v899j	d1aafb70c066a2068b02786f8929fd9c900897fb	PMC	Clinical features of culture-proven Mycoplasma...	10.1186/1471-2334-1-6	PMC35282	11472636	no-cc	OBJECTIVE: This retrospective chart review des...	2001-07-04	Madani, Tari A; Al-Ghamd Aisha
1	02tnwd4m	6b0567729c2143a66d737eb0a2f63f2dce2e5a7d	PMC	Nitric oxide: a pro-inflammatory mediator in l...	10.1186/rr14	PMC59543	11667967	no-cc	Inflammatory diseases of the respiratory tract...	2000-08-15	Vliet, Alber van de Eiserich, Jaso P; Cros.
2	ejv2xln0	06ced00a5fc04215949aa72528f2eeaae1d58927	PMC	Surfactant protein-D and pulmonary host defense	10.1186/rr19	PMC59549	11667972	no-cc	Surfactant protein-D (SP-D) participates in th...	2000-08-25	Crouch, Erik
3	2b73a28n	348055649b6b8cf2b9a376498df9bf41f7123605	PMC	Role of endothelin-1 in lung disease	10.1186/rr44	PMC59574	11686871	no-cc	Endothelin-1 (ET-1) is a 21 amino acid peptide...	2001-02-22	Fagan, Kare A; McMurtry Ivan F Rodman, Davi
4	9785vg6d	5f48792a5fa08bed9f56016f4981ae2ca6031b32	PMC	Gene expression in epithelial cells in respons...	10.1186/rr61	PMC59580	11686888	no-cc	Respiratory syncytial virus (RSV) and pneumoni...	2001-05-11	Domachowske Joseph E Bonville Cynthia A Ro.

Figure 4: Head of the metadata

To be more precise, the number of files that we can work with in the directory is approximately over 300000 json files. The explanation for this is that some papers in the metadata dataset is not available in the json format for us to process and some of them are duplicated.(to fix it)

3.2 Language status of the dataset

During this project, we agreed to work only with english-written articles. That is why we made an analysis that you can see on an image below. As can be observed most of the articles are meeting the requirements. In addition to that, papers that do not respond to the criteria(*check the spelling*) will be deleted in the preprocessing part.

To detect the language that is written in the paper, we use a library called langdetect to do it. To speed up the language detect process we will not detect the language of the whole body but only detect on

the first part.

- Try to detect on the first 50 words of the body text, if the number of words is lower than 50, take the whole text instead.

```
if len(text) > 50:
        lang = detect(" ".join(text[:50]))
elif len(text) > 0:
        lang = detect(" ".join(text[:len(text)]))
```

- If first 50 words does not work, try to detect with the whole body.
- If we cannnot detect the language using the body part, try to detect on the abstract.

```
try:
        lang = detect(df.iloc[ii]['abstract_summary'])
except Exception as e:
        lang = "unknown"
```

- In other case, mark the language as unknown.

This is the final result when we randomly pick 10000 papers. It is obviously that the most out of dataset is written in English.

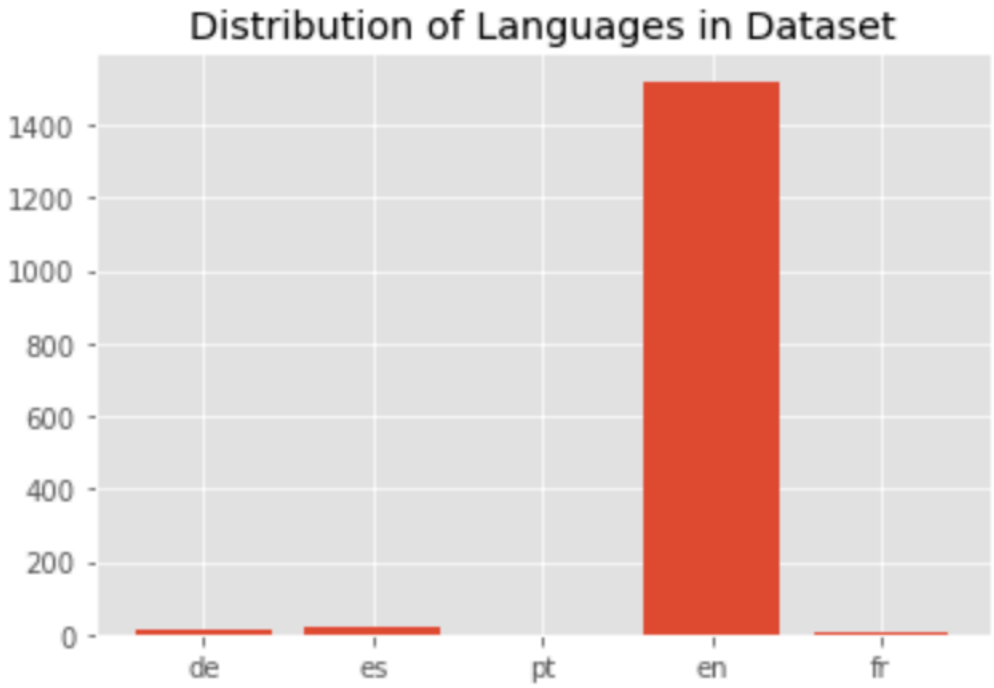

Figure 5: Language percentage in the dataset

4 Preprocessing

The second part of this project is preprocessing. This step will first clean the data, including transfer the JSON file that be used in the dataset to pandas dataframe, which is more common to process. In the preprocessing block, we will remove the row with duplicated and empty abstract.

4.1 Handling multiple languages

As in the data explore, more than 95% of papers are written in English. We have already add a new column named `language` in the dataframe. It could be easily filtered out with

```
df = df_covid[df_covid['language'] == 'en']
```

4.2 Change from JSON to a more convinient format

4.3 Special characters and number remove

Convert to lowercase and remove punctuations and characters and then strip.

```
text = re.sub(r'[^\w\s]', '', str(text).lower().strip())
    pat = r'\d+'
    text = re.sub(pat, '', text)
```

4.4 Tokenization

Simply using `split()` to tokenize the data.

4.5 Stemming

```
if flg_stemm == True:
        ps = nltk.stem.porter.PorterStemmer()
        lst_text = [ps.stem(word) for word in lst_text]
```

4.6 Lemmatisation

```
if flg_lemm == True:
        lem = nltk.stem.wordnet.WordNetLemmatizer()
        lst_text = [lem.lemmatize(word) for word in lst_text]
```

5 Clustering

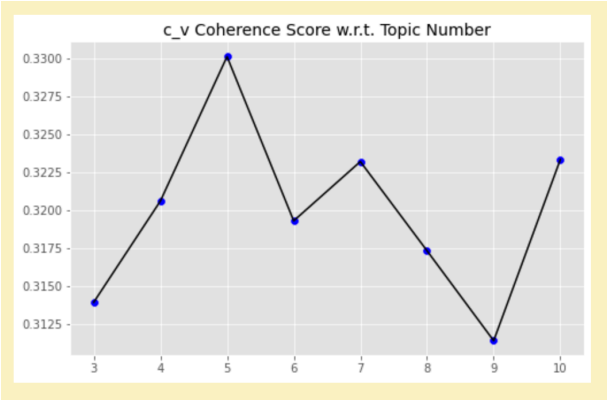

Figure 6: LDA 1

t

t

t

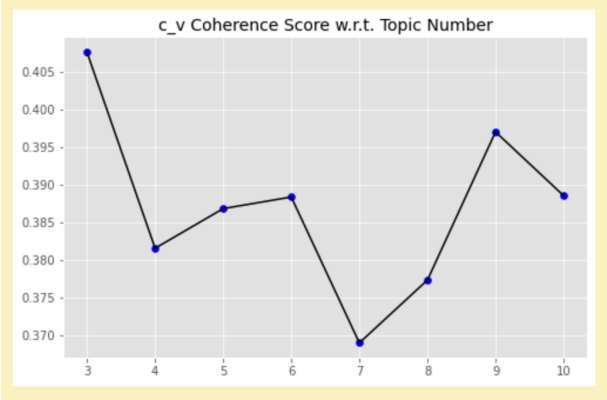

Figure 7: LDA 2

6 Named-identity recognition

Scispacy t t

t t

t

7 Future improvement

Knowledge graph

8 References

https://github.com/allenai/cord19 - this for the metadata desrp