Les systèmes de recherche d'information

Un Système de Recherche d'Information (SRI)est un système qui permet de retrouver les documents pertinents à une requête d'utilisateur, à partir d'une base de documents volumineuse. Le but de la RI est de mettre en correspondance des informations disponibles d'une part, et les requêtes (besoins) de l'utilisateur d'autre part. Ce processus cherche alors à mettre en relation des besoins utilisateurs et des informations, afin que le Système de Recherche d'Information (SRI), ne retourne à l'utilisateur le maximum de documents pertinents par rapport à son besoin (et le minimum de documents non-pertinents).

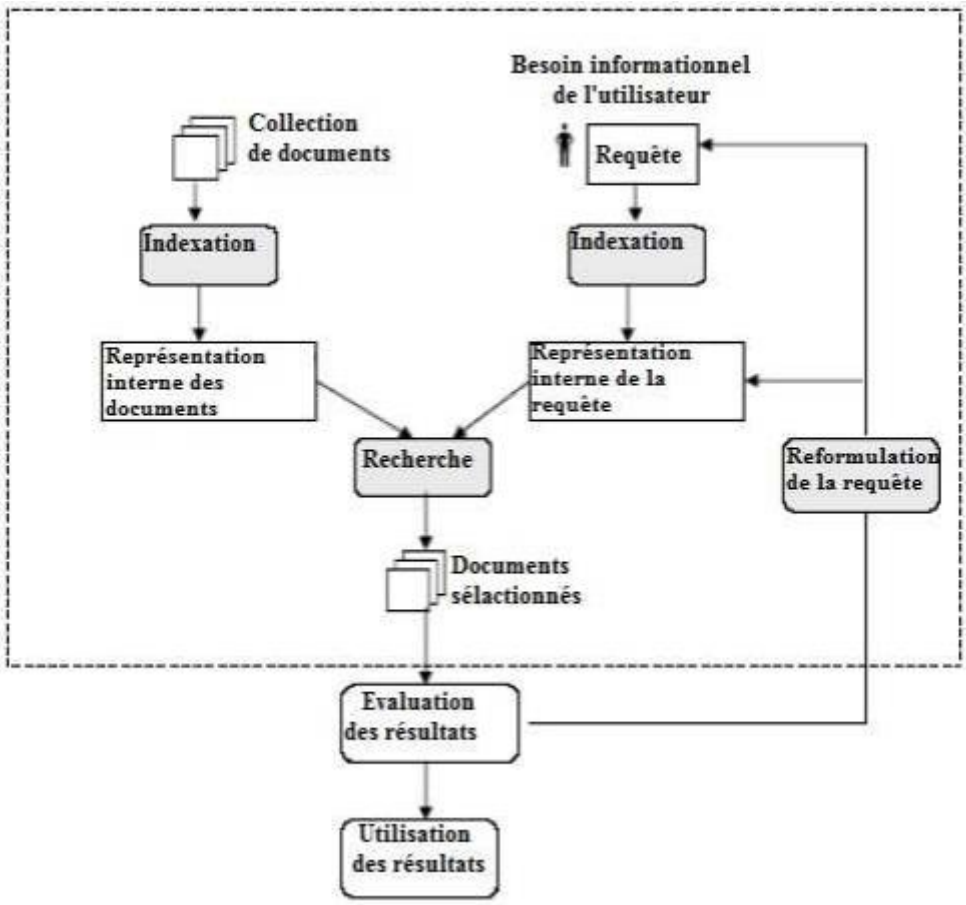


Figure 1. Processus en U de la recherche d'information

Le processus d'indexation

Indexer un document est une façon de rendre des informations accessibles et facilement exploitables par le système lors du processus de la recherche en analysant chaque document de la collection afin de créer un ensemble de mots-clés. L'indexation permet donc de réduire le cout de la recherche en créant une représentation des documents dans le système.

L'indexation suit plusieurs étapes :
- L'analyse lexicale, - L'élimination des mots vides, - La lemmatisation, - La pondération des termes.

1.L’analyse lexicale

Le but de l’analyse lexicale est de convertir le texte d’un document en une suite de termes. Un terme est une unité lexicale ou un radical. Ce processus permet de reconnaitre les espaces de séparation des mots, des chiffres, les ponctuations, etc.

2. L’élimination des mots vides

Cette étape a pour objectif de supprimer les termes non significatifs (pronoms personnels, prépositions,...) ou mots athématiques (les mots qui peuvent se retrouver dans n’importe quel document parce qu’ils exposent le sujet mais ne le traitent pas, comme par exemple contenir, appartenir, etc.).

Deux techniques pour éliminer les mots vides :

- L’utilisation d’une liste de mots vides (aussi appelée anti-dictionnaire),
- L’élimination des mots dépassant un certain nombre d’occurrences dans la collection.

L’élimination des mots vides permet donc de réduire le nombre de termes d’indexation, mais aussi peut réduire le taux de rappel, c'est-à-dire le rapport de documents pertinents renvoyés par le système sur l’ensemble des documents pertinents, qu’on va détailler plus tard.

3. La lemmatisation ou La troncature

Souvent on trouve des mots dans un texte qui ne sont pas dans leur forme canonique. Ces différentes formes peuvent avoir le même sens ou des sens très similaires et par suite, on n’a pas besoin d’indexer tous ces mots. Il suffit alors d’indexer les racines, et donc substituer les termes par leur lemme.

La lemmatisation permet alors d’indexer la forme canonique du terme qui regroupe les différentes variables du mot et ses dérives. Par exemple le lemme de « cheval » et celui de « chevaux » sont les mêmes. Cette forme est l'infinitif pour les verbes, la forme masculine singulière pour les noms, etc. Cela permet alors à l’utilisateur d’éviter de devoir entrer les formes plurielles des noms ou les formes conjuguées des verbes dans sa requête.

D’autre part, la lemmatisation peut, dans quelques cas, supprimer le sens original du mot. Par exemple le mot « fils » possède deux lemmes très différents, « fil » et « fils ». Ainsi une requête avec le mot « fils » va retourner certainement des documents non pertinents, ce qui diminue le taux de précision (le rapport de document pertinents sur le nombre de documents total renvoyés par le système).

4. La pondération des termes

Le poids d’un terme indique l’importance du terme dans la caractérisation d’un document. L’objectif est toujours de trouver les termes représentant le mieux le contenu du document. Ainsi, une bonne formule de pondération est celle qui assure à la fois un rappel et une précision élevés. La plupart des techniques de pondération sont basées sur les facteurs « TF » et « IDF » qui permettent de combiner les pondérations locale et globale d’un terme :

- TF (Term Frequency), est la fréquence d’occurrence d’un terme dans un document. Ainsi un terme présent fréquemment dans un document est considère important, et son poids doit être élevé. (Pondération locale).

- IDF (Inverse Document Frequency), est l'importance d'un terme dans toute la collection (Pondération globale). Un terme ayant une fréquence élevée mais qui n'est pas concentrée dans un nombre limité de document mais plutôt dans toute la collection, ne doit pas avoir le même impact ou le même poids qu'un terme moins fréquent.

IDF = Log (N / df)

N = nombre total de documents dans la collection,
df = nombre de documents contenant le terme.

La mesure TF*IDF donne une bonne approximation de l'importance du terme dans le document.

L'appariement document-requête

Le but de tout SRI est de retourner à l'utilisateur le plus grand nombre possible de documents pertinents. La pertinence est basée sur une fonction d'appariement (matching) qui effectue une comparaison entre les représentants des documents et ceux des requêtes construits lors de la phase d'indexation. La pertinence du document vis-à-vis de la requête est représentée par un calcul de score.

Ce score est calculé à partir d'une fonction ou d'une probabilité de similarité notée RSV(Q,d) (Retrieval Status Value), ou Q est une requête et d un document. Cette mesure tient compte du poids des termes dans les documents, détermine en fonction d'analyses statistiques et probabilistes. La fonction de similarité permet d'ordonner les documents renvoyés à l'utilisateur. Cet ordonnancement joue un rôle primordial puisque l'utilisateur se contente la plupart du temps d'examiner les premiers documents affiches

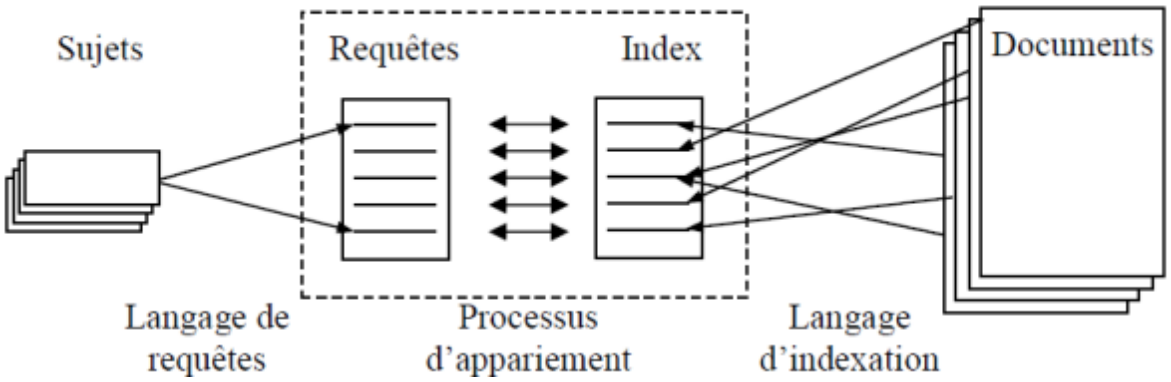


Figure 2. Appariement document-requête

Les différents modèles de la Recherche d'Information

Le modèle de RI a pour rôle de déterminer le comportement clé d'un Système de Recherche d'Information. Il fournit donc un cadre théorique pour la modélisation de la mesure de pertinence.

1 Le modèle booléen

Le modèle booléen a été le premier à être utiliser dans le monde de la RI. C'est le plus simple des modèles de RI, et permet de faire une recherche très restrictive et obtenir, pour un utilisateur expérimenté, une information exacte et spécifique. Il est basé sur l'algèbre de Boole et considère ainsi une requête comme une expression logique. Il considère aussi que les termes de l'index sont soit présents soit absents d'un document. En conséquence, les poids des termes dans l'index sont binaires c'est-à-dire wij= {0,1}.

Dans ce modèle :

Un document d est représenté comme une conjonction logique des termes non pondérés.

Un document d est représenté comme une conjonction logique des termes non pondérés.

Exemple: d = $t_1 \wedge t_2 \wedge t_3 \wedge \ldots \wedge t_n$.

Une requête q est composée de termes lies par 3 connecteurs logiques ET, OU, NON.

Exemple :

q = ($t_1 \wedge t_2$) $\vee$ ($t_3 \wedge \neg t_4$). où q : la requête, t_i: terme d'indexation.

La similarité entre un document et une requête est définie par :

$$RSV(q, d) = \begin{cases} 1 & Si\ d\ appartient\ a\ l'ensemble\ decrit\ par\ la\ requête \\ 0 & Sinon \end{cases}$$

Ainsi, le modèle booléen affirme que chaque document est soit pertinent soit non pertinent, donc répond exactement à la requête qui a été formulée. Il n'y a pas de notion de réponse partielle aux conditions de la requête.

D'autre part, il est souvent très difficile pour l'utilisateur d'exprimer son besoin en information avec des expressions booléennes ce qui ne permet pas d'utiliser au mieux les caractéristiques de ce modèle.

Enfin, tous les termes dans un document ou dans une requête sont pondérés de la même façon simple (0 ou 1), donc tous les termes ont la même importance dans le document. Cela ne correspond pas à ce qu'on souhaite avoir en RI.

Très peu de systèmes de nos jours utilisent le modèle booléen standard, et c'est plutôt une extension de ce modèle qui est implémentée. Il est connu qu'une pondération non binaire des termes de l'index peut amener à des améliorations notables des performances. La pondération de ces termes nous amène donc à introduire le modèle vectoriel.

2. Le modèle vectoriel

Le modèle vectoriel standard a été proposé par Salton dans le système de recherche documentaire SMART (Salton's Magical Automatic Retriever of Text. Le modèle vectoriel représente les documents et les requêtes par des vecteurs dans un espace à n dimensions, les dimensions étant constituées par les termes du vocabulaire d'indexation. L'espace vectoriel est défini par l'ensemble de termes que le système a rencontré durant l'indexation.

Soit l'espace vectoriel suivant <t_1 , t_2 , … , t_n>. Chaque document et requête sont respectivement représentés par un vecteur document et un vecteur requête comme suit:

$d_j = (w_{1j}, w_{2j}, \dots, w_{nj})$ w_{ij} = poids du terme t_i dans le document d_j.

$q = (w_{1q}, w_{2q}, \dots, w_{nq})$ w_{iq} = poids du terme t_i dans la requête q.

La figure 1.3 montre un exemple d'espace vectoriel compose des trois termes t_1, t_2, t_3. Les index de deux documents d_1 et d_2 et une requête sont représentés dans cet espace.

Etant donné ces deux vecteurs, la pertinence est traduite en une similarité vectorielle : un document est d'autant plus pertinent à une requête que le vecteur associé est similaire à celui de la requête, c'est-à-dire autant que la distance et l'angle entre les vecteurs documents et requêtes sont petits

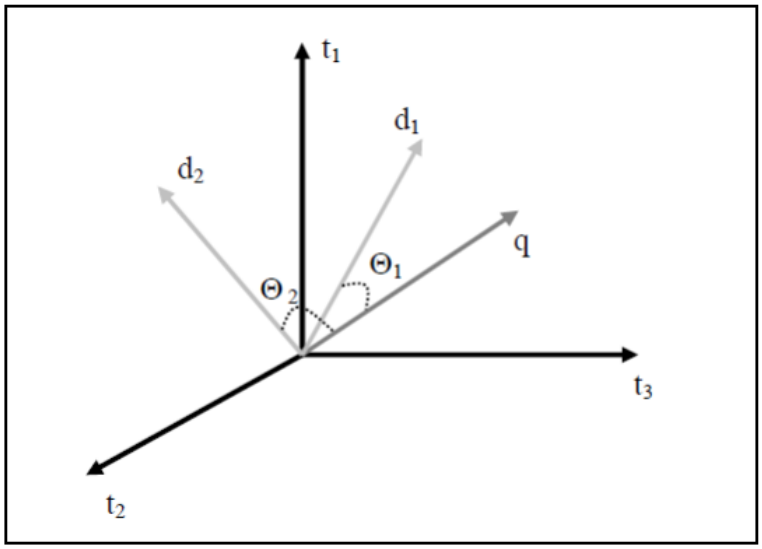


Figure 3. Représentation vectorielle de deux documents et d'une requête (q) dans un espace compose de trois termes

Plusieurs formules sont utilisées pour calculer la similarité entre ces deux vecteurs :

- Produit interne (Inner product) : $sim\left(\vec{d_j}, \vec{q}\right) = \sum x_i * y_i$

- Coefficient de Dice : $sim\left(\vec{d_j}, \vec{q}\right) = \frac{2 * \sum x_i * y_i}{\sum x_i^2 + \sum y_i^2}$

- Mesure du Cosinus : $sim\left(\vec{d_j}, \vec{q}\right) = \frac{\sum x_i * y_i}{\sqrt{\sum x_i^2 + \sum y_i^2}}$

- Mesure de Jaccard : $sim\left(\vec{d_j}, \vec{q}\right) = \frac{2 * \sum x_i * y_i}{\sum x_i^2 + \sum y_i^2 - \sum x_i * y_i}$

Un tel modèle présente de nombreux avantages : La pondération des termes augmente les performances du système et améliore les résultats de recherche. La mesure de similarité ou la fonction d'appariement permet d'ordonner et de trier les documents selon leur pertinence vis-

à-vis de la requête.
D'autre part, le modèle vectoriel présente l'inconvénient de considérer que les termes de l'index sont indépendants. Cependant, malgré sa simplicité, les résultats d'une telle approche peuvent être compares à de nombreuses nouvelles méthodes d'ordonnancement, ce qui fait de ce modèle le plus populaire en Recherche d'Information.

3. Le modèle probabiliste

Ce modèle est fonde sur le calcul de la probabilité de pertinence d'un document pour une requête. On se rapproche ici de la notion de classification probabiliste. L'idée est de retrouver des documents qui ont en même temps une forte probabilité d'être pertinents, et une faible probabilité d'être non pertinents.
En effet, le RI est un processus incertain et imprécis. L'incertitude se présente dans la représentation des informations tandis que l'imprécision se manifeste dans l'expression des besoins. Ce modèle tend alors à estimer la probabilité qu'un document donné soit pertinent pour une requête donnée, donc de mesurer cette incertitude et cette imprécision. Pour ce faire, le processus de décision complète le procédé d'indexation probabiliste en utilisant deux probabilités conditionnelles :

- P (wji/pert) : Probabilité que le terme ti apparait dans le document Dj sachant que ce dernier est pertinent pour la requête.
- P (wji/Nonpert) : Probabilité que le terme ti apparait dans le document Dj sachant que ce dernier n'est pas pertinent pour la requête.

Si on suppose l'indépendance des variables documents « pertinents » et « nonpertinents », la fonction de recherche peut être obtenue en utilisant la formule de Bayes.

Soit $d_j(t_1, t_2, \dots, t_n)$ où $t_i = \begin{cases} 1 \ Si \ t_i \ \ indexe \ le \ document \ d_j \\ 0 \ Sinon \end{cases}$

$$P(\ pert / d_j) = \frac{p\,(d_j/pert) * P\,(pert)}{p\,(d_j)}$$

$$P\,(nonpert / d_j) = \frac{p\,(d_j/nonpert) * P\,(nonpert)}{p\,(d_j)}$$

Avec : P(pert/dj) est la probabilité de pertinence du document d_j sachant sa description.

$P(d_j) = P(d_j \ /pert) * P(pert) + P(d_j \ /nonpert) * P(nonpert)$

$P(d_j \ /pert)$ (respectivement $P(d_j \ /nonpert)$) est la probabilité d'observer le document d_j sachant qu'il est pertinent (respectivement non pertinent).

Si l'on considère l'indépendance des termes :

$P(pert/\ d_j) = P(t_1/pert) * P(t_1) \dots P(t_N/pert) * P(t_N)$

$P(Nonpert/\ dj) = P(t_1/nonpert) * P(t_1) \dots P(t_N/nonpert) * P(t_N)$

Où $P\,(t_1/pert\,) = \frac{r_i}{R}$ et $P\,(t_1/pert\,) = \frac{m_i - r_i}{M - R}$

M : nombre total de documents dans la collection.

R : nombre de documents pertinents pour une requête.

r_i : nombre de documents pertinents dans lesquels le terme t_i apparait.

m_i: nombre total de documents dans lesquels le terme t_i apparait.

Les modèles probabilistes retournent de bons résultats par rapport aux modèles booléens et sont indépendants du domaine d'application. Cependant, ils présentent un obstacle majeur dans les méthodes d'estimation des probabilités utilisées pour évaluer la pertinence.

Évaluation des systèmes de recherches d'information

Un protocole regroupe la description des conditions et des étapes de déroulement d'une expérience, dans notre cas l'évaluation des systèmes de recherche d'information. La description doit être suffisamment claire afin que l'expérience puisse être reproduite et il doit faire l'objet d'une analyse critique pour détecter les limites afin d'évoluer les systèmes.

Corpus de test (Collection de tests)

Pour évaluer le degré de pertinence des réponses d'un système de recherche d'information à une requête, il est nécessaire de connaître l'ensemble des documents pertinents pour une requête donnée. C'est à cette fin que des collections de tests ont été élaborées. Une collection de tests comprend : des corpus de documents, une liste de requêtes prédéfinies et des jugements de pertinence.

Corpus de documents :

C'est un ensemble de documents à indexer, il s'agit de l'ensemble des informations accessibles et exploitables sur lesquelles le système sera évalué. Un groupe d'experts dans un domaine participent généralement à la constitution d'un corpus documentaire homogène et cohérent couvrant le domaine. Dans le cas général et pour un souci d'optimalité, la base documentaire constitue des représentations simplifiées mais suffisantes pour ces documents.

Requête :

C'est une représentation d'un besoin d'information. Il est souhaitable que le protocole d'évaluation prend en compte une requête directement dans la forme où elle a été soumise au système, et non un besoin ou une question exprimée sous forme libre et détaillée (éventuellement beaucoup plus riche d'informations).

Jugements de pertinence:

Ils sont manuellement établis et constituent la liste des documents pertinents pour chaque requête. Ils peuvent être portés par l'utilisateur lui-même, ou par un observateur expert 'extérieur' qui se concentrera plus facilement sur la « pertinence ». Ils peuvent prendre des formes booléennes variées (le document est pertinent ou il ne l'est pas) ou plus nuancées, généralement par l'usage de pondérations.

Mesures d'évaluation

Précision :

La précision est le rapport de documents pertinents trouves sur l'ensemble des documents restitués par le système. Elle mesure la capacité du système à rejeter tous les documents non pertinents à une requête donnée.

$$précision = \frac{nombre\ de\ documents\ pertinents\ trouvés}{nombre\ de\ documents\ trouves}$$

Rappel :
Le rappel est le rapport de documents pertinents restitués par le système sur l'ensemble des documents pertinents contenus dans la base documentaire. Elle mesure la capacité du système à retrouver tous les documents pertinents répondant à une requête.

$$rappel = \frac{nombre\ de\ documents\ pertinents\ trouvés}{nombre\ de\ documents\ pertinents}$$

F – mesure :
Plusieurs indicateurs de synthèse ont été créés à partir de deux mesures de Rappel et de la Précision, mais le plus célèbre est la F-mesure. Cette mesure correspond à une moyenne harmonique de la précision et du rappel. Cette moyenne diminue lorsque l'un de ses paramètres est petit et augmente lorsque les deux paramètres sont proches tout en étant élevés

$$F\text{- }mesure = \frac{(1+\beta^2)\ précision * rappel}{(\beta^2 * précision) + rappel}$$

Le paramètre β permet de pondérer la précision ou le rappel, il est égal généralement à la valeur 1. Pour effectuer ces mesures, il faut disposer des réponses idéales aux requêtes en question.

L'évaluation des performances d'un SRI, peut être effectuée encore en mesurant différents paramètres et critères, tels que :
Précision interpolée (Courbe Rappel/Précision) :
La performance d'un SRI peut être représentée par une courbe Rappel/Précision.
Lorsque les valeurs exactes de rappel ne peuvent pas être atteintes, Il est fréquent d'employer une interpolation sur ces courbes, qui consiste à lisser la courbe initiale pour qu'elle soit décroissante. La valeur interpolée de la précision pour un point de rappel i est la précision maximale obtenue pour un point supérieur ou égal à $i\ P(R_i) = \max_{R_i \leq R} P(R)$. L'avantage est de définir la précision sur des valeurs standardisées.

Le calcul de la précision et du rappel s'effectue pour chaque élément de la liste des documents retrouvés par le SRI. Pour évaluer la performance globale d'un système, il est utile de disposer d'une mesure unique qui réunie en une seule grandeur la performance du SRI. La précision moyenne interpolée IAP (Interpolated Average Precision) est une mesure décrivant la précision globale du système évalué sur une requête. Elle consiste à calculer la précision des résultats sur onze points de rappel qui vaut 0, 10, 20, 30, 40, 50, 60, 70, 80, 90 et 100%. Si ces points ne sont pas atteints, les mesures sont alors interpolées. La moyenne de ces 11 précisions forme la précision moyenne interpolée.
La courbe rappel-précision calcule, pour une requête, la précision obtenue pour un nombre donné de documents retrouvés par le système. Ce nombre est fixé pour chaque requête en fonction du nombre de documents pertinents dans la collection. La courbe rappel-précision est intéressante lorsque la collection comporte un nombre considérable de documents pertinents.
Soit une requête *Q*, et soit *N* = (d3, d8, d12, d11, d1, d210) l'ensemble des documents

que l'on sait pertinents pour la requête *Q*. Soit *S* un SRI qui retourne les documents du tableau 1 en réponse à la requête *Q*.

[1]	d12*	[6]	d5	[11]	d88	[16]	d31
[2]	d17	[7]	d33	[12]	d77	[17]	d72
[3]	d1*	[8]	d11*	[13]	d4	[18]	d23
[4]	d15	[9]	d3*	[14]	d210*	[19]	d2
[5]	d8*	[10]	d18	[15]	d7	[20]	d13

Tableau 1. Liste des documents restitués par un SRI pour la requête Q

Dans le tableau 1, les documents sont ordonnés par pertinence décroissante. Les chiffres entre crochets représentent le rang du document dans la liste restituée. Les documents suivis du symbole "*" correspondent aux documents pertinents restitués par le système. Le premier document de la liste (document d12) est pertinent. On en déduit que d12 correspond un taux de rappel de 0.167 puisque seulement un document pertinent sur 6 a été retrouvé à ce moment.

Lorsque le système *S* retrouve le document d12 au rang 1, il obtient une précision de 100% au taux de rappel de 16.7%. Les listes de documents restitués par les SRI sont en généraux ordonnés par degré de pertinence système. Il est alors possible d'examiner les listes en partant du document ayant obtenu le meilleur score. À chaque nouveau document analysé, on calcule alors le taux de rappel réel correspondant (R).

En reprenant l'exemple présenté dans le tableau 1, le prochain document pertinent que *S* restitue (après le document d12) est le document d17 et il se situe au rang 2. Le système obtient donc un taux de précision de 50% (1 document pertinent sur 2 documents retournés) au taux de rappel de 16.7% (1 documents pertinent retrouvé sur l'ensemble des 6 documents pertinents pour la requête). Les valeurs de rappel et précision ainsi calculées sont représentées à l'aide d'une courbe rappel/précision aux 11 points de rappel standards. La figure 4, illustre cette courbe.

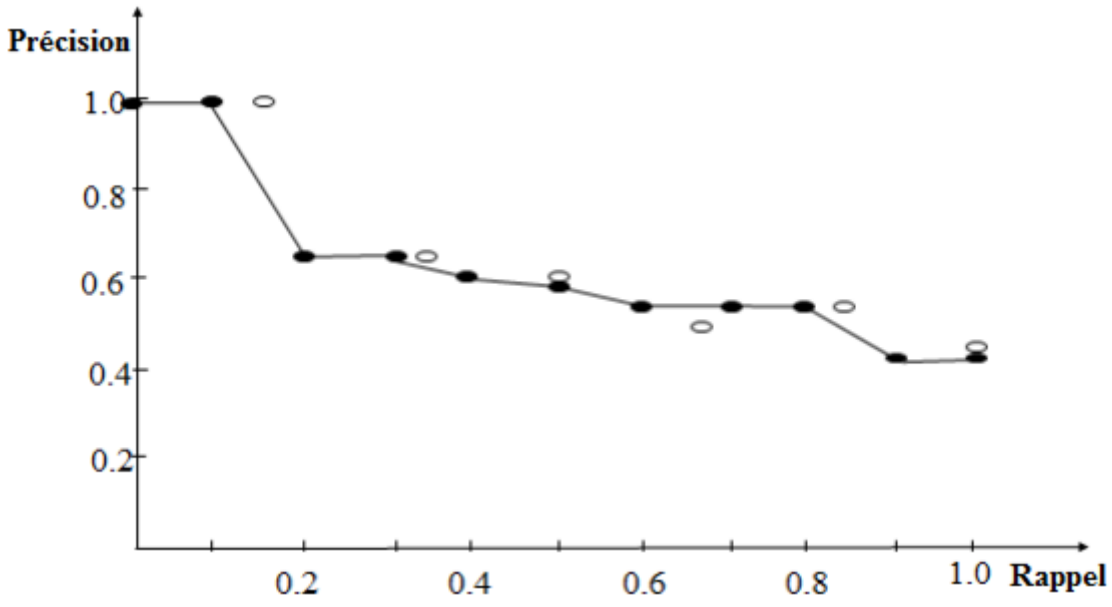


Figure 4. Précision aux 11 points standards de rappel.

PageRank

Quelques moteurs de recherche, dont le plus connu est Google, ont pris le pari d'utiliser un autre mode de classement des résultats. Les pages Web sont ordonnées selon leur popularité, une page qui est la cible d'un très grand nombre de liens est probablement non seulement une page validée (page parcourue par un grand nombre de lecteurs, qui ont jugé bon de la citer en référence) mais aussi une page détenant un contenu utile à un grand nombre d'utilisateurs. L'approche du PageRank qui a fait la spécificité du moteur de recherche Google, repose sur la notion de propagation de popularité. Le principe consiste à évaluer l'importance d'une page en fonction de chacune des pages pointant vers elle. La propagation met en avant les pages qui jouent un rôle particulier dans le graphe des liens, avec l'hypothèse suivante : *"une page est importante quand elle est beaucoup citée ou citée par une page très importante*". La mesure de PageRank (PR)] est une distribution de probabilité sur les pages. Elle mesure en effet la probabilité PR, pour un utilisateur navigant au hasard, d'atteindre une page donnée. Elle repose sur un concept très simple : un lien émis par une page A vers une page B est assimilé à un vote de A pour B. Plus une page reçoit de votes, plus cette page est considérée comme importante. Le PageRank se calcule de la façon suivante :

- Soient $T1$, $T2$, ..., Tn : n pages citant une page A. Notons $PR(Tk)$ le PageRank de la page Tk, $S(Tk)$ le nombre de liens sortants de la page Tk, et d'un facteur compris entre 0 et 1, fixé en général à 0.85. Ce facteur d représente la probabilité de suivre effectivement les liens pour atteindre la page A, tandis que (1-d) représente la probabilité d'atteindre la page A sans suivre de liens. Le PageRank de la page A se calcule à partir du PageRank de toutes les pages Tk de la manière suivante :

$$PR(A) = (1 - d) + d \quad \frac{PR(T)}{S(T)}$$

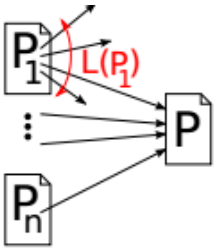
Initialement, toutes les pages sont équiprobables, leur valeur de PR est alors égale à 1/n, n Étant le nombre de documents de la collection.

Ex :

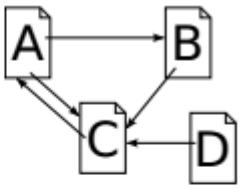
Plus précisément[3], le « PageRank » PR d'une page P ayant n autres pages P_1, ..., P_n qui la citent, est défini par :

$$\mathrm{PR}(P) = 0.15 + 0.85 \left(\frac{\mathrm{PR}(P_1)}{L(P_1)} + \ldots + \frac{\mathrm{PR}(P_n)}{L(P_n)} \right)$$

où $L(x)$ est le nombre de liens sortant de la page x vers d'autres pages.



Par exemple, si :



— la page A cite les pages B et C,
— la page B cite la page C,
— la page C cite la page A,
— la page D cite la page C,

alors on a $L(A) = 2$ et $L(B) = L(C) = L(D) = 1$ (nombre de liens sortant) et :

$$\begin{aligned}
\mathrm{PR}(A) &= 0.15 + 0.85\,\mathrm{PR}(C) \\
\mathrm{PR}(B) &= 0.15 + 0.85\,\frac{\mathrm{PR}(A)}{2} \\
\mathrm{PR}(C) &= 0.15 + 0.85\left(\frac{\mathrm{PR}(A)}{2} + \mathrm{PR}(B) + \mathrm{PR}(D)\right) \\
\mathrm{PR}(D) &= 0.15 + 0.85 \times 0 = 0.15
\end{aligned}$$

Sur un exemple aussi simple, on peut très facilement résoudre les équations ; mais dans la réalité d Web le nombre de variables est bien trop grand ! PageRank calcule donc ces scores de façon approchée
et itérative. Il fonctionne de la façon suivante :
— au départ, les notes PageRank de toutes les pages sont égales à l'inverse du nombre total de pages (0.25 dans l'exemple ci-dessus),
— ensuite on recalcule la note de chaque page en utilisant les notes précédentes,
— et on itère comme cela sans arrêt (les notes sont constamment recalculées, le Web évoluant tout le temps).
Pour l'exemple précédent cela donne :

	étape 1	étape 2	étape 3	étape 4	...		...
$\mathrm{PR}(A)$	0.25	0.36250	0.72906	0.70197	...	1.49011	...
$\mathrm{PR}(B)$	0.25	0.25625	0.30406	0.45985	...	0.78330	...
$\mathrm{PR}(C)$	0.25	0.68125	0.64937	0.84580	...	1.57660	...
$\mathrm{PR}(D)$	0.25	0.15	0.15	0.15	...	0.15	...

Crawler

Un **robot d'indexation** (***web spider***) est un logiciel qui explore automatiquement le Web. Il est généralement conçu pour collecter les ressources (pages Web, images, vidéos, documents Word, PDF, etc.), afin de permettre à un moteur de recherche de les indexer.

En français, depuis 2013, *crawler* est remplaçable par le mot *collecteur*. Il existe aussi des collecteurs analysant finement les contenus afin de ne ramener qu'une partie de leur information.

Pour indexer de nouvelles ressources, un robot procède en suivant récursivement les hyperliens trouvés à partir d'une page pivot. Par la suite, il est avantageux de mémoriser l'URL de chaque ressource récupérée et d'adapter la fréquence des visites à la fréquence observée de mise à jour de la ressource. Toutefois, si le robot respecte les règles du fichier robots.txt, alors de nombreuses ressources échappent à cette exploration récursive. Cet ensemble de ressources inexploré est appelé Web profond ou Web invisible.

Un fichier d'exclusion (`robots.txt`) placé dans la racine d'un site Web permet de donner aux robots une liste de ressources à ignorer. Cette convention permet de réduire la charge du serveur Web et d'éviter des ressources sans intérêt. Toutefois, certains robots ne se préoccupent pas de ce fichier.