RICM 4

HMUL8R6B: Accès et recherche d'information

Modèles de recherche & mesures d'évaluation

Massih-Reza Amini

Université Joseph Fourier
Laboratoire d'Informatique de Grenoble
Massih-Reza.Amini@imag.fr

Les différents modèles standard

- Modéle booléen
- Modéle vectoriel
- Modéles probabilistes

Notations

x_w^q	Nbre occurrences du mot w dans q
x_w^d	Nbre occurrences du mot w dans le document d
t_w^d	Version normalisée de x_w^d (poids)
N	Nbre de documents dans la collection
M	Nbre de termes dans la collection
F_w	Nbre d'occ. total de w : $F_w = \sum_d x_w^d$
N_w	Fréquence documentaire de w : $N_w = \sum_d I(x_w^d > 0)$
y_d	Longueur du document d
m	Longueur moyenne dans la collection
L	Longueur de la collection
RSV	Retrieval Status Value (score)

Le modèle booléen (1)

Modèle simple fondé sur la théorie des ensembles et l'algèbre de Boole, caractérisé par :

- Des poids binaires (présence/absence)
- Des requêtes qui sont des expressions booléennes
- Une pertinence binaire
- Pertinence système : satisfaction de la requête booléenne

Le modèle booléen (2)

Exemple

$q =$ programmation $\wedge$ langage $\wedge$ (C $\vee$ java)

($q =$ [prog. $\wedge$ lang. $\wedge$ C] $\vee$ [prog. $\wedge$ lang. $\wedge$ java])

	programmation	langage	C	java	$\cdots$
d_1	3 (1)	2 (1)	4 (1)	0 (0)	$\cdots$
d_2	5 (1)	1 (1)	0 (0)	0 (0)	$\cdots$
d_0	0 (0)	0 (0)	0 (0)	3 (1)	$\cdots$

Score de pertinence

$RSV(d_j, q) = 1$ si $\exists\, q_{cc} \in q_{dnf}$ tq $\forall w, t_w^d = t_w^q$; 0 sinon

Le modèle booléen (3)

Considérations algorithmiques

Quand la matrice documents-termes est creuse (lignes et colonnes), utiliser un fichier inverse pour sélectionner le sous-ensemble des documents qui ont un score de pertinence non nul avec la requête (sélection rapidement réalisée). Le score de pertinence n'est alors calculé que sur les documents de ce sous-ensemble (généralisation à d'autres types de score).

	d_1	d_2	d_3	$\cdots$
programmation	1	1	0	$\cdots$
langage	1	1	0	$\cdots$
C	1	0	0	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	

Le modèle booléen (4)

Avantages et désavantages

\+ Facile à développer

\- Pertinence binaire ne permet pas de tenir compte des recouvrements thématiques partiels

\- Passage d'une besoin d'information à une expression booléenne

Remarque À la base de beaucoup de systèmes commerciaux

Le modèle vectoriel (1)

Revient sur deux défauts majeurs du modèle booléen : des poids et une pertinence binaires
Il est caractérisé par :

- Des poids positifs pour chaque terme dans chaque document
- Mais aussi des poids positifs pour les termes de la requête
- Une représentation vectorielle des documents et des requêtes

Le modèle vectoriel (2)

On considère donc que les documents et les requêtes sont des vecteurs dans un espace vectoriel de dimension M dont les axes correspondent aux termes de la collection

Similarité Cosinus de l'angle entre les deux vecteurs

$$RSV(d_j, q) = \frac{\sum_w t_w^d t_w^q}{\sqrt{\sum_w (t_w^d)^2}\sqrt{\sum_w (t_w^q)^2}}$$

Propriété Le cosinus est maximal lorsque document et requête contiennent exactement les mêmes termes, dans les mêmes proportions ; minimal lorsqu'ils n'ont aucun terme en commun (*degré de similarité*)

Le modèle vectoriel (3)

Calcul des poids schémas *tf-idf*

Qu'est-ce qui décrit bien un document ?

$\rightarrow$ ses termes fréquents ($\text{tf}_w^d = \frac{x_w^d}{\max_{w'} x_{w'}^d}$)

Qu'est-ce qui distingue un document des autres ?

$\rightarrow$ ses termes spécifiques ($\text{idf}_w = \log \frac{N}{N_w}$)

$\rightarrow t_w = \text{tf}_w^d \times \text{idf}_w$

Le modèle vectoriel (4)

Avantages et désavantages

\+ Schémas de pondération permettant de prendre en compte différentes propriétés des index

\+ Un appariement partiel qui permet de retrouver les documents qui répondent en partie à la requête

\+ Un ordre total sur les documents qui permet de distinguer les documents qui abordent pleinement les thèmes de la requête de ceux qui ne les abordent que marginalement

\- Difficulté d'aller plus avant dans le cadre vectoriel (modèle relativement simple)

Complexité : comme le modèle booléen, linéaire sur le nombre de documents qui contiennent les termes de la requête (similarité requête-document plus coûteuse)

Les différents modèles probabilistes

- *Binary Independence Model* et BM25 (S. Robertson & K. Sparck Jones)
- *Inference Network Model* (Inquery) - *Belief Network Model* (Turtle & Croft)
- *Statistical Language Models*
  - *Query likelihood* (Ponte & Croft)
  - *Probabilistic distance retrieval model* (Zhai & Lafferty)
- *Divergence from Randomness* (Amati & Van Rijsbergen)

Généralités

Modèle booléen $\rightarrow$ pertinence binaire
Modèle vectoriel $\rightarrow$ degré de similarité
Modèle BIR $\rightarrow$ degré de pertinence
probabilité d'un document d'être pertinent

- R variable aléatoire binaire qui indique la pertinence : $R = r$ (*relevant*) ou $\bar{r}$ (*not relevant*)
- $P(R = r|d, q)$: probabilité que R prenne la valeur r pour le document d et la requête q considérés
- $RSV(q, d) = \log \frac{P(R=r|d,q)}{P(R=\bar{r}|d,q)}$

Deux points de vue peuvent être adoptés ici pour ré-écrire ces quantités : le point de vue *génération du document* (BIR) ou le point de vue *génération de la requête* (LM)

Le modèle BIR (1) : Hypothèses

H_1. la première stipule que les documents et les requêtes sont représentés sous forme de vecteurs binaires de même tailles que le vocabulaire de la collection. Si un terme du vocabulaire est présent dans un document (ou une requête), la caractéristique associée à ce terme dans le vecteur représentatif du document (ou de la requête) est fixée à 1 et 0 dans le cas contraire;

H_2. la deuxième correspond à l'hypothèse sac-de-mots présentée qui stipule que les termes présents dans un document sont mutuellement indépendants;

H_3. la dernière hypothèse spécifie que tous les termes non présents dans la requête sont uniformément répartis dans les documents pertinents et non-pertinents par rapport à cette dernière.

Le modèle BIR (2)

Avec ces hypothèses :

$$
\begin{aligned}
RSV(q,d) &= \log \frac{P(R=r|d,q)}{P(R=\bar{r}|d,q)} \\
&= \log \frac{P(d|R=r,q)}{P(d|R=\bar{r},q)} + \log \frac{P(R=r,q)}{P(R=\bar{r},q)} \\
&=_{\text{rang}} \log \frac{P(d|R=r,q)}{P(d|R=\bar{r},q)}
\end{aligned}
$$

Et on a,

$$\forall r \in \{0,1\}; P(d \mid R=r,q) = P(\mathbf{d}=(w_{1d},...,w_{Vd}) \mid R=r,\mathbf{q})$$

Le modèle BIR (3)

En posant

- $\rho_i = P(w_{id} = 1 \mid R = 1, q)$ (resp. $1 - \rho_i = P(w_{id} = 0 \mid R = 1, q)$) est la probabilité que le terme d'indice i du vocabulaire apparaisse (resp. n'apparaisse pas) dans un document pertinent par rapport à la requête q;
- $\tau_i = P(w_{id} = 1 \mid R = 0, q)$ (resp. $1 - \tau_i = P(w_{id} = 0 \mid R = 0, q)$) représente la probabilité que le terme d'indice i du vocabulaire apparaisse (resp. n'apparaisse pas) dans un document non-pertinent par rapport à la requête q.

Le modèle BIR (4)

Et comme, on a supposé que les termes du vocabulaire qui ne sont pas présents dans la requête sont uniformément répartis dans les documents pertinents et non-pertinents (hypothèse $\mathtt{H}_3$), on peut ne tenir compte, dans les produits de l'équation ci-dessus que les probabilités des termes présents dans la requête, soit:

$$RSV(q,d) = \prod_{i:w_{id}=w_{iq}=1} \frac{\rho_i}{\tau_i} \times \prod_{i:w_{iq}=1,w_{id}=0} \frac{1-\rho_i}{1-\tau_i}$$

Ce qui, en multipliant le terme de droite par $\prod_{i:w_{id}=w_{iq}=1} \frac{1-\rho_i}{1-\tau_i} \times \frac{1-\tau_i}{1-\rho_i} = 1$, donne:

$$RSV(q,d) = \prod_{i:w_{id}=w_{iq}=1} \frac{\rho_i(1-\tau_i)}{\tau_i(1-\rho_i)} \times \prod_{i:w_{iq}=1} \frac{1-\rho_i}{1-\tau_i}$$

Le modèle BIR (5) : Estimation des paramètres

On considère une collection de documents $\mathcal{C} = \{d_1, ..., d_N\}$ et un ensemble de requêtes $\mathcal{Q} = \{q_1, ..., q_{|\mathcal{Q}|}\}$ donnés, où pour chaque couple $(d, q) \in \mathcal{C} \times \mathcal{Q}$ on dispose un jugement de pertinence binaire $R_{d,q}$ (Section **??**). On suppose de plus que chaque document $d \in \mathcal{C}$ est représenté par un vecteur *binaire* de dimension V, $\mathbf{d} = (w_{1d}, ..., w_{Vd})$. Posons $\rho_i = P(w_i = 1 \mid R = 1, q)$ (respectivement $\tau_i = P(w_i = 1 \mid R = 0, q)$) la probabilité que le terme d'indice i du vocabulaire apparaisse dans un document pertinent (respectivement non-pertinent) par rapport à la requête q.

- Pour une requête fixe q, quelles sont les lois de probabilités suivies par le terme d'indice i du vocabulaire, si ce dernier apparaît dans un document $d \in \mathcal{D}$, pertinent ou non-pertinent par rapport à cette requête?
- Soit d (respectivement d') un document jugé pertinent (respectivement non-pertinent) pour une requête de $\mathcal{Q}$. Montrer que $\forall w_{id} \in \mathbf{d}$, $P(w_{id} \mid \rho_i) = \rho_i^{w_{id}}(1-\rho_i)^{1-w_{id}}$ et $\forall w_{id} \in \mathbf{d}'$, $P(w_{id'} \mid \tau_i) = \tau_i^{w_{id'}}(1-\tau_i)^{1-w_{id'}}$
- On note $\mathcal{R}$ (respectivement $\bar{\mathcal{R}}$) le sous-ensemble des documents de $\mathcal{C}$ jugés, pertinent au-moins une fois (respectivement jamais jugé pertinent), par rapport à une requête de $\mathcal{Q}$ (i.e. $\mathcal{C} = \mathcal{R} \cup \bar{\mathcal{R}}$). On suppose de plus que les termes apparaissant dans n'importe quel document de $\mathcal{C}$ sont indépendants les uns des autres. Donner les expressions de $P(\mathbf{d} \mid \rho)$ pour $d \in \mathcal{R}$ et $P(\mathbf{d}' \mid \tau)$ pour $d' \in \bar{\mathcal{R}}$.

Le modèle BIR (5) : Estimation (suite)

- Il existe différentes méthodes statistiques pour estimer les paramètres $\boldsymbol{\rho} = (\rho_1, ..., \rho_V)$ et $\boldsymbol{\tau} = (\tau_1, ..., \tau_V)$; parmi lesquelles la méthode du *maximum de vraisemblance* (*MV*) qui est la plus utilisée dans la littérature. Nous allons estimer $\boldsymbol{\rho}$ et $\boldsymbol{\tau}$ respectivement sur les sous-ensembles $\mathcal{R}$ et $\bar{\mathcal{R}}$. Pour une collection de documents $\mathcal{X} = \{d_1, ..., d_{|\mathcal{X}|}\}$ ($\mathcal{X}$ étant $\mathcal{R}$ ou $\bar{\mathcal{R}}$), la méthode du *MV* consiste à trouver l'ensemble des paramètres $\boldsymbol{\lambda}^{MV}$ ($\boldsymbol{\rho}^{MV}$ ou $\boldsymbol{\tau}^{MV}$) qui maximise la vraisemblance des données $P(\mathcal{X} \mid \boldsymbol{\lambda})$. Dans le cas où on suppose que les documents sont tous indépendamment distribués, donner l'expression de $P(\mathcal{X} \mid \boldsymbol{\lambda})$.
- Soit le tableau de contingence suivant, comptabilisant la présence et l'absence du terme d'indice *i* du vocabulaire dans les sous-ensembles $\mathcal{R}$ et $\bar{\mathcal{R}}$

	$\mathcal{R}$	$\bar{\mathcal{R}}$	Total
Terme t_i présent, $w_i = 1$	n_{i1}	$\mathrm{df}_{t_i} - n_{i1}$	df_{t_i}
Terme t_i absent, $w_i = 0$	$\lvert\mathcal{R}\rvert - n_{i1}$	$(N - \mathrm{df}_{t_i}) - (\lvert\mathcal{R}\rvert - n_{i1})$	$N - \mathrm{df}_{t_i}$
Total	$\lvert\mathcal{R}\rvert$	$N - \lvert\mathcal{R}\rvert$	N

Montrer alors que

$$\forall i \in \{1, .., V\}, \rho_i^{MV} = \frac{n_{i1}}{|\mathcal{R}|}, \tau_i^{MV} = \frac{\mathrm{df}_{t_i} - n_{i1}}{N - |\mathcal{R}|}$$

Le modèle BIR (6)

Avantages et désavantages

+ Une notion claire et théoriquement fondée du degré de pertinence

+ Le processus de recherche d'information est un processus itératif qui implique l'utilisateur

- Sensibilité aux valeurs initiales

- Procédure d'estimation certes intéressante mais coutêuse

Complexité similaire au modèle vectoriel à chaque itération ; un peu plus complexe en général

Le modèle vectoriel: score cosinus

Algorithm 1 Moteur de recherche textes avec le score cosinus

Une collection $\mathcal{C}$ de N documents et son index inversé associé
Un tableau contenant les facteurs de normalisation $(Norm_d[j])_{j\in\{1,\ldots,N\}}$
$q = \{w_1^q, w_2^q, ..., w_K^q\}$: une requête constituée de K termes.
Initialiser le tableau $\forall j \in \{1, ..., N\}, s[j] \leftarrow 0$
Initialiser le facteur de normalisation associé à la requête $Norm_q \leftarrow 0$.
for $i \in \{1, ..., K\}$ **do**
    $Norm_q \leftarrow Norm_q + (t^q_{w_i^q})^2$
    for all j dans la liste des identifiants de w_i^q **do**
        $s[j] \leftarrow s[j] + t^q_{w_i^q} \times t^{d_j}_{w_i^q}$
    end for
end for
for $j \in \{1, ..., N\}$ **do**
    if $s[j] \neq 0$ **then**
        $s[j] \leftarrow s[j] / \left(\sqrt{Norm_q} \times \sqrt{Norm_d[j]}\right)$
    end if
end for

Modèle BM25

- Le modèle OKAPI BM25 (ou simplement BM25) est devenu une référence dans le développement des systèmes de recherche,
- L'idée de départ de modèle : un bon descripteur de document est un terme assez fréquent de ce document mais qui est relativement rare dans la collection,
- Cette idée est fondée sur le constat que beaucoup de termes apparaissent avec une fréquence assez basse dans beaucoup de documents d'une collection, alors qu'ils apparaissent avec une fréquence élevée dans un groupe distinct de documents (groupe élite) $\rightarrow$ BIM.

Modèle BM25

- Ce constat a motivé la modélisation du groupe élite avec la loi de Poisson de paramètre λ ainsi que la modélisation du groupe non élite avec la loi de Poisson de paramètre μ avec $\mu < \lambda$.
- La probabilité d'apparition d'un terme w apparaissant x_w^d fois dans un document d peut être exprimée par une loi de mélange de paramètre α_w et β_w suivant que ce document a été jugé pertinent ou non par rapport à une requête q :

$$P(x_w^d \mid R = 1, q) = \alpha_w \frac{\lambda^{x_w^d} e^{-\lambda}}{x_w^d!} + (1 - \alpha_w) \frac{\mu^{x_w^d} e^{-\mu}}{x_w^d!}$$

$$P(x_w^d \mid R = 0, q) = \beta_w \frac{\lambda^{x_w^d} e^{-\lambda}}{x_w^d!} + (1 - \beta_w) \frac{\mu^{x_w^d} e^{-\mu}}{x_w^d!}$$

Modèle BM25

- La fonction de score de BM25 est inspirée de celle du modèle BIM à la différence que les probabilités de présence et d'absence des termes dans les documents pertinents et non pertinents sont calculées d'après les lois 2-Poisson précédentes soit

$$s(q,d) = \sum_{w \in q \cap d} \ln \delta_w$$

avec, $\forall t$:

$$\delta_w = \frac{(\alpha_w \lambda^{x_w^d} e^{-\lambda} + (1-\alpha_w)\mu^{x_w^d} e^{-\mu})(\beta_w e^{-\lambda} + (1-\beta_w)e^{-\mu})}{(\beta_w \lambda^{x_w^d} e^{-\lambda} + (1-\beta_w)\mu^{x_w^d} e^{-\mu})(\alpha_w e^{-\lambda} + (1-\alpha_w)e^{-\mu})}$$

Modèle BM25

- Avec les conditions $\forall w, \alpha_w > \beta_w$ et $\mu < \lambda$ on peut montrer que

$$\forall w, \lim_{x_w^d \to +\infty} \ln \delta_w = \ln \left(\frac{\alpha_w}{\beta_w} \frac{1-\beta_w}{1-\alpha_w} \right)$$

- Dans une collection de N documents, le choix des paramètres des mélanges s'opère généralement d'une part en considérant, sans connaissance préalable sur les paramètres α_w et d'autres part, pour n'importe quelle requête, la majorité des documents d'une collection donnée sont non-pertinents, d'où :

$$\forall w, \alpha_w = 0.5, \beta_w = \frac{N_w + 0.5}{N}$$

Modèle BM25

- Dans ce cas, nous avons

$$\forall w, \ln \delta_w = \ln \frac{N - N_w + 0.5}{N_W + 0.5}$$

- (Robertson et Walker 1994) ont introduit modifications au calcul de la fonction de score :
  1. La prise en compte du nombre d'occurrences normalisé des termes de la requête dans les documents,
  2. La prise en compte du nombre d'occurrences normalisé des termes de la requête q dans la requête elle-même.

Implémentation du score cosinus avec deux modèles simples

- Modèle BM25

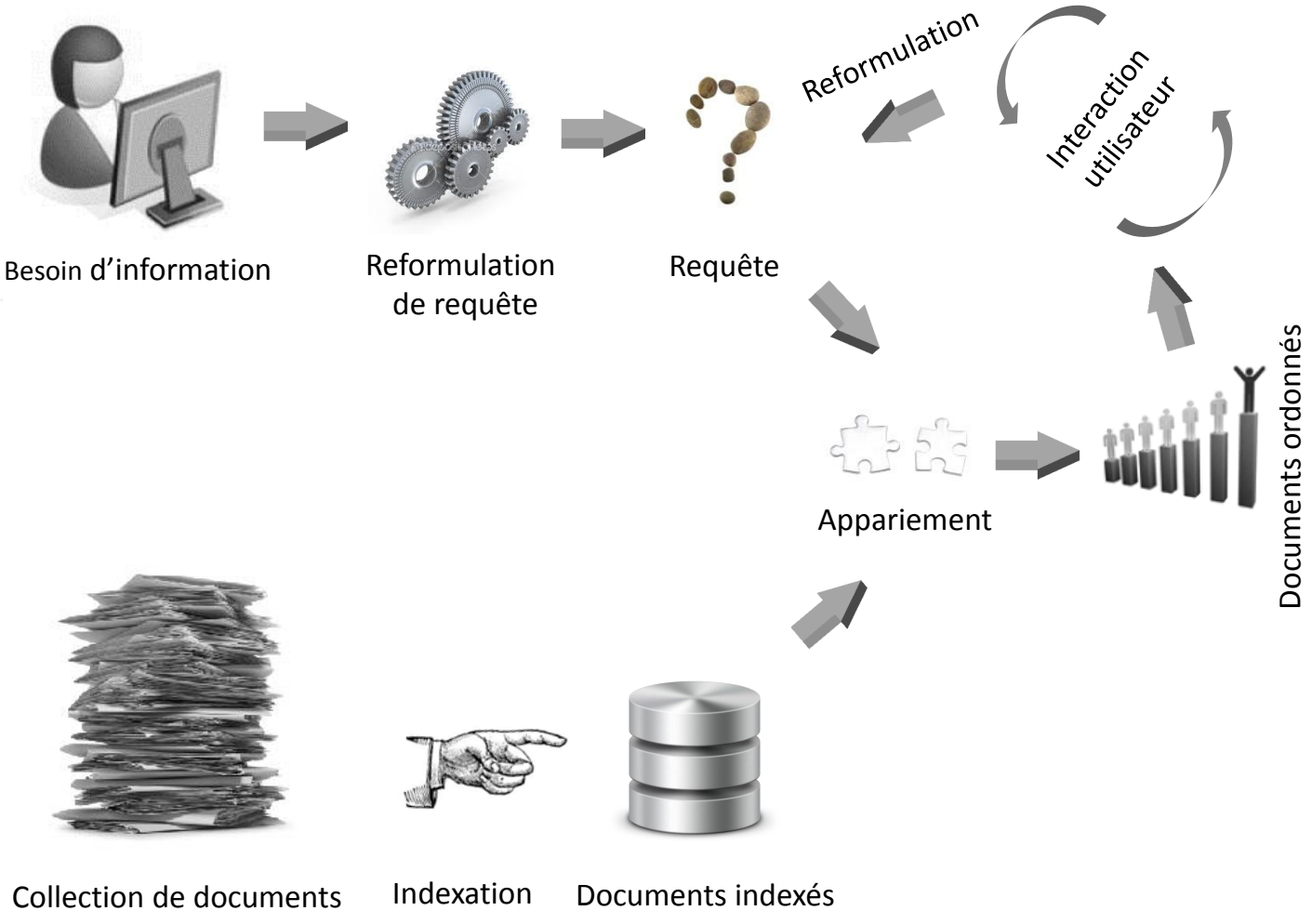
$$RSV(q,d) = \sum_{w \in d \cap q} \underbrace{\frac{(k_1+1)x_w^d}{k_1((1-b)+b\frac{y_d}{m})+x_w^d} \times \ln\left(\frac{N-N_w+0.5}{N_w+0.5}\right)}_{=t_w^d} \times \underbrace{\frac{(k_3+1)x_w^q}{k_3+x_w^q}}_{=t_w^q}$$

$$k_1 \in [1;2], b = 0.75, k_3 \in [0;1000]$$

- Modèle $tf - idf$

$$RSV(q,d) = \sum_{w \in d \cap q} \underbrace{x_w^d \times idf_w}_{=t_w^d} \times \underbrace{x_w^q}_{=t_w^q}$$

Expansion de requêtes



Algorithme de Rocchio

- Dans un premier temps, on cherche, à partir des ensembles de documents jugés *pertinents*, $\mathcal{D}_p$, et *non-pertinents*, $\mathcal{D}_{np}$, la requête unitaire $q^\star$ qui maximise la différence de score entre les ensembles *pertinents* et *non-pertinents*:

$$q^\star = \underset{q}{\text{argmax}} \left(s(q, \mathcal{D}_p) - s(q, \mathcal{D}_{np}) \right), \ \ s.t. ||q^\star|| = 1$$

- Dans le cas où la mesure de similarité utilisée est la fonction *cosinus*, on peut montrer que cette condition se traduit par la relation vectorielle suivante :

$$\mathbf{q}^\star \propto \frac{1}{||\mathcal{D}_p||} \sum_{d \in \mathcal{D}_p} \mathbf{d} - \frac{1}{||\mathcal{D}_{np}||} \sum_{d' \in \mathcal{D}_{np}} \mathbf{d}'$$

où $||\mathcal{D}_p|| = || \sum_{d \in \mathcal{D}_p} ||$ (la relation pour $||\mathcal{D}_{np}||$ étant similaire).

- La formule de Rocchio consiste alors à enrichir la requête initiale q_0 avec les termes de la requête $q^\star$, cet enrichissement étant contrôlé par des poids qui peuvent être réglés automatiquement (ou manuellement) sur de nouvelles collections:

$$\mathbf{q}^{\text{new}} = \alpha \mathbf{q}_0 + \beta \frac{1}{||\mathcal{D}_p||} \sum_{d \in \mathcal{D}_p} \mathbf{d} - \gamma \frac{1}{||\mathcal{D}_{np}||} \sum_{d' \in \mathcal{D}_{np}} \mathbf{d}'$$

avec α, β et γ des réels positifs ou nuls.

Les jugements/annotations les plus fréquents

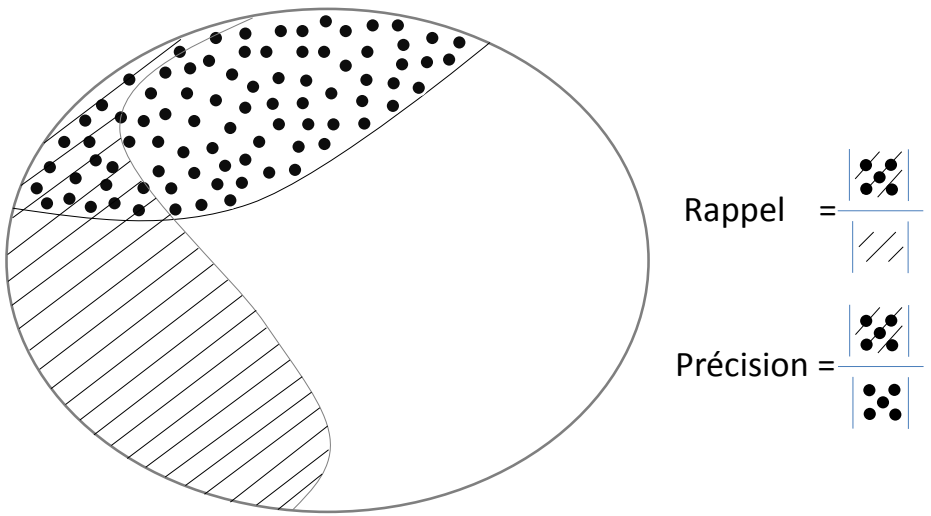
- Jugements binaires : ce document est pertinent (1) ou non (0) pour cette requête
- Jugements multi-valués :
  $Parfait > Excellent > Bon > Correct > Mauvais$
- Paires de préférence : document d_A plus pertinent que document d_B pour cette requête

Mesures d'évaluations, jugements binaires

Les deux mesures d'évaluation les plus utilisées en RI sont le *rappel* et la *précision*:

$$\text{Rappel} = \frac{\text{Nbre de documents pertinents retournés par le système}}{\text{Nbre de documents pertinents}}$$

$$\text{Precision} = \frac{\text{Nbre de documents pertinents retournés par le système}}{\text{Nbre de documents retournés}}$$

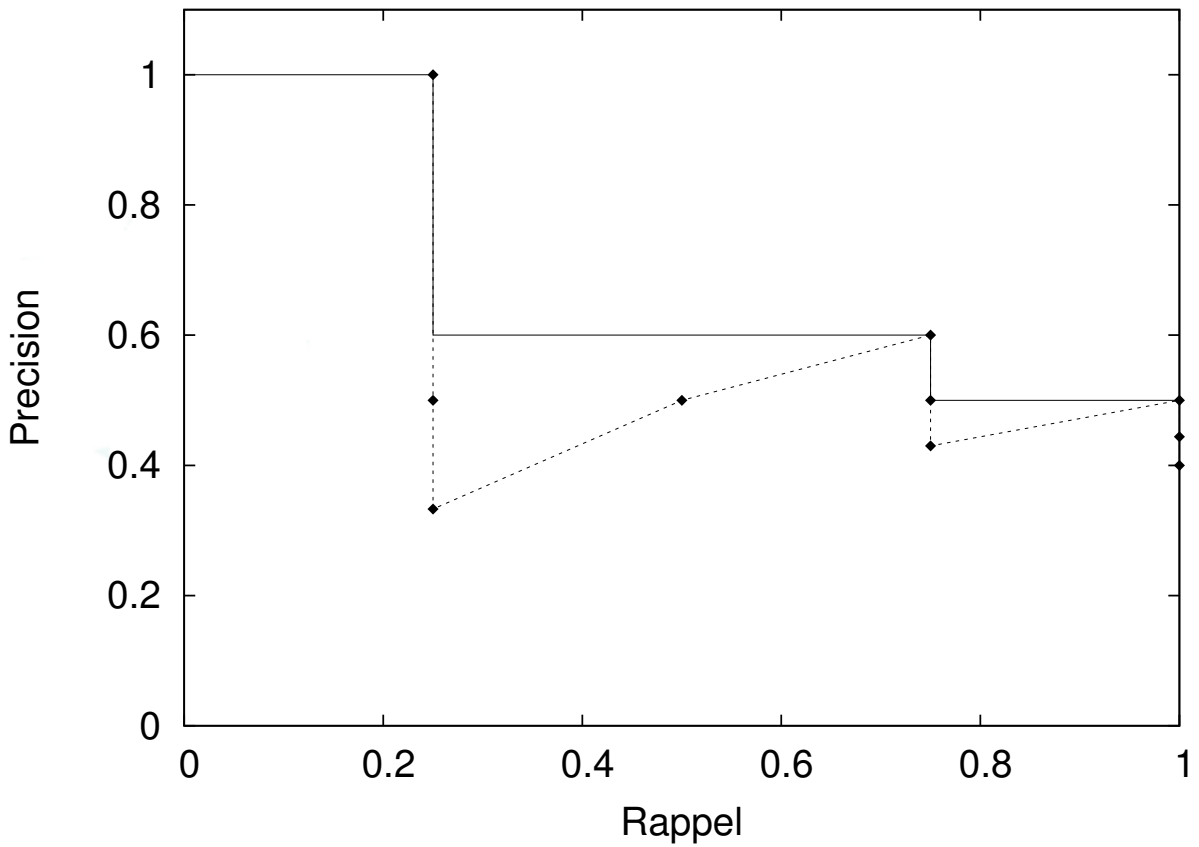


Courbes précision/rappel

rg	Réponse de $\mathcal{M}$	$R_{d_{rg},q}$	rappel, r	précision, $\mathtt{P}(r)$
1	0.95	1	$1/4$	1
2	0.82	0	$1/4$	$1/2$
3	0.75	0	$1/4$	$1/3$
4	0.7	1	$1/2$	$1/2$
5	0.65	1	$3/4$	$3/5$
6	0.5	0	$3/4$	$1/2$
7	0.4	0	$3/4$	$3/7$
8	0.35	1	1	$1/2$
9	0.2	0	1	$4/9$
10	0.1	0	1	$2/5$

Table : Mesures de *rappel* et de *précision* sur un ensemble de 10 documents ordonnés d'après la réponse d'un moteur de recherche $\mathcal{M}$ pour une requête fictive q.

Courbes Précision/Rappel



Lissage avec

$$\forall \rho \in [0,1], \Pi(\rho) = \max\{\mathbb{P}(r) \mid r \geq \rho\}$$

D'autres mesures classiques

- La précision moyenne(*Average Precision* en anglais) d'un système de recherche $\mathcal{M}$ pour une requête q donnée, notée souvent *AveP*, est la moyenne des valeurs de précision des documents pertinents par rapport à q dans la liste ordonnée des réponses:

$$AveP(q) = \frac{1}{n_+^q} \sum_{k=1}^{N} R_{d_k,q} \times \mathtt{P@}k(q)$$

où $n_+^q = \sum_{i=1}^{N} R_{d_i,q}$ est le nombre total de documents pertinents par rapport à q.

- Mean Average Precision

$$MAP = \frac{1}{|Q|} \sum_{j=1}^{|Q|} AveP(q_j) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{n_+^{q_j}} \sum_{k=1}^{N} R_{d_k,q_j} \times \mathtt{P@}k(q_j)$$