Université de Paris

# PPD

## Fouille de texte pour l'exploration des facteurs de sévérité du COVID-19

**Proposé par :**

**Séverine Affeldt & Lazhar Labiod**

**Réalisé par :**

- **Mohammed Erifai MAAMIR - MLDS**
- **Kamel MESBAHI -MLDS**
- **Ryad lotfi MAHTAL - MLDS**
- **Ahmed Seyfeddine GOUMEIDA - MLDS**

# Table des matières :

**Table des matières :** 2

**Introduction** 3

**Motivations** 4

**Objectifs** 6

**Rudiments & Etat de l'art** 6

**Chargement de données** 7

Collecte de données 8

Approche 1 8

Approche 2 9

Approche 3 9

**Prétraitement de données** 9

Suppression des valeurs null 9

Multi-langues traitement 10

Histogramme des langues d'articles. 10

Stopwords 10

Ponctuations 11

Lowercase 11

Lemmatization 11

Liens et HTML tags 11

Frequent words 12

Rare words 12

Nombre de 'mots' / 'mots uniques' par articles 12

**Vectorization** 13

**CoClustering & Entités nommées** 14

CoClustering 14

Number de CoClusters 15

Les entités nommées (NER) 16

**Résultats et interprétations** 16

Premier jeu de données 16

Deuxième jeu de données 20

Troisième jeu de données 23

**Conclusion** 33

**Bibliographie** 34

**Références / Sources** 35

# Introduction

Le text mining, également appelé traitement automatique du langage, peut être défini comme étant un ensemble de techniques issues de l'intelligence artificielle, alliant plusieurs domaines : la linguistique, la sémantique, le langage, les statistiques et l'informatique. Combinées ensemble, ces techniques permettent d'extraire des données pour recréer de l'information à partir de corpus de textes en les classifiant et les analysant de manière à établir des tendances. Le text mining est notamment employé dans le secteur du marketing, mais également dans de nombreux autres domaines tels que la communication, les sciences politiques, la recherche et la médecine.

L'intelligence artificielle est désormais capable de classifier automatiquement les textes par sentiment, par sujet ou par intention. Un algorithme de Text Mining est par exemple capable de passer en revue les commentaires sur un produit pour déterminer s'ils sont principalement positifs, neutres ou négatifs. Il est aussi possible de repérer les mots-clés les plus fréquemment employés. Même s'il est loin d'être un concept nouveau, le text mining connaît aujourd'hui un nouvel essor. L'émergence du Big Data où le stockage des données numériques n'est plus un problème et où les sources de données disponibles se multiplient font de l'analyse des données textuelles un enjeu crucial.

Alors même que des études scientifiques sont publiées chaque jour et que de nouveaux projets de recherche sont lancés dans le monde entier, nous avons tous plus de questions que de réponses sur le nouveau coronavirus et le COVID-19. Et les questions s'accumulent. Historiquement, ce type de questions trouve des réponses au fil du temps grâce à des recherches approfondies qui sont publiées, examinées et utilisées pour aider à prévenir et à traiter les itérations futures d'une maladie infectieuse.

Avec le COVID-19, nous n'avons pas le luxe du temps. Bien qu'il y ait beaucoup de recherches et de données partagées sur la maladie, le temps nécessaire pour lire, comparer et comprendre toutes les recherches tout en combattant la maladie est un problème croissant. Le problème est le suivant : quelle stratégie analytique pouvons-nous employer pour mettre en relation les bonnes recherches et les bonnes personnes afin de répondre à certaines de nos questions les plus urgentes ? et là l'exploitation de la littérature scientifique s'introduit pour explorer certains profils de patients afin d'identifier les comorbidités et les facteurs de sévérité du COVID-19 afin de proposer des traitements innovants susceptibles de réduire la sévérité de la maladie.

# Motivations

Depuis le début de la crise sanitaire , plusieurs études ont été faites sur les facteurs déclencheurs de la sévérité du coronavirus. les symptômes qui apparaissent chez les patients atteints de la covid semblent être les mêmes symptômes pour certaines maladies comme la toux et l'écoulement nasal dans le cas de la grippe, les difficultés respiratoires comme l'asthme, d'autres symptômes tels que la faiblesse et la fatigue apparaissent également chez les diabétiques et la fièvre comme réaction du système immunitaire à un corps étrangé , la perte de goût et d'odorat semblent avoir une relation avec d'autres maladies . La gravité de la maladie est parfois liée à l'âge et aux maladies chroniques. néanmoins les patients en réanimation sont parfois des jeunes qui n'ont aucun antécédent pathologique , les causes principales de la sévérité du covid restent mystérieuses pour la communauté scientifique, une réponse à cette question guidera les médecins à adopter des soins pertinents à chaque patient et éventuellement ralentir les contaminations et arrêter la propagation du virus .

Les informaticiens et les staticiens quand à eux, ont contribué fortement à la simplification de la tâche pour le corps médical en mettant en place un ensemble de solutions informatiques pour le diagnostic de masse , la détection des nouveaux cas contaminés et la traçabilités des personnes suspectées, la réalisation des tableaux de bord pour l'analyse et le suivi de la propagation du virus dans le monde , des outils et des moyens de protection et de gestion d'urgences au sein des hôpitaux .

les méthodes avancées de l'intelligence artificielles et du machine Learning sont par contre peu exploitées dans la fouille des données de covid et l'extraction d'une information utile que les médecins n'arrivent pas à détecter avec leur moyens de travail habituels , certaines applications ont été faites sur l'étude de la voix et la détection des personnes suspectées , d'autres travaux ont été fait sur la classification d'images radio pour distinguer la covid et la pneumonie . Vu le taux d'informations qui se publient par la communauté scientifique dans le monde, Les méthodes du machine learning pourront être intéressantes pour traiter cette masse de données textuelles et extraire des informations les plus fréquentes sur la pandémie telles que les symptômes et les personnes vulnérables. regrouper ces informations en groupes homogènes en utilisant les méthodes de clustering , et éventuellement détecter les relations entre ces groupes en utilisant des méthodes de co-clustering est une approche pertinente pour résumer toute l'information qui circule sur la covid et apporter des précisions par rapport à plusieurs problématiques .

# Objectifs

L'objectif de ce projet est d'utiliser les algorithmes avancés d'apprentissage non supervisé tels que le co-clustering  afin d'extraires les features à partir des données textuelles permettant :

- de chercher les maladies qui ont un rapport avec les sévérité de coronavirus ;
- retrouvez les symptômes communs  entre la covid et d'autres maladies ;
- détecter les facteurs déclencheurs de la sévérité du virus ;
- mettre en place un système de collecte de données pertinentes autour du covid ;
- créer un dashboard pour visualiser les résultats en changeant les paramètres des méthodes à la volée.

# Rudiments & Etat de l'art

La Covid-19 a eu un impact significatif sur la société depuis son apparition ("Novel Coronavirus(2019-nCoV)" 2020). A la fois en raison des effets graves de la maladie, mais aussi car il est difficile d'établir les différentes mesures afin de limiter sa propagation et protéger les plus vulnérables. L'association de l'analyse de texte avec l'apprentissage machine est une technique largement utilisée durant les dernières années (Ebadi 2020) (*Annual Reviews* 2021) (*NCBI* 2020). Avec l'aide des différents sites web comportant des articles parlant de la Covid-19, plusieurs outils ont en analyse de texte ont été développés pour résumer et tirer des informations sur cette maladie (*arXiv* 2020) ("LitCovid" 2020)

Jingqi Wang et al. (NCBI 2020) ont développé un outil efficace qui puisse reconnaître avec précision les concepts et symptômes cliniques importants de la Covid-19 à partir du texte brut trouvé dans les documents médicaux électroniques. Une approche hybride combinant des modèles basés sur l'apprentissage profond, des lexiques conservés et des règles basées sur des motifs a été appliquée pour construire rapidement l'outil SygnSym COVID-19, avec des performances optimisées.

Greg M. Silverman (Silverman 2021) ont utilisé l'apprentissage profond pour l'extraction des symptômes à partir des données non structurées en vue de leur utilisation dans des modèles de classification. Pour cela, deux méthodes ont été utilisées. Les deux méthodes

utilisent un lexique dérivé à partir du Centre de contrôle et de prévention des maladies - Cov 19 (CDC). La première méthode utilise un modèle word2vec pour l'expansion de cette liste à l'aide d'un dictionnaire correspondant au système de langage médical unifié (UMLS). La deuxième méthode a utilisé le lexique étendu comme un répertoire géographique basé sur des règles et l'UMLS. Ces méthodes ont été évaluées par rapport à une référence annotée manuellement (f1-score de 0,87 pour l'ensemble basé sur UMLS ; et 0,85 pour la nomenclature à base de règles avec UMLS).

Billie S Anderson (Anderson 2021) a utilisé le Clustering pour classifier les articles parlant du Covid-19 par thème. L'étude analyse 83 264 résumés d'articles relatifs au Covid-19. Les données textuelles ont été analysées grâce à la Décomposition en valeures singulières (SVD) et l'algorithme EM. Les résultats suggèrent que le regroupement de textes peut à la fois révéler des thèmes de recherche cachés dans la littérature publiée relative à COVID-19, et réduire le nombre d'articles que les chercheurs doivent parcourir pour trouver du matériel pertinent à leur domaine d'intérêt.

Veysel et al. (Kocaman and Talby 2020) ont utilisé le NLP pour l'amélioration de la compréhension sur les articles du covid-19, à l'aide de SPARK. Le système développé peut reconnaître plus de 100 entités dont les déterminants sociaux de la santé, l'anatomie, les facteurs de risque et les événements indésirables. Deuxièmement, le pipeline de traitement de texte comprend la détection de l'état des assertions, pour distinguer les faits cliniques qui sont présents, absents, conditionnels, ou d'une personne autre que le patient. Troisièmement, les modèles d'apprentissage profond utilisés sont plus précis que ceux disponibles précédemment, en tirant parti d'un pipeline intégré de modèles de reconnaissance d'entités nommées pré-entraînés de pointe, et en améliorant les précédents repères les plus performants pour la détection de l'état d'assertion.

Cependant, et au mieux de notre connaissance, il n'existe pas encore d'articles scientifiques abordant l'exploration des sévérités et symptômes du covid-19 en utilisant des méthodes de co-clustering.

# • Chargement de données

Pour la récupération des données on a choisi de le faire d'une manière dynamique qui va nous permettre de jouer sur le nombre d'articles voulus, les termes utilisés pour chercher les articles et même la mise à jour de données.

Pour ce faire, on a utilisé le module Entrez du package Bio, ce module permet d'accéder aux bases du NCBI (à la fois les bases type séquences et autre, mais aussi PubMed par exemple).

La plupart des requêtes récupère les infos sous forme de XML (Nous on a choisi le format json) et il y a un parseur intégré que l'on peut appeler avec Entrez.read qui transforme le xml en structure python (avec dictionnaires et listes) puis on met les données dans un Dataframe pour pouvoir les exploiter dans les prochaines étapes.

Pour la recherche on a choisi de faire la requête avec les termes "Covid", "Coronavirus", "Covid19", "Severety" avec un nombre d'articles égale à 5 000 et en prenant que les abstracts des articles, on a rapidement constaté que 10 000 enregistrement c'est très peu pour notre étude donc pour récupérer plus de 10 000 UID, on a soumis plusieurs demandes de recherche tout en augmentant la valeur de restart (restart est le paramètre qui définit l'index séquentiel du premier UID dans le jeu récupéré à afficher dans la sortie de la requête effectuée).

Le module Entrez enregistre une adresse mail et un nom d'outil de façon à pouvoir nous contacter en cas de problème (option facultative).

```
max_searchs = 5000
aterms = "coronavirus covid covid19 severity"

Entrez.email = 'A.N.Other@example.com'
ids=[[]]
print("max_searchs : ",max_searchs)
for i in range(0,max_searchs,100):
  print(i," ", end='')
  h = Entrez.esearch(db='pubmed', retmax=100,retstart=i, term=term)
  result = Entrez.read(h)
  ids.append(result['IdList'])
h    =    Entrez.efetch(db='pubmed',    id=ids,    rettype='medline',
retmode='json')
records = Medline.parse(h)
```

## ● Collecte de données

Comme expliqué précédemment, la collecte des données a été faite en utilisant une Api python sur la plateforme PubMed regroupant des articles médicaux. Afin de récupérer des données parlant d'une thématique particulière, on utilise la recherche par mot clé. Dans notre cas, nous cherchons des articles parlant du virus Covid 19, le but étant de trouver les maladies qui ont une relation avec ce virus et qui peuvent éventuellement causer une

gravité chez certains malades. Pour ce faire, nous avons suivi trois approches différentes. Nous les résumons en dessous.

### Approche 1

Une première approche consiste à rechercher des articles par le mot clé “covid” ou “coronavirus” , cette recherche retourne un grand nombre d’articles et le filtrage a été fait d’une manière arbitraire, les résultats des méthodes de clustering et de Co-clustering ne sont pas prometteurs et le clusters résultants ne sont pas forcément caractérisés par les noms des maladies ni par les symptômes qui leurs sont associés , nous avons quand même réussi à identifier certains clusters parlant des maladies psychologiques, des symptômes de Coronavirus, mais nous n’avons pas un moyen de mesurer la proximité entre les maladies et d’identifier les symptômes communs entre elles .

### Approche 2

Une deuxième approche consiste à rechercher tous les articles qui parlent du coronavirus et d’une autre maladie , un premier test a été fait avec les mots clés “covid” et “obésité” , les clusters résultant semblent avoir plus de sens et certaines conclusions ont été faites dans les chapitres suivants sur chaque étude de cas, l'inconvénient de cette approches est le fait que les maladies qui ont un rapport avec le covid ne sont pas toutes connues et l’objectif de notre étude justement est d’identifier ces maladies d’une manière automatique sans faire appel à un expert métier .

### Approche 3

Dans la troisième approche, nous avons utilisé les entités nommées pour récupérer une liste des maladies chroniques et des symptômes les plus connues, et nous les avons injectés séquentiellement dans la requête de recherche sur l’API PubMed accompagnés par les mots clés “covid” et "severity".

## Prétraitement de données

Le Data Cleaning (nettoyage de données) est l’étape la plus importante avant d’analyser ou modéliser des données, maintenant que nous avons chargé notre jeu de données, nous devons nettoyer le texte pour améliorer les performances de classification ou du clustering. Dans ce qui suit, nous allons parcourir un certain nombre de tâches de Data Cleaning.

# 1. Suppression des valeurs null

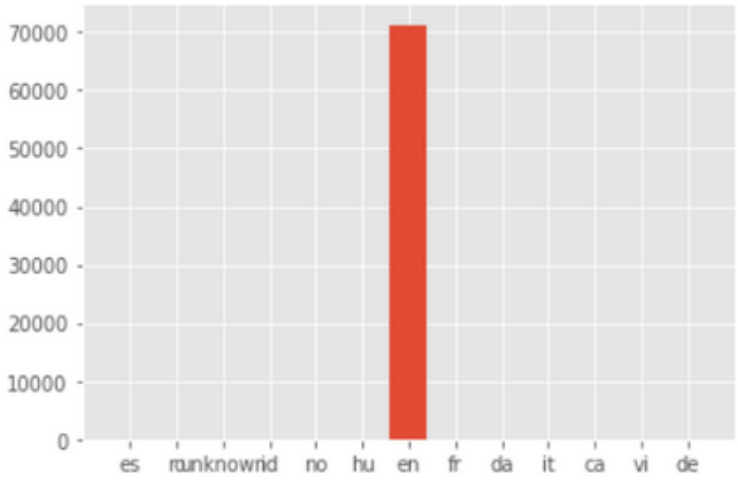
Dans le monde réel, les données sont rarement propres et homogènes. Les données peuvent être manquantes pendant l'extraction ou la collecte des données. Les valeurs manquantes doivent être traitées car elles réduisent la qualité de nos mesures de performance. Elles peuvent également conduire à une prédiction ou à une classification erronée et causer un biais élevé pour tout modèle utilisé.

Pour se débarrasser des valeurs null on a utilisé la fonction dropna de Pandas qui supprime chaque ligne contenant une valeur nulle. Bien évidemment qu'il existe plusieurs méthodes pour traiter les valeurs nulls, notre choix de supprimer la ligne qui contient au moins une valeur null est le plus adéquat par rapport au type de nos données , donc on peut pas se servir des autres méthodes  utilisées dans le cas numérique (par exemple : remplacement par la valeur moyenne de la colonne ).

# 2. Multi-langues traitement

Puisque on veut travailler qu'avec des articles en anglais, nous allons déterminer la langue de chaque article dans le cadre de données. Toutes les sources ne sont pas en anglais et la langue doit être identifiée pour que nous sachions quoi garder et quoi supprimer.

Pour ce faire, on a utilisé le package "langdetect" qui supporte 55 langues dont pour chaque article on a détecté la langue et on l'a mémorisé dans une liste qu'on va l'insérer après comme une nouvelle colonne "language" dans notre Dataframe. Cette démarche va nous permettre de connaître les articles à garder (ceux qui ont la valeur "en" dans le champ language). La figure ci-dessous montre la distribution des langues d'articles dans notre corpus.




*Histogramme des langues d'articles.*

## 3. Stopwords

Une partie du prétraitement consistera à trouver et à supprimer les stopwords (mots qui serviront de bruit dans l'étape de clustering et topic modeling). Pour cela on s'est servi de la liste "STOP_WORDS" du package spacy qui contient les stopwords qui sont utilisés dans n'importe quel texte d'anglais, par contre les documents de recherche utilisent souvent des mots qui ne contribuent pas réellement au sens et qui ne sont pas considérés comme stopwords dans la liste "STOP_WORDS" qu'on vient de citer, donc on a décidé de rajouter à cette liste une deuxième liste qui contient les stopwords généralement contenus dans les documents de recherche. On a créé une fonction qu'on a appelé "spacy_process()" qui va enlever tous les mots qui appartiennent à la liste des stopwords.

## 4. Ponctuations

Dans le même contexte et pour faire plusieurs traitement sur le texte, on a utilisé le package "en_core_web_lg" (English pipeline optimized for CPU, qui contient les composantes: tok2vec, tagger, parser, senter, ner, attribute_ruler, lemmatizer) pour enlever les ponctuations de notre texte en utilisant la liste punctuations du package "string" dont pour chaque mot on va tester s'il est dans la liste "punctuation", si oui on le supprime.

## 5. Lowercase

Toujours dans la fonction "spacy_process()" et en utilisant la fonction prédéfinie "lower()" du package "en_core_web_lg", on a transformé tous nos mots en minuscule (lowercase).

## 6. Lemmatization

Le processus de lemmatisation consiste à représenter les mots (ou lemmes ) sous leur forme canonique. Par exemple pour un verbe, ce sera son infinitif. Pour un nom, son masculin singulier. L'idée étant encore une fois de ne conserver que le sens des mots utilisés dans le corpus.

Dans notre démarche, on a exécuté la lemmatisation en deux étapes : d'abord en utilisant le lemmatizer du package "en_core_web_lg" qu'on a appelé en exécutant la fonction "spacy_process()". Ensuite on l'a fait une deuxième fois avec "WordNetLemmatizer" du package "nltk.stem". Un échantillon du résultat est donné en dessous.

```
Worksites with on-site operations have experienced coronavirus disease (COVID-19) outbreaks. We analyzed data

worksite site operation experience coronavirus disease covid-19 outbreak we analyze datum 698 nonresidential
```

# 7. Liens et HTML tags

Lorsqu'il s'agit d'analyser du HTML, on ne souhaite probablement pas traiter de JavaScript ou de CSS intégrés, et on n'est intéressé que par le texte, l'idée est donc de construire une expression régulière qui peut trouver tous les caractères "< >" comme première incidence dans un texte, et ensuite, en utilisant la fonction sub, nous pouvons remplacer tout le texte entre ces symboles par une chaîne vide, c'est exactement ce qu'elle va faire la fonction "remove_html(text)" qu'on a définit.

Même principe avec Html tags et en utilisant la fonction "remove_urls(text)" qui, à l'aide d'une expression régulière (package re), trouve et supprime les liens.

# 8. Frequent words

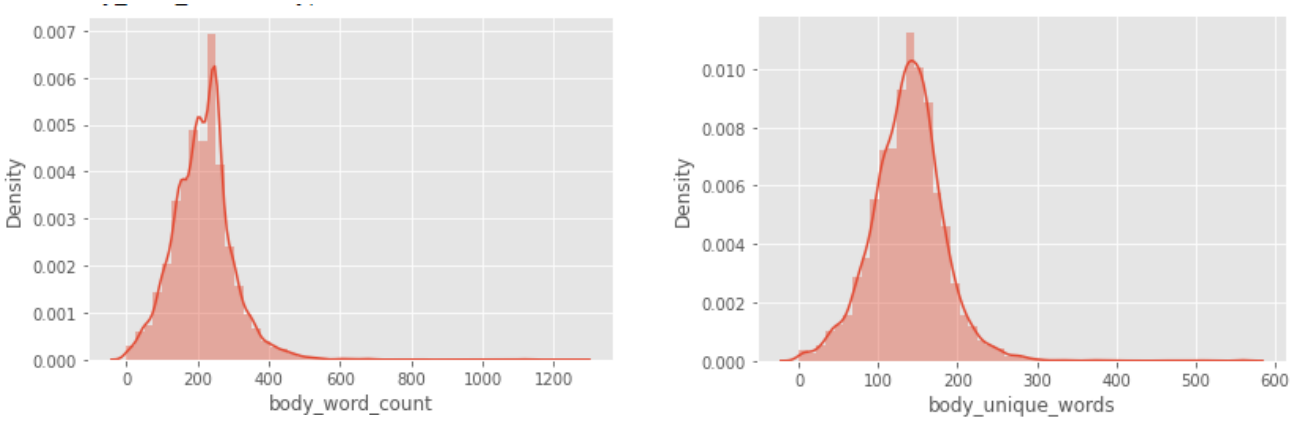
Cette étape consiste à supprimer les mots fréquents dans le corpus donné. Pour ce faire, on calcule le nombre d'occurrences de chaque mot puis on supprime les mots les plus fréquents. Finalement on a décidé de ne pas enlever les frequent words car on a constaté qu'il existe dans les résultats plusieurs mots qui peuvent être intéressant pour nous dans les prochaines étapes de clustering et topic modeling.

# 9. Rare words

Cette étape est très similaire à l'étape précédente, mais nous allons supprimer les mots rares du corpus au lieu des mots les plus fréquents.

## Nombre de 'mots' / 'mots uniques' par articles

Ces deux graphes nous donnent une bonne idée du contenu auquel nous avons affaire. La plupart des articles ont une longueur de 150 à 250 mots. Les longues queues dans les deux graphes sont dues à des valeurs aberrantes. En fait, 98% des articles font entre 150 et 250 mots, tandis qu'un petit nombre d'entre eux font plus de 250 ou moins de 150 mots. Le nombre de mots moyen est environ 200 mots par article par contre seulement 130 mots unique par article. Les histogrammes montrant la distribution des mots sont présentés en dessous.




*Histogramme des langues*

## • Vectorization

Maintenant que nous avons prétraité les données, il est temps de les convertir dans un format qui peut être traité par nos algorithmes. Dans l'apprentissage automatique, les features sont essentiellement des attributs numériques à partir desquels on peut effectuer des opérations mathématiques telles que la factorisation matricielle, le produit scalaire, etc. Mais il existe plusieurs scénarios dans lesquels les ensembles de données ne contiennent pas d'attributs numériques, par exemple l'analyse sentimentale d'un utilisateur de Twitter/Facebook. Dans notre cas, le jeu de données contient des chaînes de caractères, dont la conversion de ce type de features en features numériques est appelée featurisation. Plus clairement, le processus de conversion du texte en vecteur est donc appelé vectorisation.

À cette fin, nous utiliserons tf-idf. Nous allons convertir nos données de format chaîne de caractères en une mesure de l'importance de chaque mot pour l'instance dans l'ensemble de la littérature. TF-IDF est l'abréviation de Term Frequency-Inverse Document Frequency, qui indique l'importance d'un mot dans un corpus ou un ensemble de données. TF-IDF contient deux concepts : la fréquence des termes (TF) et la fréquence inverse des documents (IDF).

Term Frequency (TF) : est définie comme la fréquence à laquelle le mot apparaît dans le document ou le corpus. Comme chaque phrase n'a pas la même longueur, il est possible qu'un mot apparaisse plus souvent dans une longue phrase que dans une phrase plus courte. La fréquence des termes peut être définie comme suit :

$$TF = \frac{No\ of\ time\ word\ appear\ in\ the\ document}{Total\ no\ of\ word\ in\ the\ document}$$

Inverse Document Frequency(IDF) : La fréquence inverse des documents est un autre concept utilisé pour déterminer l'importance d'un mot. Il est basé sur le fait que les mots moins fréquents sont plus informatifs et plus importants. L'IDF est représentée par la formule :

$$IDF = log_{10} \frac{Number\ of\ Document}{Number\ of\ document\ in\ which\ word\ appear}$$

une partie d'affichage du résultat TF-IDF :

```
(0, 1709)     0.04484975965315016
(0, 3837)     0.044530922261825145
(0, 2935)     0.08022674988354439
(0, 3548)     0.06071357656554389
(0, 2323)     0.05814969524243637
(0, 326)      0.08110920426122992
(0, 1250)     0.09034782837160994
(0, 862)      0.03221473841317942
(0, 451)      0.06777780342426314
(0, 97)       0.0603347176243218
(0, 807)      0.058204274149323255
(0, 1544)     0.07252571620646368
(0, 3526)     0.11266822409977244
```

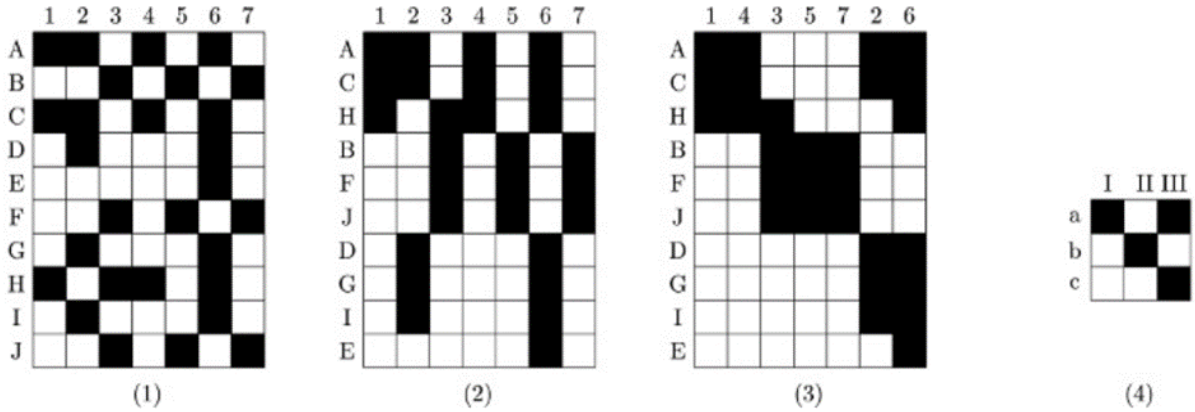
# • CoClustering & Entités nommées

Nous présentons dans cette partie le Co Clustering .

## • CoClustering

La classification double ou « Biclustering » est une technique d'exploration de données non supervisée permettant de segmenter simultanément les lignes et les colonnes d'une matrice. Cette approche  a été utilisé massivement en biologie - par exemple dans l'analyse de l'expression génétique, mais aussi dans d'autres domaines tels que la compression d'image de synthèse, l'analyse médicale, la caractérisation d'émetteurs de pourriels (« spam »), l'analyse du mouvement, l'analyse des termes publicitaires sur internet.

Un exemple de donnée Co Clustering appliqué à une matrice (1) est donnée dans la matrice (3) dans l'image ci-dessous.

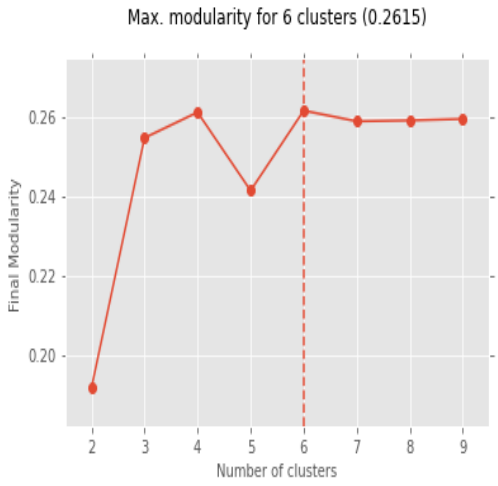


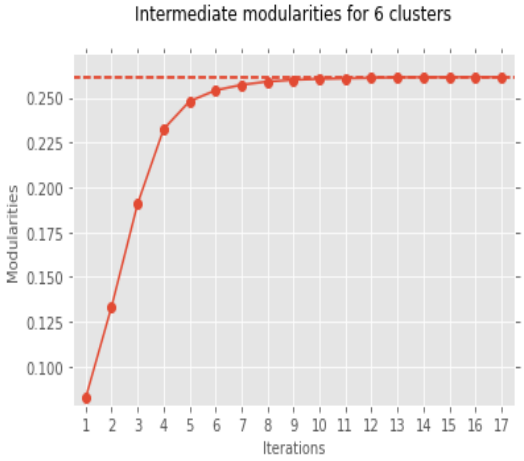

## ● Number de CoClusters

Pour définir le nombre idéal des co clusters, nous avons fait appelle à la fonction prédéfinie "*best_modularity_partition*()", en lui donnant en paramètre min_cluster_nbr et max_cluster_nbr. Pour la visualisation, nous avons utilisé deux fonctions :

**1- "plot_max_modularities()" :** qui prend en paramètre le résultat de la fonction best_modularity_partition pour déterminer le nombre de cocluster qui a le maximum des modalités ;

**2- "plot_intermediate_modularities()" :** qui prend en paramètre le model récupéré de la fonction best_modularity_partition pour visualiser intermédiate modalities pour le nombre de co clusters déterminé dans la phase précédente.




A titre d'exemple, les figures en-dessus montrent que le nombre de clusters optimal est de 6.

## ● Les entités nommées (NER)

Pour aller plus loin dans l'exploitation des clusters et avec l'utilisation de l'approche NER (Named Entity Recognition), il est possible d'extraire les entités dess différentes catégories. Il existe plusieurs modèles de base, pré-entraînés, comme **"en_ner_bc5cdr_md"**, que nous utiliserons et qui est capable de reconnaître les personnes, les lieux, les dates et même les maladies. Vi qu'on est intéressé que par les maladies on va nous limiter par chercher les mots qui ont ***label = DISEASE***

# ● Résultats et interprétations

La troisième approche étant la plus adaptée pour l'extraction des données, nous l'avons adoptée pour toutes les expériences que nous allons aborder. Dans ce chapitre, nous présentons et interprétons chaque résultat. Les visualisations sur lesquelles nous allons nous baser sont : **(1)** les mots les plus fréquents par co-coclust , **(2)** le graph de similarité et **(3)** les entités nommées par co cluster. Toutes les visualisations ont été générées grâce au dashboard développé à cet effet.

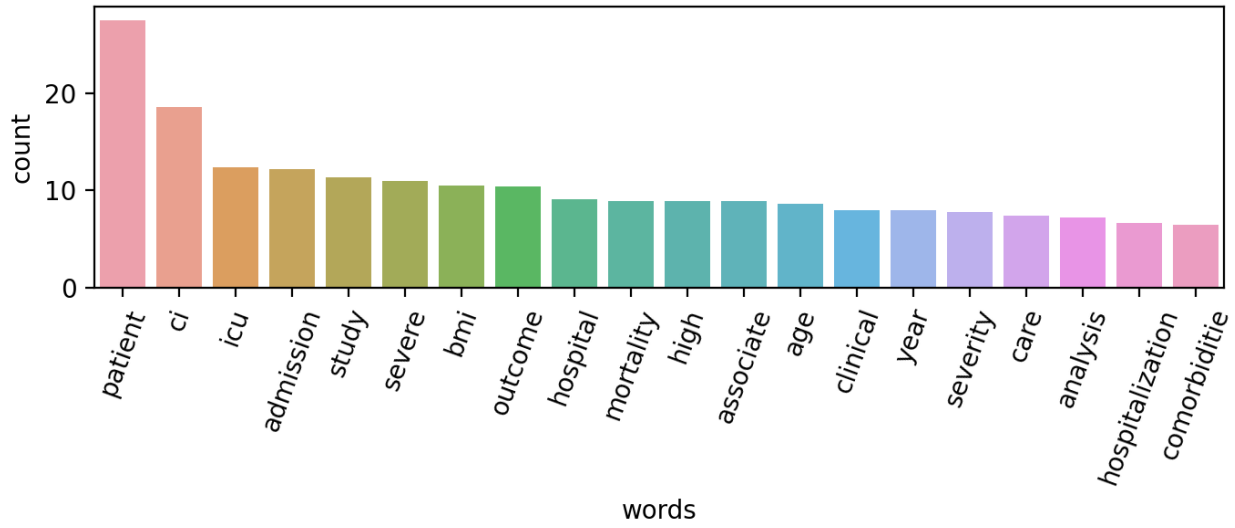
## ● Premier jeu de données

Le premier jeu de données a été extrait en utilisant les mot clés suivants : "covid19", "severity" & "obesity". Le but derrière cette extraction c'est de savoir s'il y a une quelconque relation entre la covid-19 et l'obésité, voir aussi si les personnes atteintes de l'obésité sont plus vulnérables au covid-19 et d'autres informations. En suivant la méthode précédemment décrite, nous avons opté pour 4 co-clusters.

Après avoir appliquer le pre processing ainsi que les algorithmes de co-clustering, nous avons obtenu les résultats suivants, résumés dans une première figures comportant les mots les plus fréquents dans chaque cluster, une deuxième figure qui montre la relation entre les différents mots clés présents dans chaque co-cluster et enfin une figure résultant les entitées nommées présentes dans chaque co-cluster.

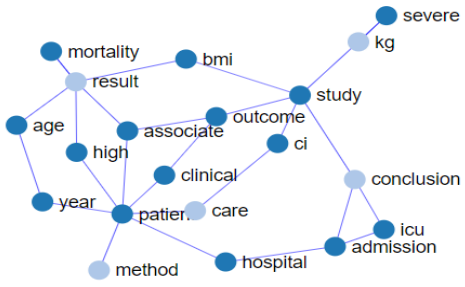
### 1. Premier cocluster

Dans le premier cocluster, nous pouvons voir des signes d'obésité et des symptômes de covid qui sont regroupés dans le même coclust. Nous voyons par exemple, qu'il y a une corrélation entre l'âge, la mortalité, la sévérité du covid.



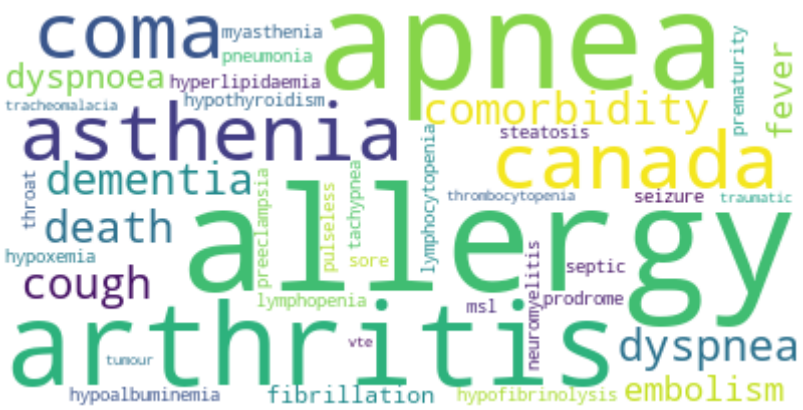

*Fréquences des mots dans le premier coclust*

Dans le graph de similarité ci-dessous, nous pouvons voir une forte similarité entre le mot sévère et kg, ce qui nous indique une existence entre la sévérité et le poids de la personne atteinte. Aussi, nous pouvons voir une chaîne qui commence par study - admission - year - âge- mortalité : ceci nous indique que les personnes les plus âgées sont celles qui décèdent le plus.




*Graph de similarité dans le premier coclust.*

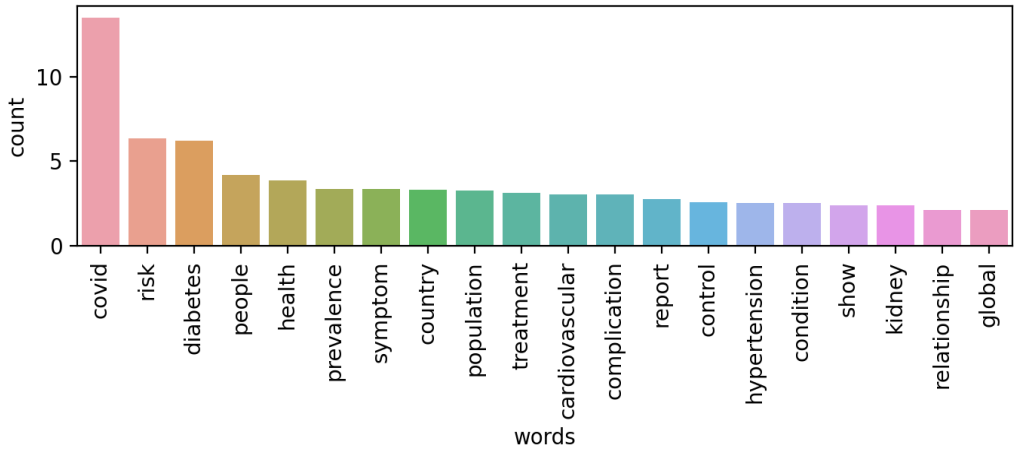
Avec le word-cloud généré à partir des entités nommées, nous pouvons confirmer notre analyse sur la relation entre l'âge et la sévérité du covid. Par exemple, nous pouvons voir les mots clés tels que le Coma, Fibrillation et Death. Et puisque nous avons vu que ce cocluster inclut l'âge, nous pouvons donc conclure que les personnes âgées passent par le coma, et passent par le défibrillateur qui est relié au maladies cardiovasculaire.




*Wordcloud des entités nommées du premier coclust.*

## 2. Deuxième coclust

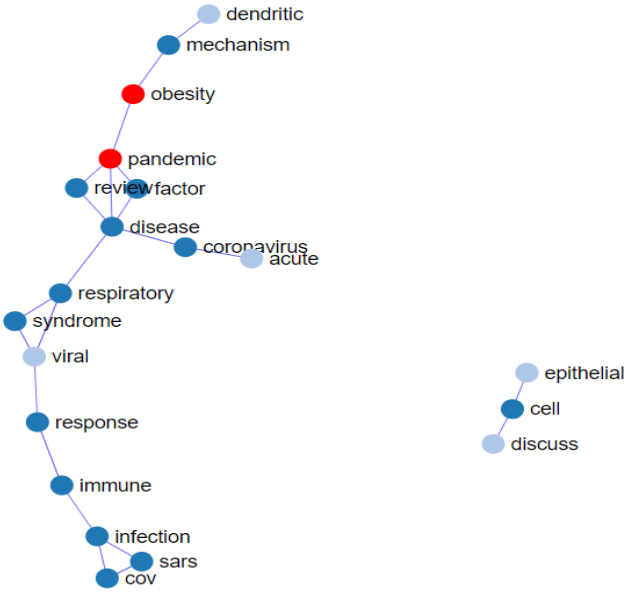
Dans le deuxième co-cluster, nous pouvons voir des symptômes, tels que l'hypertension, des maladies pulmonaires, complications cardiovasculaires et des douleurs. Nous pouvons conclure ainsi que les personnes atteintes par la maladie de l'obésité et qui attrapent la covid-19 développent les symptômes cités.




*Fréquence des mots dans le deuxième coclust.*

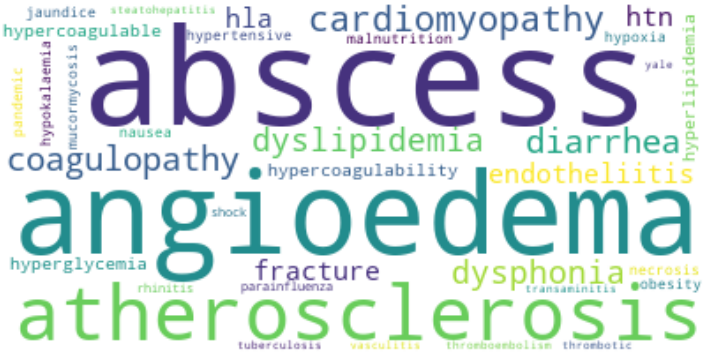
Avec le graph de similarité, donné en dessous, nous pouvons voir qu'il existe deux sous graphs reliés. Le premier inclut les mots en relation avec la Covid-19 (pandemic) et le deuxième inclut les mots en relation avec l'obésité. De là, nous pouvons conclure, qu'il y a

une relation directe entre la covid-19 et l'obésité : ce qui s'interprète par le fait qu'il existe une grande similarité entre les deux maladies et leurs symptômes. Par exemple, nous pouvons voir une forte similarité entre le mot disease et coronavirus et respiratory.. Ce qui veut dire que la sévérité du covid est reliée à une maladie chronique qui est dans notre cas l'obésité.




*Graph de similarité du deuxième coclust.*

Avec le graphe des entités nommées ci-dessous, nous pouvons voir des entités nommées relatives aux symptômes du covid tels que : la diarrhée, et hypercoagulation et également certains facteurs et signes de l'obésité et du diabète comme : malnutrition , hyperglycémie.




*Wordcloud des entités nommées du deuxième coclust.*

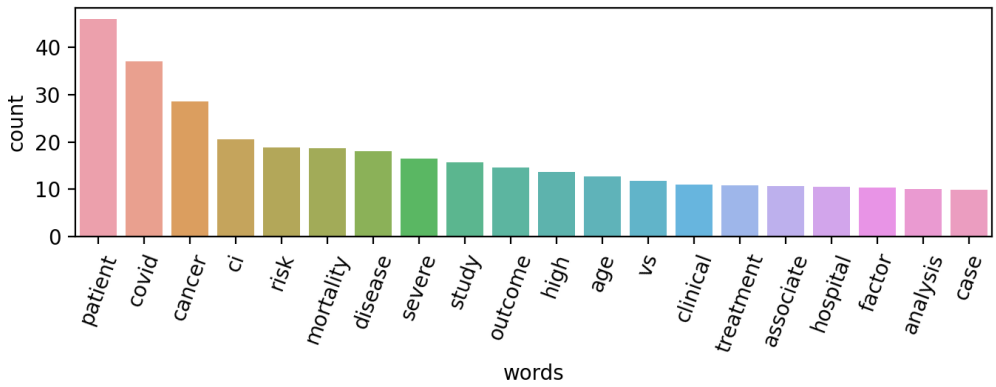
3. **Conclusion :** A partir de ce jeu de données, nous avons pu voir des co-clusters intéressants. En effet, nous pouvons voir par exemple la relation entre le diabète, l'obésité et les symptômes du covid-19.

# ● Deuxième jeu de données

Dans ce deuxième jeu de données, nous avons utilisé les mots clés “severité”, “covid19” et “Cancer”.

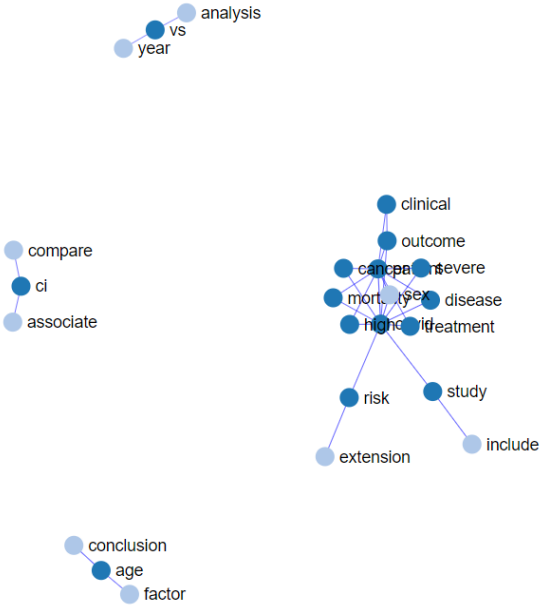
### 1. Premier coclust

Dans le premier cocluster, nous pouvons voir des mots en relation avec la Covid et la maladie du Cancer. Dans la figure des fréquence des mots par cocluster, nous pouvons voir des termes tels que covid, cancer, mortality, disease, severe, âge & hospital. Le colcuster informe ainsi qu’il y a une forte mortalité chez les cancéreux les plus âgés atteints du covid-19.




*Fréquence des mots dans le premier coclust.*

Dans le graph de similarité ci-dessous, nous pouvons voir effectivement une forte similarité entre la mortalité, l’âge, le cancer et le risk.



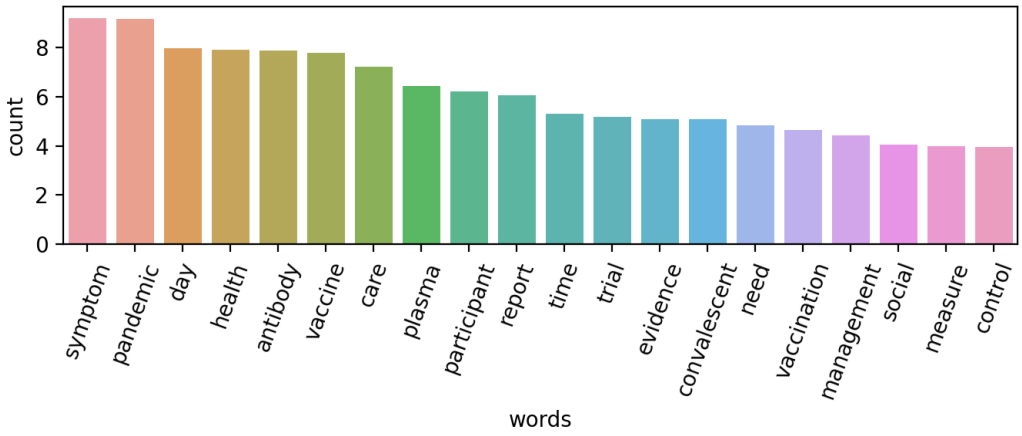

*Graph de similarité dans le premier coclust.*

Dans le graph des entités nommées ci-dessous, nous pouvons voir les signes de la sévérité comme bleeding, death, hypertension.


*Wordcloud des entités nommées dans le premier colcust.*

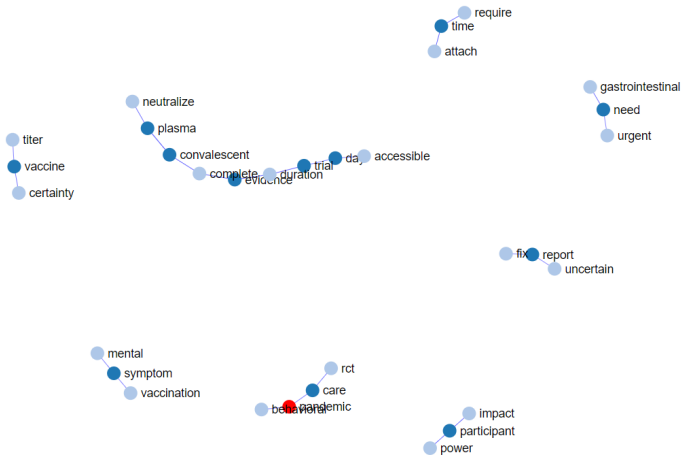
### 2. Deuxième coclust

Dans le deuxième coclust, nous pouvons voir les traitements et la prévention pour les cancéreux atteints du covid-19. Nous pouvons voir par exemple des mots comme “care, measure, et contrôle, vaccination, measure. Les fréquences des mots sont données en dessous.



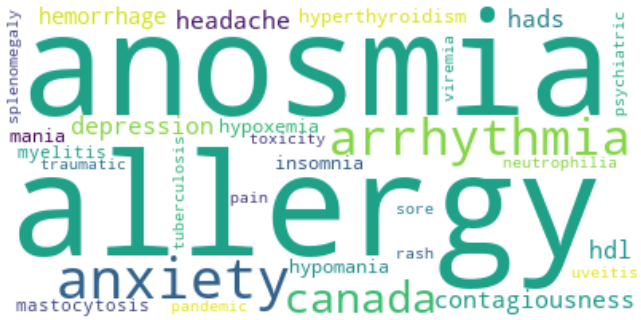

*Fréquences des mots dans le deuxième coclust.*

Dans le graph de similarité, nous pouvons voir une similarité entre *require* et *time* nous pouvons conclure ainsi que les personnes cancéreuses nécessitent plus de temps pour se rétablir.




*Graph de similarité dans le deuxième coclust.*

Dans le graph des entités nommées, nous avons pu constater la présence de mots en relation avec des maladies psychiatriques, telles que dépression, psychiatrique, anxiety. Nous pouvons ainsi conclure que les personnes cancéreuses atteintes du covid souffrent des de dépression et nécessitent ainsi un soin psychiatrique.




*Graph des entités nommées dans le deuxième coclust.*

**3. Conclusion :**

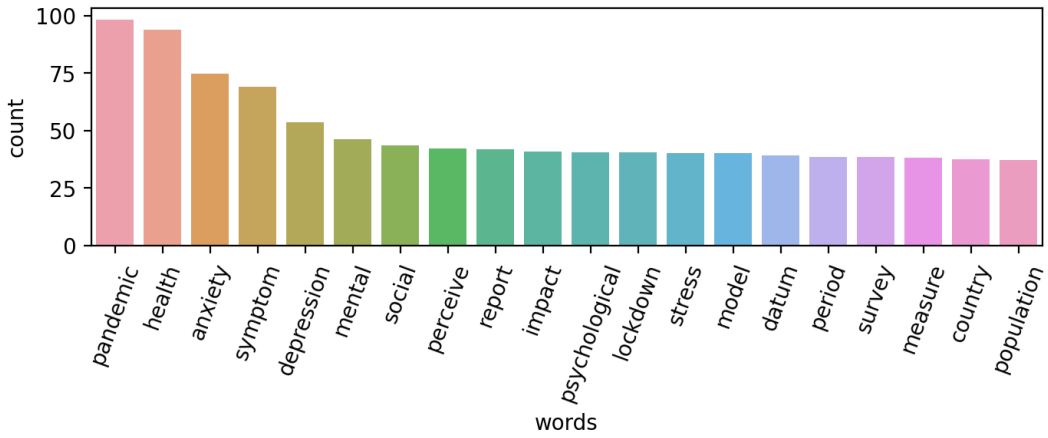
A partir de ce jeu de données, nous pouvons conclure que la mortalité est plus présente chez les personnes âgées cancéreuses. Aussi, nous pouvons voir que les personnes cancéreuses atteintes du covid sont susceptibles de développer des dépressions et des maladies mentales.

## ● Troisième jeu de données

Dans ce jeu de données, nous avons inclus les mots “sévérité”, “covid” & “pneumonie”.

**1. Premier coclust**

Dans le graph des fréquences de mots, nous pouvons voir que les maladies psychiatriques sont les plus présentes dans un même cocluster. Les mots présents sont par exemple symptômes, anxiety, stress. Nous pouvons ainsi conclure que parmi les symptômes des personnes atteintes de pneumonie, c’est les maladie psychiatriques.

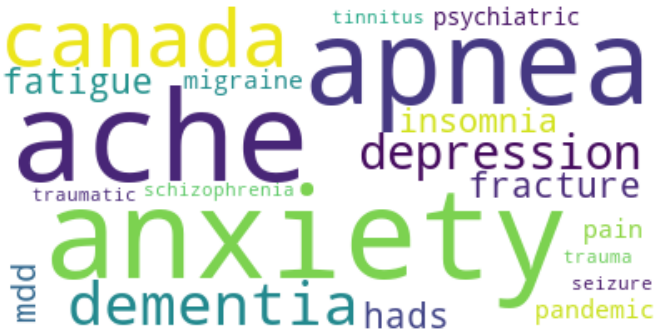



*Fréquence des mots dans le premier coclust.*

Dans le graph de similarité, nous pouvons effectivement voir une forte similarité entre les différentes maladies psychiatriques.


*Graph de similarité pour le premier cluster.*

Dans le graphique des entités nommées, nous pouvons voir en clair les noms de ces maladies/symptômes. Par exemple : dépression, anxiety, psychiatrique, schizophrène.

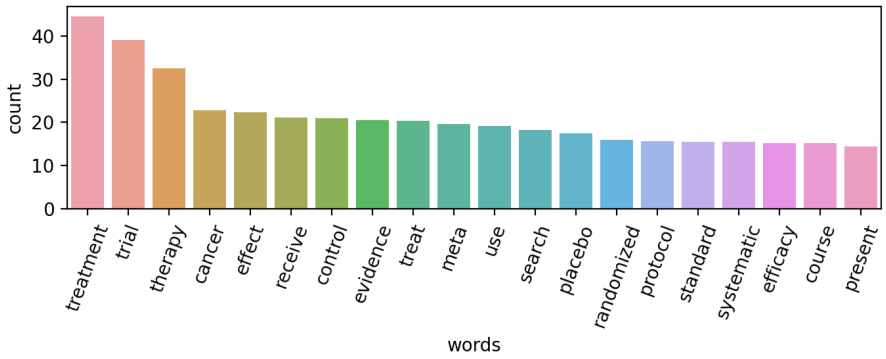



*Graph des entités nommées dans le premier cluster.*

### 2. Deuxième cocluster

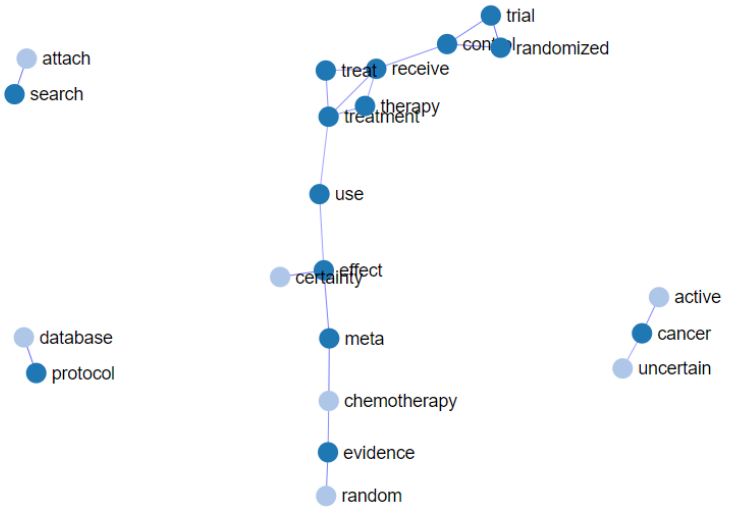
Dans ce deuxième co-cluster, nous pouvons voir des termes relatifs au cancer, ses sévérités et traitement. Des termes tels que  treatment, cancer, effect. Nous pouvons ainsi

conclure que les personnes atteintes de la pneumonie sont souvent atteintes du cancer et développent des formes sévères du covid-19. Le graph des mots fréquents ci-dessous le montre bien.




*Fréquence des mots dans la deuxième coclust.*

Dans le graph de similarité, nous pouvons voir effectivement des mots relatifs au cancer. Nous voyons par exemple que tous les traitements sont regroupés dans la même zone tel que la  chimiothérapie.




*Graph de similarité du deuxième coclust.*

### 3. Conclusion :

De ce jeu de données, nous avons pu voir une relation directe entre la Covid-19, la maladie du pneumania, le cancer et les maladies psychiatriques.

# Conclusion

Pour développer le NLP biomédical COVID-19, des scientifiques des données, des ingénieurs logiciels, des cliniciens et des chercheurs en médecine sont réunis afin de permettre une approche éclairée et de développer un processus d'extraction des connaissances bien équilibré. L'effort multidisciplinaire émule le raisonnement inductif clinique du monde réel qui utilise une approche par étapes pour évaluer, extraire et hiérarchiser les connaissances afin de permettre une médecine fondée sur les preuves.

Alors que notre NLP biomédical extrait les entités pertinentes de la littérature biomédicale et fournit un haut degré de cohérence des clusters, il existe des limitations associées à la bibliothèque de reconnaissance des entités nommées biomédicales. Cela restreint la capacité d'extraire toutes les entités biomédicales alignées nécessaires (ça se voit clairement dans nos résultats). La reconnaissance des concepts biomédicaux est un domaine de recherche actif, et des méthodes améliorées ciblant un large éventail de types d'entités et de concepts peuvent être substituées.

La pandémie de COVID-19 a créé une situation unique où il est nécessaire d'accéder rapidement à des connaissances cliniques en constante évolution à partir des publications de qualité variable qui se multiplient de manière exponentielle. Dans ce projet on a pu exploiter la littérature scientifique disponible en ligne  à l’aide d’approches de fouilles de texte avancés qui nous a également permis d’explorer les relations entre les maladies, l’étape prochaine dans cette recherche sera d'exploiter le nouveau corpus créé afin de faire la relation entre le covid-19 et des nouveaux traitements efficaces.

# 1. Bibliographie


**Anderson, Billie S. 2021**. “Article Menu Download PDF [PDF] Free EPUB View Accessing resources off campus can be a challenge. Lean Library can solve it [Lean Library] Full Article Content List Abstract 1. Introduction 2. Literature review 3.” *SAGE Journals*, Mars 16, 2021. https://journals.sagepub.com/doi/full/10.1177/01655515211001661.

***Annual Reviews*. 2021**. “Artificial Intelligence in Action: Addressing the COVID-19 Pandemic with Natural Language Processing.” Mai 14, 2021. https://www.annualreviews.org/doi/abs/10.1146/annurev-biodatasci-021821-061045.

***arXiv*. 2020**. “[2004.10706] CORD-19: The COVID-19 Open Research Dataset.” Avril 22, 2020. https://arxiv.org/abs/2004.10706.

**Ebadi, Ashkan. 2020**. “Understanding the temporal evolution of COVID-19 research through machine learning and natural language processing.” *SpringerLink*, Novembre 19, 2020. https://link.springer.com/article/10.1007/s11192-020-03744-7.

**Kocaman, Veysel, and David Talby. 2020**. “[2012.04005] Improving Clinical Document Understanding on COVID-19 Research with Spark NLP.” *arXiv*, December 7, 2020. https://arxiv.org/abs/2012.04005.

**“LitCovid.” 2020**. NCBI. https://www.ncbi.nlm.nih.gov/research/coronavirus/.

*NCBI*. 2020. “COVID-19 SignSym – A fast adaptation of general clinical NLP tools to identify and normalize COVID-19 signs and symptoms to OMOP common data model.” Juillet 13, 2020. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7480086/.

**“Novel Coronavirus(2019-nCoV).” 2020**. WHO | World Health Organization. https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200211-sitrep-22-ncov.pdf.

**Silverman, Greg. 2021.** “NLP Methods for Extraction of Symptoms from Unstructured Data for Use in Prognostic COVID-19 Analytic Models.” *Journal of Artificial Intelligence Research*, October 14, 2021. https://www.jair.org/index.php/jair/article/view/12631.

# 2. Références / Sources

- towardsdatascience.com
- stackoverflow.com
- github.com
- ia-data-analytics.fr
- datascientest.com
- blogdigital.beijaflore.com/text-mining/
- fr.coursera.org/learn/python-text-mining
- www.udemy.com/course/project-based-text-mining-in-python/
- https://datascientest.com/text-mining-definition
- https://ia-data-analytics.fr/logiciel-data-mining/text-mining/text-mining-traiter-vos-donnees-textuelles/
- https://www.sas.com/en_us/insights/articles/analytics/covid-19-research-with-text-analytics.html
- https://www.researchgate.net/
- https://intellica-ai.medium.com/comparison-of-different-word-embeddings-on-text-similarity-a-use-case-in-nlp-e83e08469c1c
- https://medium.com/apprentice-journal/pca-application-in-machine-learning-4827c07a61db
- https://mrmint.fr/algorithme-k-means
- https://towardsdatascience.com/latent-semantic-analysis-deduce-the-hidden-topic-from-the-document-f360e8c0614b
- https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24

- https://medium.com/voice-tech-podcast/topic-modelling-using-nmf-2f510d962b6e
- https://medium.com/@paritosh_30025/natural-language-processing-text-data-vectorization-af2520529cf7
- https://moncoachdata.com/blog/nettoyage-de-donnees-python/
- http://www.python-simple.com/python-biopython/utilisation-entrez.php
- https://medium.com/@maheshdmahi/scispacy-for-bio-medical-named-entity-recognition-ner-63ed548f1df0
- https://www.datacamp.com/community/tutorials/stemming-lemmatization-python
- https://pubmed.ncbi.nlm.nih.gov/