L2 - Données web

Francois.Rioult@unicaen.fr

6 mars 2019

Chapitre 1

Ingrédients pour un moteur de recherche

1.1 TF-iDF

On dispose d'un *corpus* constitué d'une liste de *texte*, issus de l'aspiration de pages web. Les éléments HTML ont été retirés, mais il subsiste une mise en forme avec de la ponctuation et des lignes vides pour délimiter les paragraphes.

Pour valoriser ces textes, un moteur de recherche est mis en place. Il répond à une requête de l'utilisateur, qui est une liste de mots. Dans un premier temps, le moteur doit renvoyer la liste des textes contenant les mots de la requête, mais ce n'est pas suffisant pour l'utilisateur : il faut lui présenter d'abord les documents les plus *pertinents*.

1.1.1 Pretinence d'un mot dans un document

Pour un texte retourné par le moteur, la pertinence d'un mot de la requête, relativement au corpus, est défini par le score TF-iDF du mots, construit à partir des mesures suivantes :

1. soit $TF(word, doc)$ (*term frequency*) le nombre d'occurrences du mot dans le document
2. soit $DF(word)$ (*document frequency*) le nombre de documents dans lequel le mot apparaît
3. soit $nDocument$ le nombre de documents du corpus

Un mot est d'autant plus pertinent dans un document que :
— sa fréquence est élevée ;
— le nombre de document qui le contient est faible.

La pertinence doit donc croître avec le TF mais décroître avec le DF, soit croître avec l'inverse du DF. Dans la pratique, on utilise la formule suivante :

$$TFiDF(word, doc) = TF(word, doc) * \log \frac{nDocuments}{DF(word)}$$

Quelques propriétés

- le TF-iDF est toujours positif ;
- le TF-iDF n'a pas de borne supérieure ;
- un mot présent dans *tous* les documents a un iDF nul ;
- le TF-iDF permet de comparer :
  - plusieurs documents pour un mot donné ;
  - plusieurs mots à l'intérieur d'un document.

1.1.2 Exemple

Tiré de `https://fr.wikipedia.org/wiki/TF-IDF`
On donne trois documents :

1. Son nom est célébré par le bocage **qui** frémit, et par le ruisseau **qui** murmure, ces vents l'emportent jusqu'à l'arc céleste, l'arc **de** grâce et **de** consolation que sa main tendit dans **les** nuages.
2. À peine distinguait-on deux buts à l'extrémité **de** la carrière : des chênes ombrageaient l'un, autour **de** l'autre des palmiers se dessinaient dans l'éclat du soir.
3. Ah ! le beau temps **de** mes travaux poétiques ! **les** beaux jours que j'ai passés près **de** toi ! **Les** premiers, inépuisables **de** joie, **de** paix et **de** liberté ; **les** derniers, empreints d'une mélancolie **qui** eut bien aussi ses charmes.

mots	iDF	TF_1	TF_2	TF_3
qui	$\log \frac{3}{2}$	2	0	1
les	$\log \frac{3}{2}$	1	0	3
de	0	2	2	5
autour	$\log \frac{3}{1}$	0	1	0

On constate :

- le document 1 est le plus pertinent pour *qui*
- dans le document 3, le mot le plus pertinent est *les*
- $TF - iDF(qui, 2) = 0.34, TF - iDF(autour, 2) = 0.47$

1.2 Pertinence d'un document pour une requête

Il est nécessaire de déterminer, parmi les documents qui contiennent la requête, ceux qui ont la meilleure similarité avec la requête. La requête va donc être considérée comme un document, à comparer aux autres.

Un document est représenté par le vecteur des TF-iDF des mots de la requête. La similarité entre deux vecteurs est donnée par le cosinus [1] de l'angle entre les deux vecteurs, qui fournit une valeur entre -1 et 1 :

- 1 : c'est la meilleure similarité
- 0 : les vecteurs sont orthogonaux, il n'y a rien de plus dissemblable
- -1 : les vecteurs suivent des directions opposées

1. dit de Salton

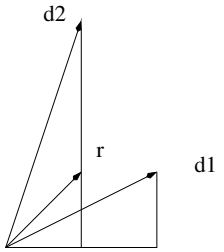

FIGURE 1.1 – Vecteurs des documents et requête

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

Pour le vecteur de la requête, on prend la liste des iDF des mots ($TF = 1$).

Exemple

Sur l'exemple précédent, on suppose que la requête de l'utilisateur est *qui les*. À $\log \frac{3}{2}$ près :

— Les coordonnées de d_1 selon les mots (qui, les) sont $d_1 = (2, 1)$
— les coordonnées de d_2 sont $d_2 = (1, 3)$
— les coordonnées de la requête sont $r = (1, 1)$
— $\cos(d_1, r) = \frac{2+1}{\sqrt{(4+1)}\sqrt{(1+1)}} = 0.94$ soit 18^o
— $\cos(d_2, r) = \frac{1+3}{\sqrt{(1+9)}\sqrt{(1+1)}} = 0.89$ soit 26^o

1.3 Discussion sur le TF-iDF

Quelle base pour le logarithme ? Normalement ce devrait être une base 2, mais cela a peu d'importance dans la pratique, car on *compare* deux TF-iDF, on n'examine pas cette mesure dans l'absolu : un facteur linéaire ne modifie pas les relations.

Pourquoi utiliser un logarithme ?

— il y a un rapport direct avec la loi de Zipf : on constate empiriquement que la fréquence des mots suit une loi de puissance de leur rang (voir section suivante).
— on veut pouvoir additionner les scores de plusieurs mots
— il y a un rapport avec l'entropie de Shannon.

1.4 Lois scalantes

Ce sont des lois :

— constatées empiriquement
— lieu de phénomène d'invariance échelle (fractal)

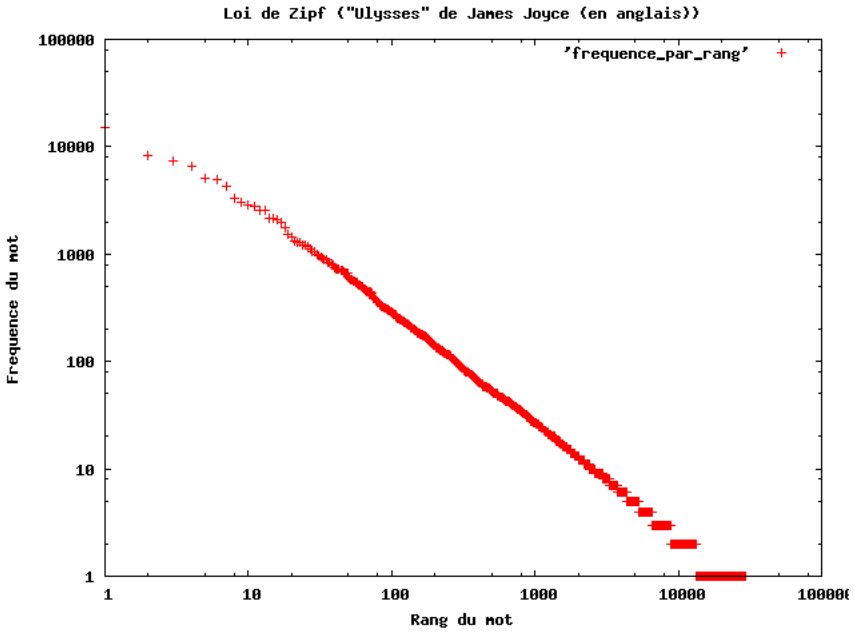

FIGURE 1.2 – Loi de Zipf (https://fr.wikipedia.org/wiki/Loi_de_Zipf)

1.4.1 Loi de Zipf

Il s'agit d'une observation empirique de la fréquence des mots dans un texte : si f est la fréquence du n-ième mot le plus fréquent, le $2n$-ième mot le plus fréquent aura une fréquence de $\frac{f}{2}$.

La fonction de masse d'une loi de Zipf de paramètres $N \in \mathbb{N}^*$ (nombre de mots) et $s > 0$ est

$$f(k) = \frac{1}{\sum_1^N \frac{1}{n^s}} \frac{1}{k^s}$$

Il s'agit d'un cas particulier de la loi de Mandelbrot stipulant l'existence de constantes a, b, c, K telles que :

$$f(n) \times (a + bn)^c = K.$$

1.4.2 Loi de Benford

La fréquence f d'apparition de d comme premier chiffre d'un nombre en base b tend vers :

$$f = \log_b(1 + \frac{1}{d}).$$

Cela concerne les nombres provenant de mesures dont l'unité est variable.

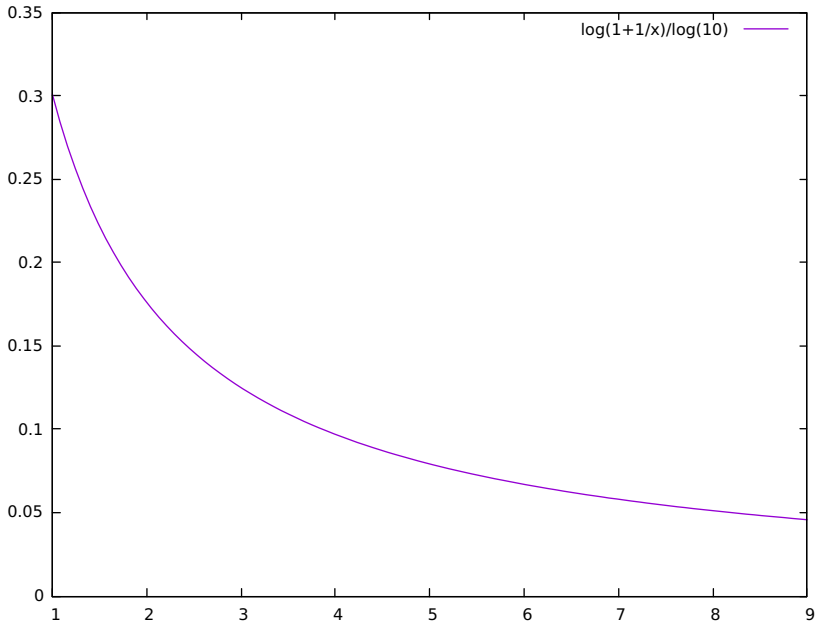

FIGURE 1.3 – Loi de Benford

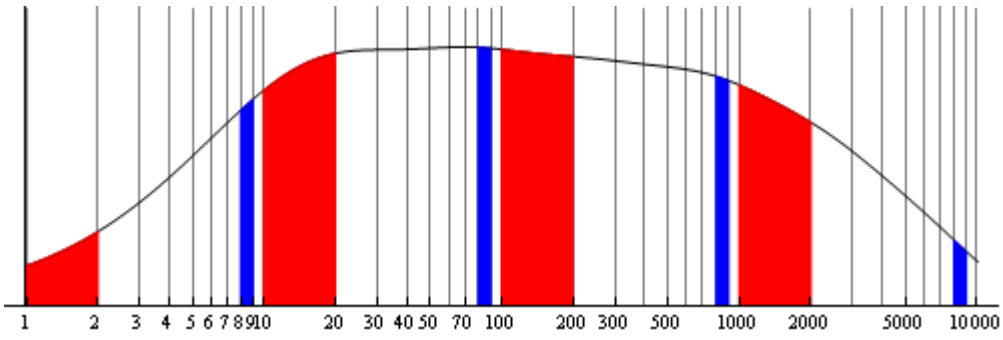

FIGURE 1.4 – Échelle logarithmique

Chapitre 2

Page Rank

L'algorithme du Page Rank consiste à classer les sommets d'un graphe orienté selon leur probabilité de visite par un surfeur aléatoire qui constitue une mesure de popularité.

2.1 Puissance de la matrice d'incidence

Soit $G = (V, E)$ un graphe, de matrice d'incidence M. Soit G_k le graphe dont la matrice d'incidence est M^k : les valeurs non nulles dans cette matrice indiquent l'existence d'un chemin de longueur k entre les deux sommets de G.

2.2 Chaîne de Markov

C'est un processus *stochastique* à temps discret tel que l'état suivant ne dépende que du précédent.

Si l'état courant X au temps n est un vecteur de coordonnées $(x_1 \dots x_p)$ l'état à $n + 1$ est défini par une matrice de transition M :

$$X_n = M \cdot X_{n+1}.$$

Cette matrice est stochasique : la somme des termes de toute ligne vaut 1. On se restreint au cas particulier où la chaîne est apériodique et irréductible.

2.3 Cas simple à deux états

Voir [1] pour des exemples à trois et quatre états ou [2] pour des exercices de maths.

Une mouche se situe dans une pièce A au temps 0. Soient $a(n)$ et $b(n)$ les probabilités de présence de la mouche en A ou B au temps n.

1. `http://cpc.cx/luT`
2. `http://cpc.cx/luS`

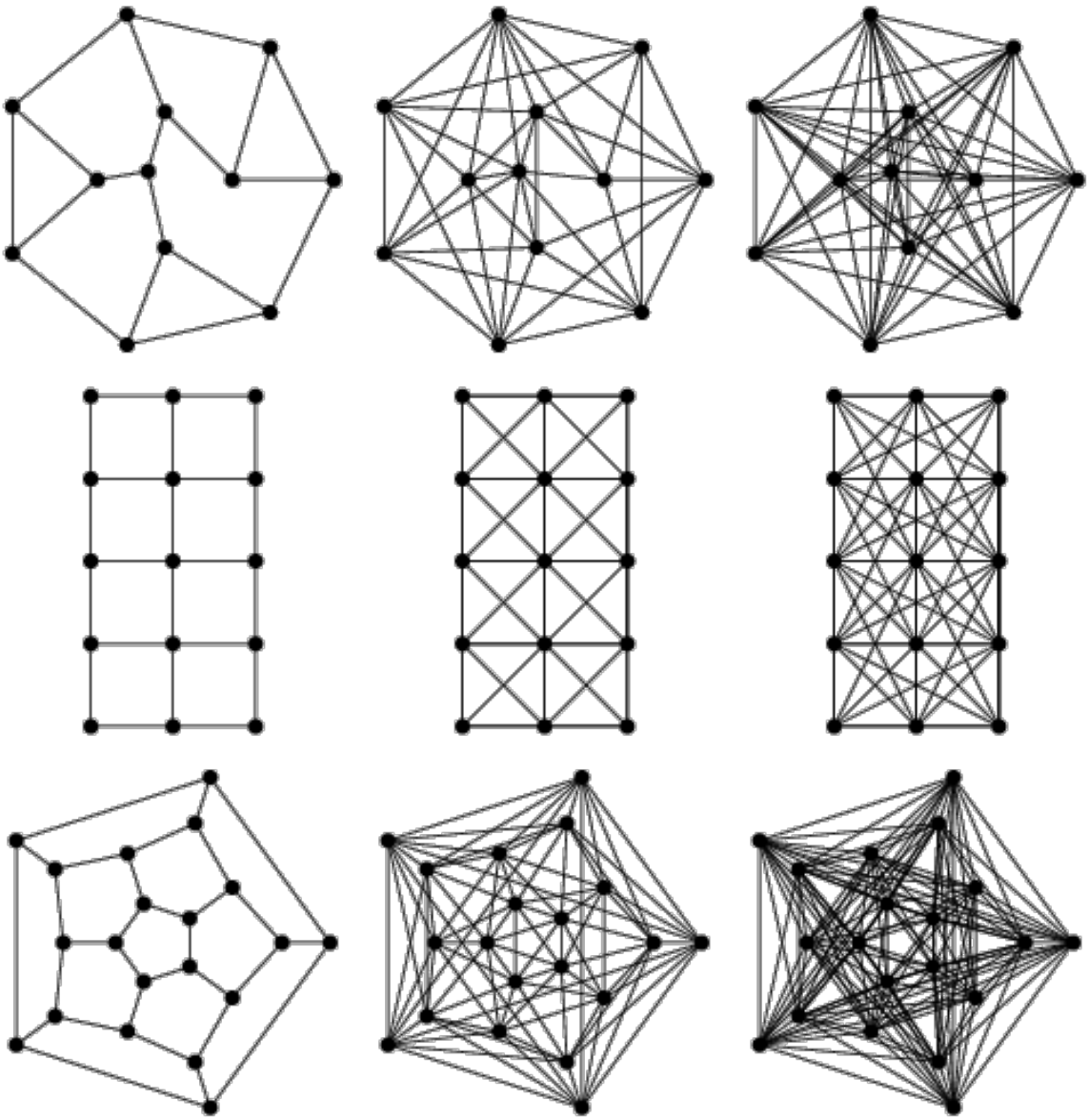

FIGURE 2.1 – Graphes pour les premières puissance de la matrice d'incidence (source `http://mathworld.wolfram.com/GraphPower.html`)

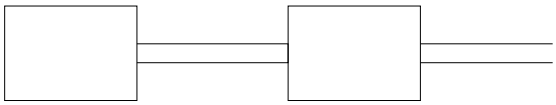

FIGURE 2.2 – Le problème de la mouche

Xcas en ligne. Tapez une instruction dans cette
bouée).

M:=[[1/3, 1/4], [2/3,1/2]];

$$\begin{pmatrix} \frac{1}{3} & \frac{1}{4} \\ \frac{2}{3} & \frac{1}{2} \end{pmatrix}$$

eigenvals(M);

$$[0, \frac{5}{6}]$$

eigenvects(M);

$$\begin{pmatrix} -3 & 1 \\ 4 & 2 \end{pmatrix}$$

inv(eigenvects(M));

$$\begin{pmatrix} \frac{-1}{5} & \frac{1}{10} \\ \frac{2}{5} & \frac{3}{10} \end{pmatrix}$$

FIGURE 2.3 – Calcul des valeurs propres

On définit la matrice de transition comme suit :
— en A, la mouche a un chance sur 1/3 d'y rester à l'instant suivant 2/3 d'aller en B
— en B, elle a 1/4 de retourner en A, 1/2 de rester en B, 1/4 de sortir.
On a donc les relations :

$$\begin{cases} a_{n+1} = \frac{1}{3}a_n + \frac{1}{4}b_n \\ b_{n+1} = \frac{2}{3}a_n + \frac{1}{2}b_n \end{cases}$$

Soit $X_n = \binom{a_n}{b_n}$ l'état du système au temps n. La matrice de transition est donc

$$\begin{pmatrix} \frac{1}{3} & \frac{1}{4} \\ \frac{2}{3} & \frac{1}{2} \end{pmatrix}$$

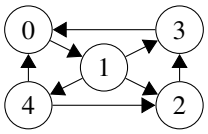

FIGURE 2.4 – Graphe orienté des pages web

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

FIGURE 2.5 – Matrice d'incidence

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0.02 & 0.92 & 0.02 & 0.02 & 0.02 \\ 0.02 & 0.02 & 0.32 & 0.32 & 0.32 \\ 0.02 & 0.02 & 0.02 & 0.92 & 0.02 \\ 0.92 & 0.02 & 0.02 & 0.02 & 0.02 \\ 0.47 & 0.02 & 0.47 & 0.02 & 0.02 \end{pmatrix}$$

FIGURE 2.6 – Incidence + Téléportation -> matrice de transition normalisée

Convergence de la matrice de transition

$$\begin{pmatrix} 0.02 & 0.92 & 0.02 & 0.02 & 0.02 \\ 0.02 & 0.02 & 0.32 & 0.32 & 0.32 \\ 0.02 & 0.02 & 0.02 & 0.92 & 0.02 \\ 0.92 & 0.02 & 0.02 & 0.02 & 0.02 \\ 0.47 & 0.02 & 0.47 & 0.02 & 0.02 \end{pmatrix}^n \rightarrow \begin{pmatrix} 0.268 & 0.262 & 0.14 & 0.22 & 0.09 \\ 0.268 & 0.262 & 0.14 & 0.22 & 0.09 \\ 0.268 & 0.262 & 0.14 & 0.22 & 0.09 \\ 0.268 & 0.262 & 0.14 & 0.22 & 0.09 \\ 0.268 & 0.262 & 0.14 & 0.22 & 0.09 \end{pmatrix}$$

Vector probabiliste à l'équilibre

$$\begin{pmatrix} 0.268 & 0.262 & 0.14 & 0.22 & 0.09 \end{pmatrix}$$

Chapitre 3

Document électronique

3.1 Qu'est-ce qu'un document ?

D'après `https://www.youtube.com/watch?v=5ICyFJouHv4`

C'est un objet matériel (livre, CD) ou électronique (disponible sur le web) qui enregistre des informations. En première approche, cet objet possède deux dimensions : la forme et un contenu (le fond), c'est-à-dire l'information. On peut considérer d'autres dimensions comme le statut (penser au workflow d'un document dans le cadre de l'édition.

Classiquement, les informations sont :

— texte
— image
— sons

Mais nous verrons d'autres formes de document électronique : une partie d'échecs, de jeu vidéo, etc.

Les fonctions principales d'un document sont de transmettre et de prouver une information (référence).

On peut également considérer les dimensions suivantes :

1. la forme : le vu. Cette forme est anthropologique et concerne notre rapport physique au document. Il doit pouvoir être repéré et vu.
2. le contenu : le lu. C'est une dimension intellectuelle et concerne le rapport de notre cerveau au contenu.
3. le medium : le su. C'est une dimension sociale et concerne le rapport du document à la société.

Cette approche documentaire permet d'analyser le positionnement des firmes majeures du web. Parmi les GAFA, on peut dire que certaines firmes se sont spécialisées dans ces trois dimensions :

1. Apple pour le vu : cette société vend surtout des machines,
2. Google pour le contenu, son savoir-faire concerne l'ingénierie linguistique et vend des mots-clé.

3. Facebook pour le médium, relie les internautes grâce à un graphe social. L'homme est un document comme un autre.

La théorie du document conseille la mise en cohérence de ces trois dimensions.

Exemple : le smartphone est un objet physique, permettant d'accéder à des documents sur le web, et valorise un carnet d'adresses.

Quelques exemples d'objets pouvant être considérés comme des documents électroniques, mettant en place une navigation dans l'information (IHM, étiquettes) :

sport : analyse des trajectoires, outils pour l'observation

les médias (texte, video, musique : outre leur contenu, on associe à ces documents les modalités d'interaction des utilisateurs. Par exemple le temps passé sur une page, la position de la souris, les films vus, pendant combien de temps, etc.

les jeux : les parties de jeu vidéo sont le lieu de nombreuses interactions humaines. Noter également les parties d'échecs (`https://lichess.org/`, de go (`https://gobooks.com/fr/`).

Chapitre 4

TP1

On souhaite appréhender les textes comme des sacs de mots, que l'on appellera des *documents*, *i.e.* comme des ensembles de mots avec leur nombre d'occurrences. On commence donc par traiter les fichiers originaux pour n'en conserver qu'une liste de mots en minuscules que l'on va compter.

4.0.1 Création du corpus

Récupérez le mini-corpus : une collection de quelques textes. `https://ecampus.unicaen.fr/mod/resource/view.php?id=99694`

Pour tester les scripts réalisés, on pourra également se constituer une petite collection d'exemples jouets.

4.0.2 Calcul des documents

En utilisant `sed`, nettoyer le texte et le transformer en une liste de mots.

— `cat ...` tr [:upper :] [:lower :]| passe en minuscule

4.0.3 Compter les mots d'un texte

Réaliser une commande comptant le nombre d'occurrences de chaque mot en triant le document (`sort`) puis en comptant le nombre de ligne contenant chaque mot [1]

4.0.4 Compter les mots du corpus : DF

Réalisez les opérations précédentes sur tout le corpus, pour obtenir les DF.

4.0.5 Calculer les TF-iDF

Fusionner les résultats en combinant les TF de chaque documents aux DF.

1. en awk, détecter le changement de mot et afficher le compte du précédent, ne pas oublier le dernier mot

4.0.6 Analyse et commentaire

— Quels sont les mots les plus pertinents pour chaque document ? [2]
— Quels sont les mots les moins pertinents ?
— En temps, quelle est la complexité empirique de cette méthode ? [3]
— En théorie, comment se comporte le calcul de TF-iDF selon le nombre de documents ?
— Quelles sont les pistes d'amélioration en terme de filtrage initial ? [4]

4.1 TP2

4.1.1 Calcul de l'index

Pour répondre à une requête, il faut disposer d'un index associant les mots aux documents.

— pour chaque document, générer la liste des mots associés à l'identifiant du document
— fusionner(`cat`), trier (`sort`), compter [5]

4.1.2 Répondre à une requête

On considère une requête [6] comme une liste de mots, un par ligne, dans un fichier. Écrire une commande qui calcule le TF-iDF de la requête en allant piocher dans les DF.

4.1.3 Calculer la pertinence de la requête

4.2 Tip

La bonne commande pour renommer un ensemble de fichiers.

```
ls -f| sed 1,2d | (j=1; while read i; do mv "$i" $j.txt; j=$(($j + 1)); done)
```

2. utiliser `sort -k2,2nr` pour définir une clé de tri reverse (r) numérique (n) sur la deuxième colone (de 2 à 2)
3. Testez sur 10, 100, 1000, 10000, 100 000 textes
4. taille des mots, nombre d'occurrence, etc.
5. en awk, signaler le changement dans la première colonne et sortir la liste mémorisée
6. ex. « music again »