UNIVERSITÉ du SUD
Toulon-Var

Master Sciences de l'information et des systèmes

Spécialité Ingénierie des systèmes d'information

Parcours Sécurité des systèmes d'information

Année universitaire 2007-2008

M1 – UE C2 – Projet

Recherche de similarités de documents décrits par des mots-clefs.

Grégory ANNE, Yoann JANSZEN, Benoît THEVENET

(responsable : Marie-Christine GONTARD)

Sujet :

Recherche de similarités de documents décrits par des mots-clefs.

Ce TER se composera de l'étude d'un modèle dans le domaine de la recherche d'information et de son implantation, puis d'une comparaison de différentes approches classiques.

Remerciements :

Nous tenons à remercier Mme Marie-Christine Gontard, pour nous avoir encadré sur ce TER, pour ses conseils et sa disponibilité. Nous remercions aussi Mrs Jacques Le Maitre et Hervé Glotin qui nous ont également orienté dans notre travail.

. Sommaire

1 La Recherche d'Information

Face à la quantité croissante des données sur les réseaux internes et mondiaux (intranets, internet, etc.) la question de l'accès à l'information est un des plus grands enjeux d'actualité mais aussi un des plus délicats. Dans ce contexte, il est nécessaire de pouvoir accéder au contenu des documents par des moyens rapides et efficaces.

La **Recherche d'Information** (RI) ou recherche documentaire, propose de retrouver parmi une masse volumineuse de documents textuels (corpus textuel, page internet...) , ceux qui correspondent au besoin informationnel d'un utilisateur généralement formulé par une requête en langage naturel. Les premiers modèles de RI ont établi la notion de triplet (document, besoin, correspondance). Il existe un grand nombre de **modèles de recherche d'information**, et ces modèles diffèrent principalement sur la façon dont les informations disponibles sont représentées, et sur la façon d'interroger la base.

Pour cela, la plupart des modèles demandent à l'utilisateur d'exprimer son besoin en utilisant le langage de *requête* du modèle. À partir de la requête, la *fonction de correspondance* du modèle extrait de la base les informations qui sont susceptibles de répondre au besoin. Habituellement, cette fonction n'utilise pas les informations de la base, mais les *indexations*, qui sont des représentations des informations, dont le but est d'améliorer les performances de la fonction de correspondance (temps et qualité des résultats). Ainsi, le principal problème en recherche d'information concerne le choix de la représentation des informations, car de ce choix dépend la fonction de correspondance, et donc la qualité des résultats et la satisfaction des utilisateurs.

1.1 Architecture d'un Système de Recherche d'Information (SRI)

Le but d'un SRI est de présenter à l'utilisateur des documents répondant à ses besoins. Selon [**Baeza-Yates** & **Ribeiro-Neto**, 1999], de manière globale, un SRI se modélise par le quadruplet SRI=<D,Q,M,P> (voir la figure 1.1) où :

– D est l'ensemble des documents du corpus ;
– Q est un langage de requête destiné à représenter les besoins d'information de l'utilisateur. Ce langage définit l'ensemble des requêtes que peut formuler directement ou indirectement un utilisateur d'un SRI ;
– M est un modèle de RI qui sert à décrire les documents de D et à exprimer les requêtes de Q.
– P est une fonction qui associe une valeur de pertinence à toute requête qi de Q et tout document d de D. Cette fonction peut fournir un ordonnancement des documents par rapport à la requête qi

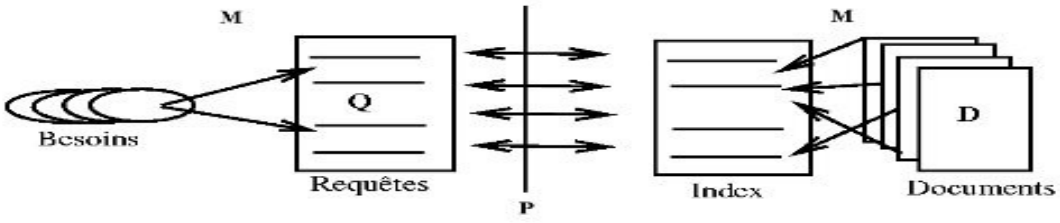


FIG. 1.1 – Exemple d'architecture d'un SRI

Le SRI se décompose essentiellement en deux processus de base :

– Le **processus d'indexation** qui consiste à identifier dans un document certains éléments significatifs qui serviront de clés pour retrouver ce document au sein d'une collection. Il produit une représentation des documents (l'index). Pour faciliter le processus d'interrogation, le même processus d'indexation s'applique généralement aux requêtes.

– Le **processus d'interrogation** (ou de recherche) qui consiste à comparer les représentations des requêtes avec celles des documents.

Cette architecture peut être enrichie par un retour arrière sur pertinence (relevance feedback) qui affine la recherche et améliore la qualité des résultats en tenant compte de l'évaluation de l'utilisateur qui classe les documents en pertinents et non pertinents. Il est aussi possible d'avoir recours à l'expansion de requêtes qui permet d'étendre la requête (en rajoutant des termes) ou la réécrire.
Les processus d'indexation et de recherche sont dépendants l'un de l'autre. En effet, la phase d'indexation est une phase importante qui a un impact direct sur la recherche. Si un document est mal indexé, il risque de ne plus être retrouvé et donc perdu. La norme AFNOR NF Z 47-102 1996, définit l'indexation de la manière suivante :

> **L'indexation** est l'opération qui consiste à décrire et à caractériser un document à l'aide de représentations des concepts contenus dans ce document, c'est-à-dire à transcrire en langage documentaire les concepts après les avoir extraits du document par une analyse.

1.2 Les Modèles de recherche d'informations

Lors de notre recherche documentaire nous avons remarqué qu'il y avait différents modèles de recherche d'information, nous allons présenter ces modèles et nous nous interesserons plus particulièrement aux modèles suivants: le modèle booléen pur, le modèle booléen étendu basé sur la théorie des ensembles flous et le modèle vectoriel. En effet , nous avons décider d'implanter ces trois modèles pour notre moteur de recherche d'information.

Si c'est l'indexation qui choisit les termes pour représenter le contenu d'un document ou d'une requête, c'est au modèle de leur donner une interprétation. Étant donné un ensemble de termes pondérés issus de l'indexation, le modèle remplit les deux rôles suivants:
- créer une représentation interne pour un document ou pour une requête basée sur ces termes;
- définir une méthode de comparaison entre une représentation de document et une représentation de requête afin de déterminer leur degré de correspondance (ou similarité).

Le modèle joue un rôle central dans la RI. C'est le modèle qui détermine le comportement clé d'un système de RI. Dans ce paragraphe, nous allons décrire quelques modèles souvent utilisés dans la RI.

1.2.1 Modèle Matching score

Le modèle « matching score » a été l'un des premiers modèles utilisé dans la RI. La représentation d'un document se fera par un ensemble de termes pondérés par leur fréquence. Une ***requête*** sera aussi un ensemble de termes, pondérés à 1. Le degré de correspondance entre un document et une requête est la somme des fréquences 'f_i' des termes 't_i'de la requête 'q' dans le document 'd':

$$R(d, q) = \Sigma i\ f_i$$

Le résultat calculé est appelé le "**matching score**". Il s'agit simplement de parcourir le document est de regarder combien de fois les termes de la requête apparaissent. Plus le matching score calculé est élevé, plus le document correspond à la requête, et ainsi il sera classé plus haut dans la réponse. C'est un modèle assez primitif car il utilise directement le résultat de l'indexation sans aucune réorganisation ou modélisation.

1.2.2 Modèle booléen

Ce modèle (1960) est le plus simple des modèles de RI, il repose sur l'algèbre de Boole. Un document est représenté par une conjonction de termes (non pondérés) :

$$d = t1 \wedge t2 \wedge ... \wedge tn$$

Une requête sera formée d'une expression logique de termes en utilisant les opérateurs AND ($\wedge$), OR ($\vee$) et NOT ($\neg$),
par exemple :

$$q = (t1 \wedge t2) \vee (t3 \wedge \neg t4)$$

Pour que le document corresponde à la requête, il faut que l'implication **d → q** soit valide. La correspondance R(d, q) entre le terme et la requête est déterminée par:

R(d, ti) = 1 si ti ∈ d; 0 sinon.
R(d, q1 ∧ q2) = 1 si R(d, q1) = 1 et R(d, q2) = 1; 0 sinon.
R(d, q1 ∨ q2) = 1 si R(d, q1) = 1 ou R(d, q2) = 1; 0 sinon.
R(d, ¬q1) = 1 si R(d, q1) = 0; 0 sinon.

Les documents retournés par le système sont considérés à pertinence égale. La **conjonction** est très contraignante (un document qui contient quelques uns des termes recherchés est rejeté au même titre qu'un document qui ne contient aucun terme) et la **disjonction** très permissive (les documents contenant seulement un terme sont aussi bons qu'un document qui satisfait tous les termes). Les termes dans le document ou dans la requête ont une pondération binaire (1 si présent et 0 si absent), il n'est pas possible d'exprimer qu'un terme est plus important qu'un autre.

De plus, la formulation booléenne des requêtes complexes n'est pas évidente pour des utilisateurs non expérimentés.

Toutes ces raisons font que le modèle booléen standard est rarement utilisé de nos jours, les extensions proposées pour corriger ses lacunes seront plutôt privilégiées.

1.2.3 Modèle Booléen basé sur des ensembles flous

Voici donc une extension au modèle booléen standard qui vise à tenir compte de la **pondération** des termes dans les documents. Du côté requête, elle reste toujours une expression booléenne classique.
Avec cette extension, un **document** est représenté comme un ensemble de termes pondérés comme suit:

d = {..., (ti, ai), ...}

L'évaluation d'une **requête** peut prendre plusieurs formes. Une d'elles est la suivante:

R(d, ti) = ai
R(d, q1 ∧ q2) = min(R(d, q1), R(d, q2)).
R(d, q1 ∨ q2) = max(R(d, q1), R(d, q2)).
R(d, ¬q1) = 1 - R(d, q1).

Dans cette évaluation, les opérateurs logiques ∧ et ∨ sont évalués par **min** et **max** respectivement.
C'est cette représentation de **norme** et **co-norme** que nous allons utiliser dans notre implantation du modèle booléen étendu.
C'est une des évaluations classiques proposées par **L. Zadeh** dans le cadre des ensembles flous. Cependant, cette évaluation n'est pas parfaite. Par exemple, on n'a pas R(d, q∧¬q) ≡0 et R(d,q∨¬q) ≡1. Du point de vue théorique, c'est gênant. Du point de vue pratique, quand on évalue une requête en forme de conjonction, on ne s'intéresse qu'à la partie la plus difficile, et quand on évalue une requête en forme de disjonction, c'est la partie la plus facile qui domine.
Intuitivement, on aimerait plutôt que les deux parties jouent toutes les deux un rôle dans l'évaluation. Ainsi, beaucoup d'autres formes d'évaluation ont été proposées. Une des formes est l'évaluation de **Lukasiewicz** qui est la suivante:

R(d, ti) = ai
R(d, q1 ∧ q2) = R(d, q1) * R(d, q2).
R(d, q1 ∨ q2) = R(d, q1) + R(d, q2) - R(d, q1) * R(d, q2).
R(d, ¬q1) = 1 - R(d, q1).

Dans cette évaluation, on voit que les deux parties d'une conjonction ou d'une disjonction contribuent en même temps, contrairement à celle de Zadeh. Cependant, elle a le même problème qui est R(d, q∧¬q) ≠0 et R(d, q∨¬q) ≠1. En plus, R(d, q∧q) ≠R(d, q) ≠R(d, q∨q).
Si on compare ces extensions avec le modèle standard, il est assez facile de voir les avantages.
Le plus important est qu'on peut mesurer le degré de correspondance entre un document et une requête dans [0, 1]. Ainsi, on peut ordonner les documents dans l'ordre décroissant de leur correspondance avec la requête. L'usager peut parcourir cette liste ordonnée et décider où s'arrêter. Au niveau de la représentation, on a également une représentation plus raffinée. On peut exprimer dans quelle mesure un terme est important dans un document.
Ces évaluations ont été proposées à la fin des années 1970 et au début des années 1980. Maintenant, ces extensions sont devenues standard: la plupart des systèmes booléens utilisent un des ces modèles étendus.

1.2.4 Modèle vectoriel

Le modèle vectoriel (1960) a été introduit par G.**Salton.** Dans ce modèle, un document, ainsi qu'une requête, sont représentés par un **vecteur de poids**. Chaque poids dans le vecteur désigne l'importance d'un terme correspondant dans ce document ou dans la requête.
Plus formellement, on considère l'espace vectoriel:

< t1, t2, t3, ..., t_i >

où les t_i sont les termes que le système a rencontré lors de l'indexation.
Un **document** dj et une **requête** q sont représentés par :

$$d_j = [w_{1,j} , ... , w_{T,j}]$$

$$q = [w_{1,q} , ... , w_{T,q}]$$

où $w_{1,j}$ et $w_{1,q}$ correspondent aux **poids du terme** ti dans d_j et q.

Le degré de correspondance entre d_j et q est déterminé par leur **coefficient de similarité**. Le classement des documents trouvés par ordre décroissant de leurs degrés de similarité permet d'avoir une estimation plus fiable de la pertinence de ces documents.

La notion de similarité est définie par le cosinus de l'angle entre deux vecteurs. Cette définition est valide pour la similarité entre deux documents ou entre une requête et un document :

$$\text{sim}(d_j, q) = \frac{\vec{d_j} \bullet \vec{q}}{|\vec{d_j}| * |\vec{q}|} = \frac{\sum_{i=1}^{T} u}{\sqrt{\sum_{i=1}^{T} w_{i,j}^2}}$$

Il reste à définir quels poids associer aux termes des documents. L'une des stratégies les plus répandues est la stratégie nommée **tf-idf** (term frequency - inverse document frequency) qui se base sur la fréquence $\Phi_{i,j}$ du terme t_i dans le document d_j et la fréquence normalisée $f_{i,j}$ calculée par :

$$f_{i,j} = \frac{\phi_{i,j}}{\max_{d_i}(\phi_{i,i})}$$

Le poids $\omega_{i,j}$ est alors calculé par :

$$w_{i,j} = f_{i,j} \times \log \frac{N}{n_i}$$

où ni est le nombre de documents dans lequel le terme t_i apparaît. La seconde partie de l'équation représente la fréquence inverse du document (inverse document frequency) du terme t_i .

Cette approche du modèle vectoriel est prise dans le système **SMART** (System for the Mechanical Analysis and Retrieval of Text) qui est un système de RI expérimental construit entre 1968 et 1970 (la première version). Dans les années 1980, il a été réécrit et réorganisé. Les travaux sur SMART ont été dirigés par le professeur G. Salton.

1.2.5 Modèle p-norme

Le modèle p-norme (Salton, Fox et Wu) est proposé pour résoudre certains problèmes observés dans le modèle booléen standard:

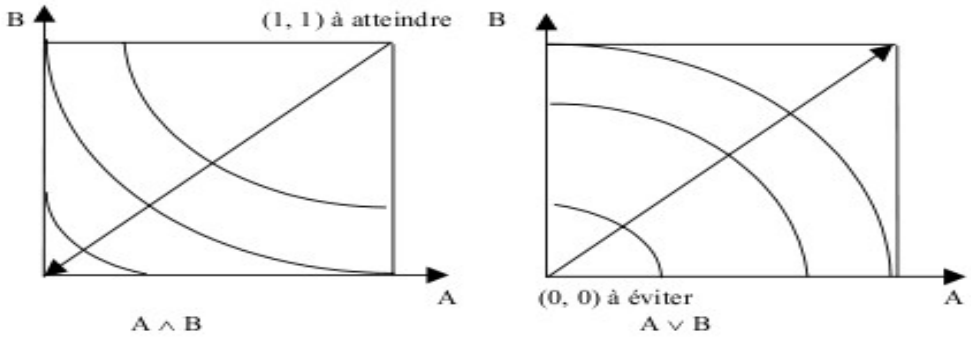
- taille de réponse non contrôlable;

- les réponses non-ordonnées;
- tous les termes ont la même importance;
- pour une requête qui est une longue conjonction, un document qui satisfait la majorité de termes est aussi mauvais qu'un document qui ne satisfait aucun terme; pour une requête qui est une longue disjonction, un document qui satisfait un terme est aussi bon qu'un document qui satisfait tous les termes;

L'approche proposée tente d'étendre le modèle booléen standard sur plusieurs aspects. D'abord, observons la table de vérité utilisée pour l'évaluation booléen standard:

A	B	A∧B	A∨B
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Dans la colonne de A ∧ B, l'objectif est d'atteindre le cas de la dernière ligne. Dans la colonne de A ∨ B, c'est plutôt la première ligne qu'il faut éviter. Ainsi, une façon de floufier (fuzzify) l'évaluation stricte consiste à calculer une sorte de distance entre le points à éviter ou à atteindre. Selon cette distance, on va déterminer l'évaluation de la conjonction ou de la disjonction. L'idée de base correspond aux figures suivantes:



Dans ces figures, étant donné une évaluation de A et de B, on détermine un point dans l'espace A-B. Dans la première figure, on cherche à évaluer dans quelle mesure ce point est proche de (1, 1) - le point à atteindre. Ce rapprochement peut être mesuré par le complément de la distance entre le point et le point (1, 1): plus cette distance est grande, moins A ∧ B est satisfaite à ce point. Pour les points qui se situent sur une même courbe, ils ont la même distance avec (1, 1).
Dans le cas de A ∨ B, on cherche plutôt à éviter le point (0, 0). Plus on est loin de (1, 0), plus A ∨ B est satisfaite.
Basée sur cette intuition, l'évaluation suivante est proposée. On admet la pondération de termes dans les documents: pi est le poids de ti dans d.

$$R(d, t_i) = p_i$$
$$R(d, q1 \wedge q2) = 1 - ([(1-R(d, q1))^2 + (1-R(d, q2))^2] / 2)^{1/2}.$$
$$R(d, q1 \vee q2) = [(R(d, q1)^2 + R(d, q2)^2) / 2]^{1/2}.$$
$$R(d, \neg q1) = 1 - R(d, q1).$$

nous avons aussi la formule avec p:

$$sim(q_{OR}, d_j) = (x_1^p + x_2^p + ... + x_m^p)^{1/p} / m$$
$$sim(q_{AND}, d_j) = 1 - ((1-x_1)^p + (1-x_2)^p + ... + (1-x_m)^p)^{1/p} / m$$

Plus généralement, dans le cas où p=1, on généralise de modèle vectoriel, et lorsque p= infini, on généralise le modèle booléeen pur.
Le modèle p-norme est intéressant non pas pour sa performance en pratique (bien que les expérimentations montrent qu'il est meilleur que le modèle vectoriel et le modèle booléen séparé), mais pour son cadre unificateur. Cela nous aide à comprendre la différence entre le modèle vectoriel et le modèle booléen: un modèle vectoriel peut être considéré comme un modèle booléen dans lequel la différence entre la conjonction et la disjonction est effacée; ou bien la relation entre deux termes dans un vecteur est une relation et-ou neutre. C'est la seule relation qu'on peut exprimer dans le modèle vectoriel.

2 Notion de pertinence

La pertinence s'avère être la notion centrale de la Recherche d'Information car toutes les évaluations s'articulent autour d'elle. Il semble toutefois que cette notion soit peu connue, malgré de nombreuses études à ce sujet. Distinguons ainsi diverses définitions de la notion en question.

La pertinence est:

- la correspondance entre un document et une requête, une mesure d'informativité du document à la requête;
- un degré de relation (chevauchement, relativité, …) entre le document et la requête;
- un degré de la surprise qu'apporte un document, qui a un rapport avec le besoin de l'utilisateur;
- une mesure d'utilité du document pour l'utilisateur;
- …

Même dans ces définitions, les termes utilisés (informativité, relativité, surprise, …) restent très vagues. Pourquoi arrive-t-on à cette situation ? Parce que les utilisateurs d'un système de RI ont des besoins très variés ainsi que des critères très différents pour juger si un document est pertinent ou pas. Donc, la notion de pertinence est utilisée pour recouvrir un très vaste éventail des critères et des relations.

3 Travail effectué

3.1 Mise en place du corpus

Avant d'implanter ou de tester un quelconque moteur de recherche d'information, il est nécessaire de constituer un ensemble de documents (corpus documentaire). Celui-ci se constitue essentiellement de textes récupérés sur le Web dont le sujet sera unique, donc dont les champs lexicaux seront proches voire identiques ; car les requêtes qui seront effectuées devront avoir plusieurs réponses dans le dit corpus. Nous avons trouvé et organisé un corpus sur le thème du nucléaire.

3.2 Premier objectif , les scripts

Notre premier objectif avant l'implantation était de formater tous les documents récupérés pour nos tests. En effet , nous avons crée un script (qui sera intégré dans le programme) qui permet de placer chaque document de notre corpus sur une ligne d'un même fichier. Par exemple, notre corpus est composé de 30 documents texte, donc le fichier de sortie sera composé de 30 lignes.
Sur ce fichier , un autre script enlèvera les mots vides car ils n'ont aucune importance dans le calcul de similarité lors de la recherche d'information.

3.3 Deuxième objectif, l'implantation des modèles

Notre TER est axé plus particulièrement sur l'étude du modèle booléen étendu basé sur la théorie des ensembles flous.

Comme nous l'avons précisé dans les paragraphes précédents, nous avons décidé d'implanter les trois modèles, booléen pur, booléen étendu et le modèle vectoriel.
Nous avons choisi cette approche dans le but d'effectuer différents tests de recherche d'information par mots-clés suivant les différents modèles. Ainsi, suivant les résultats obtenus, nous pourrons comparer ces trois modèles et donc nous pourrons situer le modèle booléen étendu par rapport aux deux autres modèles implantés. De plus, nous pourrons dire, d'après nos tests, quel modèle nous semble le plus efficace suivant les besoins de l'utilisateur.

3.3.1 Evaluation des différents systèmes

Le but de la RI est de trouver des documents pertinents à une requête, et donc utiles pour l'utilisateur. La qualité d'un système doit être mesurée en comparant les réponses du système avec les réponses idéales que l'utilisateur espère recevoir. Plus les réponses du système correspondent à celles que l'utilisateur espère, meilleur est le système.

Pour la construction d'un corpus de test, les jugements de pertinence constituent la tâche la plus difficile, puisqu'elle n'est évaluée qu'à la lecture des documents par l'utilisateur.

3.3.2 Evaluation Précision-Rappel

Après avoir crée notre corpus de documents nous avons établi un ensemble de requêtes dans le but de tester nos différents modèles. Ensuite nous avons examiné chaque document du corpus et jugé s'il était pertinent en fonction de la requête afin d'obtenir une liste des documents idéaux. En exécutant le programme, constitué de trois modes d'exécution correspondants aux trois modèles, nous avons obtenu la liste des documents répondant à chacune de nos requêtes. La méthode que nous utilisons pour comparer les SRI est basée sur des calculs de précisions et de rappels, définis comme ceci:

Rappel: la capacité du système à trouver tous les documents pertinents

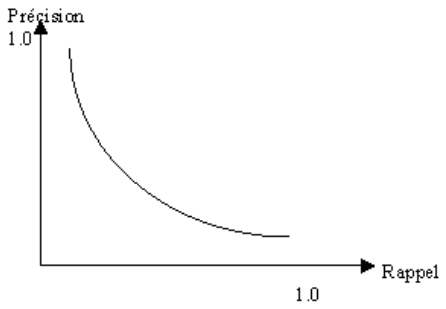
$$\text{rappel} = \frac{\#\text{documents pertinents retrouvés}}{\#\text{documents pertinents dans la base}}$$

Précision: la capacité du système a ne trouver que les documents pertinents

$$\text{précision} = \frac{\#\text{documents pertinents retrouvés}}{\#\text{documents retrouvés}}$$

Idéalement, on voudrait qu'un système donne de bons taux de précision et de rappel en même temps. En effet, un système qui aurait 100% pour la précision et pour le rappel signifie qu'il trouve tous les documents pertinents (rappel) et rien que les documents pertinents (précision). Cela signifie que les réponses du système à chaque requête sont constituées de tous et seulement les documents idéaux que identifiés.

Les deux valeurs de ce couple ne sont pas indépendantes. Il y a une forte relation entre elles: quand l'une augmente, l'autre diminue. Cela ne signifie rien de parler de la qualité d'un système en utilisant seulement une de ces valeurs. En effet, il est facile d'avoir 100% de rappel: il suffirait de donner toute la base comme la réponse à chaque requête. Cependant, la précision dans ce cas-ci serait très basse. De même, on peut augmenter la précision en donnant très peu de documents en réponse, mais le rappel sera atténué. Il faut donc utiliser les deux métriques ensemble.
Les mesures de précision-rappel ne sont pas statiques non plus (c'est-à-dire qu'un système n'a pas qu'une mesure de précision et de rappel). Le comportement d'un système peut varier en faveur de précision ou en faveur de rappel (en détriment de l'autre métrique). Ainsi, pour un système, on a une courbe de précision-rappel qui a en général la forme suivante:



Nos résultats des couples Rappel/Précision ont été utilisés pour établir de telles courbes. En effet, l'ensemble de couples obtenu pour chaque modèle et chaque requête se base sur le classement, par pertinence, des documents réponses.

- pour le modèle booléen pur, il n'y a aucun classement. C'est le concept du tout ou rien, ne peut être attribué à un document que vrai ou faux suivant la requête.
- pour le booléen étendu, des poids sont préalablement attribués à chaque mot distinct du corpus selon leur nombre d'occurrences. Le classement, par pertinence décroissante, s'opère selon ces poids et la présence des termes de la requête dans le document.
- pour le modèle vectoriel, le classement s'effectue selon les valeurs des tf/idf calculés.

Si on veut comparer deux systèmes de RI, il faut les tester avec le même corpus de test (ou plusieurs corpus de test). Un système dont la courbe dépasse (c'est-à-dire qu'elle se situe en haut à droite de) celle d'un autre est considéré comme un meilleur système.

3.3.3 Exemples de tests

Nos tests ont été effectués sur un ensemble de treize requêtes. Les résultats ont tous été répertoriés dans un tableur pour nous permettre un calcul automatique de rappel et de précision, ainsi que la génération de courbes (R/P) pour chaque requête.

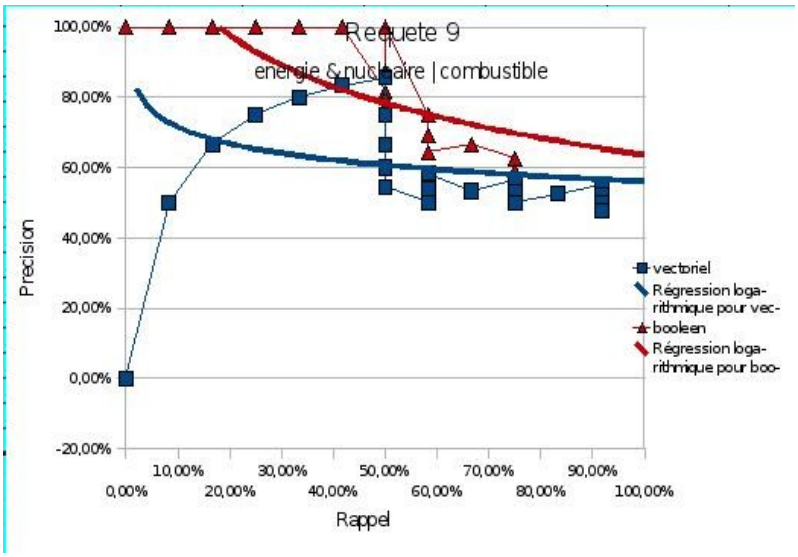
NB: les courbes des modèles booléen et vectoriel sont superposées dans le même repère. De plus, des courbes de régression logarithmique ont été ajoutées pour donner une allure statistique de répartition des couples rappel/précision pour chaque modèle.

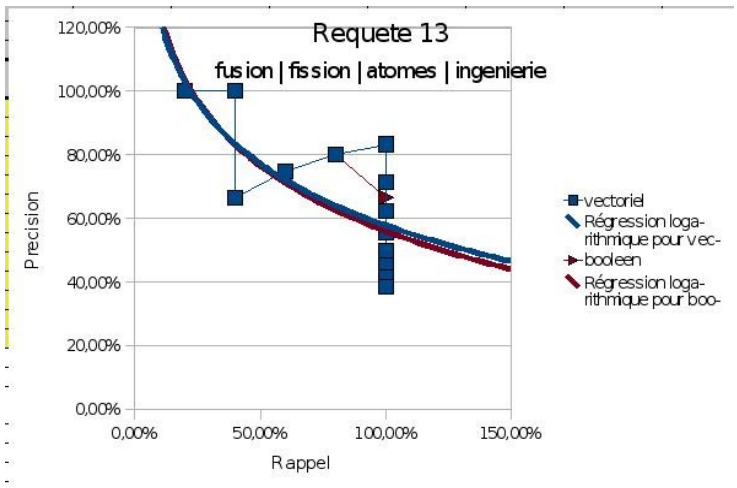
Voici une partie du tableau représentant deux des treize requêtes:

		modèle Booléen Pur	modèle Vectoriel	modèle Booléen étendu	vectoriel		booleen		*pertinents retrouvés (vectoriel)*	*docs retrouvés (dr)*	vectoriel		*pert bool*	dr	booleen	
	requetes :	*reponses systeme*			ordre	pert	ordre	pert			rappel	prec			rappel	prec
9	energie & nucleaire | combustible	1	0,0878	**0,1688**	7		4	1	0	1	0,00%	0,00%	1	1	0,00%	100,00%
		2	0,0395	0,0512	6	1	6	1	1	2	8,33%	50,00%	2	2	8,33%	100,00%
		3	0,0733	**0,1688**	18	1	18	1	2	3	16,67%	66,67%	3	3	16,67%	100,00%
		4	0,0196	**0,1688**	1	1	1	1	3	4	25,00%	75,00%	4	4	25,00%	100,00%
		5	0,0705	0,0512	19	1	3	1	4	5	33,33%	80,00%	5	5	33,33%	100,00%
		6	0,1117	**0,1688**	12	1	15	1	5	6	41,67%	83,33%	6	6	41,67%	100,00%
		7	**0,1195**	0,0512	3	1	14	1	6	7	50,00%	85,71%	7	7	50,00%	100,00%
		8	0,0620	0,0512	5		13		6	8	50,00%	75,00%	7	8	50,00%	87,50%
		10	0,0165	0,0512	8		21		6	9	50,00%	66,67%	7	9	50,00%	77,78%
		11	0,0262	0,0512	25		19	1	6	10	50,00%	60,00%	8	10	50,00%	80,00%
		12	0,0734	0,0512	13		16	1	6	11	50,00%	54,55%	9	11	50,00%	81,82%
		13	0,0461	0,0512	16	1	8		7	12	58,33%	58,33%	9	12	58,33%	75,00%
		14	0,0222	0,0512	2		7		7	13	58,33%	53,85%	9	13	58,33%	69,23%
		15	0,0266	0,0512	23		2		7	14	58,33%	50,00%	9	14	58,33%	64,29%
		16	0,0457	0,0512	9	1	12	1	8	15	66,67%	53,33%	10	15	66,67%	66,67%
		18	0,1046	**0,1688**	15	1	11		9	16	75,00%	56,25%	10	16	75,00%	62,50%
		19	0,0809	0,0512	22		5		9	17	75,00%	52,94%	10	17	75,00%	58,82%
		21	0,0167	0,0512	11		10		9	18	75,00%	50,00%	10	18	75,00%	55,56%
9bis	vectoriel : « energie nucleaire combustible »	9	0,0323	0,0000	14	1	9		10	19	83,33%	52,63%				
		17	0,0071	0,0000	4	1	23		11	20	91,67%	55,00%				
		22	0,0265	0,0000	21		25		11	21	91,67%	52,38%				
		23	0,0356	0,0000	10		17		11	22	91,67%	50,00%				
		25	0,0469	0,0000	17		22		11	23	91,67%	47,83%				
13	fusion | fission | atomes | ingenierie	2	0,0274	0,2640	8	1	3	1	1	1	20,00%	100,00%	1	1	20,00%	100,00%
		3	0,1347	**0,3617**	3	1	21	1	2	2	40,00%	100,00%	2	2	40,00%	100,00%
		7	0,0339	0,2640	15		20		2	3	40,00%	66,67%	2	3	40,00%	66,67%
		8	**0,2811**	0,3256	24	1	12	1	3	4	60,00%	75,00%	3	4	60,00%	75,00%
		10	0,0043	0,2127	12	1	8	1	4	5	80,00%	80,00%	4	5	80,00%	80,00%
		12	0,0924	0,3256	21	1	15		5	6	100,00%	83,33%	4	6	100,00%	66,67%
		13	0,0468	0,2640	13		24	1	5	7	100,00%	71,43%	5	7	100,00%	71,43%
		15	0,1163	0,2640	20		2		5	8	100,00%	62,50%	5	8	100,00%	62,50%
		20	0,0370	0,3256	7		7		5	9	100,00%	55,56%	5	9	100,00%	55,56%
		21	0,0810	0,3256	2		13		5	10	100,00%	50,00%	5	10	100,00%	50,00%
		22	0,0028	0,2127	10		10		5	11	100,00%	45,45%	5	11	100,00%	45,45%
		23	0,0025	0,2127	22		23		5	12	100,00%	41,67%	5	12	100,00%	41,67%
		24	0,1029	0,2640	23		22		5	13	100,00%	38,46%	5	13	100,00%	38,46%
		*1	*2	*2	*1	*3	*1	*3	*4	*4			*4	*4		

*1 : numéro de document
*2 : valeur calculée de pertinence
*3 : document pertinent ou non
*4 : colonnes utiles aux calculs automatiques de précision et de rappel

courbes:

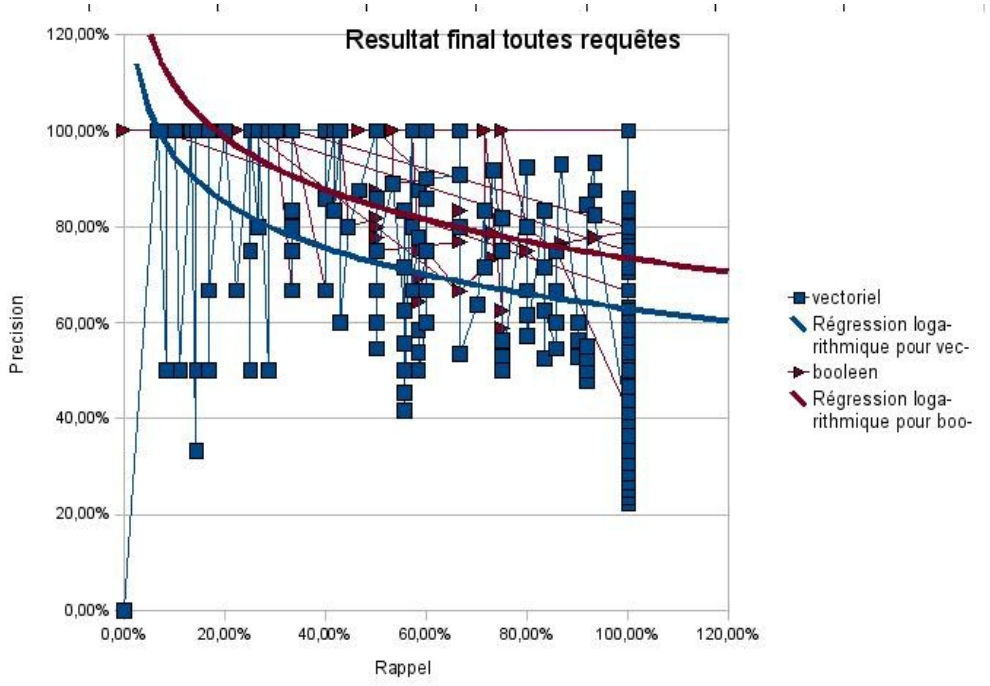




- Requête 9 : les couples (R/P) booléens sont principalement placés au-dessus des couples du modèle vectoriel. Cela peut s'expliquer de la manière suivante : le modèle booléen étendu a su repérer directement les réponses pertinentes, il s'avèrerait donc plus précis dans ce cas.
- Requête 13 : d'une part, le moteur vectoriel ne peut retourner aucun document de plus que le booléen étendu et d'autre part, les classement sont assez proches. Ceci est dû au fait que les opérateurs logiques de la requête ne sont que des « ou ». Ceci explique le fait que les courbes soient très proches l'une de l'autre.

Dans le cas où l'on ne peut distinguer clairement l'avantage d'un modèle sur l'autre, une méthode alternative est envisageable : celle des permutations. Celle-ci consiste, en référence à l'ordre idéal établi par l'utilisateur, constater le nombre de permutations nécessaires entre ce dernier et le classement obtenu pour chaque modèle afin de retrouver l'ordre idéal.

Par exemple, pour une liste ordonnée de documents pertinents <a, b, c>, soit un modèle m1 renvoyant <b, c, a> et un modèle m2 renvoyant <a, c, b>, alors nous pouvons compter le nombre de permutations pour m1 : perm(m1) = 2 et pour m2 : perm(m2) = 1. Ainsi, nous pouvons dire que le modèle m2 sera meilleur.



4 Analyse des résultats

Selon la courbe finale ci-dessus, le modèle booléen étendu est au-dessus, on la considère comme étant le meilleur ; or théoriquement, le modèle vectoriel devrait être le meilleur. Ces résultats sont à analyser avec prudence car pour notre série de tests nous avons utilisé un corpus comportant une trentaine de documents ce qui est tres peu pour une comparaison de différents SRI où habituellement ces tests sont effectués sur un corpus comportant plusieurs milliers de documents.

5 Conclusion

Nous avons pu obtenir des résultats conformes à la théorie des modèles seulement pour une minorité de requêtes. Cette théorie décrivait un modèle booléen précis et un modèle vectoriel retournant des résultats exhaustifs.
Malgré cela, nous avons pu observer que le booléen étendu permettait un tri des réponses du booléen pur, en cela il constitue pour ce dernier une extension appréciable.
Des tests sur un corpus plus important auraient sûrement pu nous permettre de situer le modèle booléen étendu comme moins performant que le vectoriel.

Réferences :

Salton, G., Fox, E. A. and Wu, H. (1983). Extended Boolean information retrieval. Communications of the ACM, 26(12): 1022-1036.

Zadeh, L. A. (1965). Fuzzy sets. Information and Control, 8: 338-353.

Baeza-Yates, R. and Ribeiro-Neto, B. 1999. Modern Information Retrieval. Addison-Wesley.

Lefèvre P. (2000). La recherche d'informations , du texte integral au thésaurus.

Bosc P. , Liétard L. , Pivert O. , Rocacher D. (2004) . Base de données. Gradualité et imprécision dans les bases de données; Ensemble flous.