Universitat Oberta de Catalunya

uoc.edu

Application of NLP to extract biomedical entities from COVID-19 papers

Student: Jin Lung Chan
Master Degree in Data Science
M2.980 TFM - Area 3

Coordinating Professor: Ferran Prados Carrasco
Supervisor: Erola Pairó Castiñeira

Submission date: 06/06/2021

FINAL PROJECT OVERVIEW

Project title:	Application of NLP to extract biomedical entities from COVID-19 papers
Author name:	Jin Lung Chan
Coordinating Professor:	Ferran Prados Carrasco
Supervisor:	Erola Pairó Castiñeira
Submission date (mm/yyyy):	*06/2021*
Degree:	*Master degree in Data Science*
Final project area	M2.980 TFM - Area 3
Project language:	*English*
keywords	*"Named Entity Recognition," "Natural Processing Language," "COVID-19"*

Abstract (in English, 250 words or less):

The project explores the current State-of-the-Art of NLP, researches different biomedical datasets, and applies the SciSpacy library to extract and recognize entities from the CORD-19 dataset, a growing data collection with over 500.000 scientific papers linked to COVID-19.

The data extraction code is written in Python and deployed in the Kaggle platform. Different visualization software such as Tableau and Gephi has been used to represent the extracted entities in the post-processing analysis.

Index

List of figures

1. Introduction

1.1. Context and project justification

The scientific community generates a vast of information through investigations, and at the same time, they do consume the findings from other researchers due to scientific literature has been the most reliable source (Mitsumori et al., 2005). This chain of exchange of information in the written articles generates large data that cannot be assessed manually.

Such is the case of the Scientifics from the biomedical field whom needs to extract relevant information from the papers to reproduce the results. In the last decades, different efforts have been performed to cope with this issue, like the BioCreAtIvE initiative, which is a competition where the participants develop algorithms and apply data mining techniques (BioCreAtIvE, 2021), or the UnitProt, which is a free database that compiles information about the proteins sequences (UnitProt, 2021).

In the era of Data Science, data mining techniques to extract information from the biological domain revolve around the Natural Processing Language (NLP), extracting structured data from unstructured information using the output to recognize and classify the entities in categories.

Nonetheless, the identification of entities in biomedical scientific papers entails a challenge because, among other reasons, there are too many abbreviations, and the paper authors do not use standard gene names (Mitsumori et al., 2005). In addition, dictionary-based recognition tools might fail to identify the entity because the exact entity mention within a sentence depends on the context, and therefore homonyms must be resolved (Kroll et al., 2020).

The main motivation to carry out the current project is to improve the scientific community capacity to deal with big data sets, allowing them to save their data pre-processing and extraction efforts to invest the time in the analysis. In addition, it is also interesting to apply the knowledge acquired in the Master Degree in Data Science at the Open University of Catalonia in the field of biomedicine because the “Biomedical research is drowning in data, yet starving for knowledge” (Holzinger & Jurisica, 2014).

1.2. Project goals

General goals:

- The project aims to apply a Name Entity Recognition (NER) model to parse the abstract of scientific papers linked to COVID-19 to automatically identify entities such genes, tissues, and cells, among other kinds.

Specific goals:

- Explore the current state-of-the-art of NLP libraries
- Collect a dataset with COVID-19 scientific papers
- Analyze the COVID-19 dataset
- Write code to extract entities with the chosen NLP library
- Analyze the outcome generated from the recognized entities

1.3. Scope and methodology

The project strategy starts evaluating the current State-of-the-Art of the NPL applied in the data recognition of medical, scientific papers to explore the latest trends, how it has improved, and what problems it might face.

The analysis will allow to obtain the best approach to tackle the recognition of the relevant entities within the texts and to chose an optimal NLP model. It has also required to select a data set from open source to test and train the model.

Afterwards, the NER model will be applied against the abstract from a bulk of scientific papers. Finally, a post-processing analysis will be carried out to hightlight the conclusions.

1.4. Project planning

The project planning has been prepared out according to the deliveries and deadlines established initially by the tutors, and it has been split into the following five stages:

1. **Project definition and planning**
   1.1. The definition of the project and its scope has been defined with the support of the responsible teacher. This phase also includes an initial exploration and collection of the information that will be used to justify and structure the project.

2. **State-of-the-art analysis**
   2.1. Research and analyze what projects have been carried out to face automatic extraction of entities from biomedical scientific papers.

3. **Design and implementation**
   3.1. Interview possible users that will use the final product.
   3.2. Research information about the biomedical domain to fulfill the primary needs of the project.
   3.3. Gather biological scientific papers to build a dataset to test
   3.4. Data pre-processing and extraction NLP

4. **Draft report**
   4.1. Compile all the project information, including tasks, findings, and conclusions.
   4.2. Draft report

5. **Project presentation**
   5.1. Prepare slides of the project
   5.2. Prepare the oral presentation

The following Gant diagram depicts the project schedule:

				February		March					April				May				June	
Tasks	**Start**	**End**	**Total days**	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18	Week 19	Week 20	Week 21	Week 22	Week 23
1. Project definition and planning	**17/02/2021**	**28/02/2021**	**11**																	
1.1 Reach out assigned teacher	17/02/2021	18/02/2021	1																	
1.2 Search and compile information	18/02/2021	22/02/2021	4																	
1.3 Project planification	22/02/2021	23/02/2021	1																	
1.3 Draft report	24/02/2021	28/02/2021	4																	
2. State of the art	**29/02/2021**	**21/04/2021**	**29**																	
2.1 Research NLP scientifier papers	21/03/2021	26/03/2021	5																	
2.2 Research NER scientifier papers in biomedical field	26/03/2021	03/04/2021	8																	
2.3 Research NER evaluation metrics	05/04/2021	05/04/2021	1																	
2.4 Research Biomedical datasets	06/04/2021	08/04/2021	2																	
2.5 Research NLP libraries	09/04/2021	14/04/2021	5																	
2.6 Draft report	15/04/2021	21/04/2021	6																	
3. Design and implement NER prototype	**22/03/2021**	**23/05/2021**	**62**																	
3.1 Dataset selection for testing purposes	22/03/2021	25/03/2021	3																	
3.1.1 Explorer selected dataset	26/03/2021	29/03/2021	3																	
3.2 Scispacy modules exploration	30/03/2021	04/04/2021	5																	
3.3 NER development	05/04/2021	16/04/2021	11																	
3.3 NER Testing phase with selected dataset	17/04/2021	28/04/2021	11																	
3.4 Exploring results	29/04/2021	14/05/2021	15																	
3.5 Draft report	15/05/2021	23/05/2021	8																	
4. Draft report	**24/05/2021**	**06/06/2021**	**13**																	
4.1 Report restructuration	24/05/2021	27/05/2021	4																	
4.2 Add additional sections (limitation and conclusions)	28/05/2021	30/05/2021	3																	
4.3 Review report	31/05/2021	06/06/2021	6																	
5. Project presentation	**07/06/2021**	**18/06/2021**	**11**																	
5.1 Powerpoint preparation	07/06/2021	10/06/2021	3																	
5.2 Presentation preparation	11/06/2021	18/06/2021	8																	

Figure 1 – Project planning diagram

2. State-of-the-art

It has been a while since the linguistic Zellig Harris proposed a theory of science sublanguages (Harris, 1982) which later in the 1980s allowed Naomi Sager and her colleagues of the New York University Linguistic String Project to prove that the language processing applied to medical records and literature, also known as medical language processing (MLP), was feasible (Sager et al., 1987).

Thereafter, biomedical literature mining has evolved, and multiple NLP approaches have been developed to solve automatic text classification problems existing nowadays a vast of options.

2.1. Named Entity Recognition

Named Entity Recognition (NER) is an NLP task to recognize an entity from unstructured data and categorize it to the type it belongs to, as the workflow process may be seen in Figure 2. NER plays a crucial role in the medical domain by extracting meaningful keywords from clinical records and reports used in processing tasks such as entity resolution, relation extraction, assertion status detection, and patient records de-identification (Kocaman & Talby, 2020).

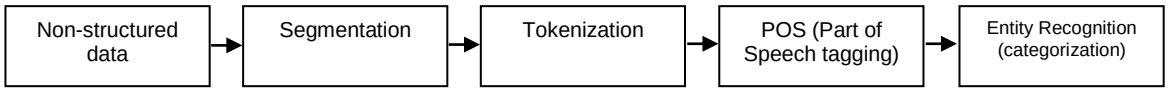

Figure 2 - Named Entity Recognition process

During the last year, despite the biomedical papers complexity, NER systems such **TaggerOne**, which joint NER and named entity normalization with Semi-Markov Models (Leaman & Lu, 2016), and **GNormPlus**, an end-to-end system that handles both gene/protein name and identifier detection in biomedical literature, including gene/protein mentions, family names, and domain names (C-H, H-Y, & Z., 2015) were combined to develop a pipeline upon the latest NER tools for Chemicals, Diseases, Genes and Species in the framework of COVID-19 research (Kroll et al., 2020).

Further on, so important is to classify the entity correctly as is also is to establish the link between the entities, if the relation exists. This task is known as Named Entity Linking (NEL), and it requires a distinct model and training data than NER, although hybrid models combining both tasks have also been developed (Bansal et al., 2020).

2.1.1. NER Challenges

The segmentation of clinical entities in biomedical NER systems is considered a difficult task because of the complex orthographic structures of named entities (Liu et al., 2015).

- **Ambiguity and abbreviations**: Recognizing words with multiple meanings or words can be a part of different sentences. Another major challenge is classifying similar words from texts (Kamath & Wagh, 2017).
- **Multiple languages:** Although the primary language used by researchers is English, other languages are also frequently used in the scientific community.
- **Name and spelling variations:** paper authors do not use standard gene names (Mitsumori et al., 2005).
- **Computational cost:** some complex NER models based on Deep Learning have an expensive computational cost turning difficult to applicate in the production environment.

2.2. Deep Learning approaches in NLP

The mathematic complexity in deep learning did not seem to be an issue for some researchers and developers who belong to different domains. However, they do not have the mathematics knowledge to understand and even less develop a neural network from scratch (López, 2021).

Within the last years, Deep learning methods have a breakthrough in the biomedical domain framework making a difference to solve abstract papers exploitation in the scientific literature achieving the state-of-the-art (SOTA) within the NLP field. Some of those methods use Neural Networks (NN) and Transformer-based models like Bidirectional Encoder Representations from Transformers (BERT) and other distilled versions (Atanassova et al., 2019).

A brief and not exhaustive background description will be given to explore Deep Learning models applied in NLP in the following subsections.

2.2.1. Convolutional Neural Network

A Convolution Neural Network (CNN) is a Deep Learning algorithm with high accuracy to solve image classification problems. Recently, it has also been proved its ability for text classification as a pipeline combined with NLP.

Several recent studies showed that deep neural networks advanced the SOTA in NER, including biomedical NER. However, the accuracy of performance and the robustness of improvements are critical, and it relies on the availability of enough extensive training data, being an issue in the biomedical field due to the small human annotations corpora (Weber et al., 2020).

2.2.2. Bidirectional Encoder Representations from Transformers

In 2018, researchers from the Google AI department introduced a breakthrough new language representation model within NLP called Bidirectional Encoder Representations from Transformers (BERT), designed to pre-train deep bidirectional representations from an unlabelled text by jointly conditioning on both left and right context in all layers (Devlin et al., 2019).

BERT is the first fine-tuning based representation model that achieves state-of-the-art performance on an extensive suite of sentence-level and token-level tasks, outperforming many task-specific architectures

The main drawback of using BERT and other big neural language models is the computational resources needed to train, tune and make inferences. These models are humongous in size. However, real-world applications demand a small model size, low response times, and low computational power wattage (Varma et al., 2020).

2.3. Biomedical NER Datasets

To know if an entity is relevant and classify it correctly, the NER model requires a training dataset within the biomedical domain, and the task accuracy relies on the quality of the data. Therefore, the scientific community has strived to establish datasets focused on specific fields within the biomedical domain.

- **BC5CDR** dataset was selected from the CTD-Pfizer corpus used in the BioCreative V chemical-disease relation task, consisting of 1500 PubMed articles with 4409 annotated chemicals and 5818 diseases (Li et al., 2016).

- **JNLPBA** the data came from the GENIA version 3.02 corpus, and it contains data about cell lines, cell types, DNAs, RNAs, proteins. The dataset was formed from a controlled search on MEDLINE using the Medical Subject Headings terms human, blood cells, and transcription factors. From the results, 2,000 abstracts were selected and hand-annotated according to a small taxonomy of 48 classes based on a chemical classification. Among the classes, 36 terminal classes were used to annotate the GENIA corpus (Kim et al., 2003).

- **BioNLP13CG** dataset was built from two event extraction tasks introduced in the BioNLP Shared Task 2013: Cancer Genetics (CG) and Path Curation (PC). The CG task focuses on cancer targets the automatic analysis of the literature on cancer genetics.

  It emphasizes the extraction of physiological and pathological processes at various levels of biological organization. The PC task targets reactions relevant to the development of biomolecular pathway models, defining its extraction targets based on established pathway representations and ontologies (Pyysalo et al., 2015).

- **CRAFT** (Colorado Richly Annotated Full Text) Corpus is a collection of 97 full-text biomedical journal articles that is being richly annotated both syntactically and semantically and is designed to be an open community resource for the development of advanced bioNLP systems (Bada et al., 2010).

  The dataset contains all most all the terminology like Cell Type Ontology, Chemical Entities of Biological Interest ontology, NCBI Taxonomy, Protein Ontology, Sequence Ontology, Entrez molecular biology database entries, and the three sub-ontologies of the Gene Ontology.

2.4. NER models evaluation

Most NER models precision and performance are evaluated with the outcome of the NER model (predictions) contrasted with the human annotations (gold standard) on the same dataset (Zhang Z., 2013). The standard measures to test the accuracy of the model are the following ones (Tsai et al., 2006):

- **Precision** is the number of named entities a system correctly detected divided by the total number of named entities identified by the system.
- **Recall** is the number of NEs a system correctly detected divided by the total number of NEs in the input text.
- **F-Measure**, also known as **F1-measure** or **F-score**, is a single score and is defined by the following equation:

$$F - score = \frac{2 \times precision \times recall}{precision + recall}$$

2.5. NLP Libraries

The NLP libraries are platforms to simplify the different tasks that involved text mining in non-structured texts. At the following subsections it will be analyzed two NLP libraries, being the applied criteria to select them:

- NLP library has been used before to solve a biomedical domain text classification problem reaching the state-of-the-art
- Programming language has to be in Python due to its simplicity and efficiency
- Free and Open Source with an active community behind who offer an extended development support
- Fit for production with no heavy computational requirements

2.5.1. Stanza

The Natural Language Processing Group at Stanford University developed Stanza, a language-agnostic fully neural pipeline to convert a string containing human language text into lists of sentences and words, including tokenization, multi-word token expansion, lemmatization, part-of-speech and morphological feature tagging, dependency parsing, and named entity recognition. The library has been trained on 112 datasets, and it achieves the SOTA (Figure 4) at each step of the pipeline (Qi et al., 2021).

The Stanza last version, 1.2, was released on 27 January 2021, and among other features, the following ones are the most relevant:

- Full neural network pipeline for robust text analytics, including tokenization, multi-word token (MWT) expansion, lemmatization, part-of-speech (POS) and morphological features tagging, dependency parsing, and named entity recognition.
- Pre-trained neural models supporting 66 (human) languages

Stanza supports various biomedical and clinical NER models pre-trained on the corresponding NER datasets (Figure 3).

Category	Corpus	Package Name	Supported Entity Types
Bio	AnatEM	anatem	ANATOMY
	BC5CDR	bc5cdr	CHEMICAL, DISEASE
	BC4CHEMD	bc4chemd	CHEMICAL
	BioNLP13CG	bionlp13cg	16 types in Cancer Genetics (* see below for a full list)
	JNLPBA	jnlpba	PROTEIN, DNA, RNA, CELL_LINE, CELL_TYPE
	Linnaeus	linnaeus	SPECIES
	NCBI-Disease	ncbi_disease	DISEASE
	S800	s800	SPECIES

Figure 3 – Stanza NER pre-trained models
Source: (https://stanfordnlp.github.io)

Category	Dataset	Domain	Stanza	BioBERT	scispaCy
Bio	AnatEM	Anatomy	**88.18**	–	84.14
	BC5CDR	Chemical, Disease	**88.08**	–	83.92
	BC4CHEMD	Chemical	89.65	**92.36**	84.55
	BioNLP13CG	16 types in Cancer Genetics	**84.34**	–	77.60
	JNLPBA	Protein, DNA, RNA, Cell line, Cell type	76.09	**77.49**	73.21
	Linnaeus	Species	88.27	**88.24**	81.74
	NCBI-Disease	Disease	87.49	**89.71**	81.65
	S800	Species	**76.35**	74.06	–

Figure 4 - NER performance comparison across different datasets in the biomedical domains[1]
Source: (Zhang et al., 2020)

To illustrate a practical case, the Stanza website provides a demo section to test its capabilities, and the following example text extracted from the SciSpaCy website has been used:

"Myeloid derived suppressor cells (MDSC) are immature myeloid cells with immuno suppressive activity. They accumulate in tumor-bearing mice and humans with different types of cancer, including hepatocellular carcinoma (HCC)"

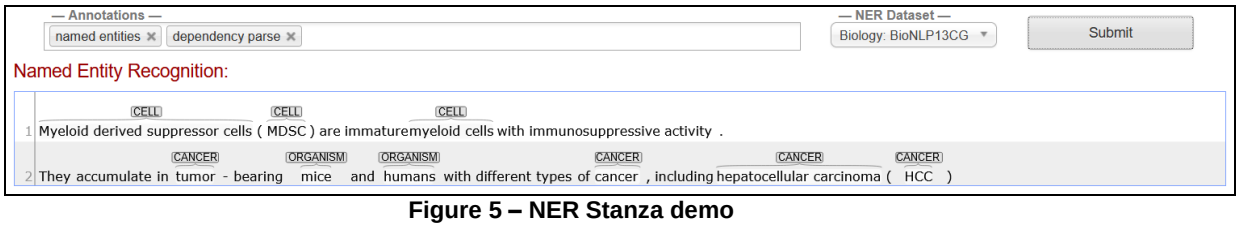

Figure 5 – NER Stanza demo
Source: (https://stanfordnlp.github.io)

Although it not very precise, as it can be seen, the NER has recognized entities as cells, organisms, and diseases. It has also considered that the result may be different depending on which NER dataset was selected.

[1] scispaCy results are from the scispacy-medium models reported in Neumann et al. (2019)

2.5.2. SpaCy

SpaCy is an open-source library for advanced NLP, which has released on 1 February 2021 version 3.0. The last update introduced state-of-the-art transformer-based pipelines.

Some of the most interesting features are:

- SpaCy has multi-task learning with pre-trained transformers, SOTA speed
- Custom pipes and models related to using SpaCy for scientific documents
- Support 64 languages

SpaCy also has its library specific for scientific documents called **SciSpaCy** (current version v0.4.0-14), a Python package containing SpaCy models for processing biomedical, scientific, or clinical text.

SpaCy allows the user to customize the tokenizer by adding tokenization rules on top of rule-based tokenizer and also includes a POS tagger and syntactic parser trained on biomedical data and an entity span detection model. There are others NER packages for more specific tasks that are listed in Figure 6.

Model	Description
en_core_sci_sm	A full spaCy pipeline for biomedical data with a ~100k vocabulary.
en_core_sci_md	A full spaCy pipeline for biomedical data with a ~360k vocabulary and 50k word vectors.
en_core_sci_lg	A full spaCy pipeline for biomedical data with a ~785k vocabulary and 600k word vectors.
en_core_sci_scibert	A full spaCy pipeline for biomedical data with a ~785k vocabulary and `allenai/scibert-base` as the transformer model. You may want to use a GPU with this model.
en_ner_craft_md	A spaCy NER model trained on the CRAFT corpus.
en_ner_jnlpba_md	A spaCy NER model trained on the JNLPBA corpus.
en_ner_bc5cdr_md	A spaCy NER model trained on the BC5CDR corpus.
en_ner_bionlp13cg_md	A spaCy NER model trained on the BIONLP13CG corpus.

Figure 6 - SpaCy pre-trained models in the biomedical field
(Source: https://spacy.io/)

Another SpaCy library relevant feature is the **EntityLinker** component which performs a string overlap-based search (char-3grams) on named entities, comparing them with the concepts in a knowledge base using an approximate nearest neighbors search (Honnibal, 2021).

According to SpaCy, their models achieve performance within 3% of published SOTA dependency parsers and 0.4% accuracy of SOTA biomedical POS taggers (Figure 7).

model	UAS	LAS	POS	Mentions (F1)	Web UAS
en_core_sci_sm	89.54	87.62	98.32	68.15	87.62
en_core_sci_md	89.61	87.77	98.56	69.64	88.05
en_core_sci_lg	89.63	87.81	98.56	69.61	88.08
en_core_sci_scibert	92.03	90.25	98.91	67.91	92.21

Figure 7 - Spacy performance on biomedical POS taggers
(Source: https://spacy.io/)

To illustrate a practical case, the following example that recognizes entities from a free text of the biomedical field has been extracted from the SciSpacy website. As it can be seen, the variable ***text*** contains several sentences which are tokenized, and finally, only the relevant entities are printed.

```
import scispacy
import spacy

nlp = spacy.load("en_core_sci_sm")
text = """
Myeloid derived suppressor cells (MDSC) are immature myeloid cells with
immunosuppressive activity. They accumulate in tumor-bearing mice and humans
with different types of cancer, including hepatocellular
carcinoma(HCC).
"""

doc = nlp(text)

print(doc.ents)
>>> (Myeloid derived suppressor cells,
     MDSC,
     immature,
     myeloid cells,
     immunosuppressive activity,
     accumulate,
     tumor-bearing mice,
     humans,
     cancer,
     hepatocellular carcinoma,
     HCC)
```

The library also allows the user to have a graphical perspective of the sentence and its structure.

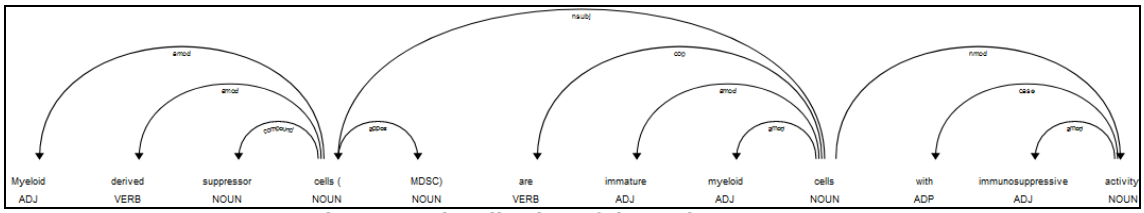

Figure 8 - Visualization of dependency parses
(Source: https://spacy.io/)

3. Design

The SciSpacy library has been chosen to test a NER model with data set of scientific papers from open-source because even the Spacy performance scores were apparently lower than Stanza, it has been just released a new version that according to its authors, it reaches the current State-of-the-Art. In addition, the creator and developer of Spacy, Allen Institute for AI (AI2), is also collaborating with the data set that will be used to test the NER model.

3.1. COVID-19 Open Research Dataset (CORD-19)

To put in place an operational test with the SciSpacy library, it has been chosen the Covid-19 Open Research Dataset, also known as CORD-19, build by the White House and a coalition with Allen Institute for AI (AI2) and other leading research groups in response to the COVID-19 pandemic. The first release was on March 2020, containing 28.000 papers about COVID-19, SARS-CoV-2, and up to today, the set includes over 500.000 scholarly articles about and related to coronaviruses (Lu Wang et al., 2020).

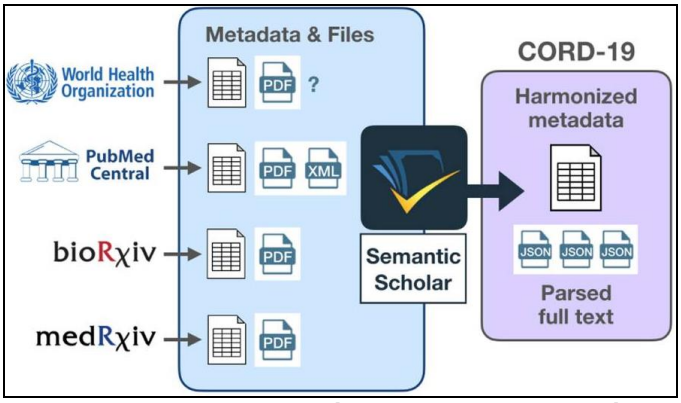

Figure 9 - CORD-19 sources (Source: Lu Wang et al., 2020)

The data set was downloaded from Kaggle[2] on 21/03/2021 with 11.7 GB, containing all the compiled papers and a summary file with 490.904 metadata rows as it is shown below (bold values will be the fields used during the extraction process):

- **CORD_UID:** Unique key for all the CORD-19 dataset.
- SHA
- Source
- **Title:** topic of the paper.
- **DOI number:** Digital Object Identifier
- **PMCID:** PubMed Central unique identifier
- License
- **Abstract:** brief summary of the paper.
- **Publication date:**
- Authors
- Journal
- Mag_id
- Who_covidence_id
- Arxiv_id
- Pdf_json_files

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

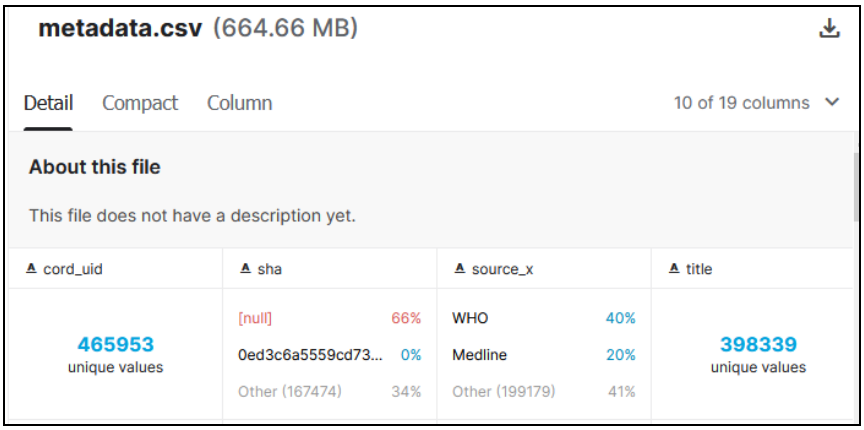

Figure 10 - CORD-19 metadata description

As shown in Figure 11, there has been a very small quantity of publications until the spike of 2020. Most of the papers published from 1962 to today were originated from the following sources as well a collection of hand-curated articles about Covid-19.

- Paper sources
- ArXiv
- BioRxiv
- Elsevier
- Medline
- MedRxiv
- PubMed Central (PMC)
- World Health Organization (WHO)

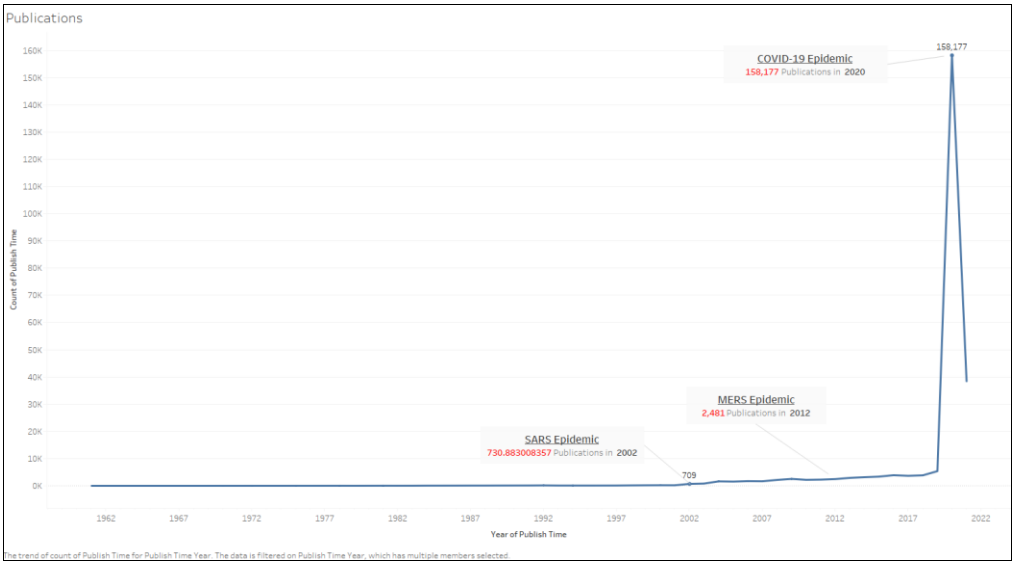

Figure 11 - Publications linked to COVID-19

3.2. Testing and development environment

The used resources to test and develop the project are listed in the following subsections.

3.2.1. Hardware

The project report, developing, and programming part has been drafted with a Laptop Dell with Intel processor i5-7200 and 8GB RAM. However, the pre-processing part has not been powerful enough in computing terms due to the testing processes were executed for more than 48 hours without finishing the execution.

Therefore, the online platform Kaggle which provides a free remote computational environment has been applied to perform the data pre-processing, modeling, and entity extraction.

Kaggle notebook runs in a remote computational environment and provides the following hardware:

Environment	**Local**	**Remote**[3]
Hardware	Intel Core i5-7200 CPU @ 2.50 GHZ 8GM RAM 115 GB HD	4 CPU Cores Intel(R) Xeon(R) CPU @ 2.20GHz 16 GB RAM 20GB HD **GPU** 2 CPU Cores 13 Gb RAM **TPU** 4 CPU Cores 16 GB RAM
Operating system	Windows 10 64 bits	Linux version 5.4.104+

Table 1 - Hardware specifications

3.2.2. Software

As a software, it has mainly used the following ones:

- **Jupyter Notebook 6.1.4** (Anaconda 3): to program and test the NER model
- **Tableau**: to apply visual analysis
- **Gephi**: to explore connections between entities as a graphs using social network analysis

[3] https://www.kaggle.com/docs/notebooks#technical-specifications

4. Implementation

The current section covers the programming part of the project.

4.1. NER Pseudocode

1. Clean outdated libraries already installed by default and install the latest versions
2. Import Spacy, SciSpacy, and four NER libraries (BC5CDR, BIONLP13CG, CRAFT, and JNLPBA)
3. Read the metadata file from CORD-19 that contains the paper abstracts.
4. Define a function to call the NER models, which take two arguments (model, abstractList), and it returns a table with the CORD_UID, DOI number, PMCID reference, extracted entity, its classification, and the title.
5. Each model calls the function, and it generates a CSV file with the extracted entities and their records

4.2. NER Source Code

As mentioned in section 3.2.2, at first instance, Jupyter Notebook is the programming environment used with Python 3 language. Once the code was working correctly with a small demo of the data, it was exported to the Kaggle platform to run the whole script.

Although Kaggle lease to the users such power processing, the user cannot run a kernel for more than 9 hours in a row. Due to that fact, each NER model from SciSpacy was running independently because the total time was more than the allowed one, as it can be seen below the time to execute the already trained models:

- **BIONLP13CG** 15464.951 seconds (4 hours 29 minutes)
- **BC5CDR** 15816.159 seconds (4 hours 39 minutes)
- **CRAFT** 16174.281 seconds (4 hours 49 minutes)
- **JNLPBA** 12717.181 seconds (3 hours 53 minutes)

The following screenshots were taken from the Kaggle platform (the source code is attached to the current delivery as "*Kaggle_notebook_SciSpacy_CORD-19.ipynb*":

```
# Uninstall old libraries
!pip uninstall -y en-core-web-sm
!pip uninstall -y en-core-web-lg 2.3.1
!pip uninstall -y allennlp 2.0.1

# Install Spacy and SciSpacy NLP libraries
!pip install spacy
!pip install scispacy

# Install core modules
!pip install https://s3-us-west-2.amazonaws.com/ai2-s2-scispacy/releases/v0.4.0/en_core_sci_sm-0.4.0.tar.gz
!pip install https://s3-us-west-2.amazonaws.com/ai2-s2-scispacy/releases/v0.4.0/en_core_sci_lg-0.4.0.tar.gz

# Install NER trained models
!pip install https://s3-us-west-2.amazonaws.com/ai2-s2-scispacy/releases/v0.4.0/en_ner_bc5cdr_md-0.4.0.tar.gz
!pip install https://s3-us-west-2.amazonaws.com/ai2-s2-scispacy/releases/v0.4.0/en_ner_bionlp13cg_md-0.4.0.tar.gz
!pip install https://s3-us-west-2.amazonaws.com/ai2-s2-scispacy/releases/v0.4.0/en_ner_craft_md-0.4.0.tar.gz
!pip install https://s3-us-west-2.amazonaws.com/ai2-s2-scispacy/releases/v0.4.0/en_ner_jnlpba_md-0.4.0.tar.gz
```

```
# import libraries
import scispacy
import spacy
from spacy import displacy

import pandas as pd
import time

# Import core modules
import en_core_sci_lg
import en_core_sci_sm

# Import trained models
import en_ner_bc5cdr_md
import en_ner_bionlp13cg_md
import en_ner_craft_md
import en_ner_jnlpba_md


# Timer to control running task time
start = time.time()

#  Read and load the data in a DataFrame
CORD19_df = pd.read_csv('../input/cord19/metadata.csv')
#CORD19_df = pd.read_csv('../input/cord19sample/cord_sample.csv')

# Drop N/A values
df = CORD19_df.dropna(subset=['abstract'])

end = time.time()
print("Time to read the file",round(end - start,3)," seconds")
```

```
#Load NER models
nlp_bc5cdr = en_ner_bc5cdr_md.load()
nlp_bionlp13cg = en_ner_bionlp13cg_md.load()
nlp_craft = en_ner_craft_md.load()
nlp_jnlpba = en_ner_jnlpba_md.load()

# Function to extract entities depending on the model which can be bc5cdr, bionlp13cg, craft or jnlpba

#def ner(model, abstractList, doiList, titleList):
def ner(model, abstractList):
    catalogue= {'CORD_UID':[],'DOI':[], 'PMCID':[], 'Entity':[], 'Class':[],'Title':[]}
    i = 0

    if model == 'bc5cdr':
        for docs in nlp_bc5cdr.pipe(abstractList):
            doi = doiList[i]
            title = titleList[i]
            pmcid = pmcidList[i]
            cordUid = cordUidList[i]
            for doc in docs.ents:
              catalogue["CORD_UID"].append(cordUid)
              catalogue["DOI"].append(doi)
              catalogue["PMCID"].append(pmcid)
              catalogue["Entity"].append(doc.text)
              catalogue["Class"].append(doc.label_)
              catalogue["Title"].append(title)
            i +=1
        return catalogue

    if model == 'bionlp13cg':
        for docs in nlp_bionlp13cg.pipe(abstractList):
            doi = doiList[i]
            title = titleList[i]
            pmcid = pmcidList[i]
            cordUid = cordUidList[i]
            for doc in docs.ents:
              catalogue["CORD_UID"].append(cordUid)
              catalogue["DOI"].append(doi)
              catalogue["PMCID"].append(pmcid)
              catalogue["Entity"].append(doc.text)
              catalogue["Class"].append(doc.label_)
              catalogue["Title"].append(title)
            i +=1
        return catalogue

    if model == 'craft':
        for docs in nlp_craft.pipe(abstractList):
            doi = doiList[i]
            title = titleList[i]
            pmcid = pmcidList[i]
            cordUid = cordUidList[i]
            for doc in docs.ents:
              catalogue["CORD_UID"].append(cordUid)
              catalogue["DOI"].append(doi)
              catalogue["PMCID"].append(pmcid)
              catalogue["Entity"].append(doc.text)
              catalogue["Class"].append(doc.label_)
              catalogue["Title"].append(title)
            i +=1
        return catalogue
```

```
    if model == 'jnlpba':
        for docs in nlp_jnlpba.pipe(abstractList):
            doi = doiList[i]
            title = titleList[i]
            pmcid = pmcidList[i]
            cordUid = cordUidList[i]
            for doc in docs.ents:
              catalogue["CORD_UID"].append(cordUid)
              catalogue["DOI"].append(doi)
              catalogue["PMCID"].append(pmcid)
              catalogue["Entity"].append(doc.text)
              catalogue["Class"].append(doc.label_)
              catalogue["Title"].append(title)
            i +=1
        return catalogue


# Lists
abstractList = df['abstract'].tolist()
cordUidList = df['cord_uid'].tolist()
doiList = df['doi'].tolist()
pmcidList = df['pmcid'].tolist()
titleList = df['title'].tolist()


# BC5CDR model
BC5CDR_start = time.time()
print('BC5CDR running')
# Execution
model = 'bc5cdr'
table_bc5cdr = ner(model,abstractList)
trans_df = pd.DataFrame(table_bc5cdr)
trans_df.to_csv ("bc5cdr_entities.csv", index=False)

BC5CDR_end = time.time()
print("Time to execute the execute mode bc5cdr",round(BC5CDR_end - BC5CDR_start,3)," seconds")

# BIONLP13CG model
bionlp13cg_start = time.time()
print('BIONLP13CG running')
# Execution
model = 'bionlp13cg'
table_bionlp13cg = ner(model,abstractList)
trans_df = pd.DataFrame(table_bionlp13cg)
trans_df.to_csv ("bionlp13cg_entities.csv", index=False)

bionlp13cg_end = time.time()
print("Time to execute the execute model bionlp13cg",round(bionlp13cg_end - bionlp13cg_start,3)," seconds")
```

```
# CRAFT model
CRAFT_start = time.time()
print('CRAFT running')
# Execution
model = 'craft'
table_craft =  ner(model,abstractList)
trans_df = pd.DataFrame(table_craft)
trans_df.to_csv ("craft_entities.csv", index=False)

CRAFT_end = time.time()
print("Time to execute the execute model craft",round(CRAFT_end - CRAFT_start,3)," seconds")
```

```
# JNLPBA model
JNLPBA_start = time.time()
print('JLNPBA running')
# Execution
model = 'jnlpba'
table_jnlpba = ner(model,abstractList, doiList,titleList)
trans_df = pd.DataFrame(table_jnlpba)
trans_df.to_csv ("jnlpba_entities.csv", index=False)

JNLPBA_end = time.time()
print("Time to execute the execute model jnlpba",round(JNLPBA_end - JNLPBA_start,3)," seconds")
```

4.3 WordCloud Source Code

A “Word Cloud” is a visual representation from a bag of words highlighting the keywords with more relevance from a set of strings because they appear more frequently.

Based on the output generated with the SciSpacy library, it has been drafted Word Clouds from the extracted entities from the abstract of the papers.

The following screenshots were taken from the Jupyter Notebook (the original source-code is attached to the current delivery as “*WordCloud_generator.ipynb*”:

```
import pandas as pd
import re
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS

# read extracted entities
df = pd.read_csv('bc5cdr_entities.csv')
STOPWORDS.add('®')
stopwords = set(STOPWORDS)
bag_of_words = ''

# loop csv file
for entity in df.Entity:
    entity = str(entity)
    if entity=='' or entity =='nan':
        #print("skipped")
        z=0
    else:
        tokens = entity.split()
        for i in range(len(tokens)):
            tokens[i] = tokens[i].lower()
        bag_of_words += " ".join(tokens)+" "
wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                collocations=False,
                min_font_size = 10).generate(bag_of_words)

# plot WordCloud
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
plt.show()
```

5. NER Outcomes

From 490.904 of the papers, a total of 7.452.816 entities were extracted consisting of 1.324.184 unique terms (**BC5CDR** 267.625 + **BioNLP13CG** 606.512 + **CRAFT** 124.331 + **JNLPBA** 325.716).

The following subsections will represent those entities in different ways to summarize the outcomes.

5.1. NER Model Figures

The application of the SciSpacy library to recognize and extract entities based on pre-trained models from the abstracts of a collection of papers of COVID-19 has generated four CSV files, one per set, resulting in the below figures from Table 2:

Model	Class	Extracted Entities	Total
BC5CDR	Chemical	555.717	**1.329.843**
	Disease	774.126	
BioNLP13CG	Amino acid	11.754	**2.467.152**
	Anatomical system	4.916	
	Cancer	100.603	
	Cell	147.252	
	Cellular component	57.395	
	Developing anatomical structure	439	
	Gene or Gene Product	932.690	
	Immaterial anatomic entity	13.216	
	Multi Tissue Structure	41.501	
	Pathological formation	33.763	
	Organ	134.264	
	Organism	681.265	
	Organism Subdivision	18.366	
	Organism Substance	44.745	
	Simple Chemical	215.571	
	Tissue	29.412	
CRAFT	CHEBI	498.851	**2.265.170**
	CL	214.272	
	GGP	287.649	
	GO	140.917	
	SO	414.952	
	TAXON	708.529	
JNLPBA	Cell type	99.352	**1.390.651**
	Cell line	38.953	
	DNA	298.684	
	Protein	994.796	
	RNA	8.866	
		Total	**7.452.816**

Table 2 - Summary of extracted entities by model

5.2. Visual representations

5.2.1. Tree Map

The following tree map drafted with *Tableau* summarize how is the distribution of the recognized entities by class for each model applied.

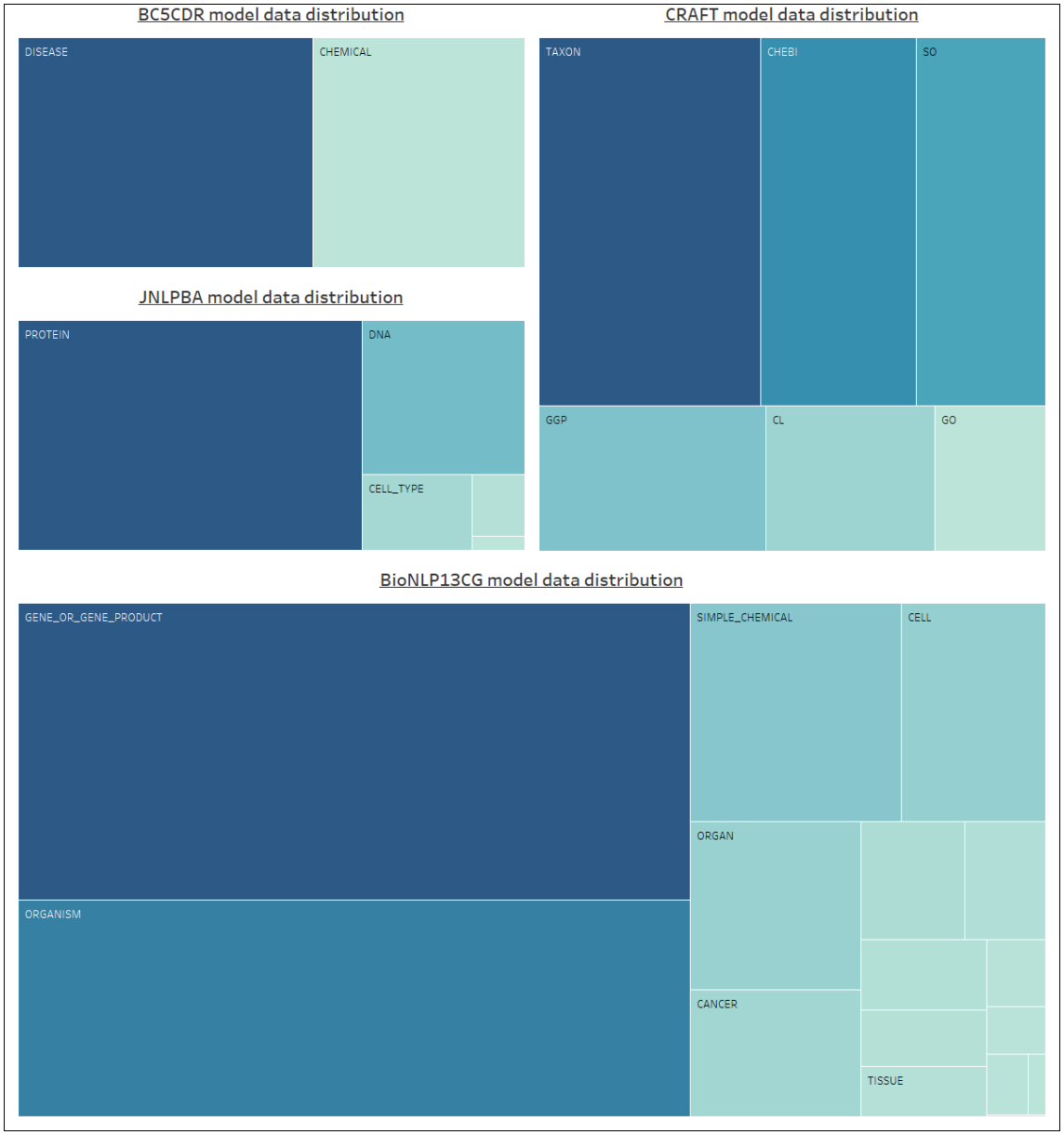

Figure 12 - Visual summary of extracted entities by model

5.2.2. Bubble Chart

The bubble chart has been drafted also with Tableau, and it has been selected to represent the extracted entities in three dimensions (Entity itself, Entity classification and the volume of the entity). The bubble charts are dynamics and can be accessed by clicking here.

For optimal visualization reasons, keywords such “COVID-19” and “patients” have been excluded because they are likely to be present in all the papers Otherwise, it will provide a bias of the importance of the rest of the entities and will not allow seeing the significance of others entities. Other keywords less relevant such “people”, “individuals” or “country” has been also excluded.

BC5CDR

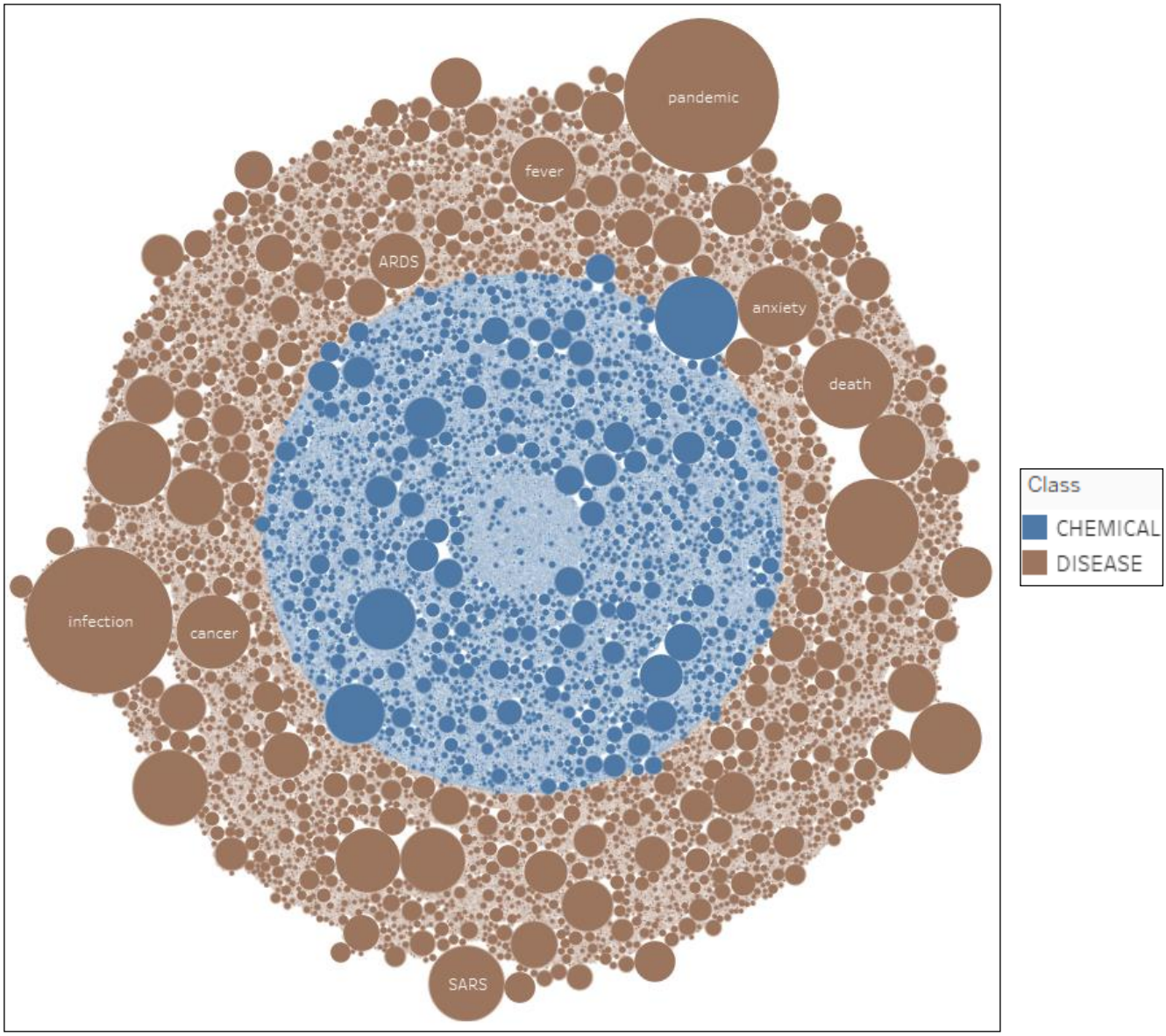

Figure 13 – BC5CDR Bubble chart

BioNLP13CG

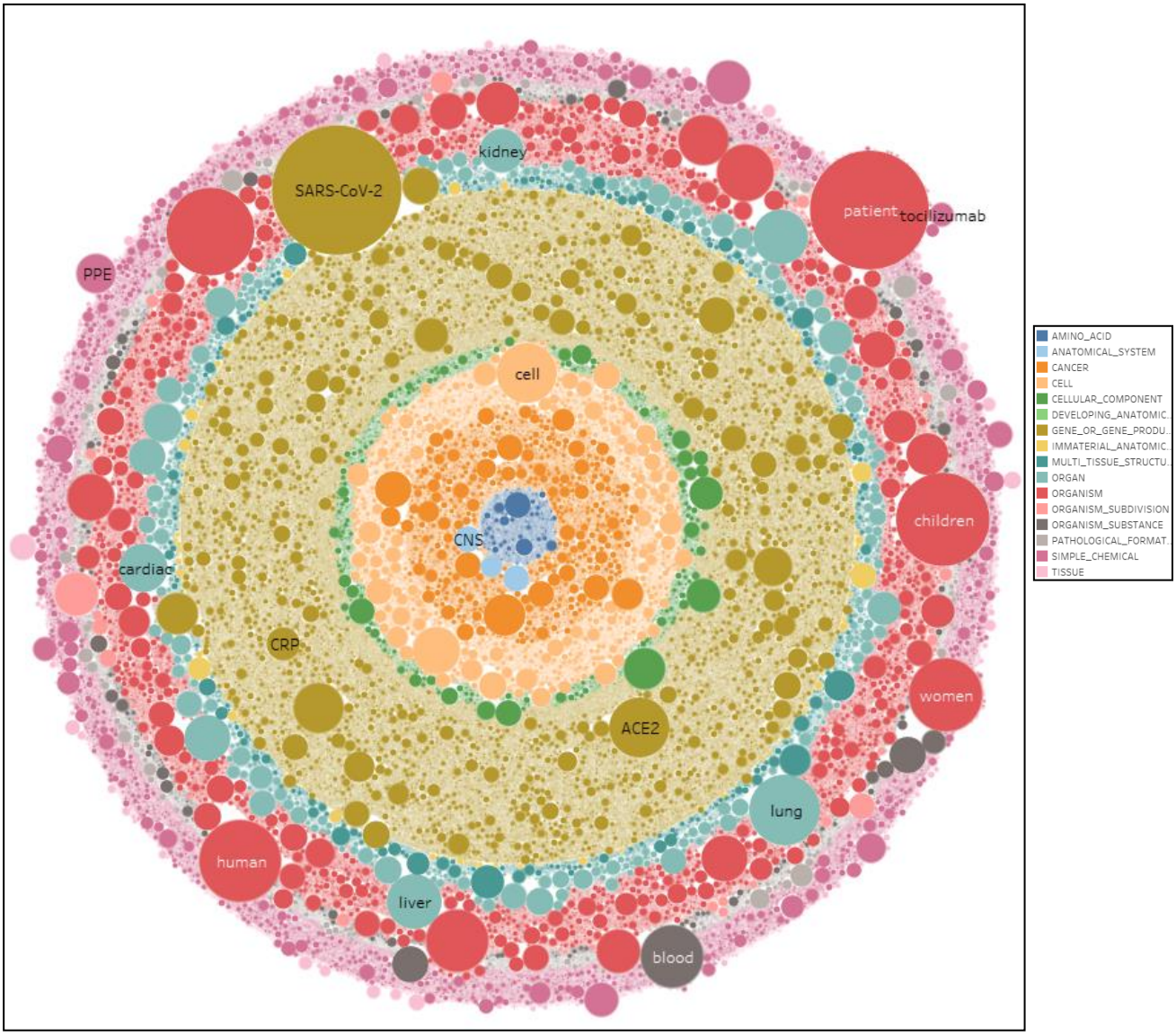

Figure 14 – BioNLP13CG Bubble chart

CRAFT

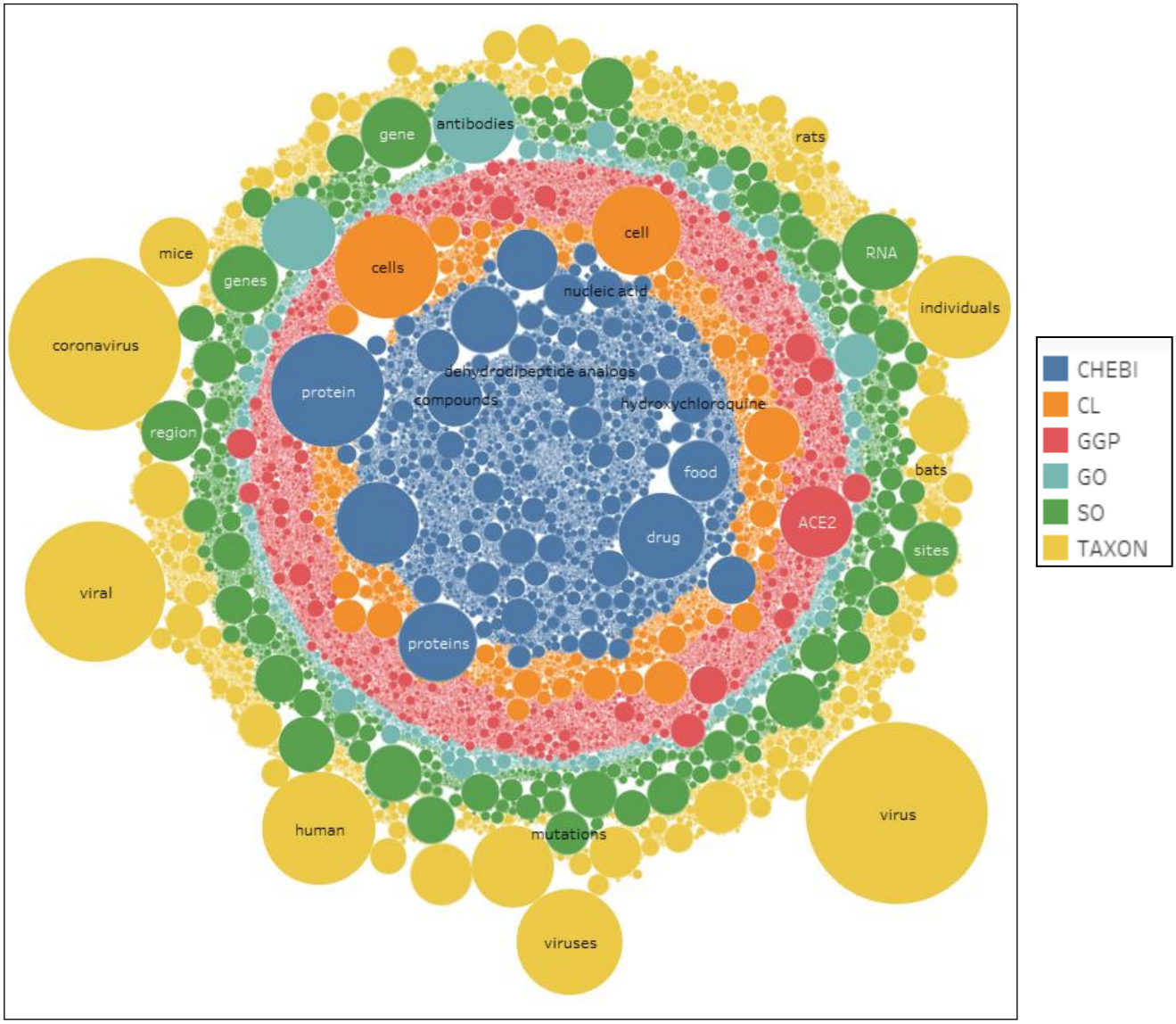

Figure 15 – CRAFT Bubble chart

JNLPBA

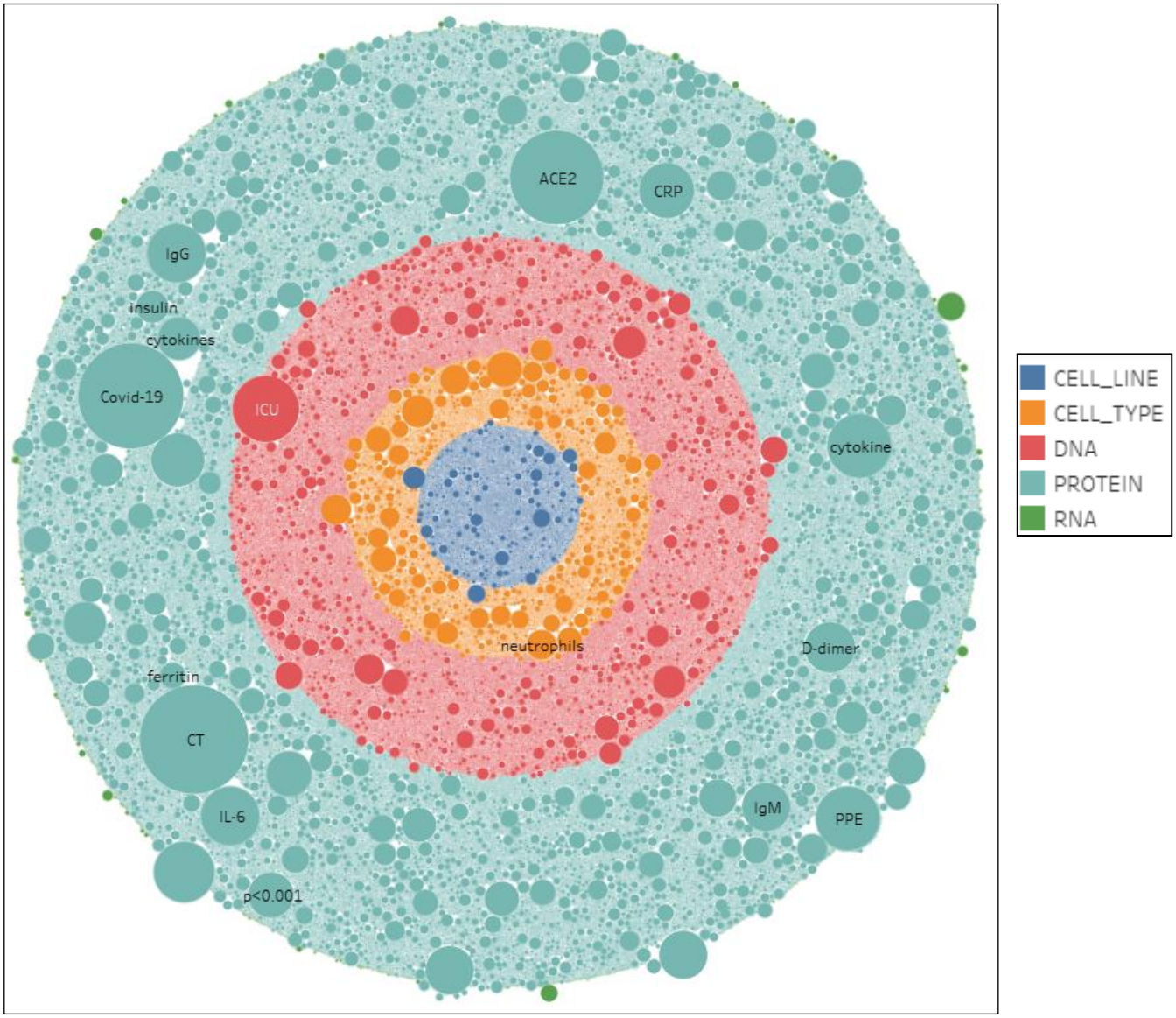

Figure 16 – JNLPBA Bubble chart

5.2.3. WordCloud

As the entities extracted by each model are different and contains specific kind of entities, the generated word cloud is also different as it can be seen below and it may provide to the user a visual summary of the abstracts:

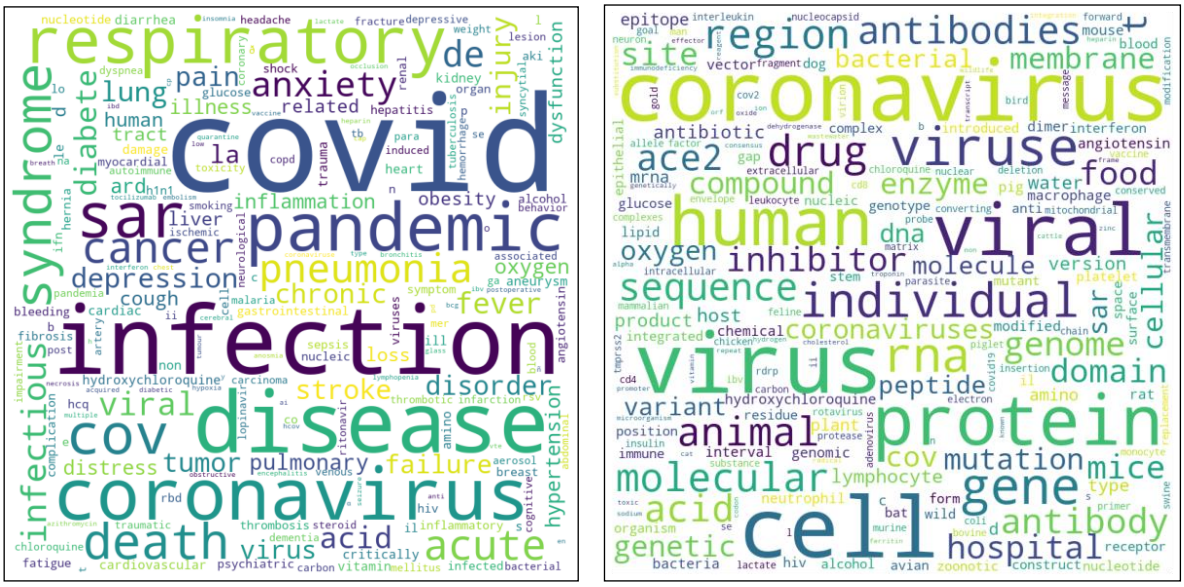

Figure 17 - BC5CDR (left) and BioNLP13CG (right) WordCloud

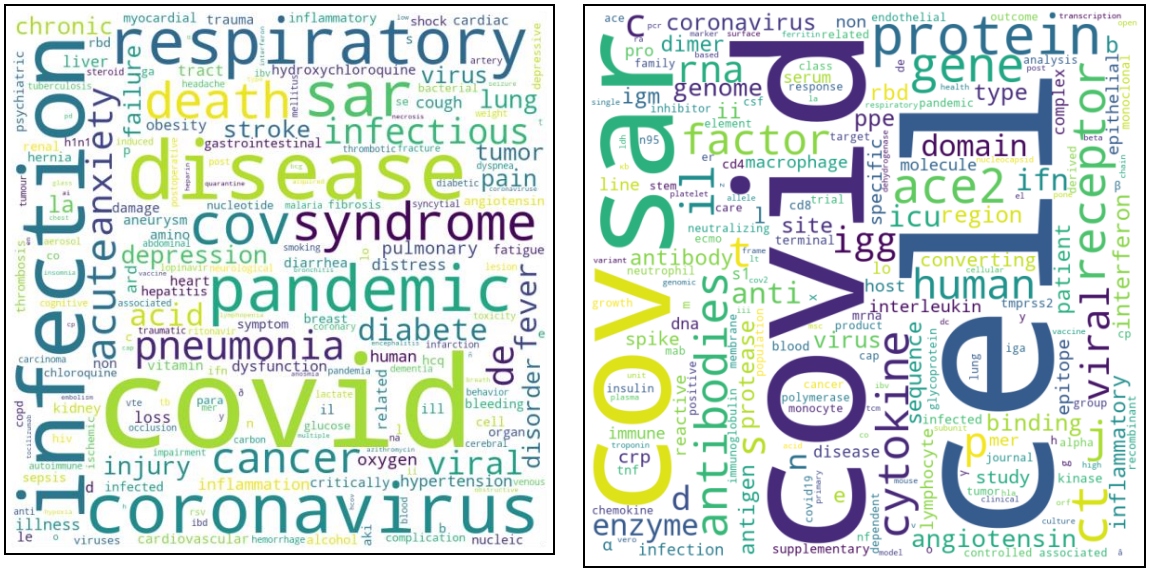

Figure 18 - CRAFT (left) and JNLPBA (right) WordCloud

5.2.4. Gephi

Gephi has been select to visualize the graphical connection between the extracted entities and their weight. To be able to import an edge table, it requires to assign at least as a node the “Source” and the “Target” and the “label” as the edge. In the project case, it has assigned:

- “CORD_UID” as a “Source”
- “Entity” as a “Target”
- “Class” as a “label.”

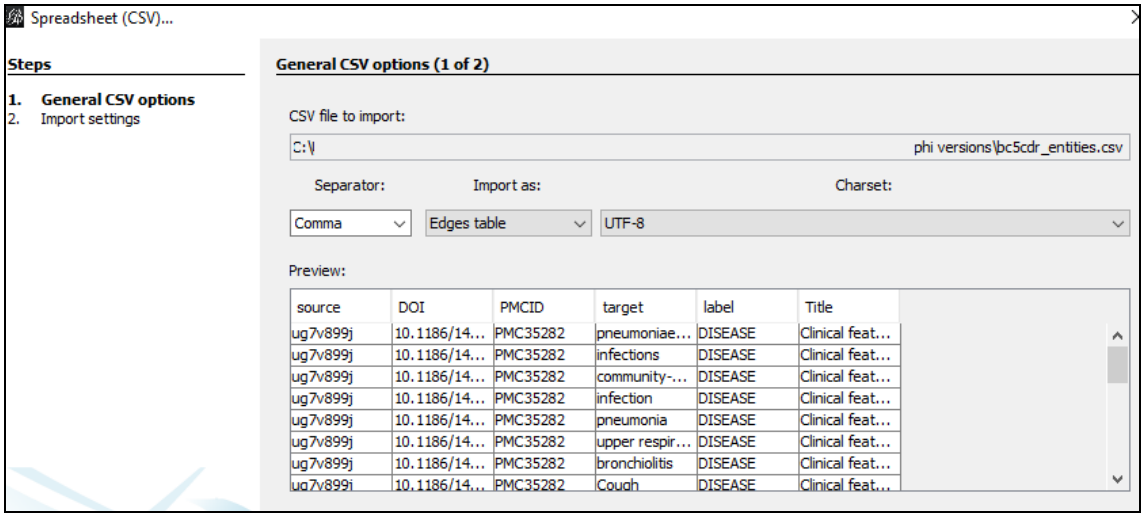

Figure 19 – Gephi import wizard

CRAFT

The CRAFT dataset recognizes entity types such as CHEBI, TAXON, CL, GGP, GO, and SO. Once the list of the entities recognized was imported and extracted by the SciSpacy library, it may be seen that 139.286 nodes and 412.978 were created. Also, the modularity class useful to detect communities currently has found 1.092.

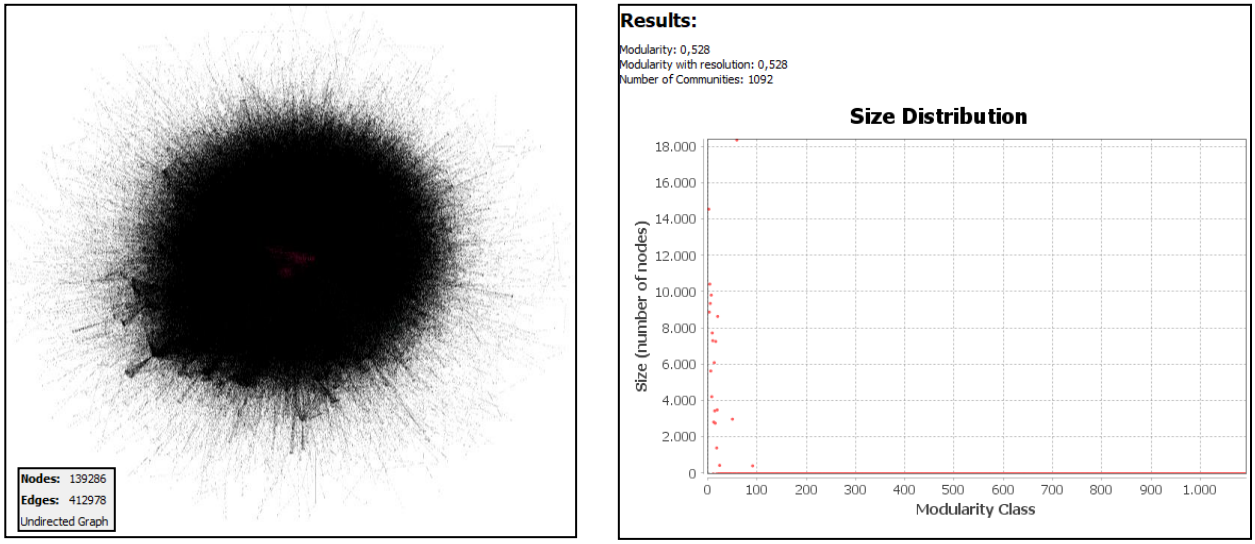

Figure 20 - CRAFT Network connections without filters (left) and CRAFT Moduliaryt Class

As it is not doable to analyze such huge figures because of the size, it needs to be escalated by applying filters the following filters:

- Select the Giant component.
- Explore the node and edges weights and then isolate the nodes less relevant. In this case, the top values were assigned to keywords such as “COVID-19” among others, so excluding them, it was possible to reduce the graph to 6 communities by filtering the weight of the edges from 9 to 24 degrees. The output was 101 nodes (0,07% of the total) with 100 edges (0,02% of the total).

Therefore, the most mentioned entities within all the papers are: “virus”, “viruses”, “food,” and “animals”.

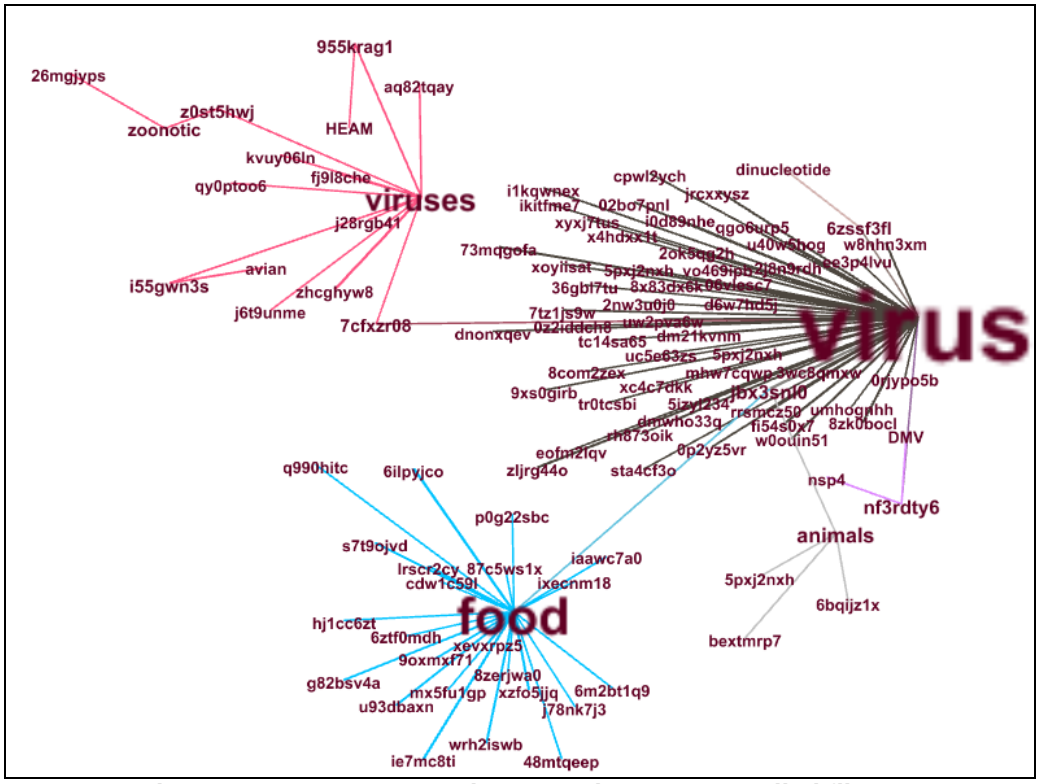

Figure 21 - CRAFT Network connections once applied filters

JNLPBA

JNLPBA model recognizes entities types such as CELL LINE, CELL TYPE, DNA, PROTEIN, and RNA. Once the list of the entities recognized was imported and extracted by the SciSpacy library, it may be seen that 196.396 nodes and 264.728 were created. Also, the modularity class useful to detect communities currently has found 1.092.

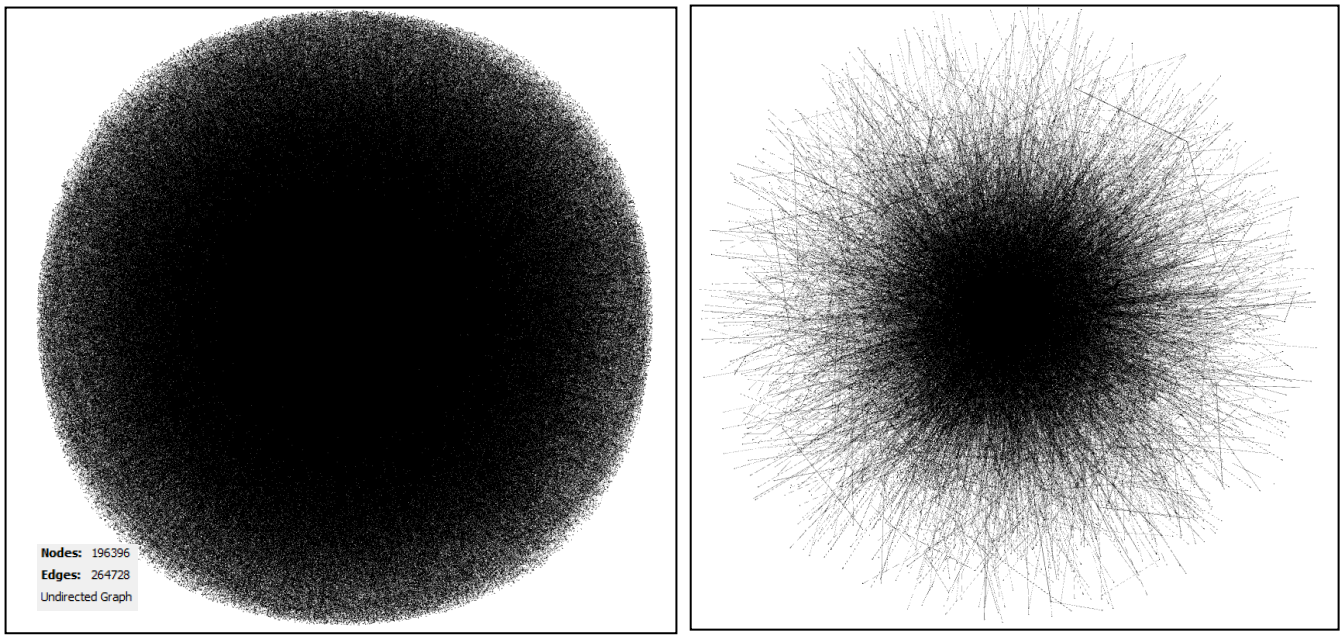

Figure 22 - JNLPBA Network connections without filters (left) and JNLPBA graph partially filtered

Reaching this point of the project, it was not possible to continue the post-processing analysis due to the lack of time and constant technical issues to analyze the networks (See 7. Limitations).

5.2.5. Other considerations

During the exploration of the generated results, after cleaning processes has applied to the data, there was still noise within the entities names because several entities were classified wrongly or the same entity was classified in more than one category.

For instance, in the JNLPBA model, the “COVID-19” entity is mainly classified as a Protein but also as a DNA. Also, in BC5CDR model, it behaves strangely with non-Latin words such “吃亏是福”, which has been classified as a chemical component, but it is a Chinese proverb that means “Disadvantage is a blessing”. Moreover, “Western site” was classified as a DNA, or DOI reference was classified as a CHEMICAL entity.

Therefore, to evaluate the accuracy of the model, several assessments should be done to evaluate each case.

6. Limitations

The limitations identified during the development are listed below, starting from the ones that have impacted the project highly.

- **Computing power limitation:** The first and most constraint that the project comes across from the beginning to the end of the project has been definitely the computing power limitation and the big data sets (Gb of data and more than 7 million entities to deal with).

  Although the pre-processing, initial evaluation of data and processing of the NLP model have been done with the Kaggle platform, which offers the user good enough hardware to perform high demanding tasks, it has a limitation of 9 hours in a row.

  For instance, the time restriction did not allow to use of NetworkX as planned initially to apply social network analysis to the extracted entities from the CORD-19 dataset. Another case was the NLP processing, which to complete the four models (BC5CDR, BioNLP13CG, CRAFT, and JNLPBA) to process the CORD-19 dataset, it needed four hours for each model. The solution was to split the code and run each model separately.

  The technical issues mentioned above cost the project much more time than expected and highly impacted the quality and final product.

  Due to NetworkX was timing out while trying to draw a graph chart and after testing several days on a row, other alternatives were sought to run a social network analysis. The first one was to Neo4j, which does not yet have an operational (and cost-free) cloud service, and after installed locally, it was not possible to load several millions of rows.

  That lead to trying to load and analyze the data with Gephi, although it allowed initially to load the data, it was very slow to apply filters and was crashing very frequently.

- **The significant data volume:** The initial starting point was over 500.000 abstracts of scientific papers, and every transformation of the data increased the volume of the entities and the size files, resulting in a high computational cost for each further step.

- **Lack of medical domain knowledge:** My first contact with **the** Biomedicine field has been in this project, and only the complexity of the vocabulary and the technical issues they are facing have also been a challenge**.**

- **Lack of NLP training:** I have learned on the fly from the basic concepts of the NLP to the application of the Spacy model against a real dataset.

- **Project scope:** The initial project scope set was to use an NLP library to extract entities from a dataset, but once achieved the goal it appeared the need to analyze and represent the result, which in my opinion, that is likely to be carried out as another project.

- **Multiple wording for the same concept:** The most obvious example is the COVID keyword as it has several ways to be written: COVID, COVID-19, coronavirus, SARS-CoV-2.

7. Conclusions

The starting point of this project was that “biomedical research is drowning in data, yet starving, for knowledge.” Carrying out all the steps to process over 500.000 papers (that were already collected and saved me time) has allowed me to understand why the scientific community is struggling and experience the meaning of the quote.

The data processing could not be possible done without the support of the Kaggle platform, and even so, it has also found severe limitations to finish the task. In addition, trying to apply social network analysis has also been a real challenge that finished without success and therefore, the post-processing results did not lead to strong conclusions.

Regarding the project planning, it has been possible to stick to the schedule only in the initial phases, where the goals were to collect information. Once reached the technical stages, the planning has been continuously modified to achieve the objectives because unexpected additional workload and continuous technical challenges has not been enough to conduct an optimal post-processing analysis, although it is also a future investigation line for other projects.

Project scope wise, initially was set to identify entities such as genes, tissues, and cells, but later was extended to others such as DNA and diseases.

In general, and taking into account that the topic of the project is the NLP, the primary goals established were achieved, and overall, the project has not only allowed me to learn an exciting and useful discipline of Data Science, but I have also applied a variety of knowledge acquire during the Master's.

8. Glossary

BERT - Bidirectional Encoder Representations from Transformers
BioNLP – Biomedical Natural Language Processing
CNN - Convolution Neural Networks (CNN)
CORD – 19 - Covid-19 Open Research Dataset
LSTM - Long Short-Term Memory
ML – Machine Learning
NLP – Natural Language Processing
NER – Name entity Recognition
World Health Organization (WHO)

9. Bibliography

Atanassova, I., Bertin, M., & Mayr, P. (2019). Mining Scientific Papers: NLP-enchanced Bibliometrics. *Frontiers in Research Metrics and Analytics 4*.

Bada, M., Hunter, L. E., Eckert, M., & Palmer, M. (2010). *An Overview of the CRAFT Concept Annotation Guidelines.* Colorado: University of Colorado.

Bansal, T., Verga, P., Choudhary, N., & Andrew, M. (2020). *Simultaneously Linking Entities and Extracting: Relations from Biomedical Text without Mention-Level Supervision.* Massachusetts: University of Massachusetts.

BioCreAtIvE. (21 / 02 / 2021). Recollit de BioCreAtIvE: https://biocreative.bioinformatics.udel.edu/

C-H, W., H-Y, K., & Z., L. (2015). GNormPlus: An Integrative Approach for Tagging Gene, Gene Family and Protein Domain. Text Mining for Translational Bioinformatics special issue. *BioMed Research International Journal*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (24 / May / 2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* Google AI.

Harris, Z. S. (1982). *A grammar of English on mathematical principles.* New York: Wiley.

Holzinger, A., & J. I. (2014). Knowledge Discovery and Data Mining in Biomedical Informatics: The Future Is in Integrative, Interactive Machine Learning Solutions. Berlin: Springer.

Honnibal, M. (2021). *SpaCy*. Recollit de SpaCy: https://spacy.io/

Kamath, S., & Wagh, R. (2017). Named Entity Recognition Approaches and Challenges. *International Journal of Advanced Research in Computer and Communication Engineering*.

Kim, J.-D., Ohta, T., Tsuruoka, Y., & Tateisi, Y. (2003). *Introduction to the Bio-Entity Recognition Task at JNLPBA.* Tokyo: University of Tokyo.

Kocaman, V., & Talby, D. (2020). *Biomedical Named Entity Recognition at Scale.* Lewes: John Snow Labs Inc.

Kroll, H., Pirklbauer, J., Ruthmann, J., & Balke, W.-T. (2020). *A Semantically Enriched Dataset based on Biomedical NER for the COVID19 Open Research Dataset Challenge.* Braunschweig, Germany: Institute for Information Systems TU Braunschweig.

Leaman, R., & Lu, Z. (2016). TaggerOne: joint named entity recognition and normalization with semi-Markov Models. A *Bioinformatics* (p. 2839–2846). Oxford Academic.

Li, J., Sun, Y., Johnson, R. J., Sciaky, D., Wei, C.-H., Leaman, R., . . . Lu, Z. (2016). BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Oxford Journal*.

Liu, S., Tang, B., Chen, Q., & Wang, X. (2015). Effects of semantic features on machine learning-based drug. *Information*, 848–865.

López, J. M. (2021). *Introducció al deep learning aplicat al processament del llenguatge natural.* Barcelona: Universitat Oberta de Catalunya.

Lu Wang, L. L. (2020). CORD-19: The Covid-19 Open Research Dataset.

Mitsumori, T., Fation, S., & Murata, M. e. (2005). *Gene/protein name recognition based on support vector machine using dictionary as features.* Japan: BMC Bioinformatics.

Neumann, M., King, D., Beltagy, I., & Ammar, W. (2019). ScispaCy: Fast and robust models for biomedical natural language processing. *Proceedings of the 18th BioNLP Workshop and Shared Task*.

Project NLTK. (13 / April / 2020). *Natural Language Toolkit*. Recollit de Natural Language Toolkit: https://www.nltk.org/

Pyysalo, S., Ohta, T., & R. R. (2015). Overview of the Cancer Genetics and Pathway Curation tasks of BioNLP Shared Task 2013. *BMC Bioinformatics*.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & D. Manning., C. (2021). Stanza: A Python Natural Language Processing Toolkit. *Stanford University*. Recollit de https://stanfordnlp.github.io/stanza/

Sager, N., Friedman, C., & Lyman, M. S. (1987). *Medical Language Processing: Computer Management of Narrative Data.* Massachusetts: Addison-Wesley, Reading, MA.

Tsai, R., Wu, S., & Chou, W. (2006). Various criteria in the evaluation of biomedical named entity recognition. *BMC Bioinformatics*.

UnitProt. (23 / 02 / 2021). Recollit de UnitProt: https://www.uniprot.org/

Varma, V., Damani, S., Damani, S., & Narahari, K. N. (2020). *Compression of Deep Learning Models for NLP.*

Weber, L., Münchmeyer, J., Rocktäschel, T., Habibi, M., & Leser, U. (2020). HUNER: improving biomedical NER with pretraining. A *Bioinformatics* (p. 295–302). Oxford University Press.

Zhang, Y., Zhang, Y., & Qi, P. (2020). *Biomedical and Clinical English Model Packages in the Stanza Python NLP Library.* Standford: Standford University.

Zhang, Z. (2013). *NAMED ENTITY RECOGNITION CHALLENGES IN DOCUMENT ANNOTATION, GAZETTEER CONSTRUCTION AND DISAMBIGUATION.* Sheffield: University of Sheffield.