

Recherche d'Information (RI)

- Fondements -

Philippe Mulhem
LIG
Philippe.Mulhem@imag.fr

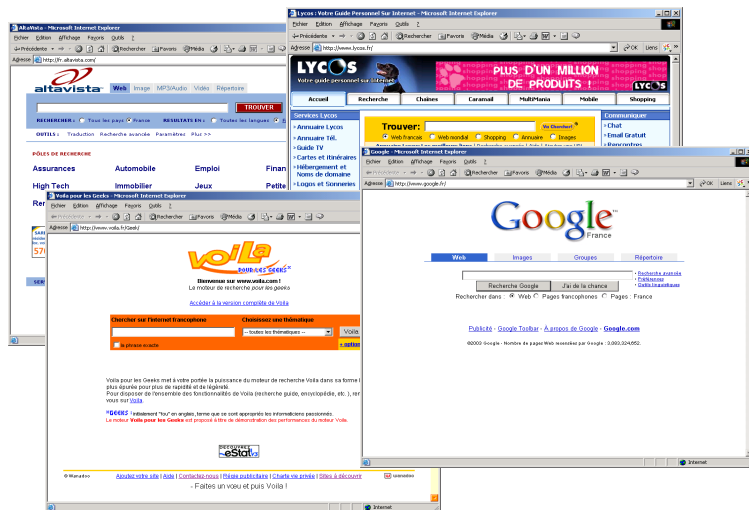
http://lig-membres.imag.fr/mulhem/wp-content/uploads/sites/11/2017/01/Fondements_RI.pdf

1

Plan

1. Introduction
2. Eléments clés en RI
3. Modèle de RI
 - Modèle Booléen
 - Modèle Booléen Pondéré
 - Modèle Vectoriel
4. Systèmes de recherche d'information (SRI)
 - Architecture
5. Evaluation de SRI
6. Conclusion

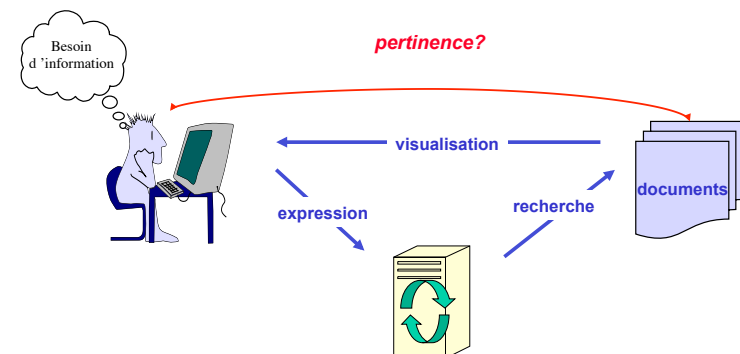
2



3

1. Introduction

- Problématique de la recherche d'information :
 - Accès par le contenu à des documents satisfaisant un besoin d'information d'un utilisateur



4

1. Introduction

- Contextes d'utilisation
 - Bureautique
 - Applications techniques : maintenance de matériel
 - Informatique médicale
 - Génie logiciel
 - Vente en ligne
 - Tourisme
 - Recherche scientifique
 - etc.

5

1. Introduction

- Points abordés sur le cours
 - Fondements de la Recherche d'Information
 - La RI textuelle sur le Web

6

2. Éléments clés en RI

- Quels éléments sont centraux pour la Recherche d'Information
 - Documents
 - Contenu des documents
 - Besoin d'information d'un utilisateur
 - Satisfaction

7

2. Éléments clés en RI

- Documents
 - Différents médias :
 - Texte, Image, vidéo, documents structurés
 - Différents types d'information par média
 - Texte : livre, article, lettre...
 - Image : Images par rayons X, Photographies, Graphiques, ...

8

2. Éléments clés en RI

- Contenu des documents
 - 2 classes d'information
 - Méta-Information (information à propos du document)
 - Attributs : titre, auteur, date de création, etc.
 - Structure (organisation du contenu) : structure logique, liens, etc.
 - Contenu
 - Contenu brut : le document initial
 - Contenu symbolique : information extraite du contenu brut

9

2. Éléments clés en RI

- Besoin d'information d'un utilisateur
Utilisation de requêtes suivant un langage fixé
 - Contraintes sur la méta information
 - Attributs : « roman écrit par Victor Hugo »
 - attribut de type de document et auteur
 - Structure : « article de football contenant une photographie »
 - Structure de lien entre texte et image
 - Contraintes sur le contenu
 - Contenu brut : « lettre avec le texte "Je suis venu, j'ai vu, j'ai vaincu" » : recherche sur des chaînes de caractères
 - Contenu symbolique : « documents au sujet de recherche d'information documentaire » : recherche sur des descriptions symboliques

10

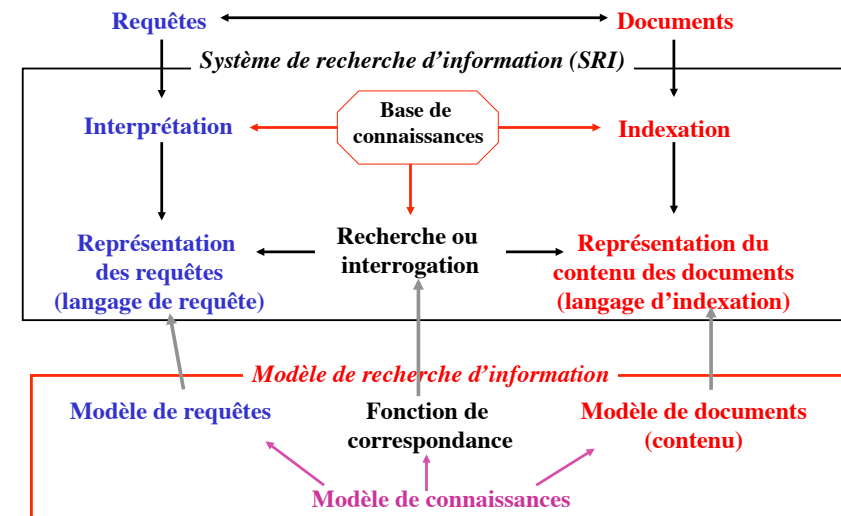
2. Éléments clés en RI

- Satisfaction de l'utilisateur
Le système doit
 - être simple à utiliser
 - fournir les meilleures réponses possibles, et ces réponses doivent être « pertinentes » pour l'utilisateur
 - Pertinence système versus pertinence utilisateur
 - fournir un nombre raisonnable de réponses
 - Le système doit fournir des réponses rapides

Difficile de satisfaire tous ces points à la fois...

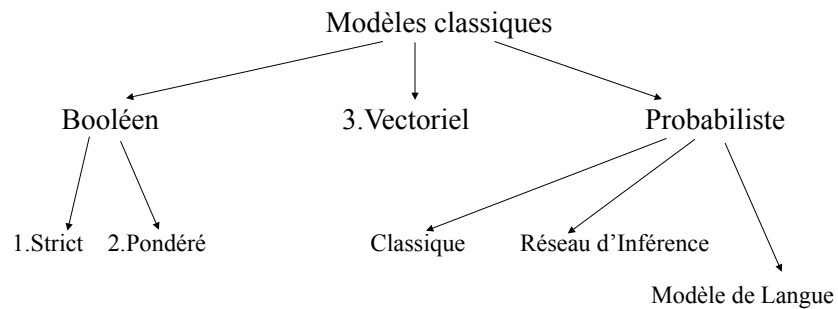
12

3. Modèles de RI



13

3. Modèles de RI



14

3. Modèles de RI

- Le modèle booléen strict (1)
 - Modèle de connaissances : $T = \{t_i\}, i \in [1, .. N]$
 - Termes t_i qui indexent les documents
 - Le modèle de documents (contenu) est une expression booléenne dans la logique des propositions avec les t_i considérés comme des propositions :
 - Un document $D1$ est représenté par une formule $\mathcal{D}I$

$$\mathcal{D}I = t_1 \wedge t_3 \wedge t_{250} \wedge t_{254} (\wedge_{j \neq 1,3,250,254} \neg t_j)$$
 - Une requête Q est représentée par une formule logique Q

$$Q = (t_1 \wedge t_3) \vee (t_{25} \wedge t_{145} \wedge \neg t_{134})$$

15

3. Modèles de RI

- Le modèle booléen strict (2)
 - La fonction de correspondance est basée sur l'implication logique en logique des propositions :
 - Un document D répond à une requête Q si et seulement si

$$\mathcal{D} \supset Q$$
 - Utilisation de déduction par
 - » Axiomes : $(a \wedge b) \supset a, (a \wedge b) \supset b, a \supset (a \vee b), b \supset (a \vee b), \dots$
 - Exemple : $\mathcal{D} = t_1 \wedge t_3 \wedge \neg t_2 \wedge \neg t_4$ et $Q = t_1 \vee t_4$
 - Déduction :
 1. $t_1 \wedge t_3 \wedge \neg t_2 \wedge \neg t_4 \supset t_1$ (équivalent à $\mathcal{D} \supset t_1$)
 3. $t_1 \supset t_1 \vee t_4$ (équivalent à $t_1 \supset Q$)
 - » Q est donc dérivable à partir de $\mathcal{D}$, donc $\mathcal{D} \supset Q$, donc le document répond à la requête.

16

3. Modèles de RI

- Le modèle booléen strict (3)
 - Remarques (1)
 - Correspondance stricte : Oui/Non
 - $Q = t_1 \wedge t_3 \wedge t_4$
 - $\mathcal{D}I = t_1 \wedge t_4 \wedge \neg t_2 \wedge \neg t_3$
 - $\mathcal{D}I \not\supset Q$
 - Le document $D1$ (représenté par $\mathcal{D}I$) n'est pas pertinent pour la requête Q (représentée par Q) d'après le modèle, alors qu'il contient une description « proche » de la requête.

17

3. Modèles de RI

- Le modèle booléen strict (4)

- Remarques (2)

- Pas de distinction entre les documents pertinents

$$-Q = t_1 \wedge t_4$$

$$-D2 = t_1 \wedge t_4 \wedge \neg t_2 \wedge \neg t_3 \wedge \neg t_5 \wedge \neg t_6 \wedge \neg t_7$$

$$-D3 = t_1 \wedge t_3 \wedge t_4 \wedge t_5 \wedge t_6 \wedge t_7 \wedge \neg t_2$$

$$D2 \supset Q \text{ et } D3 \supset Q$$

- Le document D2 (représenté par $D2$) est-il plus ou moins pertinent que D3 (représenté par $D3$) pour la requête Q (représentée par Q) ?

18

3. Modèles de RI

- Le modèle booléen strict (5)

- Remarques (3)

- Utilisable pour des experts, mais difficile pour des utilisateurs "courants"

- Une requête $t_1 \wedge t_4 \wedge t_6$ peut donner des centaines de réponses, et la requête $t_1 \wedge t_4 \wedge t_6 \wedge t_{10}$ peut ne donner aucune réponse ...
- Il faut donc savoir jongler avec les $\wedge$ et les $\vee$

- Expression de requêtes complexe

$$-Q = ((t_1 \wedge t_4) \vee t_6) \wedge (t_8 \vee (\neg t_{10} \wedge t_{40})) \dots ???$$

- Sens du $\vee$ logique (inclusif) différent du « ou » courant (exclusif)

⇒ Ceci a amené à définir des modèles fournissant des résultats sous forme de liste

19

3. Modèles de RI

- Le modèle booléen pondéré (1)

- Extension du modèle booléen en intégrant des pondérations (dénnotant la représentativité d'un terme pour un document).

- Modèle de connaissances : $T = \{t_i\}, i \in [1, \dots N]$

- Termes t_i qui indexent les documents

- Un document D est représenté par :

- Une formule logique $\mathcal{D}$ (idem modèle booléen)
- Une fonction $W_{\mathcal{D}} : T \rightarrow [0,1]$, qui pour chaque terme de T donne le poids de ce terme dans $\mathcal{D}$. Le poids vaut 0 pour un terme non présent dans le document.

- Une requête Q est similaire au modèle booléen strict

20

3. Modèles de RI

- Le modèle booléen pondéré (2)

- Fonction de correspondance non binaire (on se passe des implications logiques)

- basée sur une similarité notée *Sim*

- Version1

- Formules inspirées de la logique floue (avec a et b des termes)

$$\gg \text{Sim}_1(\mathcal{D}, (a \wedge b)) = \min [W_{\mathcal{D}}(a), W_{\mathcal{D}}(b)]$$

$$\gg \text{Sim}_1(\mathcal{D}, (a \vee b)) = \max [W_{\mathcal{D}}(a), W_{\mathcal{D}}(b)]$$

$$\gg \text{Sim}_1(\mathcal{D}, (\neg a)) = 1 - W_{\mathcal{D}}(a)$$

- Requêtes complexes (x et y sont des sous-requêtes):

$$\gg \text{Sim}_1(\mathcal{D}, (x \wedge y)) = \min [\text{Sim}_1(\mathcal{D}, x), \text{Sim}_1(\mathcal{D}, y)]$$

- Limitation : on ne tient pas compte dans la réponse de tous les termes de la requête : ex; $\min(0.5, \min(0.3, 0.5)) = \min(1, \min(0.3, 1))$

21

3. Modèles de RI

- Le modèle booléen pondéré (3)

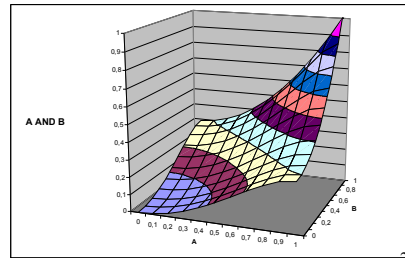
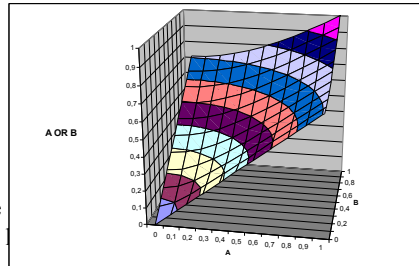
- Version 2

- Définition d'une mesure de similarité qui tient davantage compte de tous les termes de requête

$$Sim_2(D, a \wedge b) = 1 - \sqrt{\frac{(1 - W_D(a))^2 + (1 - W_D(b))^2}{2}}$$

$$Sim_2(D, a \vee b) = \sqrt{\frac{W_D(a)^2 + W_D(b)^2}{2}}$$

- Pas de formule pour la négation ...



22

3. Modèles de RI

- Le modèle booléen pondéré (4)

- Ex. avec des poids binaires pour les documents

Documents	a	b	Sim ₁		Sim ₂	
			a ∨ b	a ∧ b	a ∨ b	a ∧ b
D ₁	1	1	1	1	1	1
D ₂	1	0	1	0	1/√2	1 - 1/√2
D ₃	0	1	1	0	1/√2	1 - 1/√2
D ₄	0	0	0	0	0	0

22

3. Modèles de RI

- Le modèle booléen pondéré (5)

- Ex. avec des poids non-binaires pour les documents

Documents	a	b	Ba Sim ₁ 1)		B Sim ₂ 2)	
			a ∨ b	a ∧ b	a ∨ b	a ∧ b
D ₁	1	1	1	1	1	1
D ₂	0.8	1	1	0.8	0.91	0.86
D ₃	0	0.5	0.5	0	0.35	0.21
D ₄	0.8	0	0.8	0	0.57	0.28

24

3. Modèles de RI

- Le modèle booléen pondéré (6)

- Traitement de requête (totalement parenthésée) Sim₁

- Ex : (((a ∧ b) ∨ c) ∧ ¬ d)

- Etape 1 : génération de l'arbre de requête

- Une parenthèse (on a un opérateur binaire ∧ ou ∨) :

- un nœud non-feuille avec 2 fils, décoré avec l'opérateur
- Fils gauche : sous-arbre avec partie gauche
- Fils droit : sous-arbre avec partie droite

- Un terme : nœud feuille avec le terme

- Un ¬ : un nœud non-feuille unaire avec sous-arbre fils

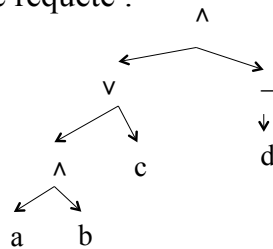
- Génération des opérations

25

3. Modèles de RI

- Le modèle booléen pondéré (7)

- Ex . $((a \wedge b) \vee c) \wedge \neg d$
- Arbre de requête :

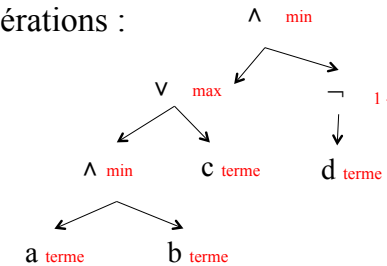


26

3. Modèles de RI

- Le modèle booléen pondéré (8)

- Ex . $((a \wedge b) \vee c) \wedge \neg d$
- Génération des opérations :



→ Opérations effectuées :
 $\min(\max(\min(WD(a), WD(b)), WD(c)), 1-WD(d))$

27

3. Modèles de RI

- Le modèle vectoriel (1)

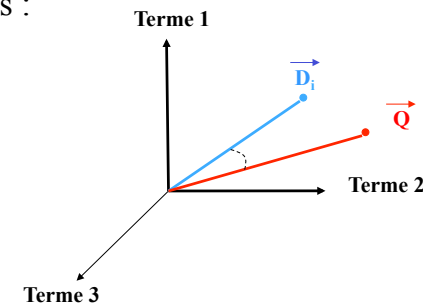
- Modèle de connaissances : $T = \{t_i\}, i \in [1, .. N]$
- Tous les documents sont décrits suivant ce vocabulaire
- Un document D_i est représenté par un vecteur $\vec{D_i}$ décrit dans l'espace vectoriel $\mathbb{R}^N$ défini par T :
 - $\vec{D_i} = (w_{i,1}, w_{i,2}, \dots, w_{i,j}, \dots, w_{i,N})$, avec w_{kl} le *poids* d'un terme pour un document
- Une requête Q est représentée par un vecteur $\vec{Q}$ décrit dans l'espace vectoriel $\mathbb{R}^N$ défini par T :
 - $\vec{Q} = (w_{Q,1}, w_{Q,2}, \dots, w_{Q,j}, \dots, w_{Q,N})$

28

3. Modèles de RI

- Modèle vectoriel (2)

- Plus les vecteurs représentant les documents/requêtes sont « proches », plus les documents/requêtes sont similaires :



29

3. Modèles de RI

- Modèle vectoriel (3)

- Comment trouver les poids des termes pour les documents :

- Un document

- « Un violon est composé de bois précieux comme l'érable, le palissandre, l'ébène... »

- Pour indexer, la première idée est de compter les mots les plus fréquents excepté les termes non significatifs comme « de », « avec », « comme »...

- « Un violon est composé de bois précieux comme l'érable, le palissandre, l'ébène... »



Termes retenus et comptés

30

3. Modèles de RI

- Modèle vectoriel (4) :

- On définit la "fréquence" d'un terme (term frequency)

- $tf_{i,j}$: la "fréquence" du terme t_j dans le document D_i est égale au nombre d'occurrences de t_j dans D_i .

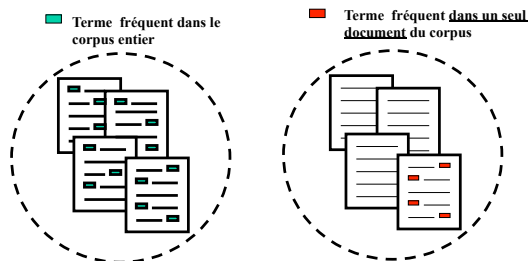
- Exemple : si violon apparaît 5 fois dans le document D_3 , avec $violon=t_{23}$, alors $tf_{3,23} = 5$

31

3. Modèles de RI

- Modèle vectoriel (5) :

- On tient compte du corpus (base de documents) entier, un terme qui apparaît beaucoup ne discrimine pas nécessairement les documents :



32

3. Modèles de RI

- Modèle vectoriel (6) :

- On définit la "fréquence" documentaire d'un terme

- df_j : la fréquence dans le corpus du terme t_j est le nombre de documents du corpus où t_j apparaît

- On utilise l'

- **inverse de la fréquence documentaire, idf_i**

- Définition simple : $idf_j = 1 / df_j$

- Définition la plus utilisée : $idf_j = \log(N_D / df_j)$, avec N_D le nombre de documents du corpus.

(idf : inverse document frequency)

33

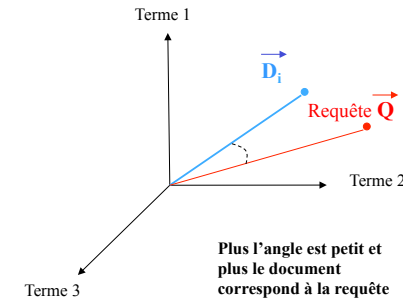
3. Modèles de RI

- Modèle vectoriel (7) :
 - Combinaison du tf et de l'idf pour un vecteur document:
 - Le poids d'un terme dénote la capacité du terme à discriminer les documents
 - Exemple le plus courant
 - $w_{i,j} = \text{tf}_{i,j} \cdot \text{idf}_j$
 - Utilisation du tf ou du **tf.idf** pour une requête

34

3. Modèles de RI

- Modèle vectoriel (8) :
 - Fonction de correspondance : fonction de l'angle entre le vecteur requête $\vec{Q}$ et le vecteur document $\vec{D}_i$



35

3. Modèles de RI

- Modèle vectoriel (9) :
 - Fonction de correspondance : une solution est de calculer le cosinus de l'angle entre le vecteur requête et le vecteur document.

$$\begin{aligned} \text{Sim}(\vec{D}_i, \vec{Q}) &= \frac{\sum_{k=1}^N (w_{i,k} \cdot w_{q,k})}{\sqrt{\sum_{k=1}^N (w_{i,k}^2) \cdot \sum_{k=1}^N (w_{q,k}^2)}} \\ &= \frac{\sum_{k=1}^N (w_{i,k} \cdot w_{q,k})}{\|\vec{D}_i\| \cdot \|\vec{Q}\|} = \frac{\vec{D}_i \cdot \vec{Q}}{\|\vec{D}_i\| \cdot \|\vec{Q}\|} \end{aligned}$$

- Note : on constate que les $w_{q,k} = 0$ et $w_{i,k} = 0$ n'impactent pas le numérateur (utile pour les fichiers inverses)

36

3. Modèles de RI

- Modèle vectoriel (10)
 - Formule utilisée dans les systèmes :
 - On normalise en amont les vecteurs documents et requêtes

$$\begin{aligned} \text{Sim}(\vec{D}_i, \vec{Q}) &= \frac{\sum_{k=1}^N (w_{i,k} \cdot w_{q,k})}{\|\vec{D}_i\| \cdot \|\vec{Q}\|} = \sum_{k=1}^N \left(\frac{w_{i,k}}{\|\vec{D}_i\|} \cdot \frac{w_{q,k}}{\|\vec{Q}\|} \right) \\ &= \vec{D}'_i \cdot \vec{Q}' \end{aligned}$$

- Le résultat est exactement celui du cosinus initial

37

3. Modèles de RI

- Modèle vectoriel (11)
 - Il existe d'autres fonctions de correspondances

- Jaccard

$$Sim_{Jaccard}(\vec{D}_i, \vec{Q}) = \frac{\sum_{k=1}^N (w_{i,k} \cdot w_{q,k})}{\sum_{k=1}^N (w_{i,k}^2) + \sum_{k=1}^N (w_{q,k}^2) - \sum_{k=1}^N (w_{i,k} \cdot w_{q,k})}$$

- Dice

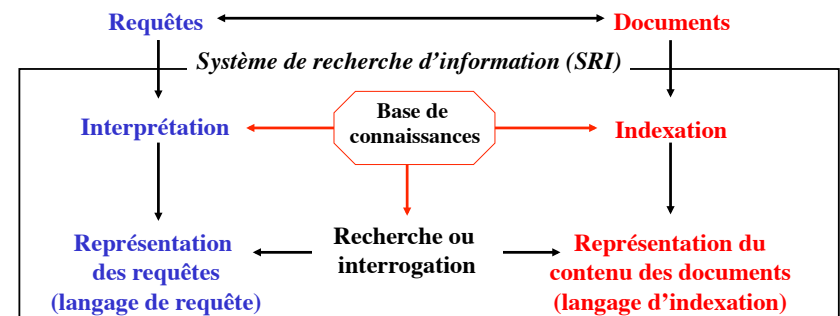
$$Sim_{Dice}(\vec{D}_i, \vec{Q}) = \frac{2 * \sum_{k=1}^N (w_{i,k} \cdot w_{q,k})}{\sum_{k=1}^N (w_{i,k}^2) + \sum_{k=1}^N (w_{q,k}^2)}$$

... mais le cosinus donne de meilleurs résultats

37

4. Systèmes de recherche d'information

- Un SRI est un système informatique qui instancie un modèle de recherche d'information



38

4. Systèmes de recherche d'information

- Indexation (1)
 - Choix des termes – Généralités
 - Une propriété souhaitée d'un bon terme d'indexation est sa capacité à distinguer les documents d'une collection les uns des autres
 - Comment faire?

39

4. Systèmes de recherche d'information

- Indexation (2)
 - Choix des termes – Occurrences
 - Objectif** : On veut trouver les mots qui représentent le mieux le contenu d'un document.
 - Hypothèse** : On admet généralement qu'un mot qui apparaît souvent dans un texte représente un concept important.
 - La première approche consiste :
 - à choisir les mots représentatifs selon leur fréquence d'occurrence.
 - à définir un seuil S_{MIN} sur la fréquence: si la fréquence d'occurrence d'un mot dépasse ce seuil, alors il est considéré important pour les documents qui le contiennent.

40

4. Systèmes de recherche d'information

• Indexation (3)

– Choix des termes – Occurrences (2)

- Cependant, quand on fait la statistique d'occurrence, on s'aperçoit que les mots les plus fréquents sont des mots fonctionnels (ou mots outils, mots vides). En français, les mots "de", "un", "les", etc. sont les plus fréquents. En anglais, ce sont "of", "the", etc.
- Ce phénomène n'est pas étrange si on connaît **la loi de Zipf**.

41

4. Systèmes de recherche d'information

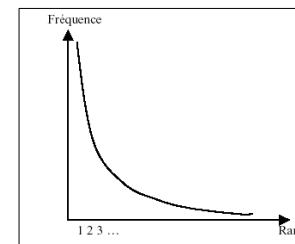
• Indexation (4)

– Choix des termes – Occurrences (3)

• La loi de Zipf

- Si on classe les mots dans l'ordre décroissant de leur fréquence, et on leur donne un numéro de rang (1, 2, ...), alors: $\text{Rang} * \text{fréquence} \approx \text{constante}$.

Rang	Mot	Fréquence	Rang * Fréquence
1	the	69 971	69 971
2	of	36 411	72 822
3	and	28 852	86 556
4	to	26 149	104 596
5	a	23 237	116 185
6	in	21 341	128 046
7	that	10 595	76 165



- La distribution de mots suit la courbe :
- Une idée est de garder les termes "utiles" : ni trop rares (place en mémoire), ni trop présents (pas discriminants)... choix difficile

42

4. Systèmes de recherche d'information

• Indexation (5)

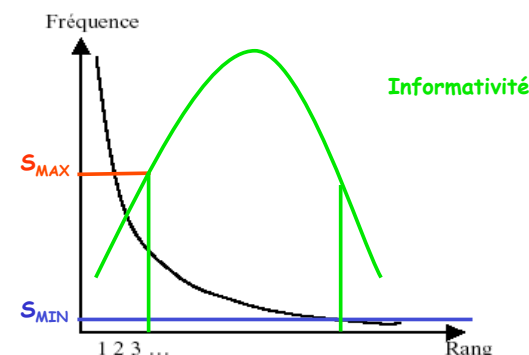
– Choix des termes – Occurrences (4)

- On ne garde pas les mots très peu fréquents car ils vont prendre la place en mémoire "pour rien" : seuil S_{MIN}
- On définit un seuil S_{MAX} sur la fréquence: si la fréquence d'occurrence d'un mot dépasse ce seuil, alors il n'est pas considéré comme important pour les documents
- Hypothèse : On admet que l'utilisation de ces deux seuils S_{MIN} et S_{MAX} donne une indication sur l'*informativité* des mots.
- Cette notion définie non précisément en RI est censée mesurer la quantité de sens qu'un mot porte

4. Systèmes de recherche d'information

• Indexation (6)

– Choix des termes – Occurrences (5)



4. Systèmes de recherche d'information

• Indexation (7)

– Choix des termes – Discrimination

- Utilisation du centre de gravité des documents

– $C = (c_1, c_2, \dots, c_N)$

– Avec

$$c_i = \frac{1}{M} \sum_{k=1}^M w_{k,i}$$

	t1	t2	t3	...	tN
d ₁	w _{1,1}	w _{1,2}	w _{1,3}	...	w _{1,N}
⋮					⋮
d _M	w _{M,1}	w _{M,2}	w _{M,3}	...	w _{M,N}

- Calcul de la densité de l'espace, $Dens$

» Utilisation d'une fonction de similarité entre les documents dans $[0,1]$, $Sim(D_i, D_j)$ (par exemple 1/distance euclidienne)

$$Dens = \sum_{i=1}^M Sim(d_i, C)$$

» Si $Dens \approx 0$ alors espace creux (bon pour la RI)

» Si $Dens \approx M$ alors espace dense (pas bon pour la RI)

4. Systèmes de recherche d'information

• Indexation (8)

– Choix des termes – Discrimination (2)

- Utilisation de la discrimination des termes

– Calcul de densité de l'espace $Dens$ pour toutes les dimensions

– Calcul de la densité de l'espace privé de la dimension d'un terme j , notée $Dens_j$.

$$dv_j = Dens_j - Dens$$

– si $dv_j > 0$ alors j est un bon terme d'indexation

– si $dv_j \leq 0$ alors j n'est pas un bon terme d'indexation

4. Systèmes de recherche d'information

• Indexation (5)

– Choix des termes – A priori

- Utilisation d'un anti-dictionnaire

– Au, aux, avec, ce, ces, dans, de, des, du, elle, en, etc, et, eux, il, je, la, le, les, leur, lui ...

– Ne garder que des termes qui ont du sens, diminuer la taille des index

- Extraction de troncatures des mots du texte :

– Algorithme de Porter (anglais) :

– Règles (exemples)

» s → /

» ed → /

» ing → /

» er → /

» e → /

Mot initial	Mot tronqué
engineered	engin
engineer	engin
engineers	engin
informing	inform
computer	comput
computing	comput

– Diminuer la taille des index, grouper les termes « similaires »

4. Systèmes de recherche d'information

• Indexation (6)

– Fichiers inverses - principe

- Après analyse de documents d'un corpus, on obtient un tableau document x termes

- Utilisation en **tableau direct**

« document → terme » possible

	t1	t2	t3	...	tN
d ₁					
⋮					⋮
d _M				...	

- Génération d'un tableau inverse « terme → document » (appelé fichier inverse)

– Avantage : rapidité lors du traitement de requête, car pas de traitement séquentiel des documents

	d ₁	d ₂	d ₃	...	d _M
t ₁					
⋮					⋮
t _N				...	

4. Systèmes de recherche d'information

- Indexation (7)

- Fichier inverse avec le modèle booléen,

	D ₁	D ₂	D ₃	...	D _M
t ₁	1	0	0		1
⋮					⋮
t _N	1	0	1	...	0

- Un 1 indique que le terme apparaît positivement dans un document, un zéro indique que le terme n'apparaît pas dans le document

45

4. Systèmes de recherche d'information

- Indexation (8)

- Fichier inverse avec les modèles pondérés (booléen pondéré, vectoriel, modèle de langue)

	D ₁	D ₂	D ₃	...	D _M
t ₁	w _{1,1}	w _{2,1}	w _{3,1}		w _{M,1}
⋮					⋮
t _N	w _{1,N}	w _{2,N}	w _{3,N}	...	w _{M,N}

- Même idée que pour les modèles non pondérés.
 - Pour un modèle vectoriel avec tf.idf, il faut calculer l'idf de ces termes.

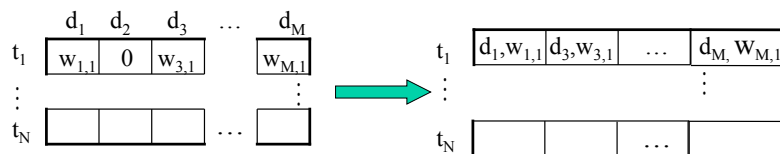
46

4. Systèmes de recherche d'information

- Indexation (9)

- En fait, les fichiers inverses ne stockent pas toutes les valeurs, car il y a beaucoup de valeurs nulles (> 90% des cases du tableau) :

- on utilise des représentations de tableaux creux (tableau avec tailles de lignes différentes, listes chaînées)



51

4. Systèmes de recherche d'information

- Indexation (10)

- Pour une implantation d'un modèle vectoriel, on stocke aussi pour chaque terme du vocabulaire son idf (utilisé pour les requêtes) : utilisation lors de la définition du vecteur requête.

47

4. Systèmes de recherche d'information

- Recherche (1)
 - Implantation du modèle booléen par fichier inverse
 - Requête booléenne transformée en une séquence d'opérateurs logiques sur les lignes du fichier inverse
 - $t_1 \wedge t_N$ traduit en : ligne_résultante = ligne(t_1) ET_{Bool_ligne} ligne(t_N)
 - Eléments à 1 dans la ligne résultante dénotent les documents pertinents pour le système

4. Systèmes de recherche d'information

- Recherche (2)
 - Implantation du modèle booléen pondéré par fichier inverse
 - Requête booléenne transformée en séquence d'opérateurs sur les lignes du fichier inverse
 - $t_1 \vee t_2$ traduit en : ligne_résultante = ligne(t_1) OU_{pond_ligne} ligne(t_2)
 - Avec OU_{pond_ligne} défini dans le modèle pour chaque document D_j :
$$res(j) = \sqrt{\frac{w_{j,1}^2 + w_{j,2}^2}{2}}$$
 - Tri des résultats par ordre de pertinence décroissant et affichage

4. Systèmes de recherche d'information

- Recherche (3)
 - Implantation du modèle vectoriel par fichier inverse
 - Requête $Q = (w_{q,1} \dots w_{q,N})$
 - Init res1(j)=0 pour j de 1 à M
 - Selection des lignes k telles que $w_{q,k} \neq 0$
 - Pour les documents D_j tels que $w_{j,k} \neq 0$
 - Calcul : $res1(j) += w_{j,k} \cdot w_{q,k}$
 - Finalisation
 - Calcul : $res2(j) = res1(j) / (\|Q\| \cdot \|D_j\|)$ pour tous les j de 1 à M
 - Tri des résultats par ordre décroissant et affichage

4. Systèmes de recherche d'information

- Recherche (1)
 - Implantation du modèle booléen strict par fichier inverse
 1. Requête booléenne transformée en un arbre d'analyse
 2. Chaque nœud non feuille de l'arbre est associé à un opérateur logique sur les lignes du fichier inverse et à une ligne intermédiaire de résultats
 3. Eléments à 1 dans la ligne résultante de la racine dénotent les documents pertinents pour le système

4. Systèmes de recherche d'information

• Recherche (2) - Implantation du modèle booléen strict

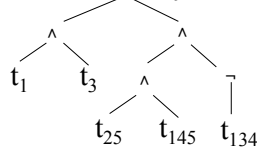
– Requête booléenne transformée en un arbre d'analyse

$$-Q = (t_1 \wedge t_3) \vee (t_{25} \wedge t_{145} \wedge \neg t_{134})$$

- On parenthèse totalement la requête pour avoir des opérateurs binaires ou unaires:

$$-Q = ((t_1 \wedge t_3) \vee ((t_{25} \wedge t_{145}) \wedge \neg t_{134}))$$

- On en déduit l'arbre d'analyse



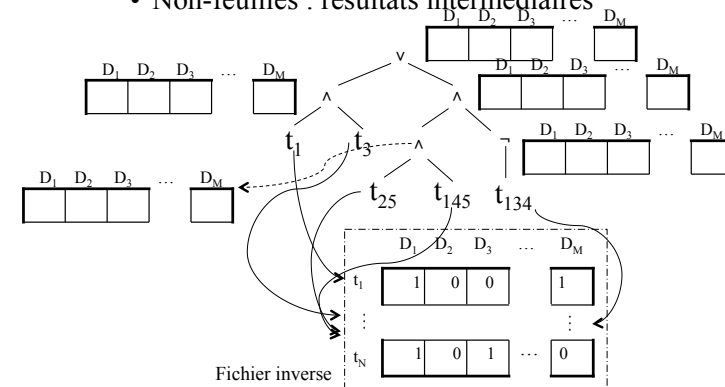
49

4. Systèmes de recherche d'information

• Recherche (3) – Impl. modèle booléen strict

– Nœuds de l'arbres associés à une ligne

- Feuilles : lignes du fichier inverse
- Non-feuilles : résultats intermédiaires



50

4. Systèmes de recherche d'information

• Recherche (4) – Impl. du modèle booléen strict

– Nœuds de l'arbres associés à une ligne : remplissage des nœuds intermédiaires des feuilles vers la racine

- Nœud $\wedge$, fils_gauche $\wedge$ fils_droit traduit en
– ligne_noeud = ligne(fils_gauche) ET_{Bool_ligne} ligne(fils_droit)
- Nœud $\vee$, fils_gauche $\vee$ fils_droit traduit en
– ligne_noeud = ligne(fils_gauche) OU_{Bool_ligne} ligne(fils_droit)

- Nœud $\neg$, $\neg$ fils traduit en
– ligne_noeud = NEG_{Bool_ligne} ligne(fils)

a	$\neg a$
0	1
1	0

a	b	$a \wedge b$	$a \vee b$
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	1

51

4. Systèmes de recherche d'information

• Recherche (5)

– Implantation du modèle booléen pondéré par fichier inverse

- Similaire au traitement du modèle booléen
 - Changement est au niveau de la génération des lignes associées aux nœuds non-feuilles
 - » Par exemple : fils_gauche $\vee$ fils_droit traduit e
ligne_résultante = ligne(fils_gauche) OU_{pond_ligne} ligne(fils_droit)
 - » avec OU_{pond_ligne} défini par :

$$res(j) = \sqrt{\frac{fils_gauche(j)^2 + fils_droit(j)^2}{2}}$$

- Ajout du tri des résultats par ordre de pertinence décroissant avant affichage

52

4. Systèmes de recherche d'information

- Recherche (6)
 - Implantation simple du modèle vectoriel par fichier inverse
 - Requête $Q = (w_{q,1} \dots w_{q,N})$
 - On garde les termes t_i tels que $w_{q,i} \neq 0$
 - Boucle pour chaque terme t_i et pour les documents D_j
 - $\text{ligne-resultante}[j] += w_{j,i} * w_{q,i}$
(utilisation du fichier inverse pour les $w_{j,i}$)
 - Calcul final : $\text{resultat}[j] = \text{ligne-resultante}[j] / (\|D_j\| \cdot \|Q\|)$
 - Tri des résultats par ordre décroissant et affichage

53

4. Systèmes de recherche d'information

- Recherche (7)
 - Implantation avancée : normalisation en amont $w \Rightarrow w'$ pour les requêtes et les documents

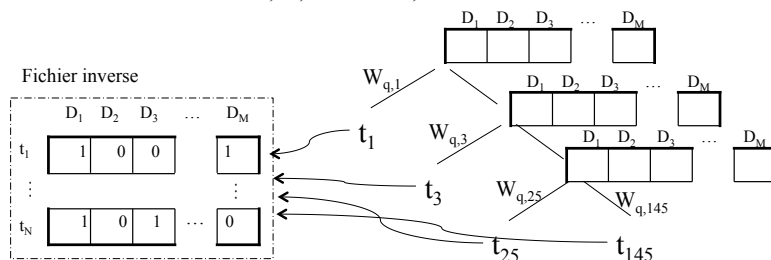
$$\begin{aligned} \text{Sim}(\vec{D}_j, \vec{Q}) &= \frac{\sum_{k=1}^N (w_{j,k} \cdot w_{q,k})}{\|\vec{D}_j\| \cdot \|\vec{Q}\|} = \sum_{k=1}^N \left(\frac{w_{j,k}}{\|\vec{D}_j\|} \cdot \frac{w_{q,k}}{\|\vec{Q}\|} \right) \\ &= \sum_{k=1}^N (w'_{j,k} \cdot w'_{q,k}) \end{aligned}$$

- On stocke dans le fichier inverse les $w'_{i,k}$, et donc :
 - Calcul final : $\text{resultat}[j] = \text{ligne-resultante}[j]$

54

4. Systèmes de recherche d'information

- Recherche (8)
 - En fait, cette implantation est similaire à celle d'un système pondéré :
 - Si on suppose que dans Q les termes w_q non-nuls sont ceux des termes, t_1, t_3, t_{25} et t_{145} , alors on fait



55

4. Systèmes de recherche d'information

- Recherche (9)
 - Modification de requête avec un modèle booléen
 - Idée : utiliser des relations entre les termes (ex. : domaine médical)
 - Relations
 - Tirées d'un thésaurus existant
 - » Par exemple MeSH ou UMLS dans le domaine médical
 - Calculées d'après un corpus existant
 - » Construction d'une matrice de co-occurrences des termes dans les documents
 - Utilisation des relations dans un modèle booléen strict ou pondéré:
 - Si un terme t_1 est relié au terme t_2 , alors une requête utilisateur de la forme : $Q = t_1 \wedge t_{23} \wedge t_{45}$, est transformée en : $Q' = (t_1 \vee t_2) \wedge t_{23} \wedge t_{45}$

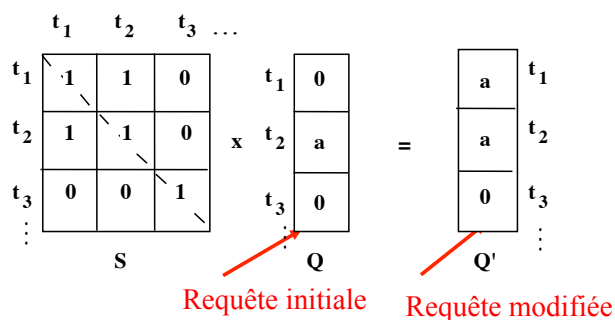
56

4. Systèmes de recherche d'information

• Recherche (10)

– Modification de la requête dans le modèle vectoriel

- Utilisation de relations entre les termes :



57

4. Systèmes de recherche d'information

• Recherche (11)

– Bouclage de pertinence (dans le modèle vectoriel)

- Idée : l'utilisateur peut montrer ce qui est pertinent pour lui-même s'il ne peut pas l'exprimer explicitement
- Mise en œuvre
 - Une requête initiale Q_0 fournie par l'utilisateur renvoie une réponse
 - L'utilisateur indique dans la liste des réponses les documents pertinents et non pertinents pour lui
 - Le système génère une nouvelle requête Q_1 qui prend en compte Q_0 et les pertinents et les non pertinents indiqués

58

4. Systèmes de recherche d'information

• Recherche (12)

– Bouclage de pertinence dans le modèle vectoriel

- Formule de Rocchio

$$Q_1 = \alpha Q_0 + \beta \left(\frac{1}{|D_R|} \sum_{d \in D_R} d \right) - \gamma \left(\frac{1}{|D_N|} \sum_{d' \in D_N} d' \right)$$

– Avec

- » D_R l'ensemble des docs marqués pertinents par l'utilisateur
- » D_N l'ensemble des docs marqués non-pertinents par l'utilisateur
- » $\alpha \geq \beta \geq \gamma$
- » valeurs possibles $\alpha=1 \beta=0.4 \gamma=0.2$, ou même 1 1 0

- Le bouclage de pertinence donne de très bons résultats.

59

5. Evaluation des SRI

- Rappel : Un système de recherche d'information doit satisfaire un besoin d'information **d'un utilisateur**

pertinence utilisateur $\neq$ pertinence système

- Pertinence utilisateur : satisfaction de l'utilisateur
- Pertinence système : estimation du système

60

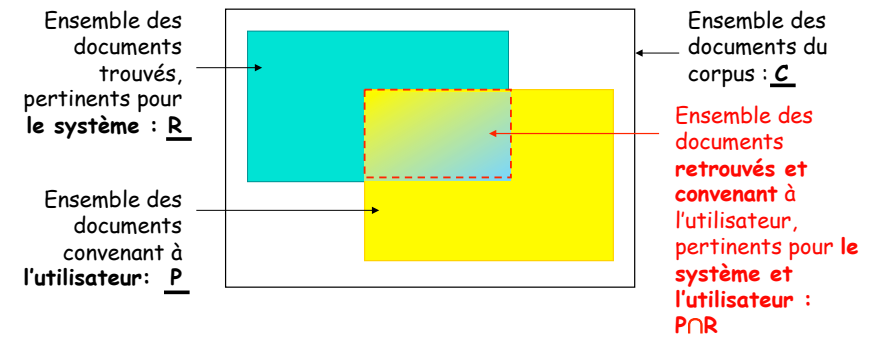
5. Evaluation des SRI

- Comparer les SRI de manière théorique (via leur modèle) est un problème non-résolu, on passe donc par des évaluations de type « boîte noire » en testant les résultats d'un système par rapport aux réponses idéales attendues.

61

5. Evaluation des SRI

- Objectif :
 - Rapprocher pertinence système et utilisateur



62

5. Evaluation des SRI

- Les critères essentiels sont :
 - Le rappel : capacité du système à fournir en réponse tous les documents pertinents pour l'utilisateur
 - La précision : capacité du système à ne fournir que des documents pertinents pour l'utilisateur en réponse.
 - Ces deux critères sont antagonistes dans la réalité...

63

5. Evaluation des SRI

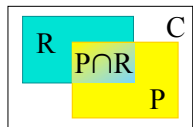
- Le rappel est le rapport du nombre de documents trouvés par le système et convenant à l'utilisateur au nombre total de documents convenant à l'utilisateur

$$rappel = \frac{|P \cap R|}{|P|} \in [0,1]$$

64

5. Evaluation des SRI

- La précision est le rapport du nombre de documents trouvés par le système et convenant à l'utilisateur au nombre de documents retrouvés par le système



$$\text{précision} = \frac{|P \cap R|}{|R|} \in [0,1]$$

65

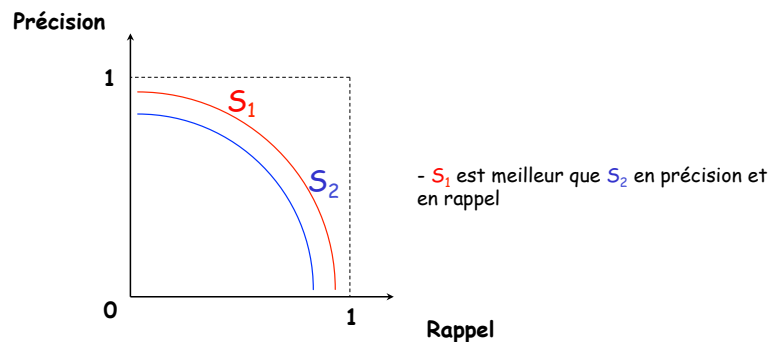
5. Evaluation des SRI

- Pour une requête et un système : 2 valeurs réelles
 - Exemple : un système retourne 5 documents, parmi lesquels 3 sont pertinents, sachant qu'il y a 10 documents pertinents dans le corpus :
 - Rappel = 3 / 10
 - Précision = 3 / 5
- Il faut des analyses plus fines des résultats
 - Courbes de rappel/précision

66

5. Evaluation des SRI

- Courbes de rappel/précision
 - Comparaison de 2 systèmes S1 et S2



67

5. Evaluation des SRI

- Courbes de rappel/précision
 - Représente l'évolution de la précision et du rappel avec des résultats triés
 - Méthode :
 - Pour chaque document retrouvé, on calcule la précision et le rappel obtenus en considérant seulement le premier document comme réponse, puis les deux premiers, puis les trois premiers etc., jusqu'à la réponse totale du système.
 - Ceci donne un tableau de rappel/précision non-normalisé

68

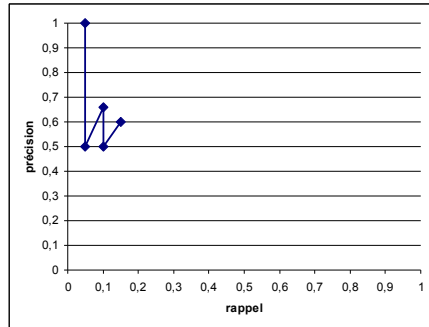
5. Evaluation des SRI

- Courbes de rappel/précision

- Exemple de tableau non normalisé (hypothèse : 20 documents pertinents)

Doc	P?	Rap.	Préc.
D ₂₃	Oui	0.05	1
D ₁₂	Non	0.05	0.5
D ₅	Oui	0.1	0.66
D ₃	Non	0.1	0.5
D ₇	Oui	0.15	0.6

Tableau non normalisé



69

5. Evaluation des SRI

- Courbes de rappel/précision

- Problème, comment faire pour fusionner les résultats de plusieurs requêtes pour un système?
- On normalise la courbe de chaque requête par la règle du maximum :

1. On fixe une valeur de rappel normalisée r (dans $[0, 0.1, 0.2, \dots, 0.8, 0.9, 1]$);
2. On garde la valeur de précision max. pour les valeurs de rappels du tableau non normalisé $\geq r$, et on la met dans le tableau normalisé pour le rappel r .

Rap.	Préc.
0.05	1
0.05	0.5
0.1	0.66
0.1	0.5
0.15	0.6

Tableau non normalisé →

Tableau normalisé →

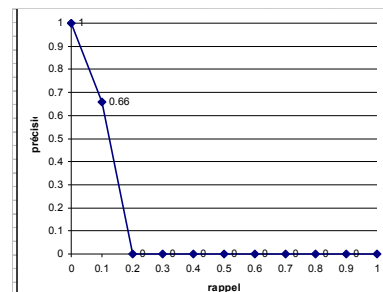
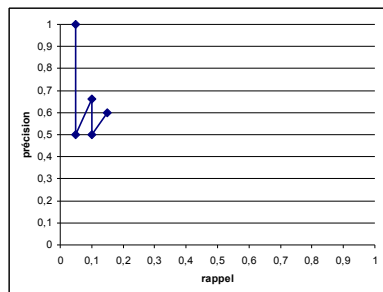
Rap.	Préc.
0	1
0.1	0.66
0.2	0
0.3	0
0.4	0
0.5	0
0.6	0
0.7	0
0.8	0
0.9	0
1	0

70

5. Evaluation des SRI

- Courbes de rappel/précision

- au niveau graphique



71

5. Evaluation des SRI

- Courbes de rappel/précision

- Pour traiter l'évaluation sur plusieurs requêtes, on calcule la moyenne aux points de rappels standards. (moyenne au point 0, au point 0.1, etc.)

- Précision moyenne à x documents

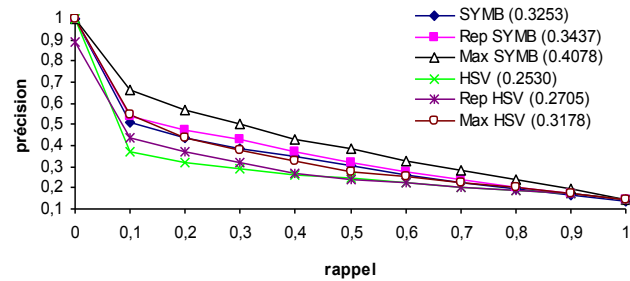
- Il est également courant de calculer le taux de précision après un nombre de documents x fixé pour une requête, puis de faire la moyenne sur toutes les requêtes

- Il existe des programmes qui génèrent les tableaux pour les courbes de rappel/précision et les précisions moyennes à 5, 10, 20, 50 et 100 documents. (trec_eval)

72

5. Evaluation des SRI

- Exemple de « vraie » courbe de rappel/précision



– Question : quel système est le meilleur pour les courbes ci-dessus?

6. Conclusion

- Nous avons traité la description de modèles de recherche d'information.
- Nous avons étudié les principes de base des systèmes basés sur ces modèles
- Nous avons compris comment réaliser une évaluation d'un système de recherche d'information.