Indexation: Phase 3 (Approche basée sur term freqency tf ,inverted document frequency idf)

La Fréquence du Terme d'Indexation ou Term Frequency (**TF**): représente la fréquence d'apparition du terme d'indexation dans l'unité documentaire, ou représente le nombre d'occurrences du terme d'indexation dans un document.

La Fréquence Inverse du Document ou Inverse Document Frequency (**IDF**): représente la fréquence inverse d'apparition du terme d'indexation dans a collection globale d'unités documentaire. Elle donne un poids plus important aux termes les moins fréquents.

La combinaison des deux mesures (**TF * IDF**) donne une bonne approximation de l'importance du terme dans le document et la discrimination du terme dans le corpus, particulièrement dans les corpus de documents de tailles homogènes.

Indexation: Phase 3 (Approche basée sur term freqency tf ,inverted document frequency idf)

- **IDF**: La fréquence inverse de document (inverse document frequency) est une mesure de l'importance du terme dans l'ensemble du corpus. Dans le schéma TF-IDF, elle vise à donner un poids plus important aux termes les moins fréquents, considérés comme plus discriminants d'où le nom inverse.
- Elle consiste à calculer le logarithme (en base 10 ou en base 2) de l'inverse de la proportion de documents du corpus qui contiennent le terme. Un terme est dit discriminant s'il distingue bien un document des autres document (ex: le terme apparait dans un seul document).

Indexation: Phase 3 (Approche basée sur term freqency tf ,inverted document frequency idf)

Un terme qui apparaît dans tous les documents n'est pas discriminant. Le **TF-IDF** est une méthode de pondération souvent utilisée en Recherche d'Information et dans la fouille de textes.

Le **TF-IDF** permet d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus. Le poids augmente proportionnellement au nombre d'occurrences du mot dans le document. Il varie également en fonction de la fréquence du mot dans le corpus.

Cette combinaison se base sur ce principe:

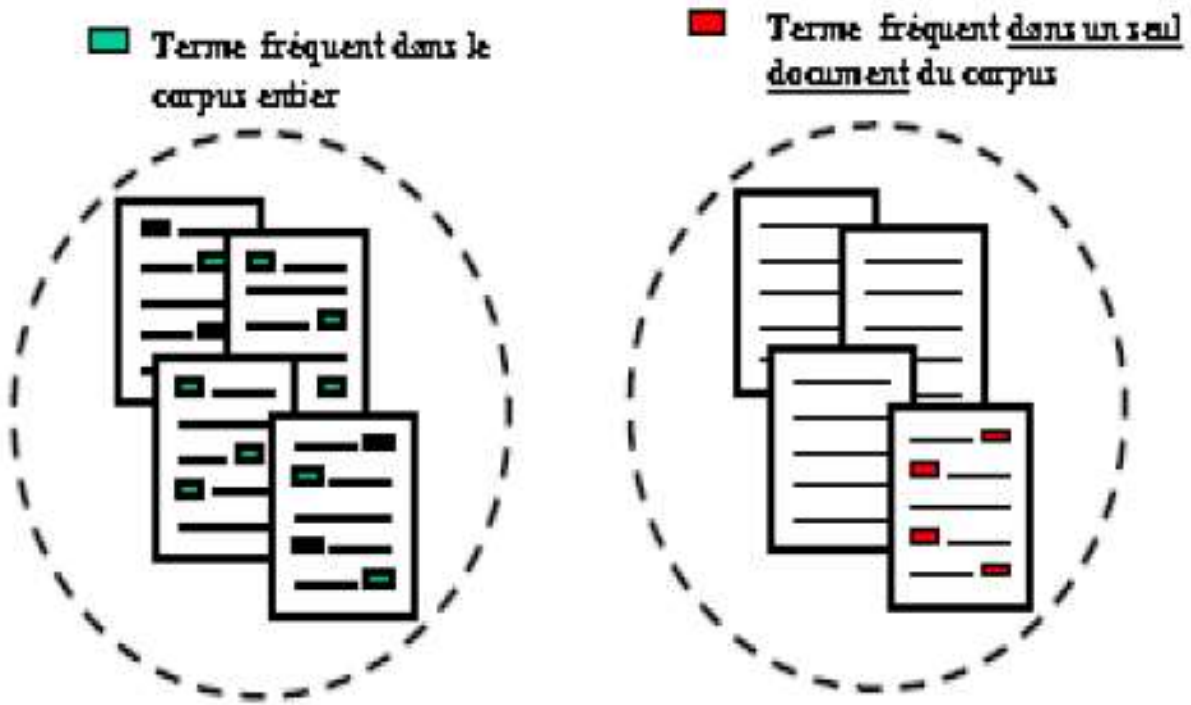
+ un terme est présent dans un document, **+** il est représentatif du contenu du document (**TF**); **-** un terme est présent dans une collection (Corpus), et **+** une occurrence de terme est significative (IDF).

Indexation: Phase 3 (Approche basée sur term freqency tf ,inverted document frequency idf)

tf: désigne l'importance d'un terme pour un document ou bien la fréquence d’un terme dans un document.

idf: mesure si le terme est discriminant (ou non-uniformément distribué), sa fréquence dans le corpus de documents

Indexation: Phase 3 (Approche basée sur term freqency tf ,inverted document frequency idf)



Indexation: Phase 3 (Approche basée sur term freqency tf ,inverted document frequency idf)

Les différentes variantes de *tf*

1. **La fréquence « brute »** d'un terme est simplement le nombre d'occurrences de ce terme dans le document.
2. **La fréquence Binaire**: Un choix plus simple, dit « binaire », est de mettre 1 si le terme apparaît dans le document et 0 sinon.
3. **Normalisation** : on peut normaliser logarithmiquement la fréquence brute pour amortir les écarts. Une normalisation courante pour prendre en compte la longueur du document est de normaliser par la fréquence brute maximale du document.

Indexation: Phase 3 (Approche basée sur term freqency tf ,inverted document frequency idf)

Tf (term frequency)

plus un terme est fréquent dans un document plus il est important dans la description de ce document

Il existe plusieurs variantes de tf

$$tf = \begin{cases} freq(t,d) \\ 1+\log(freq(t,d)) \\ \dfrac{freq(t,d)}{\max_{\forall t' \in d}(t',d)} \\ \dfrac{freq(t,d)}{\sum_{\forall t' \in d} freq(t',d)} \end{cases}$$

Taille (longueur) du document

Binaire: tf=(0,1)
0: le terme n'existe pas dans d
1: le terme existe dans d

– "Okapi tf" : K introduit pour tenir compte de la longueur des documents

$$tf / (K+tf)$$

$$tf = \frac{freq(t,d)}{k1.(1-b+b*\frac{dl}{avgdl})+freq(t,d)}$$

Indexation: Phase 3 (Approche basée sur term freqency tf ,inverted document frequency idf)

idf (inverted term frequency):*la fréquence du terme dans la collection (ensemble des documents).*

$$idf(t) = \begin{cases} \log\left(\frac{N}{n_t}\right) \\ \log\left(\frac{N - n_t}{n_t}\right) \end{cases}$$

avec
N : le nombre de documents de la collection,
n_t : le nombre de documents contenant le terme t

Indexation: Phase 3 (Approche basée sur term freqency tf ,inverted document frequency idf)

Le poids du terme dans un document

$$w(t, d) = tf * idf$$

Calcul du score d'un document

Score d'un document par rapport à une requête=Somme pondérée des termes de la requête apparaissant dans un document.

$$score(q,d) = \sum_{t \in q} w(t,d)$$

Indexation: Phase 3 (Approche basée sur term freqency tf ,inverted document frequency idf)

Exemple (Wikipedia)

Corpus (tiré d'œuvres de Friedrich Gottlieb Klopstock)[2]

Document 1	Document 2	Document 3
Son nom est célébré par le bocage **qui** frémit, et par le ruisseau **qui** murmure, les vents l'emportent jusqu'à l'arc céleste, l'arc de grâce et de consolation que sa main tendit dans les nuages.	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.	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.

L'exemple porte sur le document 1 (soit d_1) et le terme analysé est « qui » (soit t_1 = qui). La ponctuation et l'apostrophe sont ignorées.

Indexation: Phase 3 (Approche basée sur term freqency tf ,inverted document frequency idf)

Calcul de TF: la formule appliquée

$$\frac{freq(t,d)}{\sum_{\forall t' \in d} freq(t',d)}$$

$tf_{1,1}$= 2/38 **(**le terme « qui » apparait dans d_1 et d_3 seulement, d_1 contient 38 termes**)**

Calcul de idf: la formule appliquée $idf_j = \log\left(\frac{N}{df_j}\right)$

idf_1 =log(3/2)

Le poids final $W(t_1,d_1)$=tf*idf =2/38 * log(3/2) = 0.0092

Indexation: Phase 3 (Approche basée sur term freqency tf ,inverted document frequency idf)

Pour d_2 et d_3 et en suivant les même étapes

d_2 :

Le poids final **$W(t_1,d_2)=tf*idf = 0 * \log(3/2) = 0$**

d_3 :

Le poids final **$W(t_1,d_3)=tf*idf = 1/40 * \log(3/2) = 0.0044$**

Conclusion :Le premier document apparaît comme « le plus pertinent »
($W(t_1,d_1)= 0.0092$)

Indexation: Phase 3 (Approche basée sur term freqency tf ,inverted document frequency idf)

Soit le corpus suivant:

D1:<a,b,c,d,e,c,f,g,c,h>

D2:<i,j,k,l,m,n,o,p,q>

D3:<r,s,t,u,v,w,x,c,y,z>

$tf_{1,1} = 3;$

$tf_{1,2} = 0;$

$tf_{1,3} = 1;$

$idf_1 = \log_{10}(3/2) = 0.1761$

Calculer la pertinence du terme t_1 = 'c' pour D1 , D2 , D3

$W(t_1,d_1) = tf_{1,1} * idf_1 = 0,5283$

$W(t_1,d_2) = tf_{1,2} * idf_1 = 0$

$W(t_1,d_3) = tf_{1,3} * idf_1 = 0.1761$

Donc D1 est le document le plus pertinent du corpus

Les modèles de RI

Les modèles de RI manipulent plusieurs variables :

les besoins, les documents, les termes, les jugements de pertinence , les utilisateurs, …

Les modèles de RI se distinguent par le principe d'appariement (matching) :

appariement exact /approché (Exact /Best matching)

Les modèles de RI

Appariement exact	Appariement approché
Requêtes structurées et difficiles à écrire, Difficulté s'accroit avec la taille de la collection	Requêtes décrit les critères de recherche dans un document
1. La sélection d'un document est basée sur une décision binaire	Le résultat est un ensemble de documents pondérés
Le résultat est un ensemble de documents non ordonnés	Best-match donne de meilleurs performances (les meilleurs documents en premier)
Problème de collections volumineuses : le nombre de documents retournés peut être considérable	

Les modèles de RI

Appariement exact

•Requête spécifie de manière précise les critères recherchés

•L'ensemble des documents respectant exactement la requête sont sélectionnés, mais pas ordonné.

Appariement approché

• Requête décrit les critères recherchés dans un document

•Les documents sont sélectionnés selon un degré de pertinence (similarité/ probabilité) vis-à-vis de la requête et sont ordonné

Les modèles de RI

Les différent modèles de la RI

Modèle booléen (±1950)

Modèle vectoriel (±1970)

Modèle probabiliste (±1976)

Modèle connexionniste (réseaux de neurones)(±1989)

Modèle d'inférence (réseau d'inférence bayésien) (±1992)

Modèle LSI (Latent Sementic Indexing) (± 1994)

Modèle de langage (±1998)

Les modèles de RI: Le modèle booléan

Ce modèle est basé sur la théorie des ensembles

Dans ce modèle, un document est représenté comme une conjonction logique de termes (non pondérés) $d = t_1 \wedge t_2 \wedge \ldots \wedge t_n$

$d1(t_1,t_2,t_5)$; $d2(t_1,t_3,t_5,t_6)$; $d3(t_1,t_2,t_3,t_4,t_5)$

Une requête est une expression logique quelconque de termes. On peut utiliser les opérateurs et ($\wedge$), ou ($\vee$) et non ($\neg$).

$q = t_1 \wedge(t_2 \vee\neg t_3)$

- Appariement Exact basé sur la présence ou l'absence des termes de la requête dans les documents

- Appariement (d, q) = R (d, q)=1 ou 0

Les modèles de RI: Le modèle booléan

Un document est représenté comme un ensemble de termes, et une requête comme une expression logique de termes. La correspondance R(d, q) entre une requête et un document est déterminée de la façon suivante:

$$R(d, t_i) = 1 \text{ si } t_i \in d; 0 \text{ sinon.}$$

$$R(d, q_1 \wedge q_2) = 1 \text{ si } R(d, q_1) = 1 \text{ et } R(d, q_2) = 1; 0 \text{ sinon.}$$

$$R(d, q_1 \vee q_2) = 1 \text{ si } R(d, q_1) = 1 \text{ ou } R(d, q_2) = 1; 0 \text{ sinon.}$$

$$R(d, \neg q_1) = 1 \text{ si } R(d, q_1) = 0; 0 \text{ sinon.}$$

Les modèles de RI: Le modèle booléan

Exemples :

Soit la Requête

$q = t_1 \wedge (t_2 \vee \neg t_3)$

Soient les documents

$d1(t_1,t_2,t_5)$; $d2(t_1,t_3,t_5,t_6)$; $d3(t_1,t_2,t_3,t_4,t_5)$

Calculer la correspondance :

R(d1, q) = **1 et (1 ou non 0)= 1 et (1 ou 1)= 1 et 1 =1**

R(d2, q) **= 1 et (0 ou non 1)= 1 et (0 ou 0)= 1 et 0=0**

R(d3, q) = **1 et (1 ou non 1)= 1 et (1 ou 0)= 1 et 1 =1**

Les modèles de RI: Le modèle booléan

	t_1	t_2	t_3	…	t_n
D1	1	1	0		0
D2	1	0	1		0
D3	1	1	1		1
D4	0	0	1		1

avec la requête **R = t_1 AND (t_2 OR t_3) AND NOT t_n**

Q1:Quels sont les documents qui satisfont R ?
R1: D1 et D2
Q2: Quel est le document qui répond au mieux è la requête R ?
R2: impossible de le définir car le modèle booléen est linéaire

Le modèle booléen ne permet pas de définir la notion de ressemblance.

Les modèles de RI: Le modèle vectoriel

Principe

1-Représenter les documents et les requêtes sous forme de vecteurs dans l'espace vectoriel engendré par tous les termes de la collection de documents.

2- Calculer le degré de similarité entre les documents et la requête.

Un document D_i est représenté par un vecteur de dimension m :

$D_i = (w_{i1}, w_{i2}, \ldots, w_{im})$ **pour i = 1, 2, …, n.**

w_{ij} est le poids du terme t_j dans le document D_i

n est le nombre de documents dans la collection,

m est le nombre de termes dans les documents de la collection.

Une requête q_k est représentée par un vecteur dans le même espace des termes.

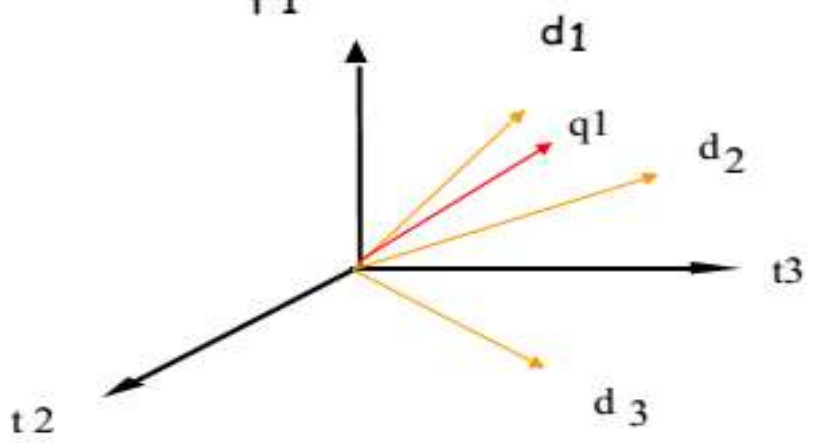
$q_k = (w_{k1}, w_{k2}, \ldots, w_{km})$ où w_{kj} est le poids de terme t_j dans la requête q_k.

Les modèles de RI: Le modèle vectoriel

Soit **T<t_1, t_2, …, t_M>** : ensemble des **M** termes de la collection

$dj = (w_{1j}, w_{2j}, \ldots, w_{Mj})$

$q = (w_{1q}, w_{2q}, \ldots, w_{Mq})$



La pertinence est traduite comme une similarité de vecteurs
un document d1 est d'autant plus pertinent à une requête que le vecteur associé est similaire à celui de la requête.

Les modèles de RI: Le modèle vectoriel

Une collection de **n** documents et **M** termes distincts peut être représentée sous forme de matrice

$$\begin{array}{c|cccc} & T_1 & T_2 & .... & T_M \\ D_1 & w_{11} & w_{21} & \dots & w_{M1} \\ D_2 & w_{12} & w_{22} & \dots & w_{M2} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ D_n & w_{1n} & w_{2n} & \dots & w_{Mn} \end{array}$$

La requête est également représentée par un vecteur.

Les modèles de RI: Le modèle booléen

- Prendre en compte l'importante des termes dans les documents et/ou dans la requête
- Possibilité d'ordonner les documents sélectionnés
- Comment étendre le modèle booléen ?

Deux modèles :

Modèle flou- fuzzy based model (basé sur la logique floue)

Modèle booléen étendu- extended boolean model

Les modèles de RI: Le modèle booléen

- Modèle booléen → Appariement exact
- Modèle booléen étendu
- Modèle booléen fuzzy

(basé sur la logique floue)

→ Appariement approché

Les modèles de RI: Le modèle booléen étendu

Le modèle booléen étendu consiste à associer des poids **w** d'indexation à chaque terme d'une requête et d'un document et de mesurer par la suite le score de pertinence « requête-document » qui est une sorte d'appariement rapproché.

Soit

Les modèles de RI: Le modèle booléen étendu

- Soient
  - **d= (w_{1j}, w_{2j} , … ,w_{tj})**
  - **q : requête à deux termes (t_1 , t_2)**

$$R(d_j, t_1 \vee t_2) = \sqrt{\frac{w_{1j}^2 + w_{2j}^2}{2}}$$

$$R(d_j, t_1 \wedge t_2) = 1 - \sqrt{\frac{(1-w_{1j})^2 + (1-w_{2j})^2}{2}}$$

Les modèles de RI: Le modèle booléen étendu

Documents			Booléen		booléen étendu	
	A	B	A ou B	A et B	A ou B	A et B
D1	1	1	1	1	?	?
D2	1	0	1	0	?	?
D3	0	1	1	0	?	?
D4	0	0	0	0	?	?

Les modèles de RI: Le modèle booléen étendu

Documents			Booléen		booléen étendu	
	A	B	A ou B	A et B	A ou B	A et B
D1	1	1	1	1	**1**	**1**
D2	1	0	1	0	**1/sqr(2)**	**1-1/sqr(2)**
D3	0	1	1	0	**1/sqr(2)**	**1-1/sqr(2)**
D4	0	0	0	0	**0**	**0**

Les modèles de RI: Le modèle booléen étendu

Généralisation

- $d_j = (w_{1j}, w_{2j}, \ldots, w_{tj})$
- q : requête composée de m termes ($t_1, t_2, \ldots, t_m$)

$$R(d_j, qor) = \left(\frac{w_{1j}^p + w_{2j}^p + \ldots + w_{mj}^p}{m}\right)^{\frac{1}{p}}$$

$$R(d_j, qand) = 1 - \frac{\left((1 - w_{1j})^p + (1 - w_{2j})^p + \ldots + (1 - w_{mj})^p\right)^{\frac{1}{p}}}{m^{\frac{1}{p}}}$$

$$R(d_j, qnot) = 1 - R(d_j, q)$$

Les modèles de RI: Le modèle booléen étendu

- Généralisation

- $d_j = (w_{1j}, w_{2j}, \ldots, w_{tj})$

- Si la requête et les documents sont pondérés $q(q_1, q_2, \ldots, q_m)$

$$R(d_j, qor) = \left(\frac{\sum q_i^p * w_{1j}^p}{\sum q_i^p}\right)^{\frac{1}{p}}$$

$$R(d_j, qand) = 1 - \left(\frac{\sum q_i^p * (1 - w_{1j})^p}{\sum q_i^p}\right)^{\frac{1}{p}}$$

Les modèles de RI: Le modèle booléen basé sur les ensembles flous

Un document d est représenté par un ensemble de termes pondérés comme suit:

$$d = \{\ldots, (t_i, w_i), \ldots\}$$

t_i est le terme, w_i est le poids associer au terme t_i

Le degré de correspondance (évaluation d' une requête) :

Évaluation 1: [Zadeh]

$$R(d, t_i) = w_i$$

$$R(d, q_1 \wedge q_2) = \min(R(d, q_1), R(d, q_2)).$$

$$R(d, q_1 \vee q_2) = \max(R(d, q_1), R(d, q_2)).$$

$$R(d, \neg q_1) = 1 - R(d, q_1).$$

Les modèles de RI: Le modèle booléen basé sur les ensembles flous

Un document d est représenté par un ensemble de termes pondérés comme suit:

$$d = \{\ldots, (t_i, w_i), \ldots\}$$

t_i est le terme, w_i est le poids associer au terme t_i

Le degré de correspondance (évaluation d'une requête) :

Évaluation 2: [Lukaswicz]

$$R(d, t_i) = w_i$$

$$R(d, q_1 \wedge q_2) = R(d, q_1) * R(d, q_2).$$

$$R(d, q_1 \vee q_2) = R(d, q_1) + R(d, q_2) - R(d, q_1) * R(d, q_2).$$

$$R(d, \neg q_1) = 1 - R(d, q_1).$$