# Introduction à la RI

## *1. Définition*

Un système de recherche d'information (RI) est un système qui permet de retrouver les documents pertinents à une requête d'utilisateur, à partir d'une base de documents volumineuse.

Dans cette définition, il y a trois notions clés: documents, requête, pertinence.

Document: Un document peut être un texte, un morceau de texte, une page Web, une image, une bande vidéo, etc. On appelle document toute unité qui peut constituer une réponse à une requête d'utilisateur. Dans ce cours, nous traitons seulement des documents textuels.

Requête: Une requête exprime le besoin d'information d'un utilisateur. Elle est en général de la forme suivante: "Trouvez les documents qui …".

Pertinence: Le but de la RI est de trouver seulement les documents pertinents. La notion de pertinence est très complexe. On verra cela plus en détail plus tard. De faon générale, dans document pertinent, l'utilisateur doit pouvoir trouver les informations dont il a besoin. C'est sur cette notion de pertinence que le système doit juger si un document doit être donner à l'utilisateur comme réponse.

Même pour des documents textuels, il existe beaucoup de formes quant à leur spécification. Un document peut être un texte sans aucune structuration à l'intérieur (on l'appelle aussi plein-texte); il peut aussi être un texte avec une partie structurée, ou complètement structuré. Dans la plupart de cas, on traite des documents partiellement structurés. Par exemple, la spécification d'un livre peut être comme suit:

> ISBN: 0-201-12227-8
> Auteur: Salton, Gerard
> Titre: Automatic text processing: the transformation, analysis, and retrieval of information by computer
> Editeur: Addison-Wesley
> Date: 1989
> …
> Contenu: <Texte du livre>

Dans cette spécification, une partie a été structurée (de ISBN à Date). Une autre partie (Contenu) ne l'est pas. Il est possible de cherche ce livre par les attributs externes comme ISBN, Auteur, …, Date. On peut aussi chercher ce livre par le contenu. Le premier type de recherche est relativement simple, étant donné la structuration existante, et le critère relativement simple pour comparer la requête avec la spécification. Par contre, la recherche par le contenu pose beaucoup de problème. C'est précisément cette dernière recherche qui est l'objet des études sur la RI (bien que la recherche via les attributs externes est aussi intégrée dans le même système).

## 2. Approches possibles

On peut imaginer quelques approches possibles pour réaliser un système de RI.

1. Une première approche très naïve consiste à considérer une requête comme une chaîne de caractères, et un document pertinent comme celui qui contient cette chaîne de caractères. À partir de cette vision simpliste, on peut imaginer l'approche qui consiste à balayer les documents séquentiellement, en les comparant avec la chaîne de caractères qui est la requête. Si on trouve la même chaîne de caractère dans un document, alors il est sélectionné comme réponse.

   Cette approche est évidemment très simple à réaliser. Cependant, elle a plusieurs lacunes:

   - Vitesse: L'opération de recherche est très lente. Pour chaque requête, on doit parcourir tous les documents dans la base. En général, il y en a des centaines de milliers, voire des millions. Il n'est donc envisageable d'utiliser cette approche que sur des collections très petites jusqu'à quelques centaines de documents.
   - Pouvoir d'expression d'une requête: Une requête étant une simple chaîne de caractères, il est difficile d'exprimer des besoins complexes comme "Trouver des documents concernant la base de données et l'intelligence artificielle utilisées dans l'industrie".

   Ainsi, cette approche n'est utilisée que dans des systèmes jouets très petits. La plupart de systèmes existants utilisent une approche différente basée sur une indexation.

2. L'approche basée sur une indexation

   Dans cette approche, on effectue certains pré-traitements sur les documents et les requêtes, ce qu'on appelle l'indexation. Cette opération vise à construire une structure d'index qui permet à retrouver très rapidement les documents incluant des mots demandés. La structure d'index est de la forme suivante (la structure de fichier inversé):

   $$\text{Mot} \rightarrow \{\ldots, \text{Doc}, \ldots\}$$

   C'est-à-dire, chaque mot est mis en correspondance avec les documents qui le contiennent.

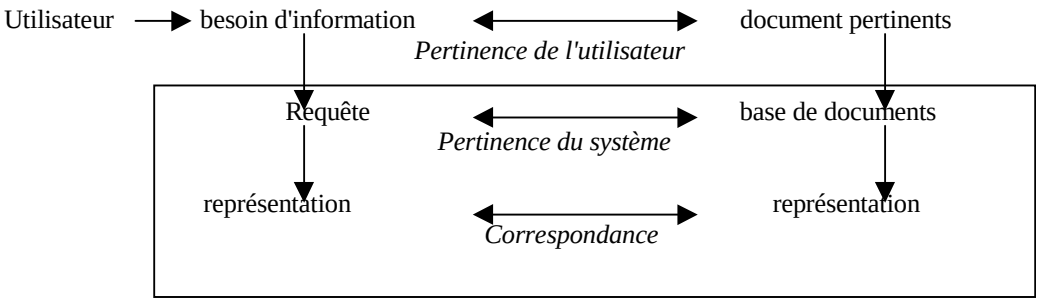
   Une requête peut être maintenant une expression plus complexe, incluant des opérateurs logiques (ET, OU, …) ou d'autres types d'opérateurs. L'évaluation est compositionnelle, c'est-à-dire, on commence par évaluer les éléments de base (par exemple, des mots) dans la requête, obtenant ainsi des listes de documents; ensuite, on combine ces listes selon l'opérateur qui relie ces éléments pour obtenir finalement une seule liste de documents.

   Par rapport à l'approche précédente, cette approche a les avantages suivants:

   - Elle est plus rapide. En effet, on n'a plus besoin du parcours séquentiel. Avec la structure d'index, on peut directement savoir quels documents contiennent tel ou tel mot.
   - L'expression des requêtes peut être très complexe, exprimant des besoins d'information complexes.

   Le prix à payer pour ces avantages est le besoin de l'espace de stockage supplémentaire pour la structure d'index. En général, cet espace correspond à 40% à 200% de la taille de collection de documents, selon la complexité de l'indexation. Mais ce besoin d'espace pose de moins en moins de problème maintenant.

   Utilisant cette deuxième approche, on peut voir les opérations et l'environnement de la RI comme suit:




On remarque qu'il y a trois niveaux différents:

- **Le niveau utilisateur**: A ce niveau, l'utilisateur a un besoin d'information dans sa tête, et il espère obtenir les documents pertinents pour répondre à ce besoin.
  La relation entre le besoin d'information et les documents attendus est la relation de pertinence (idéale, absolue, …).
- **Le niveau système**: A ce niveau, le système répond à la requête formulée par l'utilisateur par un ensemble de documents trouvés dans la base de documents qu'il possède.
  Remarquez que la requête formulée par l'utilisateur n'est qu'une description partielle de son besoin d'information. Beaucoup d'études ont montré qu'il est très difficile, voire impossible, de formuler une requête qui décrit complètement et précisément un besoin d'information. Du côté de document, il y a aussi un changement entre les deux niveau: les documents qu'on peut retrouver sont seulement les documents inclus dans la base de documents. On ne peut souvent pas trouver des documents parfaitement pertinents à un besoin. Il arrive souvent qu'aucun document pertinent n'existe dans la base.
- **Le niveau interne du système**: La requête formulée par l'utilisateur (souvent en langue naturelle) ne peut pas se comparer directement avec des documents en langue naturelle eux aussi. Il faut donc créer des représentations internes pour la requête et pour les documents. Ces représentations doivent être manipulables par l'ordinateur. Le processus de création de ces représentation est appelé l'indexation. Il est aussi a noter que les représentations créées ne reflètent qu'une partie des contenus de la requête et des documents. La technologie de nos jour de nous permet pas encore de créer une représentation complète.
  Pour déterminer si la représentation d'un document correspond à celle de la requête, on doit développer un processus d'évaluation. Différentes méthodes d'évaluation ont été développées, en relation avec la représentation de documents et de requête. C'est cet ensemble de représentation et la méthode d'évaluation qu'on appelle un *modèle* de RI.

On remarque qu'il y a des différences entre deux niveaux différents. En ce qui concerne le besoin d'information, il est transformé en une requête, puis en une représentation de cette dernière aux niveaux inférieurs. Du côté document, il y a des changements similaires. Les relations qu'on peut déterminer à chaque niveau ne sont pas pareilles non plus. Ce qu'on espère est qu'un bon système de RI puisse donner une évaluation de *correspondance* qui reflète bien la *pertinence du système*, qui à son tour, correspond bien au jugement de *pertinence de l'utilisateur*. Cependant, étant donné la différence entre les niveaux, il y a nécessairement une dégradation. Ainsi, une autre tâche de la RI est d'évaluer un système de RI une fois qu'il est construit. Cette évaluation du système tente de savoir l'écart entre les niveaux (surtout entre la second niveau et le troisième niveau).

### *3. Notion de pertinence*

Pertinence est la notion centrale dans la RI car toutes les évaluations s'articulent autour de cette notion. Mais c'est aussi la notion la plus mal connue, malgré de nombreuses études portant sur cette notion. Voyons quelques définitions de la pertinence pour avoir une idée de la divergence.

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 notions utilisées (informativité, relativité, surprise, …) restent très vaques. Pourquoi on arrive à cette situation? C'est parce que les utilisateurs d'un système de RI ont des besoins très variés. Ils ont aussi des critères très différents pour juger si un document est pertinent. Donc, la notion de pertinence est utilisée pour recouvrir un très vaste éventail des critères et des relations. Par exemple, un utilisateur qui a formulé la requête sur "système expert" peut être satisfait par un document décrivant toutes les techniques utilisées dans "MYCIN" qui est un exemple typique de système expert. Cependant, un deuxième utilisateur peut juger ce même document non-pertinent car il cherche plutôt une description non-technique. Dans les deux situations, on appelle la relation entre le document et la requête "pertinence".
Beaucoup de travaux ont été menés sur cette notion. On s'est vite aperçu que la pertinence n'est pas une relation isolée entre un document et une requête. Elle fait appel aussi au contexte de jugement. Ainsi, Tefko Saracevic propose la définition suivante pour tenir compte de cette influence multiple du contexte sur la pertinence (dans Saracevic (ed.), Introduction to information science, chap. 3 - The concept of relevance, R.R. Bowker company, 1970, bibliothéconomie, Z1001.S27-3):

> La pertinence est la A d'un B existant entre un C et un D jugé par un E.

où A = intervalle de la mesure
B = aspect de la pertinence (la pertinence absolue)
C = un document
D = contexte dans lequel la pertinence est mesurée (y compris le besoin d'information)
E = le juge (l'utilisateur)

Il reconnaît déjà l'importance du contexte sur la pertinence, ainsi que l'utilisateur lui-même. Si on varie ces facteurs, la notion de pertinence change aussi.
La question qu'on peut se poser est: à quoi sert d'étudier la notion de pertinence si on sait qu'elle est très variable? Une des raison est de tenter de trouver certains comportements communs entre les utilisateurs, et essayer de les formaliser. Si on arrive à cerne une partie de pertinence commune, on pourra l'implanter dans les systèmes pour répondre au moins à une partie commune des besoins. On connaît maintenant certains facteurs communs. Par exemple, le sujet

(ou en anglais topic) est le facteur le plus important dans la pertinence. Ainsi, on peut construire des systèmes en utilisant uniquement le critère de sujet, ce qui conduit à l'approche basée sur la *topicalité*. Une autre raison des études de la pertinence est d'essayer de comprendre exactement comment le contexte influence sur elle. Si on arrive à comprendre cela, par exemple, à trouver des contextes typiques dans lesquels un facteur devient très important, on pourra implanter des systèmes spécialisés en conséquence. Derrière ces études, il y a aussi des motivations philosophiques (de comprendre comment l'humain raisonne…). Bref, on est dans la même situation que la définition de l'intelligence en intelligence artificielle.

Dans le cadre de ce cours, nous allons d'abord considérer la pertinence du point de vue "topical" dans les approches classiques. Dans la second partie, nous allons analyser cette notion dans un contexte élargie où on peut voir l'influence des autres facteurs.

On peut lire des articles plus récents sur la pertinence dans Froehlich (ed.), Journal of the American Society for Information Science (JASIS), vol. 45, no. 3, Numéro spécial sur la pertinence, 1994 (disponible à la bibliothèque de l'Ecole de la bibliothéconomie et de science d'information - EBSI, Pavillon Lionel-Groulx).

## *4. Evaluation d'un système*

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 correspond à celles que l'utilisateur espère, mieux est le système.

### 4.1. Corpus de test (références)

Pour arriver à une telle évaluation, on doit connaître d'abord les réponses idéales de l'utilisateur. Ainsi, l'évaluation d'un système s'est faite souvent avec certains corpus de test. Dans un corpus de test, il y a:

- un ensemble de documents;
- un ensemble de requêtes;
- la liste de documents pertinents pour chaque requête.

Pour qu'un corpus de test soit significatif, il faut qu'il possède un nombre de documents assez élevé. Les premiers corpus de test développés dans les années 1970 renferment quelques milliers de documents. Les corpus de test plus récents (par exemple, ceux de TREC) contiennent en général plus 100 000 documents (considérés maintenant comme un corpus de taille moyenne), voir des millions de documents (corpus de grande taille).

L'évaluation d'un système ne doit pas se reposer seulement sur une requête. Pour avoir une évaluation assez objective, un ensemble de quelques dizaines de requêtes, traitant des sujets variés, est nécessaire. L'évaluation du système doit tenir compte des réponses du système pour toutes ces requêtes.

Finalement, il faut avoir les réponses idéales pour l'utilisateur pour chaque requête. Le dernier élément d'un corpus de test fournit cette information. Pour établir ces listes de documents pour toutes les requêtes, les utilisateurs (ou des testeurs simulant des utilisateurs) doit examiner chaque document de la base de document, et juger s'il est pertinent. Après cet exercice, on

connaît exactement quels documents sont pertinents pour chaque requête. Pour la construction d'un corpus de test, les jugements de pertinence constituent la tâche la plus difficile.

## 4.2. Précision et rappel

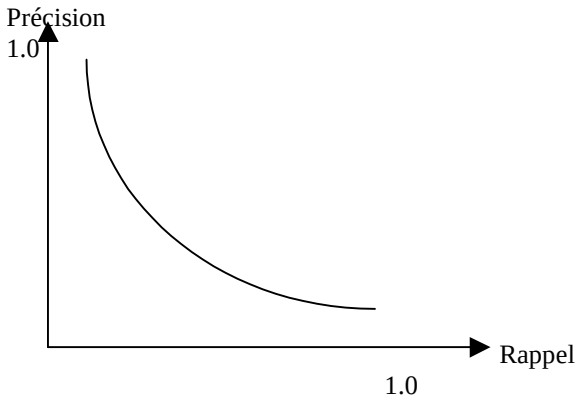
La comparaison des réponses d'un système pour une requête avec les réponses idéales nous permet d'évaluer les deux métriques suivantes:

Précision: La précision mesure la proportion de document pertinents retrouvés parmi tous les documents retrouvés par le système.
Rappel: Le rappel mesure la proportion de document pertinents retrouvés parmi tous les documents pertinents dans la base.

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

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

Idéalement, on voudrait qu'un système donne de bons taux de précision et de rappel en même temps. Un système qui aurait 100% pour la précision et pour le rappel signifie qu'il trouve tous les documents pertinents, et rien que les documents pertinents. Cela veut dire que les réponses du système à chaque requête sont constituées de tous et seulement les documents idéaux que l'utilisateur a identifiés. En pratique, cette situation n'arrive pas. Plus souvent, on peut obtenir un taux de précision et de rappel aux alentours de 30%.
Les deux métriques ne sont pas indépendantes. Il y a une forte relation entre elles: quand l'une augmente, l'autre diminue. Il ne signifie rien de parler de la qualité d'un système en utilisant seulement une des métrique. 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 souffrira. 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:




## 4.3. Comment évaluer Précision-Rappel?

La liste de réponses d'un système pour une requête peut varier en longueur. Une longue liste correspond à un taux de rappel élevé, mais un taux de précision assez basse, tandis qu'une liste courte représente le contraire. La longueur de la liste n'est souvent pas un paramètre inhérent d'un système. On peut très bien le modifier selon le besoin. Mais cette modification ne modifie pas le comportement global du système et de sa qualité. Ainsi, on peut varier cette longueur pour estimer les différents points de précision-rappel pour constituer une courbe de précision-rappel pour le système. Le processus d'évaluation est donc comme suit:

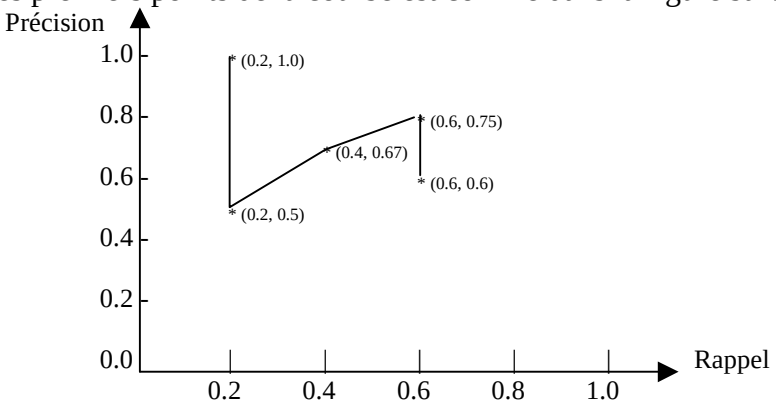
Pour i = 1, 2, … #document_dans_la_base faire:
- évaluer la précision et le rappel pour les i premier documents dans la liste de réponses du système

Par exemple, soit une requête qui a en tout 5 documents pertinents dans la base. La liste de réponse du système à cette requête est comme suit:

| Liste de réponses | Pertinence |
|---|---|
| Doc1 | (*) |
| Doc2 | |
| Doc3 | (*) |
| Doc4 | (*) |
| Doc5 | |
| … | |

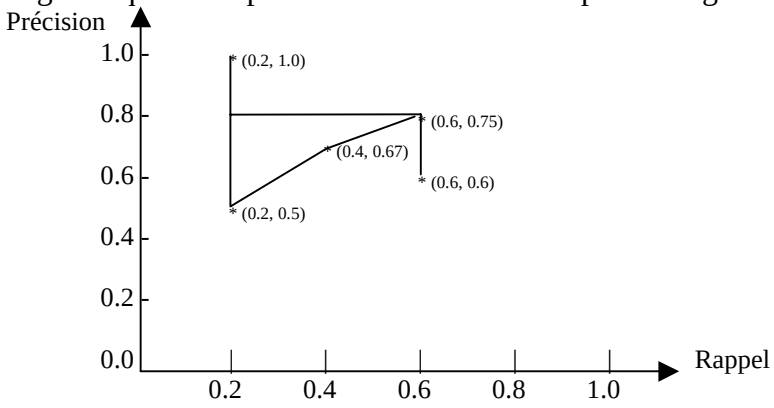
où (*) signifie que c'est un document pertinent (selon l'évaluation de l'utilisateur).
On considère d'abord le premier document Doc1 comme la réponse du système. À ce point, on a retrouvé un document pertinents parmi les 5 existants. Donc on a un taux de rappel de 0.2. La précision est 1/1. Le point de la courbe est (0.2, 1.0).
On considère ensuite les deux premiers documents comme la réponse (Doc1 et Doc2). À ce point, on a le même rappel (toujours 1/5), mais la précision devient 1/2. Ainsi le point est (0.2, 0.5).
On considère Doc1, Doc2 et Doc3, on a un rappel de 2/5, et une précision de 2/3: (0.4, 0.67).
…

Cette processus est continué jusqu'à l'épuisement de toute la liste de réponse du système (qui peut être très longue, jusqu'à inclure tous les documents de la base).
Les premiers points de la courbe est comme dans la figure suivante:




Cette courbe ne correspond pas tout à fait à la forme générale. Mais c'est juste pour une seule requête. Si on calcule la moyenne sur un ensemble de requêtes, la courbe sera plus lisse, et ressemble davantage à la forme générale.

Il arrive fréquemment qu'on applique la *interpolation* sur la courbe de chaque requête. La polarisation vise à créer une courbe qui descend (comme la forme générale). Le traitement est le suivant: Soit i et j deux points de rappel, et i<j. Si au point i, la précision est inférieure à la précision au point j, alors, on augmente la précision du point i à celle du point j. Concrètement, cela signifie qu'on remplit un creux de la courbe par une ligne horizontale comme suit:




On obtient donc une courbe en forme d'escalier. L'idée qui motive la polarisation est que les creux de la courbe ne représentent pas vraiment la performance du système. S'il existe un point à un rappel et une précision plus élevés, on peut toujours donner plus de documents dans la réponse pour augmenter la performance. Donc, le creux est surmontable.
Évidemment, on peut discuter sur cette motivation, et être en désaccord. Ce n'est pas important. L'important est qu'on compare les systèmes sur la même base. Si tous les systèmes sont mesurés avec une courbe polarisée, alors la polarisation ne donne pas plus d'avantage à un système qu'à un autre. Donc, la courbe polarisée est aussi une base équitable pour comparer des systèmes.

### 4.3. Comparaison de systèmes et Précision moyenne

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.
Il arrive parfois que les deux courbes se croisent. Dans ce cas, il est difficile de dire quel système est meilleur. Pour résoudre ce problème, on utilise aussi la *précision moyenne* comme une mesure de performance. La précision moyenne est une moyenne de précision sur un ensemble de points de rappel. On utilise soit la précision moyenne sur 10 points de rappel (0.1, …, 1.0), ou celle sur 11 points de rappel (0.0, 0.1, …, 1.0). Cette dernière est possible seulement avec la polarisation.
La précision moyenne décrit bien la performance d'un système. C'est la mesure souvent utilisée en RI.
Pour comparer deux systèmes ou deux méthodes, on utilise souvent l'amélioration relative qui est calculée comme suit :

Amélioration de méthode 2 sur méthode 1 = (performance de méthode 2 – performance de méthode 1) / performance de méthode 1.

## *5. Bref historique de la RI*

La RI n'est pas un domaine récent. Il date des années 1940, dès la naissance des ordinateurs. Au début, la RI se concentrait sur les applications dans des bibliothèques, d'où aussi le nom "automatisation de bibliothèques". Depuis le début de ces études, la notion de pertinence a toujours été un objet. Dans les années 1950, on commençait de petites expérimentations en utilisant des petites collections de documents (références bibliographiques). Le modèle utilisé est le modèle booléen. Dans les années 1960 et 1970, des expérimentations plus larges ont été menées, et on a développé une méthodologie d'évaluation du système qui est aussi utilisée maintenant dans d'autres domaines. Des corpus de test (e.g. CACM) ont été conçus pour évaluer des systèmes différents. Ces corpus de test ont beaucoup contribué à l'avancement de la RI, car on pouvait les utiliser pour comparer différentes techniques, et de mesurer leurs impacts en pratique. Le système qui a le plus influencé le domaine est sans aucun doute SMART, développé à la fin des années 1960 et au début des années 1970. Les travaux sur ce système a été dirigés par G. Salton, professeur à Cornell. Certains nouvelles techniques ont été implantées et expérimentées pour la première fois dans ce système (par exemple, le modèle vectoriel et la technique de relevance feedback). Du côté de modèle, il y a aussi beaucoup de développements sur le modèle probabiliste.
Les années 1980 ont été influencées par le développement de l'intelligence artificielle. Ainsi, on tentait d'intégrer des techniques de l'IA en RI, par exemple, système expert pour la RI, etc.
Les années 1990 (surtout à partir de 1995) sont des années de l'Internet. Cela a pour effet de propulser la RI en avant scène de beaucoup d'applications. C'est probablement grâce à cela que vous entendez parler de la RI. La venue de l'Internet a aussi modifié la RI. La problématique est élargie. Par exemple, on traite maintenant plus souvent des documents multimédia qu'avant. Cependant, les techniques de base utilisées dans les moteurs de recherche sur le web restent identiques.

## *6. Relations avec d'autres domaines*

La RI a des relations fortes avec d'autres domaines, notamment avec les bases de données et avec des systèmes de question-réponse.

### 6.1. RI et BD

On peut imaginer un système de RI comme un système de BD textuelles. Cependant, il faut souligner la différence suivante entre les deux types de système: Dans une base de données, on doit d'abord créer des schémas pour organiser les données. Ces schémas définissent des relations, ainsi que les attributs dans chaque relation. C'est en utilisant ces schémas que le système arrive à interpréter une requête de l'utilisateur. Par exemple, on peut définir la relation suivante dans une base de données:

Auteur(Livre, Nom).

où Auteur est le nom de la relation, Livre et Nom sont ses attributs qui correspondent à l'identification d'un livre et à son (un des) auteur(s).
(Ceci est juste une partie de définition). Pour trouver les livre écrits par "Knuth", on peut poser la requête suivante en SQL:

```
select Livre
from Auteur
where Nom = "Knuth"
```

Cette requête n'est valide que si la relation Auteur a été créée ainsi.
Dans la RI, une partie des spécifications de documents est structurée, notamment les attributs externes. Cette partie peut être organisée assez facilement comme une relation en BD, et ainsi utiliser des SGBD existants pour rechercher des documents selon des critères sur les attributs externes. Mais, comme ce qu'on a dit, cette partie ne représente pas le cœur de la RI. Le cœur se situe dans la recherche selon le contenu. Or, le contenu est en général sans structure, ou avec une structure très souple. Il est très difficile de créer une relation pour représenter la partie contenu de document.
Après l'indexation de document, cependant, la connexion entre la RI et les BD devient plus étroite. Le résultat de l'indexation est d'associer à chaque document un ensemble d'index. Ce résultat peut être vu comme une relation en BD:

Index(Doc, Mot).

Ainsi, il est possible de faire une requête pour sélectionner les documents contenant le mot "recherche" et le mot "information" comme suit:

```
(select Doc
 from Index
 where Mot = "recherche")
intersect
(select Doc
 from Index
 where Mot = "information")
```

ce qui signifie que l'intersection de deux sélections sera le résultat.

Noter, cependant, que les sélections ne retournent qu'un ensemble de documents sans ordonnancement. En RI, l'ordre de documents dans la liste de réponse est important. Ainsi, les BD ne permettent de réaliser qu'une partie de fonctionnalités de la RI.

## 6.2. RI et système question-réponse

Un système QR permet de répondre aux questions relatives à un petit domaine. Par exemple, on peut poser la question "quelle version de Word est disponible sous Windows 98?" à un système spécialisé sur le marché de logiciel. Pour cela, il faut qu'on crée une modélisation du domaine d'application dans lequel les concepts ou objets sont reliés par des relations sémantiques. Ce modèle permettra de retrouver le concept ou l'objet et ainsi donner une réponse directe à la question. Pour notre exemple, la réponse peut être "Word 95 et Word 98", par exemple.
On voit ici qu'il y a une différence sur la nature de réponse entre les deux types de système. Dans RI, c'est une réponse indirecte à une question: on identifie les documents dans lesquels l'utilisateur peut trouver des réponses directes à sa question. Tandis que dans un système QR, on fournit une réponse directe.
Il y a des tentatives de rapproche la RI vers des systèmes QR, mais cela s'avère très difficile. La raison principale est que la RI s'applique en général à tous les domaines sans restriction. Il est impossible, dans ce cas, de créer un modèle nécessaire pour déduire la réponse directe à une question dans un système QR. Dans certains contextes très spécialisés, la RI incorpore une base de connaissances. Elle utilise aussi des raisonnement pour déduire si un document peut être pertinent ou pas. Donc, le fonctionnement de ce type de RI ressemble un peu plus à celui d'un système QR.
Une tentative plus restreint consiste à raffiner la notion de document dans la réponse: au lieu de fournir un document complet comme une réponse, on essaie d'identifier un passage dans le document (passage retrieval). C'est une étape qui diminue un peu la distance entre la RI et la QR. Mais la différence fondamentale reste la même.