Partie 7
Ordonnancement de documents

Vincent Labatut

Laboratoire Informatique d'Avignon – LIA EA 4128
vincent.labatut@univ-avignon.fr

2019/20

M2 ILSEN
UE Ingénierie du document et de l'information
UCE3 Indexation & Recherche d'information

AVIGNON UNIVERSITÉ
CERI
Centre d'Enseignement et de Recherche en Informatique

AVIGNON UNIVERSITÉ

Plan de la séance

Section 1

Modèles à base de scores

AVIGNON UNIVERSITÉ

Modèles à base de scores
Limitations du modèle booléen

- Modèle booléen :
  - Difficile de définir des requêtes fidèles :
    - Opérateur ET → précision élevée mais rappel faible
    - (Peu de FP mais beaucoup de FN)
    - Opérateur OU → le contraire
  - Grand corpus → nombreux documents potentiellement renvoyés pour une requête
    - → Besoin de les distinguer

→ Alternative : modèle à base de score

- Permet d'ordonner des documents
- Permet les requêtes en texte libre
- Requiert une représentation sac-de-mots généralisée

Modèles à base de score
Notion de score

Score d'un document

Le **score** $S(d, q)$ est une valeur **réelle** associée à un **document** d pour une **requête** q, et quantifiant la **pertinence** de d relativement à q.

Score partiel d'un document

Le score **partiel** $s(d, t)$ mesure la pertinence du **document** d relativement au **terme** t.

- Méthode générique de calcul de $S(d, q)$:
  - $\forall t \in q$: on calcule $s(d, t)$
  - $S(d, q)$ s'obtient en combinant ces scores partiels
- Résultats (documents) ordonnés par score $S(d, q)$

Modèles à base de score
Représentation sac-de-mots

Modèle sac-de-mots

Un **document** d est représenté par un **vecteur réel** $d = (s(d, t_1), ..., s(d, t_T))$ de taille T, dont chaque valeur correspond au **score partiel** du document pour un terme du lexique.

- Deux documents dont les SdM sont proches sont supposés similaires
- Représentation SdM également appliquée à la requête
  - Requête = ensemble de mots : pas de connecteur booléen
  - → Requête en texte libre

→ On peut comparer un document d et une requête q

Modèles à base de score
Approche naïve

- Représentation naïve :
  - Utiliser la fréquence du terme comme score partiel : $s(d, t) = tf(t, d)$
  - Score du document : somme des scores partiels pour tous les termes de la requête

$$S(d, q) = \sum_{t \in q} s(d, t) = \sum_{t \in q} tf(t, d)$$

- Interprétation : plus un terme est fréquent dans un document, plus le document est pertinent pour ce terme
- Limite :
  - La fréquence du terme tf reflète son importance dans un document
  - Mais on ignore son importance dans la collection entière

Modèles à base de score
Pertinence des termes rares

→ Termes rares sont plus informatifs (≠ mots vides)

- Ex. : mot $\mathrm{voiture}$ dans une collection portant sur le secteur automobile → pas informatif
- → Donner plus de poids aux termes de la requête apparaissant peu dans la collection

- Deux possibilités (rappel) :
  - $df(t)$: nombre de documents du corpus contenant t
  - $cf(t)$: nombre total d'occurrences de t dans le corpus
    - → tient compte des occurrences multiples dans le même document
  - Exemple tiré du corpus Reuters :

Terme	cf	df
try	10 422	8 760
insurance	10 440	3 997

→ df plus adaptée car basée sur la notion de document

Modèles à base de score
Fréquence de document inverse

- Fréquence mesure le manque d'information
  - Mais on veut mesurer le contraire

  → Fréquence de document inverse

Fréquence de document inverse

La **fréquence de document inverse** d'un terme t est définie comme :

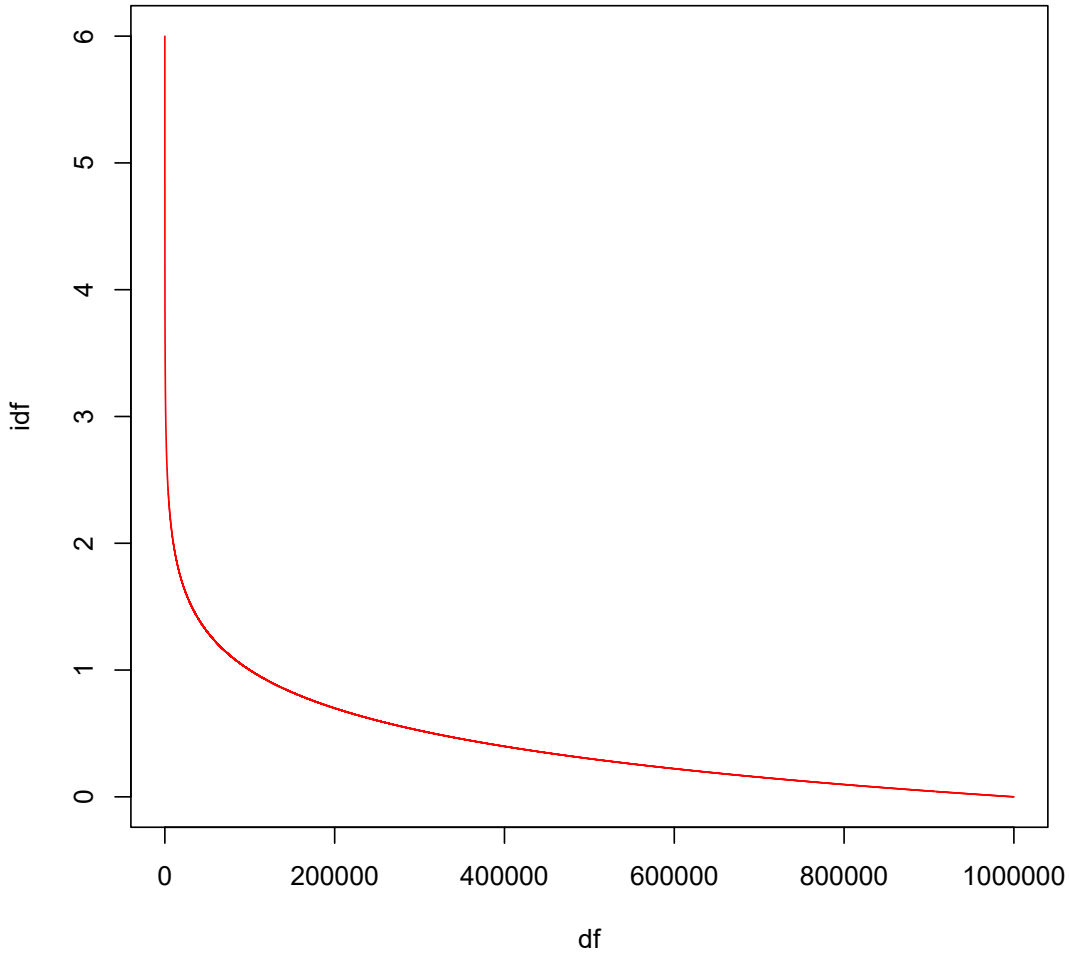
$$idf(t) = \log_{10} \frac{D}{df(t)}.$$

- Bornes :
  - Inférieure : $df(t) = D \rightarrow idf(t) = \log_{10}(1) = 0$
  - Supérieure : $df(t) = 1 \rightarrow idf(t) = \log_{10}(D)$
- Alternative robuste aux termes hors-lexique : $\log_{10} \frac{D+1}{df(t)+1}$
- Intérêt du logarithme :
  - Donner moins de contraste aux rapports proches de 1 (= termes fréquents)
  - Justifié par la loi de Zipf

Modèles à base de score
Exemple de fréquence de document inverse

Exemple : pour $D = 10^6$

df	idf
1	6
100	4
1 000	3
10 000	2
100 000	1
1 000 000	0



Modèles à base de score
Définition du score tf-idf

- Importance d'un terme t :
  - Dans un document : $tf(t, d)$
  - Dans la collection : $idf(t)$

Score tf-idf

Score **partiel** obtenu en combinant tf et idf :

$$s(d, t) = tf(t, d) \times idf(t).$$

- Comportement :
  - Valeur élevée si t fréquent dans d mais apparaît dans un petit nombre de documents
  - Plus faible si peu fréquent dans d ou apparaît dans de nombreux documents
  - Très faible si peu fréquent dans d et apparaît dans tous les documents
- Score $S(d, q)$ obtenu en sommant comme en p.7
- Note : nombreuses alternatives à tf et idf

Modèles à base de score
Exemple pour le score tf-idf

On traite la requête q suivante : $\mathbf{car\ auto\ insurance\ best}$

Supposons qu'on a déjà calculé les valeurs suivantes sur la base du corpus Reuters :

t	Terme	$df(t)$	$idf(t)$	$tf(t, d_1)$	$tf(t, d_2)$	$tf(t, d_3)$
t_1	car	18 165	1,65	27	4	24
t_2	auto	6 723	2,08	3	33	0
t_3	insurance	19 241	1,62	0	33	29
t_4	best	25 235	1,50	14	0	17

Alors les scores des documents sont :

$$\begin{aligned} S(d_1, q) &= s(d_1, t_1) + s(d_1, t_2) + s(d_1, t_3) + s(d_1, t_4) \\ &= tf(t_1, d_1)idf(t_1) + tf(t_2, d_1)idf(t_2) + tf(t_3, d_1)idf(t_3) + tf(t_4, d_1)idf(t_4) \\ &= 27 \times 1{,}65 + 3 \times 2{,}08 + 0 \times 1{,}62 + 14 \times 1{,}50 = 71{,}79 \end{aligned}$$

$$S(d_2, q) = 4 \times 1{,}65 + 33 \times 2{,}08 + 33 \times 1{,}62 + 0 \times 1{,}50 = 128{,}7$$

$$S(d_3, q) = 24 \times 1{,}65 + 0 \times 2{,}08 + 29 \times 1{,}62 + 17 \times 1{,}50 = 112{,}08$$

$\rightarrow$ le document le plus pertinent pour q est d_2

Modèles à base de score
Pondération log-fréquence et score wf-idf

- Limitation de tf :
  - La pertinence d'un document est-elle vraiment *proportionnelle* à la fréquence du terme ?
  - Ex. : un document contenant 10 fois t est-il vraiment 10 fois plus pertinent qu'un autre ne le contenant qu'une seule fois ?
- Alternative : pondération log-fréquence

Pondération log-fréquence

Normalisation **logarithmique** de tf :

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

- $tf = 0 \rightarrow wf = 0$
- $tf = 1 \rightarrow wf = 1$
- $tf = 2 \rightarrow wf = 1,3$
- $tf = 10 \rightarrow wf = 2$
- $tf = 1000 \rightarrow wf = 4$
- ...

- On obtient le score wf-idf en combinant avec idf

Section 2

Modèle vectoriel

AVIGNON
UNIVERSITÉ

Modèle vectoriel
Représentation vectorielle

- Alternative au classement par score total
- Représentation des données dans un espace vectoriel à T dimensions

Représentation vectorielle d'un document

Chaque document d est représenté par un **vecteur réel** de la **taille du lexique** $\mathcal{T}$, et dont chaque **élément** est le **score partiel** $s(d, t_j)$ associé à un terme t_j $(1 \leq j \leq T)$:

$$\vec{V}(d) = (s(d, t_1), ..., s(d, t_T)).$$

- On peut par exemple utiliser tf-idf pour calculer $\vec{V}(d)$
- Vecteurs très longs mais très creux (beaucoup de zéros)
- Requêtes représentées selon le même principe

Modèle vectoriel
Exemple

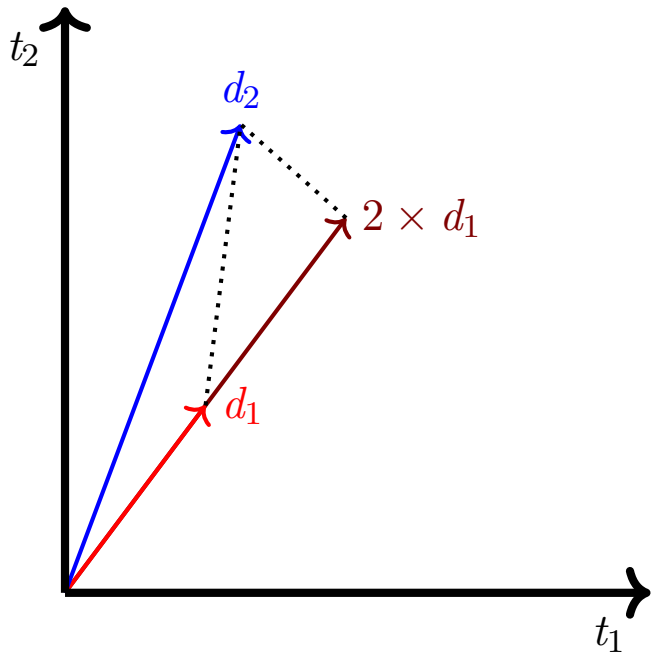
	Antony & Cleopatra	*Julius Caesar*	*The Tempest*	*Hamlet*	*Othello*	*Mac Beth*	...
Antony	5,25	3, 18	0,00	0,00	0,00	0,35	...
Brutus	1,21	6,10	0,00	1,00	0,00	0,00	...
Caesar	8,59	2,54	0,00	1,51	0,25	0,00	...
Calpurnia	0,00	1,54	0,00	0,00	0,00	0,00	...
Cleopatra	2,85	0,00	0,00	0,00	0,00	0,00	...
mercy	1,51	0,00	1,90	0,12	5,25	0,88	...
worser	1,37	0,00	0,11	4,15	0,25	1,95	...
...	...	...	...	...	...	...	...

- Utilisation :
  - Score d'un document = fonction de sa distance à la requête dans l'espace vectoriel
- Autre intérêt : comparaison de documents

Modèle vectoriel
Distance euclidienne dans l'espace vectoriel

- Approche naïve :
  - Distance euclidienne
  - (entre les points d'arrivée des vecteurs)

- Limite : effet important de la norme des vecteurs
  - Sensibilité à la fréquence absolue des termes
  - Insensibilité à la fréquence relative des termes
  - Ex. : même document dupliqué



- Solution :
  - Utiliser l'angle formé par les vecteurs à comparer

Modèle vectoriel
Similarité cosinus dans l'espace vectoriel

Similarité cosinus

La **similarité cosinus** entre deux **documents** d_1 et d_2 respectivement représentés par les **vecteurs** $\vec{V}(d_1)$ et $\vec{V}(d_2)$ est :

$$sim(d_1, d_2) = \cos\theta,$$

où θ est l'**angle** formé par les deux vecteurs.

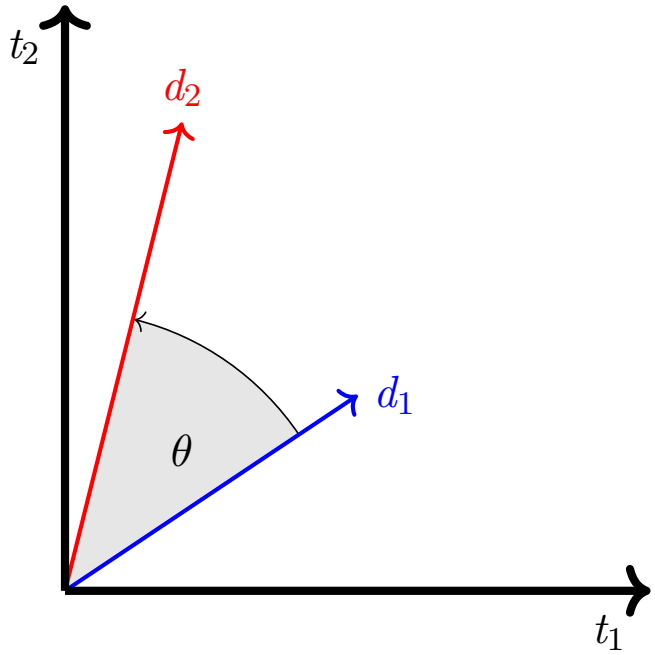
Calcul : on passe par le produit scalaire, sans utiliser directement la fonction $\cos$:

$$\cos\theta = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)| \times |\vec{V}(d_2)|} = \frac{\vec{V}(d_1)}{|\vec{V}(d_1)|} \cdot \frac{\vec{V}(d_2)}{|\vec{V}(d_2)|} = \vec{v}(d_1) \cdot \vec{v}(d_2),$$

où $|...|$ dénote la norme, et $\vec{v}(d_1)$ et $\vec{v}(d_2)$ sont des vecteurs unitaires.

Modèle vectoriel
Propriétés de la similarité cosinus

- Propriété : $\cos\theta$ est une fonction décroissante de θ sur $[0;\pi/2]$
- Comportement :
  - $-1 \rightarrow$ vecteurs opposés (impossible ici)
  - $0 \rightarrow$ vecteurs orthogonaux (indépendance)
  - $+1 \rightarrow$ vecteurs colinéaires
- Score d'un document : similarité avec la requête
  $S(d, q) = sim(d, q)$



Modèle vectoriel
Exemples de comparaison de documents

- Vecteurs représentant les documents :
  - *Non-normalisé* vs *normalisé* (i.e. vecteur unitaire)

Terme	Sense & Sensibility	Pride & Prejudice	Wuthering Heights
affection	3,06 vs. 0,789	2,76 vs. 0,832	2,30 vs. 0,524
jealous	2,00 vs. 0,515	1,85 vs. 0,555	2,04 vs. 0,465
gossip	1,30 vs. 0,335	0,00 vs. 0,000	1,78 vs. 0,405
wuthering	0,00 vs. 0,000	0,00 vs. 0,000	2,58 vs. 0,588

- (En pratique : beaucoup plus de dimensions)
- Ex : comparaison de documents

$$\begin{aligned} sim(SS, PP) &= 0{,}789 \times 0{,}832 + 0{,}515 \times 0{,}555 + 0{,}335 \times 0{,}000 + 0{,}000 \times 0{,}000 \\ &\approx 0{,}94 \\ sim(SS, WH) &= 0{,}789 \times 0{,}524 + 0{,}515 \times 0{,}465 + 0{,}335 \times 0{,}405 + 0{,}000 \times 0{,}588 \\ &\approx 0{,}79 \\ sim(PP, WH) &= 0{,}832 \times 0{,}524 + 0{,}555 \times 0{,}465 + 0{,}000 \times 0{,}405 + 0{,}000 \times 0{,}588 \\ &\approx 0{,}69 \end{aligned}$$

Modèle vectoriel
Calcul de la similarité cosinus

Input : q : String
Output : $\mathbf{S}$: Vector

// Initialisation
$\mathbf{A} \leftarrow \mathbf{0}_N$ // Produit scalaire entre chaque document d et la requête q
$\mathbf{B} \leftarrow \mathbf{0}_N$ // Normes de chaque document d
$B_q \leftarrow 0$ // Norme de la requête q

foreach $t \in q$ **do** // On traite chaque terme t de la requête q
    Calculer $s(q, t)$
    $B_q \leftarrow B_q + s(q, t)^2$ // Mise à jour de la norme de la requête
    $\ell \leftarrow$ liste de postings associée à t
    foreach $d \in \ell$ **do** // On traite chaque document d de la liste de postings ℓ
        Calculer $s(d, t)$
        $a_d \leftarrow a_d + s(d, t) \times s(q, t)$ // Màj du produit scalaire entre d et q
        $b_d \leftarrow b_d + s(d, t)^2$ // Mise à jour de la norme du document d
    end foreach
end foreach

$\mathbf{B} \leftarrow \sqrt{\mathbf{B}}$ // On finalise le calcul des normes des documents d
$B_q \leftarrow \sqrt{B_q}$ // On finalise le calcul de la norme de la requête q
$\mathbf{S} \leftarrow \mathbf{A}/(\mathbf{B} \times B_q)$ // On finalise le calcul des scores des documents

Algorithme 1 – Calcul de la similarité cosinus pour une requête q.

Section 3

Conclusion

AVIGNON
UNIVERSITÉ

Concepts abordés dans cette partie

- Score et score partiel d'un document
- Représentation sac-de-mots
- Fréquence de document inverse idf
- Similarité cosinus
- Score tf-idf
- Pondération log-fréquence wf
- Modèle vectoriel

Lectures recommandées

[MRS08] *Introduction to Information Retrieval*, chapitre 6.

[BCC10] *Information Retrieval : Implementing and Evaluating Search Engines*, chapitre 2.

[BR11] *Modern Information Retrieval : The Concepts and Technology behind Search*, chapitre 3.

[AG13] *Recherche d'information - Applications, modèles et algorithmes*, chapitre 3.

[CMS15] *Search Engines : Information Retrieval in Practice*, chapitres 2 & 7.

Références bibliographiques I

[AG13] M.-R. Amini et É. Gaussier. *Recherche d'information - Applications, modèles et algorithmes.* Paris, FR : Eyrolles, 2013. url : **https://www.eyrolles.com/Informatique/Livre/recherche-d-information-9782212673760/**.

[BR11] R. Baeza-Yates et B. Ribeiro-Neto. *Modern Information Retrieval : The Concepts and Technology behind Search.* 2nd Edition. Boston, USA : Addison Wesley Longman, 2011. url : **http://people.ischool.berkeley.edu/~hearst/irbook/**.

[BCC10] S. Büttcher, C. L. A. Clarke et G. V. Cormack. *Information Retrieval : Implementing and Evaluating Search Engines.* Cambridge, USA : MIT Press, 2010. url : **http://www.ir.uwaterloo.ca/book/**.

[CMS15] W. B. Croft, D. Metzler et T. Strohman. *Search Engines : Information Retrieval in Practice.* Pearson, 2015. url : **http://www.search-engines-book.com/**.

Références bibliographiques II

[MRS08] C. D. Manning, P. Raghavan et H. Schütze. *Introduction to Information Retrieval*. New York, USA : Cambridge University Press, 2008. url : http://www-nlp.stanford.edu/IR-book/.