

Matrice de document-terme :Vectorisation

Construction de la matrice document-terme en texte Mining

1. Représentation « bag-of-words »

meet.google.com/wvn-tsnw-rpe

2. Réduction de la dimensionnalité

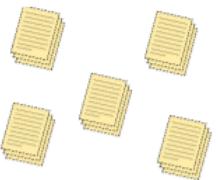
3. Pondération

4. Mesurer la similarité entre les textes

Rappel sur les principales notions



Document = individu statistique



« Base » d'apprentissage = collection de documents = **Corpus**

Enjeu : traduire la collection de documents en un tableau de données

« attributs-valeurs » propice au traitements à l'aide des algorithmes de data mining, *en minimisant au possible la perte d'information.*



Nº	V1	V2	V3	...
1				
2		?		
3				
4				

1. Que mettre en colonnes (V1, V2,...) ?
2. Quelles valeurs mettre dans le tableau ?

Transformation d'une collection de documents en un tableau de données

Représentation de Bag of words

La représentation en sac de mots : la tokenisation

Processus de « tokenization » : découpage d'un flux de caractères en phrases, symboles, mots. Nous nous intéressons plus particulièrement aux mots.

Plusieurs étapes :

1. Repérer les mots (*tokens*) présents dans les documents
2. Qui vont constituer le **dictionnaire**
3. Les mots deviennent des descripteurs (**features**, *termes*)
4. On associe alors l'absence ou la présence des mots à chaque document

L'identification du **délimiteur** de mots est important. Ce sera souvent l'espace, les ponctuations,... Certains caractères sont moins évidents (ex. « - » dans « écart-type »)

Potentiellement, le nombre de mots est très important. Il peut y avoir des redondances dans le dictionnaire, il faudra pouvoir les traiter (parfois évidentes : voiture vs. voitures ; parfois moins : voiture vs. automobile...).

Ca peut être aussi le nombre d'apparition des mots, ou autre type de valeurs. On parle de **pondération**.

La représentation en sac de mots : la tokenisation

4 documents (Coelho & Richert, 2015 ; page 55,
légèrement modifiée) :

1. Imaging databases can get huge.
2. Most imaging databases save images permanently.
3. Imaging databases store images baby.
4. Imaging databases store images. Imaging databases store images. Imaging databases store images.

Documents

Document	imaging	databases	can	get	huge	most	save	images	permanently	store	baby
1	1	1	1	1	1	0	0	0	0	0	0
2	1	1	0	0	0	1	1	1	1	0	0
3	1	1	0	0	0	0	0	1	0	1	1
4	3	3	0	0	0	0	0	3	0	3	0

- La ponctuation « . » a été retirée.
- « » a joué le rôle de séparateur.
- 4 documents → 4 lignes dans le tableau
- Le **lexique** est composé des **termes présents** dans le corpus : {imaging, databases, can, get, huge, most, save, images, permanently, store}.
- La taille du lexique n'est pas connue à l'avance. !
- Le nombre de valeurs non nulles dans la ligne dépend de la longueur du document. !

La matrice documents-termes (MDT) possède des spécificités

Document	imaging	databases	can	get	huge	most	save	images	permanently	store	baby
1	1	1	1	1	1	0	0	0	0	0	0
2	1	1	0	0	0	1	1	1	1	0	0
3	1	1	0	0	0	0	0	1	0	1	1
4	3	3	0	0	0	0	0	3	0	3	0

La MDT n'est pas un tableau de données comme les autres.

- Les valeurs du tableau sont toujours positives ou nulles. On se concentre sur les occurrences positives des termes (ex. avec les règles d'association).
- Les descripteurs (colonnes) sont exprimés dans les mêmes unités. Il n'y a pas de normalisation ou de standardisation des variables à effectuer.
- Certains termes sont présents dans peu de documents. Ou certains documents sont composés de peu de termes. De très nombreuses valeurs sont nulles.
- Mais les descripteurs sont issus des termes présents dans le corpus, une colonne composée exclusivement de la valeur 0 ne peut pas exister (sauf pondération TF-IDF)
- Il n'y a jamais de données manquantes (N/A). Un terme est forcément soit absent, soit présent dans un document.

Réduction de la dimensionnalité

Document	imaging	databases	can	get	huge	most	save	images	permanently	store	baby
1	1	1	1	1	1	0	0	0	0	0	0
2	1	1	0	0	0	1	1	1	1	0	0
3	1	1	0	0	0	0	0	1	0	1	1
4	3	3	0	0	0	0	0	3	0	3	0

Problème

La dimensionnalité est souvent très élevée. Le nombre de colonnes peut excéder le nombre de lignes. Problème pour les algorithmes de data mining souvent (ex. calcul de la matrice variance covariance). La réduction de la dimensionnalité est un enjeu crucial.

Pistes de solutions

Quelles pistes pour réduire la dimensionnalité (réduire la taille du dictionnaire) ?

- Nettoyage du corpus (ex. retrait des chiffres, correction orthographique, harmonisation de la casse, ...)
- Certains mots ne sont pas intrinsèquement porteurs de sens (ex. most) → stopwords
- Certains mots sont issus de la même forme canonique (lemmatisation) ou partagent la même racine (stemming) (ex. imaging vs. images)
- Certains mots sont synonymes ou recouvrent le même concept (ex. store vs. save)
- Certains mots apparaissent très souvent, d'autres très rarement (filtrage sur les fréquences)

Solutions pour la Réduction de la Dimensionnalité

1- Définir le dictionnaire de manière ad hoc

Dans l'idéal, on connaît les prétraitements à réaliser, on connaît la liste des mots pertinents. Tout devient facile.

1. Imaging databases can get huge.
2. Most imaging databases save images permanently.
3. Imaging databases store images baby.
4. Imaging databases store images. Imaging databases store images. Imaging databases store images.



1. image databases can get huge
2. most image databases store image permanently
3. image databases store image baby
4. image databases store image image databases store image image databases store image

Document	databases	huge	image	permanently	store
1	1	1	1	0	0
2	1	0	2	1	1
3	1	0	2	0	1
4	3	0	6	0	3

Le nombre de 0 dans la table a été réduit de manière drastique.



Sauf situations très particulières, ce type de stratégie est impraticable sur des grands corpus.

Solutions pour la Réduction de la Dimensionnalité

2- Nettoyage du texte

Le nettoyage dépend de ce que l'on souhaite exploiter par la suite, on observe les traitements suivants usuellement.

Ohio Mattress Co said its first\quarter, ending February 28, profits may be below the 2.4 mln\ndlrs, or 15 cts a share, earned in the first quarter of fiscal\n1986.\n The company said any decline would be due to expenses\related to the acquisitions in the middle of the current\quarter of seven licensees of Sealy Inc, as well as 82 pct of\nthe outstanding capital stock of Sealy.\n Because of these acquisitions, it said, first quarter sales\nwill be substantially higher than last year's 67.1 mln dlrs.\n Noting that it typically reports first quarter results in\nlate march, said the report is likely to be issued in early\nApril this year.\n It said the delay is due to administrative considerations,\nincluding conducting appraisals, in connection with the\nacquisitions.\n Reuter

Texte original



Après : harmonisation de la casse, retrait des ponctuations et de \n, retrait des nombres, retrait des espaces en trop

Texte nettoyé

ohio mattress co said its first quarter ending february profits may be below the mln dlrs or cts a share earned in the first quarter of fiscal the company said any decline would be due to expenses related to the acquisitions in the middle of the current quarter of seven licensees of sealy inc as well as pct of the outstanding capital stock of sealy because of these acquisitions it said first quarter sales will be substantially higher than last years mln dlrs noting that it typically reports first quarter results in late march said the report is likely to be issued in early april this year it said the delay is due to administrative considerations including conducting appraisals in connection with the acquisitions reuter

Solutions pour la Réduction de la Dimensionnalité

3- Utilisation d'un correcteur orthographique

Idée

Les coquilles peuvent fausser l'inventaire des termes.

Utiliser un correcteur orthographique permet de réduire la dimensionnalité.

Mode de fonctionnement d'un correcteur

L'outil s'appuie sur un dictionnaire où les termes sont correctement orthographiés. Si le mot à évaluer y est présent → OK. S'il n'y est pas, on recense les mots les plus proches. Le ou les plus proches sont proposés.

Distance entre chaînes de caractères

Une mesure spécifique aux chaînes de caractères doit être utilisé. La plus connue est celle de Levenshtein (distance d'édition).

Remarque

Attention de ne pas corriger à tort et à travers (ex. les noms propres)

Solutions pour la Réduction de la Dimensionnalité

4-Distance de Levenshtein

La **distance de Levenshtein** est une **distance**, au sens mathématique du terme, donnant une mesure de la différence entre deux chaînes de caractères. Elle est égale au nombre minimal de caractères qu'il faut supprimer, insérer ou remplacer pour passer d'une chaîne à l'autre.[wikipédia]

```
> adist("totor",c("toto","tata","tutu","tonton","tantine"))
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    3    2    5
```



La fonction renvoie un vecteur contenant la distance du mot à tester « `totor` » avec chaque mot du dictionnaire (« `toto` », « `tata` », ...).

Solutions pour la Réduction de la Dimensionnalité

Retrait des stopwords (mots vides)

Un mot vide est un mot communément utilisé dans une langue, non porteur de sens dans un document (ex. préposition, pronoms, etc.). Formellement, sa fréquence d'apparition est la même dans tous les documents. De fait, les mots vides ne permettent pas de discriminer les documents (de distinguer les documents les uns des autres), ils sont inutilisables en text mining tel que nous le concevons (catégorisation, recherche d'information, etc.) ([Wikipédia](#)).

Exemple de mots vides en français (<http://www.ranks.nl/stopwords/french>):

avoir, bon, car, ce, cela, ces, ceux, chaque, ci, comme, comment, dans, des, du, dedans, dehors, depuis, devrait, doit, donc, dos, début, elle, elles, en, encore, essai, est, et, eu, fait, faites, fois, font, hors, ici, il, ils, je, juste, la, le, les, leur, là, ma, maintenant, mais, mes, mine, moins, mon, mot, même, ni, nommés, notre, nous, ou, où, par, parce, pas, peut, peu, plupart, pour, pourquoi, quand, que, quel, quelle, quelles, quels, qui, sa, sans, ses, seulement, si, sien, son, sont, sous, soyez, sujet, sur, ta, tandis, tellement, tels, tes, ton, tous, tout, trop, très, tu, voient, vont, votre, vous, vu, ça, étaient, état, étions, été, être

Nous pouvons compléter ou restreindre la liste des mots-clés en fonction du domaine d'étude ou du contexte dans lequel nous nous situons.



Solutions pour la Réduction de la Dimensionnalité

Retrait des stopwords (mots vides)

Texte nettoyé

ohio mattress co said its first quarter ending february profits may be below the mln dtrs or cts a share earned in the first quarter of fiscal the company said any decline would be due to expenses related to the acquisitions in the middle of the current quarter of seven licensees of sealy inc as well as pct of the outstanding capital stock of sealy because of these acquisitions it said first quarter sales will be substantially higher than last years mln dtrs noting that it typically reports first quarter results in late march said the report is likely to be issued in early april this year it said the delay is due to administrative considerations including conducting appraisals in connection with the acquisitions reuter

125 termes

Texte après retrait des stopwords en anglais

ohio mattress co said first quarter ending february profits may mln dtrs cts share earned first quarter fiscal company said decline due expenses related acquisitions middle current quarter seven licensees sealy inc well pct outstanding capital stock sealy acquisitions said first quarter sales will substantially higher last years mln dtrs noting typically reports first quarter results late march said report likely issued early april year said delay due administrative considerations including conducting appraisals connection acquisitions reuter

76 termes



$$\frac{125-76}{125} = 39.2\% \text{ des termes ont été retirés.}$$

Solutions pour la Réduction de la Dimensionnalité

Attention : Les stop word peuvent avoir une signification !!!!

Les « stop word » sont en très grande partie composée de mots qui n'ont pas de sens en eux-mêmes, mais qui sont utilisés dans la construction des phrases (ex. prépositions, pronoms, verbes auxiliaires, articles). Ils sont caractéristiques d'une langue et peuvent être utilisés pour les identifier.

Algorithme

Entrée : document, liste des mots vides pour plusieurs langues

Sortie : la langue identifiée

Pour chaque langue :

Compter les occurrences des mots vides de la langue dans le document

La langue identifiée est celle pour laquelle le nombre de correspondance est le plus élevé



Attention, si la taille de la liste des mots vides est différente d'une langue à l'autre, il faut en tenir compte.

Il existe d'autres approches pour la détection de la langue, dont celles basées sur un apprentissage à partir de suites de caractères caractéristiques (n-grams, cf. plus loin).

Solutions pour la Réduction de la Dimensionnalité

La lemmatisation

La lemmatisation consiste à analyser les termes de manière à identifier sa forme canonique (lemme), qui existe réellement. L'idée est de réduire les différentes formes (pluriel, féminin, conjugaison, etc.) en une seule.

Ex. am, are, is → be
car, cars, car's, cars' → car { Ainsi, la phrase « the boy's cars are different colors » devient « the boy car be differ color ».

La technique fait à la fois référence à un dictionnaire, et à l'analyse morphosyntaxique des mots (ex. Weiss et al., 2005 ; page 24). Elle est spécifique à chaque langue. Des erreurs sont toujours possibles !

Exemple testé sur : http://www.jerome-pasquelin.fr/tools/outil_lemmatisation.php

Mignonne, pourquoi es-tu partie si loin de nos avenirs radieux, ceux où nous devions regarder le soleil se lever au milieu des chants des boeufs ?

Le mot mignon regroupe : mignon, mignonne
Le mot etre regroupe : etre, es
Le mot taire regroupe : taire, tu
Le mot partir regroupe : partir, partie
Le mot notre regroupe : notre, nos
Le mot celui regroupe : celui, ceux
Le mot devoir regroupe : devoir, devions
Le mot du regroupe : du, des
Le mot chant regroupe : chant, chants
Le mot boeuf regroupe : boeuf, boeufs

Rahma Boujelben

Solutions pour la Réduction de la Dimensionnalité

Le stemming ou (la racinisation)

Le stemming consiste à réduire un mot à sa racine (stem), **qui peut ne pas exister**.

L'algorithme de Porter est un des plus connus pour la langue anglaise. Il applique une succession de règles (mécaniques) pour réduire la longueur des mots c.-à-d. supprimer la fin des mots

```
ohio mattress co said first quarter ending february profits may  
mln dlrs cts share earned first quarter fiscal company said  
decline due expenses related acquisitions middle current  
quarter seven licensees sealy inc well pct outstanding capital  
stock sealy acquisitions said first quarter sales will substantially  
higher last years mln dlrs noting typically reports first quarter  
results late march said report likely issued early april year said  
delay due administrative considerations including conducting  
appraisals connection acquisitions reuter
```

Avant stemming : 549 caractères

```
ohio mattress co said first quarter end februari profit may mln dlrs  
cts share earn first quarter fiscal compani said declin due expens  
relat acquisit middl current quarter seven license seal inc well pot  
outstand capit stock seali acquisit said first quarter sale will  
substanti higher last year mln dlrs note typic report first quarter  
result late march said report like issu earli april year said delay due  
administr consider includ conduct apprais connect acquisit reuter
```

Après stemming : 477 caractères

- Le stemming est un traitement final, qui n'autorise plus de post-traitements sur les mots
- Le stemming peut conduire à des regroupements erronés (ex. marmite_marmaille → marm)

Solutions pour la Réduction de la Dimensionnalité

Retour à l'exemple

Document	imaging	databases	can	get	huge	most	save	images	permanently	store	baby
1	1	1	1	1	1	0	0	0	0	0	0
2	1	1	0	0	0	1	1	1	1	0	0
3	1	1	0	0	0	0	0	1	0	1	1
4	3	3	0	0	0	0	0	3	0	3	0

Dictionnaire initial = {baby, can, databases, get, huge, images, imaging, most, permanently, save, store} → 11 termes

Dictionnaire après retrait des stopwords = {baby, can, databases, get, huge, images, imaging, permanently, save, store} → 10 termes

Dictionnaire après stemming = {babí, can, databas, get, huge, imag, perman, save, store} → 9 termes



Document	imag	databas	can	get	huge	save	perman	store	babi
1	1	1	1	1	1	0	0	0	0
2	2	1	0	0	0	1	1	0	0
3	2	1	0	0	0	0	0	1	1
4	6	3	0	0	0	0	0	1	0

Solutions pour la Réduction de la Dimensionnalité

→ **N_gram sur les mots**

Un n-gramme (*n-gram* en anglais) est une séquence de *n* termes adjacents (consécutifs ou apparaissant dans une fenêtre restreinte) que l'on extrait en tant que descripteur d'un corpus. L'idée est que l'association des termes introduit une signification différente de celles qu'ils véhiculent individuellement.

Ex. bi-gramme « écart type », « bien être », « bon vivant »,... ; trigramme « bec de lièvre »

→ Le problème est de pouvoir les identifier. On ne peut pas tous les tester. On peut se baser sur les cooccurrences, fréquences ou corrélations,... mais en tenant compte de l'adjacence des termes.

→ $n > 2$, la complexité de calcul devient très vite pesante, et la pertinence n'est pas toujours au rendez vous,

Solutions pour la Réduction de la Dimensionnalité

→ **N_gram sur les lettres**

Un n-gramme (*n-gram* en anglais) est une séquence de *n* caractères consécutifs (contigus) que l'on extrait en tant que descripteur d'un corpus.

Exemple de 4-grams extrait du document « image databases can get huge »
→ ‘imag’, ‘mage’, ‘age’, ‘ge d’, ‘e da’, ‘dat’, ‘data’, ‘atab’, ...

Quel intérêt ?

- C'est une manière d'effectuer une racinisation (stemming) très simplement
- L'approche est indépendante de la langue... mais l'interprétation des résultats n'est pas toujours évidente

En pratique

- Elle donne d'excellents résultats dans les applications, par exemple dans la détection des spams, des plagiats,...
- L'approche peut s'étendre à des domaines où une tokenisation naturelle n'est pas possible et/ou trop limitée (ex. séquence ADN, description des protéines à l'aide des acides aminés, etc.)

Solutions pour la Réduction de la Dimensionnalité

Filtrage par la fréquence

Fréquence : nombre de documents où le terme apparaît au moins une fois rapporté au nombre total de documents.

Document	databases	huge	image	permanently	store
1	1	1	1	0	0
2	1	0	2	1	1
3	1	0	2	0	1
4	3	0	6	0	3

Fréquence trop élevée (termes présents dans presque tous les documents) : permet peut-être de cerner le domaine, mais ne permet pas de différencier les documents (ex. databases, image).

Fréquence trop faible (termes présents dans de très rares documents) : ne permet pas de caractériser une différence significative entre les documents.



Le choix des seuils reste arbitraire.

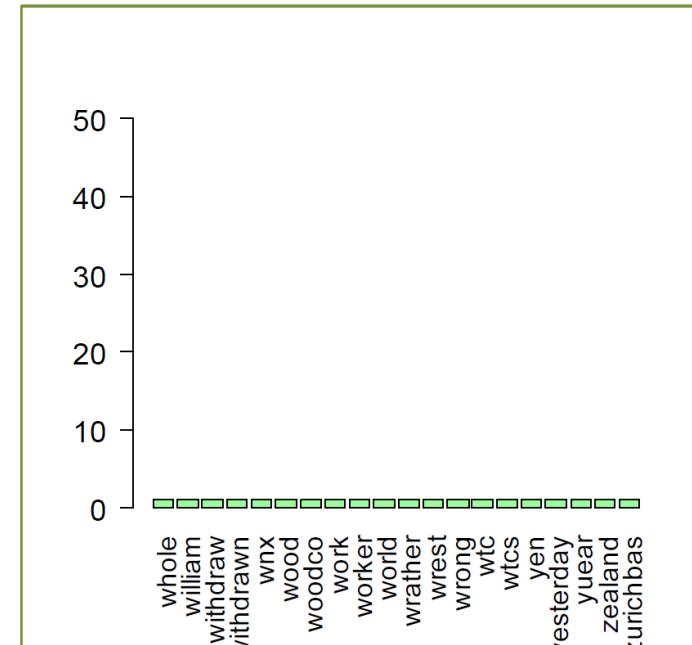
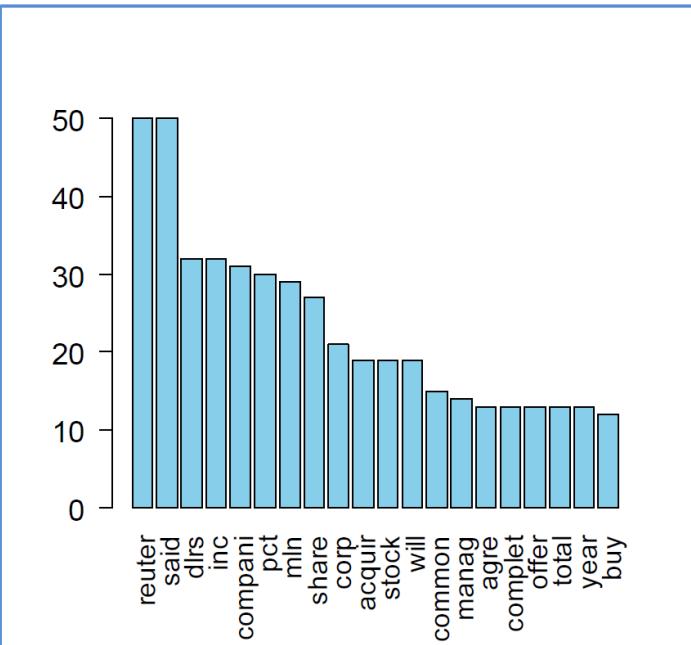
Solutions pour la Réduction de la Dimensionnalité

Filtrage par la fréquence

50 documents. Après prétraitements, le dictionnaire est composé de 1193 termes.

➡ 2 apparaissent dans tous les documents

➡ 761 n'apparaissent que dans un seul document



Il ne reste plus que 430 termes si on les retire.

Pondération : Chaque document est représenté par un vecteur : *featurevector*

Pondération : pondération binaire

Comptabiliser la présence de chaque terme dans le document,
sans se préoccuper du nombre d'occurrences (de la répétition)

1. image databases can get huge
2. most image databases store image permanently
3. image databases store image baby
4. image databases store image image databases
store image image databases store image



Document	databases	huge	image	permanently	store
1	1	1	1	0	0
2	1	0	1	1	1
3	1	0	1	0	1
4	1	0	1	0	1

Avantages

- Simplicité
- Forme de « lissage » de l'information en donnant la même importance à tous les termes
- Adaptée à certaines techniques (ex. règles d'association) et mesures de distance (ex. Jaccard)

Inconvénients

- Une partie de l'information n'est pas captée (perte d'information), dont pourrait tirer profit certaines catégories de techniques statistiques
- Pourquoi donner la même importance à tous les termes ? Rahma Boujelben

Pondération : Fréquence TF

Pour un document, comptabiliser le nombre d'occurrence des termes. **Indicateur de l'importance du terme dans le document.**

1. image databases can get huge
2. most image databases store image permanently
3. image databases store image baby
4. image databases store image image databases
store image image databases store image



Document	databases	huge	image	permanently	store
1	1	1	1	0	0
2	1	0	2	1	1
3	1	0	2	0	1
4	3	0	6	0	3

Avantages

- On capte plus d'information, la répétition d'un terme dans le document est prise en compte
- Des techniques savent prendre en compte ce type d'information (calcul matriciel)

Inconvénients

- Les écarts entre documents sont exagérés (ex. si on utilise une distance euclidienne)
- On ne tient pas compte de la prévalence des termes dans l'ensemble des documents (cf. IDF)

Pondération : Fréquence TF

Pour un document, comptabiliser le nombre d'occurrence des termes. **Indicateur de l'importance du terme dans le document.**

1. image databases can get huge
2. most image databases store image permanently
3. image databases store image baby
4. image databases store image image databases
store image image databases store image



Document	databases	huge	image	permanently	store
1	1	1	1	0	0
2	1	0	2	1	1
3	1	0	2	0	1
4	3	0	6	0	3

Avantages

- On capte plus d'information, la répétition d'un terme dans le document est prise en compte
- Des techniques savent prendre en compte ce type d'information (calcul matriciel)

Inconvénients

- Les écarts entre documents sont exagérés (ex. si on utilise une distance euclidienne)
- On ne tient pas compte de la prévalence des termes dans l'ensemble des documents (cf. IDF)

Pondération : Fréquence TF _Différentes normalisations

Document	databases	huge	image	permanently	store
1	1	1	1	0	0
2	1	0	2	1	1
3	1	0	2	0	1
4	3	0	6	0	3

On simplifie en considérant que seuls ces termes sont apparus.

Normalisation logarithmique

$$tf(t, d) = \begin{cases} 0 & \text{si } f_{t,d} = 0 \\ 1 + \log_{10} f_{t,d} & \text{sinon} \end{cases}$$

Double normalisation 0.5

$$tf(t, d) = 0.5 + 0.5 \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$$

On pondère la fréquence par le nombre d'apparition du terme le plus fréquent du document (une manière de tenir compte de la longueur)

Normalisation simple

$$tf(t, d) = \frac{f_{t,d}}{\sum_{t'} f_{t',d}}$$

On pondère la fréquence par le nombre de termes présents dans le document (une autre manière de tenir compte de la longueur)

Les normalisations des fréquences permettent d'amortir les écarts et/ou de tenir compte de la longueur des documents.



Document	databases	huge	image	permanently	store
1	1.00	1.00	1.00	0.00	0.00
2	1.00	0.00	1.30	1.00	1.00
3	1.00	0.00	1.30	0.00	1.00
4	1.48	0.00	1.78	0.00	1.48



Document	databases	huge	image	permanently	store
1	1.00	1.00	1.00	0.50	0.50
2	0.75	0.50	1.00	0.75	0.75
3	0.75	0.50	1.00	0.50	0.75
4	0.75	0.50	1.00	0.50	0.75



Document	databases	huge	image	permanently	store
1	0.33	0.33	0.33	0.00	0.00
2	0.20	0.00	0.40	0.20	0.20
3	0.25	0.00	0.50	0.00	0.25
4	0.25	0.00	0.50	0.00	0.25

Pondération : IDF

Un terme présent dans presque tout le corpus (D) influe peu quand il apparaît dans un document. A l'inverse, un terme rare apparaissant dans un document doit retenir notre attention. **L'IDF mesure l'importance d'un terme dans un corpus.**

$$idf(t, D) = \log_{10} \frac{N}{n_t}$$

N : nombre de documents dans le corpus

n_t : nombre de documents où le terme apparaît

Document	databases	huge	image	permanently	store
1	1	1	1	0	0
2	1	0	2	1	1
3	1	0	2	0	1
4	3	0	6	0	3
n_t	4	1	4	1	3
$idf(t, D)$	0.000	0.602	0.000	0.602	0.125

« *databases* » et « *image* » sont des termes « évidents » qui ne peuvent pas servir à la différenciations des documents.

Le filtrage sur les fréquences permet d'éviter l'influence exagérée des termes très rares.

Remarque : Une formule lissée est utilisée parfois pour éviter d'obtenir un IDF nul

$$idf(t, D) = \log_{10} \left(1 + \frac{N}{n_t} \right)$$

Rahma Boujelben

Pondération : TF*IDF

Relativiser l'importance d'un terme dans un document (TF) par son importance dans le corpus (IDF).

$$\text{tfidf}(t, d, D) = tf(t, d) \times idf(t, D)$$

Dans sa forme la plus utilisée
(mais des variantes sont possibles)

$$\text{tfidf}(t, d, D) = f_{t,d} \times \log_{10} \frac{N}{n_t}$$

Le TF-IDF d'un terme dans un document est élevé quand :

Le terme apparaît beaucoup dans le document.

Il se fait rare par ailleurs.



Document	databases	huge	image	permanently	store
1	0.00	0.60	0.00	0.00	0.00
2	0.00	0.00	0.00	0.60	0.12
3	0.00	0.00	0.00	0.00	0.12
4	0.00	0.00	0.00	0.00	0.37

Remarque : Dans le package 'tm', par défaut nous avons :

$$\text{tfidf}(t, d, D) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \times \log_2 \frac{N}{n_t}$$

Pondération : TF*IDF

Python

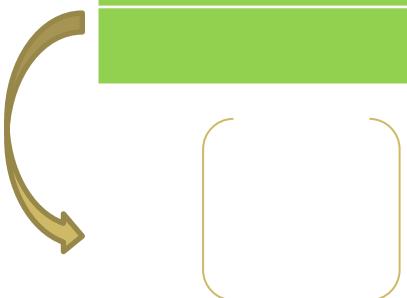
```
from sklearn.feature_extraction.text import TfidfVectorizer, TfidfTransformer
train = tv.fit_transform(X_train)
print(tv.get_feature_names())
# ngram de lettre
tv = TfidfVectorizer(ngram_range=(3,4),use_idf=True,analyzer='char')
# ngram de word
tv = TfidfVectorizer(ngram_range=(1,1),use_idf=True,analyzer='word')
```

Pondération : TF*IDF

Python

```
from sklearn.feature_extraction.text import TfidfVectorizer, TfidfTransformer  
tv = TfidfVectorizer(min_df = 0.05, max_df = 0.5, max_features=1500,  
stop_words = 'english')  
X = tv.fit_transform(new['review_tilte'])  
vocab = tv.get_feature_names()
```

review title	Rate	Classe
Texte	Float	1
Texte	Float	0
Texte	Float	-1
		...



Pondération : TF*IDF

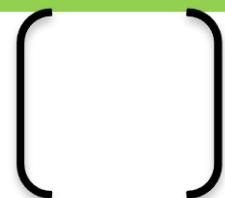
Python

I
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
new['rate'] = le.fit_transform(new['rate'])

II
from scipy.sparse import hstack
X_train_dtm = hstack((X,np.array(new['rate'])[:,None]))

```
print(X.shape)  
print(X_train_dtm.shape)
```

review title	Rate	Classe
Texte	Float	1
Texte	Float	0
Texte	Float	-1
		...



Mesure de similarité

Qu'est-ce que la similarité textuelle ?

Mesurer la similarité entre deux phrases consiste à évaluer jusqu'à quel point le sens de ces phrases est proche en termes de proximité de surface “similarité lexicale” et de signification “similarité sémantique”.

Pourquoi calculer la similarité sémantique entre les textes?

- Les moteurs de recherche doivent modéliser la pertinence d'un document pour une requête. Par exemple, les sites de questions-réponses tels que les forums doivent déterminer si une question a déjà été posée auparavant.
- Dans les services à la clientèle, le système d'IA devrait être en mesure de comprendre les requêtes sémantiquement similaires des utilisateurs et de fournir une réponse uniforme. Par exemple, si l'utilisateur demande “Qu'est-il arrivé à ma livraison” ou “Ma commande n'a pas encore été livrée.”, l'utilisateur attendra la même réponse.

Les mesures de similarité sont sous-jacents à de nombreuses méthodes de data mining (visualisation, classification supervisée et non supervisée). Elles caractérisent les ressemblances entre les objets.

Dans le cas particulier du text mining, nous devons mesurer des similarités entre documents

1. image databases can get huge
2. most image databases store image permanently
3. image databases store image baby
4. image databases store image image databases store image image databases store image

Dans quelle mesure les documents 1 et 2 se ressemblent ?

Documents qui ont été traduits en vecteurs de pondération (feature vector)

Document	databases	huge	image	permanently	store
1	1	1	1	0	0
2	1	0	2	1	1
3	1	0	2	0	1
4	3	0	6	0	3

Sachant que les calculs doivent être effectués à partir des valeurs fournies par la matrice documents-termes.

Propriété des mesures de similarité

Propriétés d'une mesure de similarité
entre deux vecteurs u et v

Non-négativité $s(u, v) \geq 0 \quad \forall u, v$

Un indice de similarité
est toujours positif.

Symétrie $s(u, v) = s(v, u) \quad \forall u, v$

La similarité ne dépend pas de
l'ordre de présentation de l'objet

Maximalité $s(u, u) = s(v, v) = s_{\max} \quad \forall u \neq v$

L'indice prend sa valeur max quand
on compare un objet avec lui-même

Normalisation $\begin{cases} s(u, v) = 1 \Leftrightarrow u = v \quad \forall u, v \\ s(u, v) < 1 \end{cases}$

L'indice est borné à 1. Il prend cette
valeur uniquement lorsqu'on
compare un objet avec lui-même

Dissimilarité

Une mesure de dissimilarité caractérise les différences entre les objets.

Déduire une dissimilarité à partir d'une similarité

$$d(u, v) = 1 - s(u, v)$$

Une distance est aussi une mesure de dissimilarité

$$\begin{cases} d(u, v) = 0 \Leftrightarrow u = v \\ d(u, v) = d(v, u) \end{cases}$$

Mesures de similarité : Distance Euclidienne

Mesure très populaire en data mining...

$$d(u, v) = \sqrt{\sum_{j=1}^p (u_j - v_j)^2}$$

p est le nombre de termes

u_j (v_j) est la pondération du terme j pour l'objet u (v)

Document	databases	huge	image	permanently	store
1	1	1	1	0	0
2	1	0	2	1	1
3	1	0	2	0	1
4	3	0	6	0	3



$$d(3,4) = \sqrt{(1-3)^2 + \dots + (1-3)^2} = 4.9$$

- ... mais problématique dans le contexte du text mining car ne tient pas compte de la longueur des documents (documents courts : nombreuses valeurs à 0)
- La distance n'est pas bornée ($d \geq 0$, mais pas de limite max.), peu adaptée aux grandes dimensions

Mesures de similarité : Similarité et Distance Cosinus

Très populaire en text mining car ne s'intéresse qu'aux cooccurrences, la normalisation permet de comparer des documents de longueurs différentes. !

$$\cos(u, v) = \frac{\langle u, v \rangle}{\|u\| \cdot \|v\|}$$

$0 \leq \cos \leq 1$ puisque les pondérations sont toujours positives ou nulles. Nous avons donc une mesure normalisée.

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

Mesures de similarité : Similarité et Distance Cosinus

Document	databases	huge	image	permanently	store
1	1	1	1	0	0
2	1	0	2	1	1
3	1	0	2	0	1
4	3	0	6	0	3



$$\cos(3,4) = \frac{1 \times 3 + \dots + 1 \times 3}{\sqrt{1^2 + \dots + 1^2} \times \sqrt{3^2 + \dots + 3^2}} = 1$$

Malgré le choix de pondération qui n'est pas très judicieux ici, cos détecte bien la ressemblance entre les documents 3 et 4

Déduire une distance à partir
de la similarité cosinus

$$d(u, v) = 1 - \cos(u, v)$$

$0 \leq d \leq 1$ la distance
est normalisée (bornée)

Mesures de similarité : Indice de Jaccard

→ Similarité entre des ensembles binaires

Indice très populaire. Adapté à la pondération binaire. S'intéresse aux cooccurrences et bénéficie également d'un mécanisme de normalisation.

$$J(u, v) = \frac{|u \cap v|}{|u \cup v|} = \frac{M_{11}}{p - M_{00}}$$

p est le nombre de termes

M_{11} est le nombre de co-présence

M_{00} est le nombre de co-absence

($0 \leq J \leq 1$) par définition

Document	databases	huge	image	permanently	store
1	1	1	1	0	0
2	1	0	1	1	1
3	1	0	1	0	1
4	1	0	1	0	1



$$J(2,3) = \frac{3}{4} = \frac{3}{5-1} = 0.75$$

De même ici, on constate que $J(3,4) = 1$.

Déduire une distance à partir
de l'indice de Jaccard

$$d(u, v) = 1 - J(u, v)$$

$0 \leq d \leq 1$ la distance
est normalisée (bornée)