Page article : <https://arxiv.org/abs/2010.01600>

On utilise un modele de « topic » (topic modeling) qui compare des topics qu’on identifie comme soit :

* Long-lasting trends
* Short-lasting topic

Le modèle doit aussi savoir localiser ses topics dans le temps avec clarté.

Cet article présente différents modèles.

On compare la « variabilité » (variability) des tailles/longueur de divers topics.

Ces longueurs étant celles obtenues en utilisant différents modèles (de topic) assez connus :

* Latent Dirichlet allocation (LDA)
* Nonnegative matrix factorization (NMF) (sujet de mon stage)
* Nonnegative CANDECOMP/PARAFAC tensor decomposition (NCPD et Online NCPD)

Le NCPD est censé être la « tensor counterpart » de la NFM. Cela veut surement dire que c’est des modèles similaires mais que le NCPD utilise les tensor contrairement au NFM. ( ?)

L’article veut nous montrer que seules les méthodes utilisant des tensor arrivent à détecter les short-lasting topics (topics de courte de durée) avec succès.

De plus, les méthodes utilisant les tensors sont plus précises (accurate) pour connaitre le moment d’apparition et de disparition d’un topic. Elles sont plus précises que la LDA et le NMF.

L’article propose de faire des mesures « quantitatives » de la taille des topics. Cela servira à illustrer la performance du NCPD lorsqu’il faut trouver/découvrir des topics. Les topics utilisés seront pour certains « semi-synthetic » ou issues de données réels (real-world data). Les données comprennent des intitulés de « news » (news headlines) ainsi que des tweets liés au covid-19.

Les données « semi-synthetic » sont surement des données en partie inventées. Peut être à partir d’autre données réelles ?

## Introduction

Quel est le format des données ?

Modèle LDA :

* Modélise les topics ou les classe par probability de distribution de « set de mots »(set of words).
* Ces sets de mots évoluent selon un « scheme Bayesien » (Bayesian scheme) en donnant au « batches » une matrice « words x documents ». C’est surement un tableau contenant le nombre de fois qu’un mot apparait pour chaque document.
* Il semble qu’on reçoit en sortie deux « représentations » de type « words x topics » et « topics x documents). Cela veut surement dire qu’on a en sortie deux autres matrices créées par notre modèle. Le modèle à du créer lui-même les « topics » en analysant les mots par documents. Puis il a dû faire des associations et compter le nombre de topics par document ainsi que la relation entre un mot et un topic identifié.

Modèle NMF :

* C’est aussi un modèle qui utilise des matrices (comme le LDA). Il va prendre la matrice « words x documents » et la « décomposera » en « words x topics » et « topics x documents »
* Les documents peuvent avoir un « timestamp ». C’est surement un fichier ou un marquage qui permet d’indiquer le moment ou est publié le texte. Si le « timestamp » est ordonné en fonction du temps, la matrice « topics x document » pourra trier et présenter les topics dans cet ordre.

Il y a deux façons de faire que le NMF prenne en compte les « tranches » (slices) temporelles simultanées ou simultanément

* On peut factoriser chaque tranche temporelle

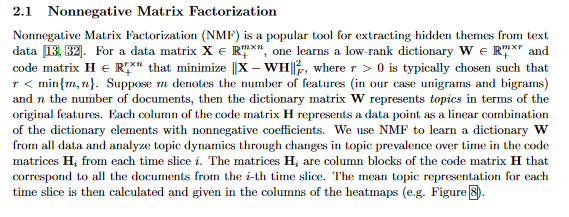
There are two basic methods of using NMF for dynamic topic modeling in a way that accounts  
to the time slices that are essentially simultaneous for the considered dynamic. First, one can  
factorize each time slice independently using NMF and obtain topic structure for each time slice  
[35, 38, 3, 36]. Second, one can concatenate all the time slices along the documents dimension and  
decompose the resulting matrix using NMF with a fixed dictionary to obtain common topics. The  
evolution of such topics is then found by computing their contribution in each time slice [18]. In  
this work, we do the latter to post-process both NMF and LDA for better visualisation and more  
interpretable results on the time scale. We discuss the methods in more details in Section 2. »

Concatenate all the time slices

On peut encoder le corpus de documents dans un tensor à 3 dimensions

Qu’est-ce que le entrywise Froebenus norm

unigrams and bigrams



Comment fait l’algorithme pour extraire des « hidden themes » d’un texte ?

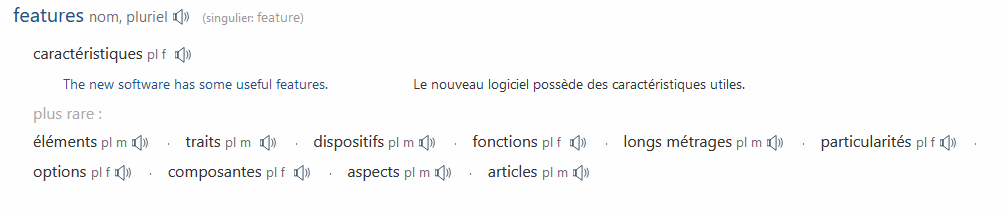
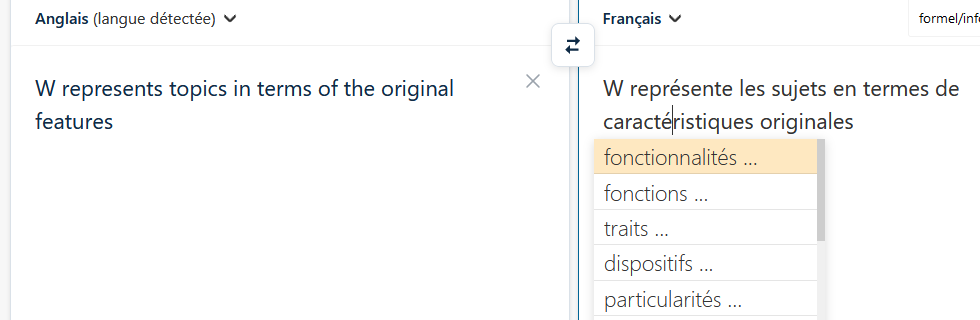
Il devrait pouvoir :

* Savoir ce qu’est un themes
* Savoir le trouver dans un texte
* Savoir l’extraire

Qu’est-ce qu’un low rank dictionary W ainsi qu’un code matrix H ?

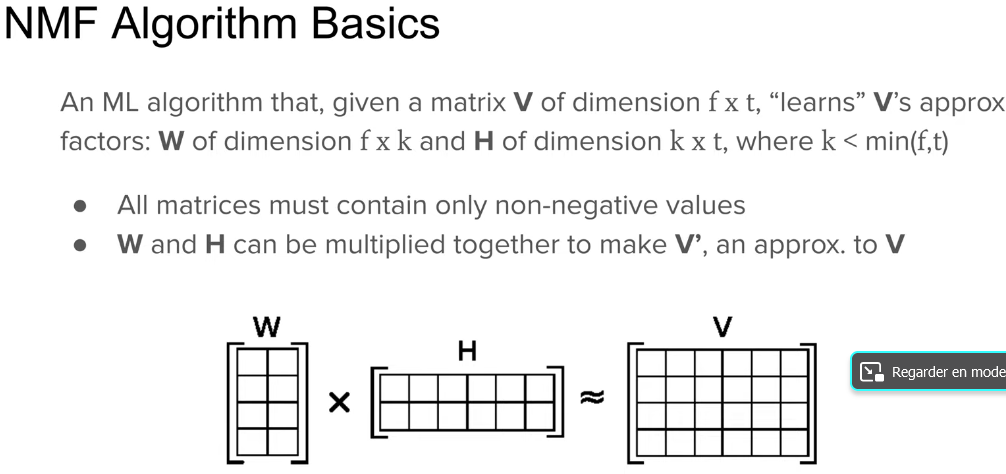
* Comment obtient-on ces deux matrices à partir de la premiere ?
* Quelle sorte d’information contiennent ces deux nouvelles matrices ?

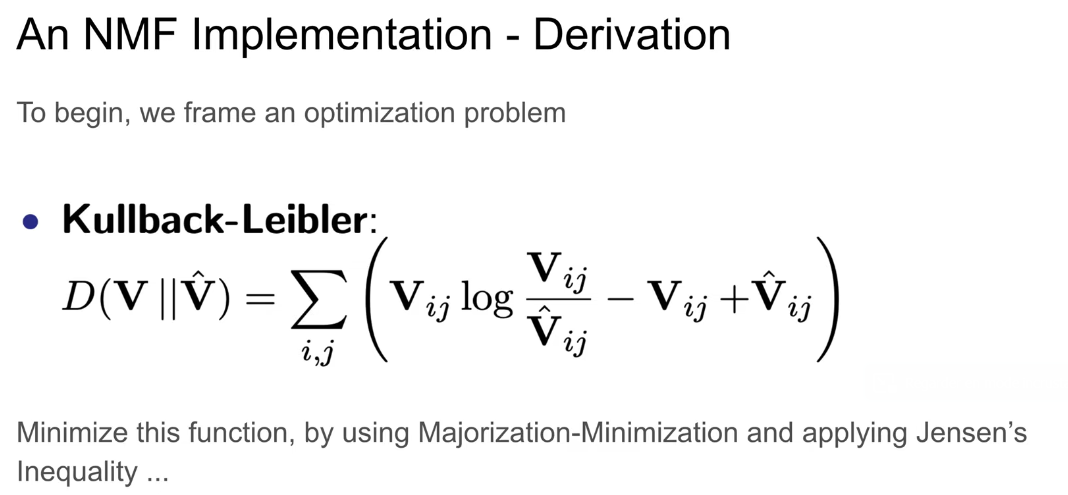
W contient les topics qui ont été repérés par notre modèle dans une étape précédente. W est de dimension m\*r. « m » étant le nombre de « features » .« r » étant choisi de sorte à être plus petit que m (nombre de features) et n.

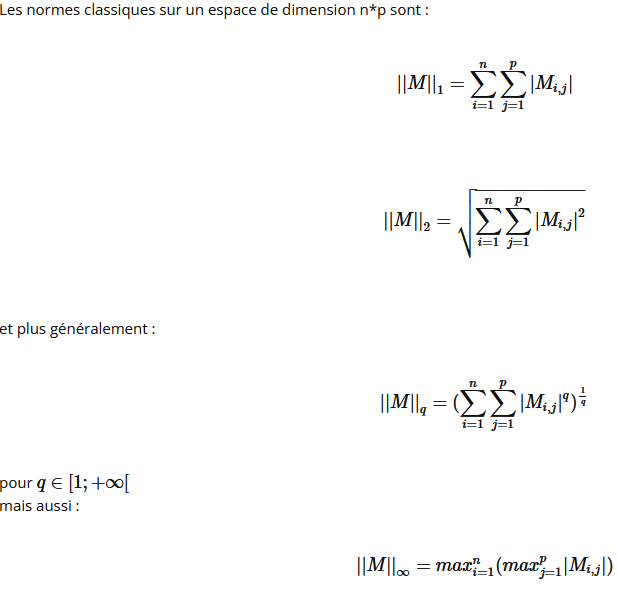
* Qu’est-ce qu’un « feature » ? On nous dit qu’ils sont ici sous forme de « unigrams and bigrams ». On nous dit aussi que W représente les topics sous forme de « original features ».
* 
* 
* J’en déduis que notre modèle après avoir trouvé les topics leur a donné des noms « features » composés d’un ou deux mots.
* 
* Wi,j représente le nombre de fois qu’un mot est utilisé dans un topic i apparait en fonction de son rang j. Soit Wi,j contient un feature. Le nombre de mot r étant
* Il semble que l’ensemble de ces matrices doit vérifier une condition, ils doivent minimiser la distance au carré entre X et WH. Je ne suis pas
* W associe à chaque topic un feature (unigram ou bigram), chaque Wi,j contient le nombre de fois que cette association est faite.
* Chaque colonne de H représente une donnée comme une combinaison linéaire d’élements du « dictionnary ».

Qu’est-ce que sont les unigrams et bigrams ?

“one learns a low-rank dictionary W” – on fait apprendre cette matrice à notre modèle ?



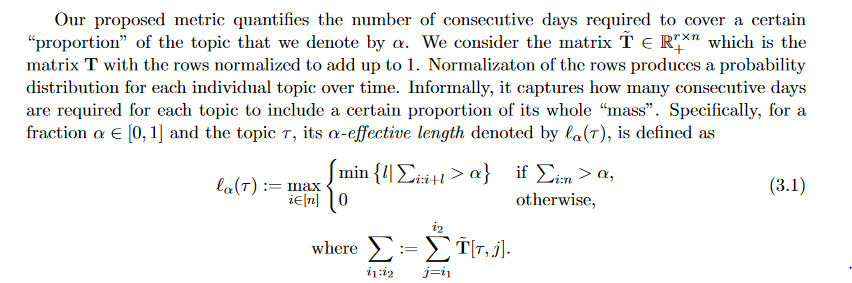




<https://www.youtube.com/watch?v=_QiTQQDrx5I>

<https://www.youtube.com/watch?v=ZspR5PZemcs>

## Calcul de la longueur d’un topic



## 01/06/2022

Recherchons dans le document des informations sur comment se fait la décomposition de la matrice initiale en deux matrices.

Rappelons qu’il faut faire plusieurs fois cette opération.

La recherche dans l’article n’a pas été fructueuse. On nous dit qu’on créé les matrices W et H mais je ne sais pas exactement comment.

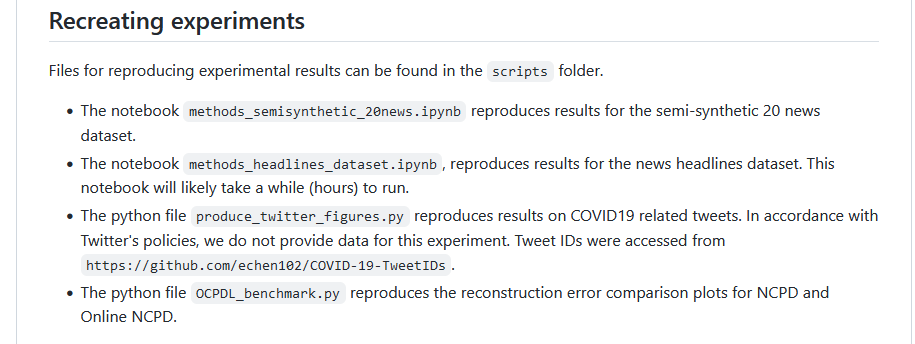
Regardons plutôt les ressources git à disposition :

* Le repo où déposer nos fichiers
* Un repo git où se trouvent le travail de Hanbaek lyu et Lara kassab sur le topic modelling
* Ainsi qu’un « notebook » qui contient un code de programme pour faire du NMF, LDA, NCPD et ONCPD.

J’ai décidé d’exécuter le code de Lyu et Lara dans un premier temps.

J’ai tout d’abord effectué une copie du repo qu’ils proposent en local puis il a fallu ajouter les données « ABC news ».

Ouvrir jupyter notebook et suivre les consignes du README.

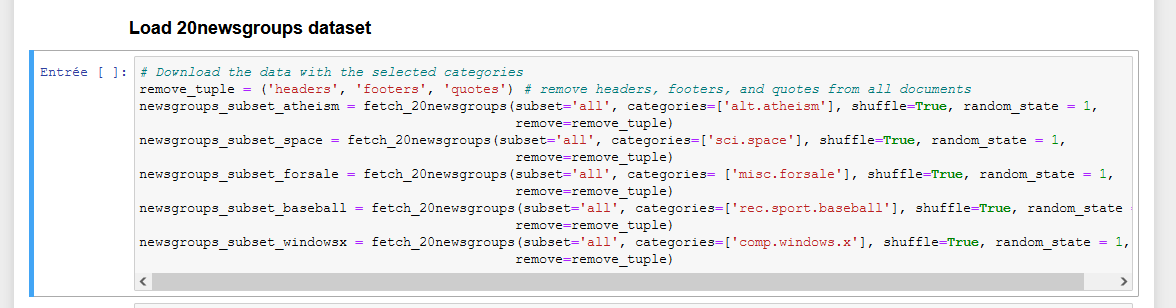


Cette partie du readme mis à disposition va nous faciliter la tache afin de tout exécuter. Nous pouvons ouvrir chacun de ces notebooks afin d’essayer une ou plusieurs méthodes de topic modeling sur des données différentes.

Voyons comment ça se passe

On décide d’exécuter le notebook « methods\_semisynthetic\_20news.ipynb »

Voyons ce qu’il fait par étape



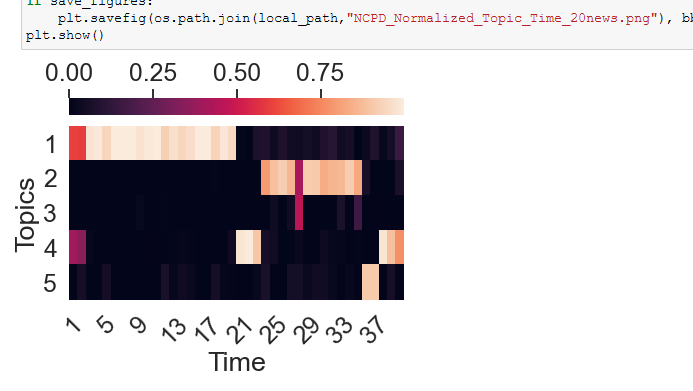
Il charge les données mais ne les fait pas d’un coup, on dirait qu’il y a une sorte de découpage et de réorganisation.

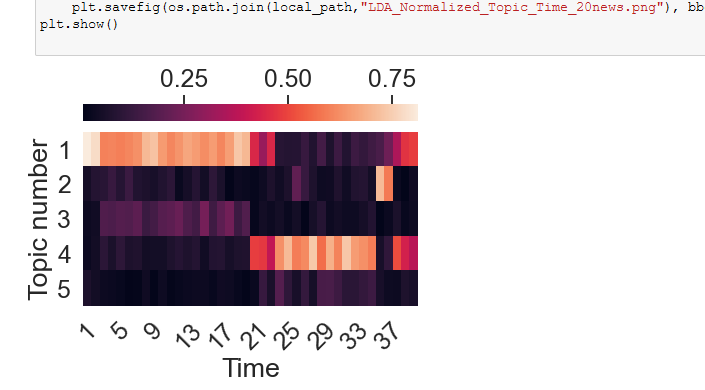
Ah on dirait qu’il ne charge pas tout le dataset mais uniquement certaines colonnes

## 03/06/2022

Nous avons pu continuer avec le même code et lancer une NMF, NCPD et LDA

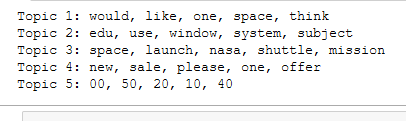






Mystères :

* La façon de découper et réorganiser les données en matrices et tensors
* La façon de choisir les topics (exemple LDA)



* La structure de départ des données

En résumé le code a eu un peu de mal à s’exécuter. Il y avait des bibliothèques à installer ainsi que quelques variables dont les noms devaient être un peu modifiées.

Essayons d’exécuter l’ONCPD en analysant de plus près les parties ou alors en montrant ce que je ne comprends pas

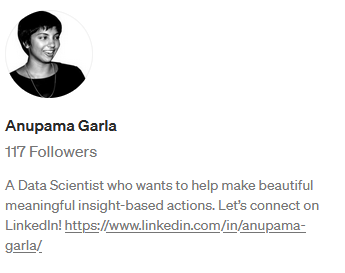
Les étapes de l’ONCPD dans le programme :

* # SVD intialization (Optional)
* # Run Online NCPD
* # Obtain the shape of the factor matrices
* # Display topics
* #Visualize topic distributions

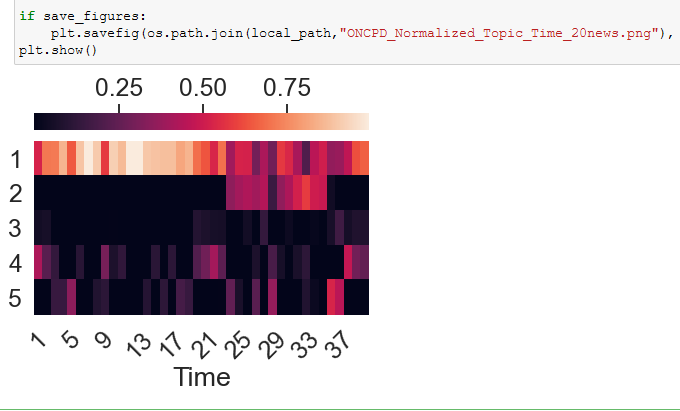
J’ai tenté de trouver une ressource expliquant la nmf et je suis tombé sur ce site :

<https://towardsdatascience.com/nmf-a-visual-explainer-and-python-implementation-7ecdd73491f8>

C’est beau. C’est de : Anupama Garla

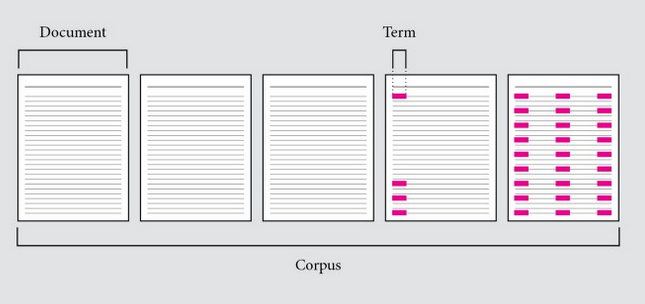


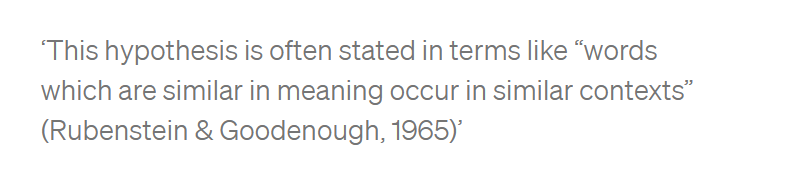
Sinon voici ce que me rend l’ ONCPD

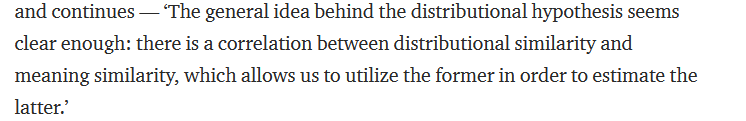


## 07/06/2022

Je suis de retour et nous allons lire l’article de anupama. Aujourd’hui j’ai malheureusement oublié ma trousse chez moi, ça ne va pas du tout. Passons.

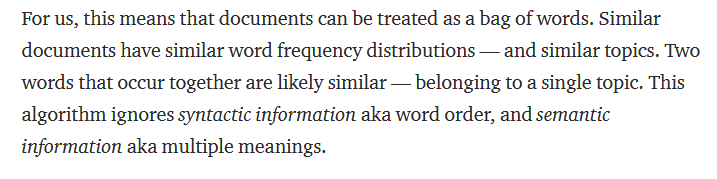


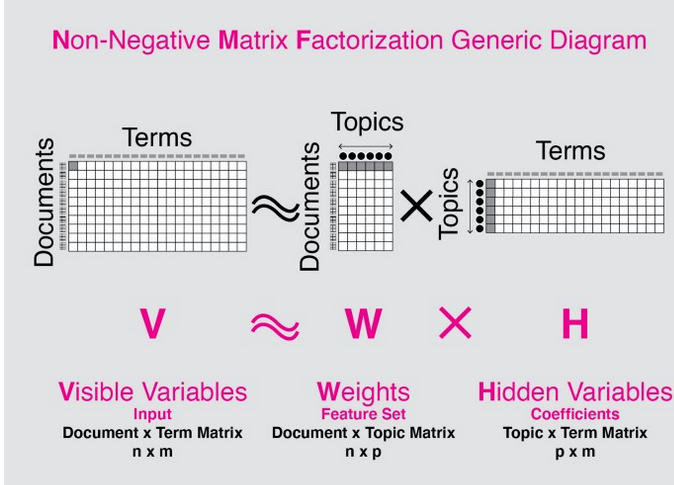




Mais c’est trop bien cette théorie.

On peut donc détecter un topic à l’aide de la distribution des termes le long d’un document ?

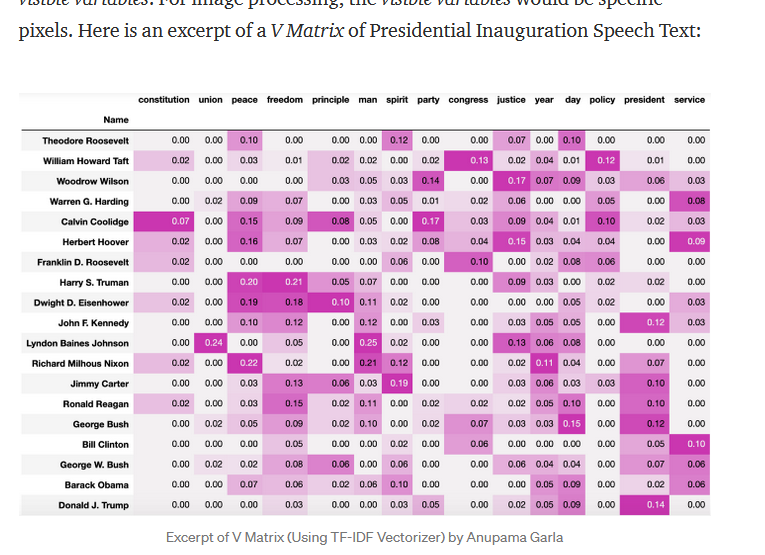




Le retour de la mystérieuse décomposition. Ce qui est mystérieux est la méthode exacte de « décomposition ». Puis comment trouve-t-on les topics ?

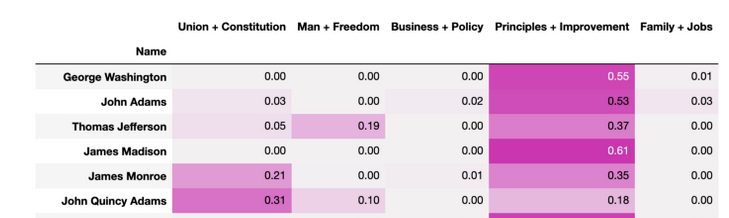
J’imaginais jusque là qu’une sorte d’analyse permet de les trouver.

Ce que je sais, c’est qu’il faut en trouver r.



Voila une chose surprenante. On a pour la première fois un exemple visuel de matrice qu’on prend en entrée pour la NMF. Cependant on y retrouve des nombres à virgule. Des proportions ?

On veut ensuite nous montrer la matrice W.



Là je reconnais les topics mis sous forme de bi-grams.

