République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab Blida

Faculté des sciences

Département d'informatique

Mémoire pour l'obtention

D'un diplôme de ***Master en informatique***

Sujet :

Développement d'un système de recherche automatique d'images

Présenté le *: 02 décembre 2009*

Par

AISSIOU Nasr Eddine

Devant le jury :

Président : - Melle. REGUIEG ***Promotrice*** *: Mlle. BENBLIDIA*

Examinateurs : - Mme. BOUMAHDI ***Encadreur*** *: Mlle. BAKALEM*

- Mme. AZZOUZ

Laboratoire de Recherche Des Systèmes Informatisés

(LRDSI)

REMRCIEMENTS

JE REMERCIE TOUT D'ABORD ALLAH QUI M'A DONNÉ L'INTELLIGENCE ET QUI M'A DONNÉ LA VOLONTÉ ET LE COURAGE POUR ACCOMPLIR CE MODESTE TRAVAIL.

UN TRÈS GRAND MERCI À MES EXAMINATEURS POUR L'HONNEUR QU'ILS ME FONT EN PARTICIPANT À CE JURY.

JE TIENS À REMERCIER TRÈS SINCÈREMENT MA PROMOTRICE : MLLE.BENBLIDIA. SON ATTENTION ET SON GOÛT POUR LA RECHERCHE SONT UN MODÈLE POUR MOI.

JE REMERCIE TOUT PARTICULIÈREMENT MON ENCADREUR : MLLE.BAKALEM QUI M'A AIDER ET ACCEPTER DE CONSACRER UNE PARTIE DE LEUR TEMPS À LA CORRECTION DE CE DOCUMENT. JE LA REMERCIE POUR SES CONSEILS ET CES REMARQUES QUI M'ONT TOUJOURS ÉTÉ PRÉCIEUX ET M'ONT PERMIS D'AMÉLIORER GRANDEMENT LA QUALITÉ DE MON TRAVAIL ET DE CE MÉMOIRE.

MERCI AUSSI À TOUTES LES PERSONNES QUI ONT PARTICIPÉ DE PRÈS OU DE LOIN À MON TRAVAIL TOUT PARTICULIÈREMENT H..., MOURAD, MOUHAMED.

JE NE PEUX ENFIN CLÔTURER CES REMERCIEMENTS SANS REMERCIER DU FOND DU CŒUR MES PARENTS, POUR M'AVOIR AIDÉ ET ENCOURAGÉ AU COURS DE CES LOOOOOOOOONGUES ANNÉES, DANS LES BONS ET MOINS BONS MOMENTS ET QUI SE SONT BEAUCOUP SACRIFIÉS POUR MOI ET SANS LESQUELS JE NE SERAIS PAS LÀ AUJOURD'HUI, J'ESPÈRE QUE VOUS SEREZ FIERS DE MOI.

Tables des matières

Tables des figures :

Table des algorithmes :

Table des tableaux :

INTRODUCTION GENERALE

Avec l'expansion de l'informatique et du multimédia, une problématique nouvelle est apparue : gérer les quantités énormes et croissantes de documents numériques aujourd'hui disponibles. Parmi ces documents, nous nous intéressons plus particulièrement aux images numériques.

Ces images aux thèmes variées (sport, actualité, art, histoire, etc.), parfois œuvres de photographes renommés, sont issues de diverses collections publiques ou privées du monde entier.

L'image sous toutes ses formes (photographie, vidéo, graphique ou dessin numérique) est donc aujourd'hui à disposition des professionnels mais aussi du grand public.

Les images fixes et les séquences d'images sont en général compressées puis archivées dans des bases de données, généralistes ou spécialisées, accessibles par les réseaux de télécommunication.

Afin de pourvoir à la réutilisation de ces données, il est donc capital de développer des outils de recherche.

Contrairement aux documents textuels, pour lesquels des méthodes d'indexation et de recherche existent depuis les années 1970 ***[Sal71],*** les outils d'analyse et d'interprétation de l'image sont souvent en décalage avec le contenu sémantique, souvent riche de celle-ci.

L'émergence de la recherche d'images par le contenu remonte au début des années 90.

Sujet de mémoire :

Nous proposons dans ce mémoire un système textuel de recherche qui se focalise sur les aspects ((image)), à savoir sur l'utilisation et l'application des méthodes pour la recherche dans de bases d'images généralistes, avec toutefois pour objectif la satisfaction de requêtes sémantiques. Pour cela, nous allons utiliser le modèle booléen comme modèle de représentation et de recherche.

De manière plus détaillée, nous nous proposons :

- D'améliorer la formulation de la requête en donnant la possibilité à un utilisateur d'entrer des mots clés en utilisant des opérateurs booléens, ces derniers vont d'un coté préciser plus la requête, et d'un autre coté d'aider l'utilisateur à exprimer mieux ce besoin.

- De vérifier la sémantique dans la représentation des requêtes, et aussi dans la représentation des images de la base.

- De déterminer de manière exacte et précise les images pertinentes à la requête et les afficher en résultat.

Organisation du document :

Ce rapport s'organise de la façon suivante :

- Le chapitre 1 fait un état de l'art concernant la recherche d'information, les systèmes de recherche des informations et les systèmes de recherche d'images, ainsi que les différentes approches de représentation d'image et les différents langages d'interrogation. Cet état de l'art permet d'élaborer notre modèle de recherche d'information dans lequel nous pouvons utiliser le modèle booléen qui est détaillé dans ce chapitre.

– Dans le deuxième chapitre nous développons une approche booléenne de recherche d'images : BMIR.

– Le chapitre 4 aborde le dernier point de notre travail : la mise en œuvre et l'évaluation de notre système.

– À la fin de ce document, une conclusion fait le bilan sur notre travail.

Partie 1 : La Recherche d'Information (RI) :

S'il est important de savoir modéliser, transmettre, et stocker de l'information, il est également important de permettre aux utilisateurs d'un Système d'Information de localiser rapidement une information recherchée *[Mar04].*

1.1 Définition :

La recherche d'information (Information Retrieval) est le champ du domaine informatique qui s'occupe de la représentation, du stockage, de l'organisation et de l'accès aux informations.

Ces dernières années ont vu une explosion du volume des données accessibles par les utilisateurs d'ordinateurs surtout à cause de l'énorme croissance du Web. De grandes quantités de données sont accessibles au public, mais la détection efficace des informations pertinentes reste toujours une tâche très difficile *[Dio05].*

1.2 Bref historique de la RI : [01]

- 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".

- 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 ont été dirigés par G. Salton, professeur à Cornell. Certaines 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.

1.3 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 ***[01].***

1.3.1 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 livres 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, 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.

1.3.2. 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 utilise aussi des raisonnements 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.

Partie 2 : Systèmes de recherche d'informations (SRI) :

2-1 Définition :

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 ***[01].***

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.

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 façon 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 donné à l'utilisateur comme réponse.

2-2 Approches possibles :

On peut imaginer quelques approches possibles pour réaliser un système de RI *[01].*

A. 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.

B. L'approche basée sur une indexation :

Dans cette approche, on effectue certains prétraitements 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 :

Mot ? {…, Doc, …}

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.

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

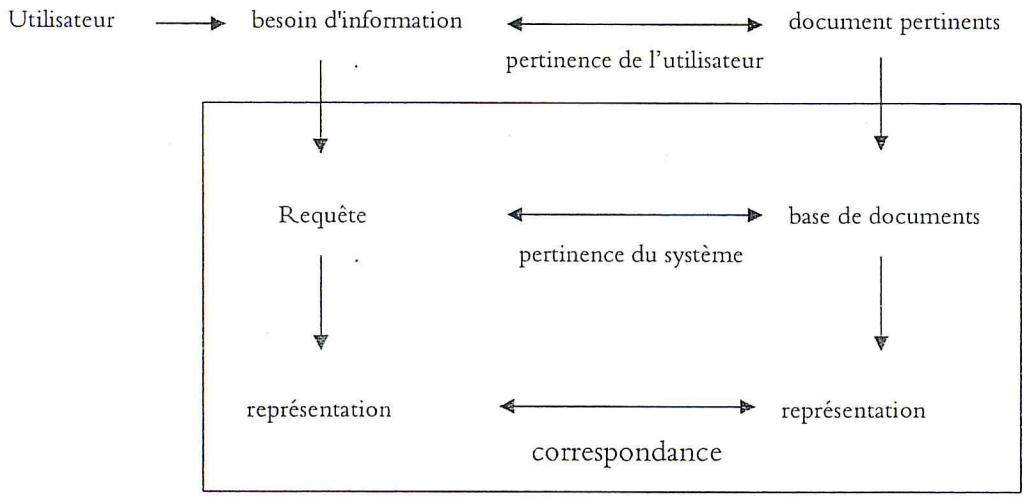


Figure 1.1 : Schéma illustrant les trois niveaux existants.

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 niveaux : 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 représentations est appelé l'indexation. Il est aussi à noter que les représentations créées ne reflètent qu'une partie des contenus de la requête et des documents. La technologie de nos jours de nous ne 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 le second niveau et le troisième niveau).

2-3 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 ***[01].***

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;
- ...

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.

Ainsi, Tefko Saracevic propose la définition suivante pour tenir compte de cette influence multiple du contexte sur la pertinence.

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)
Si on varie ces facteurs, la notion de pertinence change aussi.

2.4 Les modèles-piliers de la Recherche d'Information :

Un modèle de RI a pour rôle de fournir une formalisation du processus de recherche d'information. Il doit accomplir plusieurs rôles dont le plus important est de fournir un cadre théorique pour la modélisation de la mesure de pertinence.

Un modèle de représentation d'information définit le modèle d'indexation, le modèle de requêtes, et la méthodologie de d'évaluation de la correspondance.

Il existe plusieurs modèles de recherche d'information. Ces modèles ont été décrits dans de nombreux ouvrages sur la Recherche d'Information.

Ces modèles sont étroitement liés, mais on distingue trois principaux courants **[Sau05]**:

- ***les modèles basés sur la théorie des ensembles*** :

  Dont le représentant le plus connu est le modèle booléen. Dans ces modèles, des opérateurs logiques (OR, AND, NOT) séparent les termes de la requête et permettent d'effectuer des opérations d'union, d'intersection et de différence entre les ensembles de résultats associés à chaque terme.

- ***les modèles algébriques*** :

  Dont le premier représentant a été le modèle vectoriel : dans ces modèles, la pertinence d'un document vis-à-vis d'une requête est définie par des mesures de distance dans un espace vectoriel.

- ***les modèles probabilistes :***

  Reposant sur la théorie des probabilités : pour ces modèles, la pertinence d'un document vis-à-vis d'une requête est vue comme une probabilité de pertinence document/requête.

Les approches de RI textuelle développées depuis 30 ans ***[Sal71]*** se basent sur les mots clés utilisés comme descripteurs du contenu des documents. L'énorme avantage de cette approche est qu'elle peut être automatisée, donc rapide et applicable à de très grands corpus, comme nous le constatons avec les chercheurs sur le Web qui sont capables de retrouver des documents parmi des centaines de millions en quelques secondes ***[Bri98].***

- Nous mettons ici en évidence le fait que la base de travail d'un SRI textuel est la même que celle de l'utilisateur : les mots de la langue naturelle. Les mots clés sont extraits automatiquement, selon qu'ils sont plus ou moins représentatifs du contenu des images. Les différents modèles de RI textuelle dits *classiques* parmi lesquels figurent les modèles booléen, vectoriel et probabiliste, sont basés sur cette même notion de vocabulaire d'indexation à base de mots clés.

- La différence se situe dans les modèles de documents et de requêtes, et dans l'évaluation de la correspondance entre une image et une requête.

- Nous présentons ci-dessous le modèle booléen, qui est la base de notre travail [1].

2.5 Le modèle booléen (Boolean Model) :

Dès l'apparition des premiers SRI, le modèle booléen (1960) ***[Sal83]*** s'est imposé grâce à la simplicité et à la rapidité de sa mise en oeuvre. Il est basé sur la théorie des ensembles et l'algèbre de Boole. L'interface d'interrogation de la plupart des moteurs de recherche est basée sur ce principe et n'est composée que d'une liste de mots-clés.

- Dans ce modèle, un document est représenté comme une conjonction logique de termes (non pondérés), par exemple,

$$d = t_1 \wedge t_2 \wedge \ldots \wedge t_n \qquad (1)$$

- Une requête q est composée de termes liés par les trois connecteurs logiques AND, OR, NOT ($\wedge, \vee, \neg$) afin de déterminer au mieux le souhait d'un utilisateur. Par exemple:

$$q = (t_1 \wedge t_2) \vee (t_3 \wedge \neg t_4) \qquad (2)$$

- La fonction de correspondance est basée sur l'implication logique en logique des propositions :

  Pour qu'un document *d* réponde à une requête *q*, il faut que l'implication logique suivante soit valide:

$$d \Rightarrow q \qquad (3) \qquad \textit{[Ann08]}$$

[1] *Pour les lecteurs intéressés, l'annexe A décrit brièvement les modèles : booléen étendu, vectoriel, probabiliste, logique, les Requêtes à base de questions et les Modèles hybrides.*

- Cette déduction est utilisée par :

  - Axiomes : $(a \wedge b) \Rightarrow a$, $(a \wedge b) \Rightarrow b$, $a \Rightarrow (a \vee b)$, $b \Rightarrow (a \vee b)$, ...
  - modus ponens (MP) : si **a et a ⇒ b alors b**

Exemple : $D = t1 \wedge t3$ et $Q = t1 \vee t4$

– Déduction :
1. $t1 \wedge t3 \Rightarrow t1$ (équivalent à $D \Rightarrow t1$)
2. MP(1) : t1
3. $t1 \Rightarrow t1 \vee t4$ (équivalent à $t1 \Rightarrow Q$)
4. MP(3) : Q

  - Q est donc dérivable à partir de D, donc $D \Rightarrow Q$, donc le document répond à la requête *[Mul]*.

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

$$R(d,Q) = \begin{cases} 1 & \textit{Si Im appartient à l'ensemble décrit par Q} \\ 0 & \textit{Sinon} \end{cases} \qquad (4)$$

Exemple :

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

- d1(t1,t2,t5) ; d2(t1,t3,t5,t6) ; d3(t1,t2,t3,t4,t5)

  Rsv(d1,q)=1
  Rsv(d2,q)=0
  Rsv(d3,q)=1

↓ *Soit R* : fonction de pertinence (*Relevance*) d'un couple *(document, requête)* :

$$R(d,t) = 1 \quad Si\ t \in d \qquad (5)$$
$$R(d,t) = 0 \quad Si\ t \notin d \qquad (6)$$
$$R(d, t_1\ AND\ t_2) = R(d,t_1) \times R(d,t_2) \qquad (7)$$
$$R(d, t_1\ OR\ t_2) = R(d,t_1) + R(d,t_2) - R(d,t_1) \times R(d,t_2) \qquad (8)$$
$$R(d, t_1\ NOT\ t_2) = R(d,t_1) \times (1 - R(d,t_2)) \qquad (9)$$

Le modèle booléen considère que les termes de l'index sont présents ou absents d'un document (Appariement Exact).

En conséquence, les poids des termes dans l'index ou dans la requête sont binaires (pondération binaire), c'est-à-dire $w_{i,j} \in \{0, 1\}$ (1 si présent et 0 si absent).

Ainsi, le modèle booléen affirme que chaque document est soit pertinent soit non-pertinent (La fonction de comparaison est binaire : elle sépare le *corpus* en deux groupes – documents pertinents ou non –). Il n'y a pas de notion de réponse partielle aux conditions de la requête, Les documents retournés par le système sont considérés à pertinence égale ***[Bou06]***.

Le modèle booléen est le pionnier des systèmes de recherche d'information commerciaux. Son principal avantage est sa transparence. En effet, pour l'utilisateur, la raison pour laquelle un document a été sélectionné par le système est claire : il répond exactement à la requête qui a été formulée.

– *Remarques : [Mul]*

1) Correspondance stricte : Oui/Non

– $Q = t1 \wedge t3 \wedge t4$

– $D1 = t1 \wedge t4$,

$$D1 \not\Rightarrow Q$$

– Le document D1 (représenté par D1) n'est pas pertinent pour la requête Q (représentée par Q) d'après le modèle, alors qu'il contient une description « proche » de la requête.

Par exemple, considérons un document contenant les deux termes : information et traditional. Ce document ne sera pas pertinent pour la requête « information AND traditionnel AND model ».

2) Pas de distinction entre les documents pertinents

- $Q = t1 \wedge t4$
- $D1 = t1 \wedge t4$,
- $D2 = t1 \wedge t3 \wedge t4 \wedge t5 \wedge t6 \wedge t7$

$$D1 \Rightarrow Q \quad \text{et} \quad D3 \Rightarrow Q$$

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

Avantages :

- Simplicité de la conception du modèle ;
- Possibilité de structurer une requête avec des opérateurs logiques ;
- Bonnes performances quantitatives, même sur de très grandes collections de documents.

Partie 3 : Systèmes de recherche d'images (SRIm) :

La notion d'un document a beaucoup évolué depuis qu'il a pris la forme électronique. Les documents actuels contiennent plusieurs types de media (texte, son, image, vidéo). Pourtant la recherche d'informations a été orientée pendant longtemps vers les informations textuelles. Une raison fondamentale est que la majorité des documents étaient historiquement du texte. D'autre part la technologie disponible jusqu'au présent ne facilitait pas le traitement de grands nombres de données non textuelles. En conséquence les systèmes de recherche pour d'autres types de media ont évolués postérieurement que les SRI textuels ***[Dio05].*** Le media qui nous intéresse dans ce rapport est l'image.

3.1 Définition d'une image :

On peut trouver de nombreuses définitions d'une image :

- Le terme « image» (latin *imaginem, imago*) se définit, dans son acception la plus large, comme une représentation graphique, picturale ou sculpturale, comme représentation par la pensée, reproduction visuelle d'un objet réel, représentations mentales produites par l'esprit ou l'imagination, en rêve ou éveillé ***[02].***
- L'image est une représentation d'une personne ou d'un objet par la peinture, la sculpture, le dessin, la photographie, le film, etc ***[Kad99].***
- C'est aussi un ensemble structuré d'informations qui, après affichage sur l'écran, ont une signification pour l'œil humain ***[Kad99].***
- Elle peut être décrite sous la forme d'une fonction I(x,y) de brillance analogique continue, définie dans un domaine borné, tel que x et y sont les coordonnées spatiales d'un point de l'image et I est une fonction d'intensité lumineuse et de couleur. Sous cet aspect, l'image est inexploitable par la machine, ce qui nécessite sa numérisation ***[Kad99].***

- ***Image numérique :*** Contrairement aux images obtenues à l'aide d'un appareil photo, ou dessinées sur du papier, les images manipulées par un ordinateur sont numériques ***[Kad99].***

  On désigne sous le terme d'image numérique toute image (dessin, icône, photographie, ...) acquise, créée, traitée, stockée sous forme binaire (suite de 0 et de 1) ***[03].***

L'image numérique est l'image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs prélevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter ***[Kad99].***

Une image a une résolution définie par : le nombre de pixels par unité de longueur de la structure à numériser (classiquement en dpi (dots per inches) ou ppp (points par pouces)). La résolution d'une image numérique définit le degré de détails qui va être représenté sur cette image.

Il y a trois types des images :

- Les images binaires (deux valeurs : noir et blanc).
- Les images en niveaux de gris (renferment 256 teintes de gris).
- Les images couleurs.

Une image a un format qui spécifie sa structuration des données. Nous distinguons plusieurs types de formats :

- Le format JPEG (Joint Photo Expert Group)
- Le format GIF
- Le format TIFF (Tag Image File Format)
- BMP, PNG et Multiple-image Network Graphics, Canon RaW, SVG, TGA, PICT et PICS, ILBM, PCX, RAS, U3D et Windows Media Photo ***[04].***

Les images sont caractérisées par trois notions nécessaires qui sont : la texture, la couleur et la forme.

Ces trois caractéristiques représentent le contenu physique (visuel) de l'image.

De nombreux domaines peuvent profiter de systèmes de recherche d'images (SRIm) efficaces :

- l'éducation où les informations visuelles peuvent servir comme outils d'enseignement
- les professionnels pour lesquels la récupération des informations visuelles est primordiale (journalistes, architectes etc.)
- la protection de propriété intellectuelle qui peut être mieux assurée avec des systèmes de recherche d'images efficaces et précis ***[Dio05].***

Les systèmes de recherche d'images permettent d'améliorer à la fois la qualité des images renvoyées par le système, mais aussi de diminuer le temps qu'un utilisateur passe à essayer de trouver l'image qu'il recherche. Sans des outils de recherche les informations visuelles seront inaccessibles et donc inutilisables.

3.2 Architecture générale d'un SRIm :

La figure 1.2 décrit l'architecture générale d'un SRIm ; elle présente les éléments principaux mis en jeu dans un SRIm.

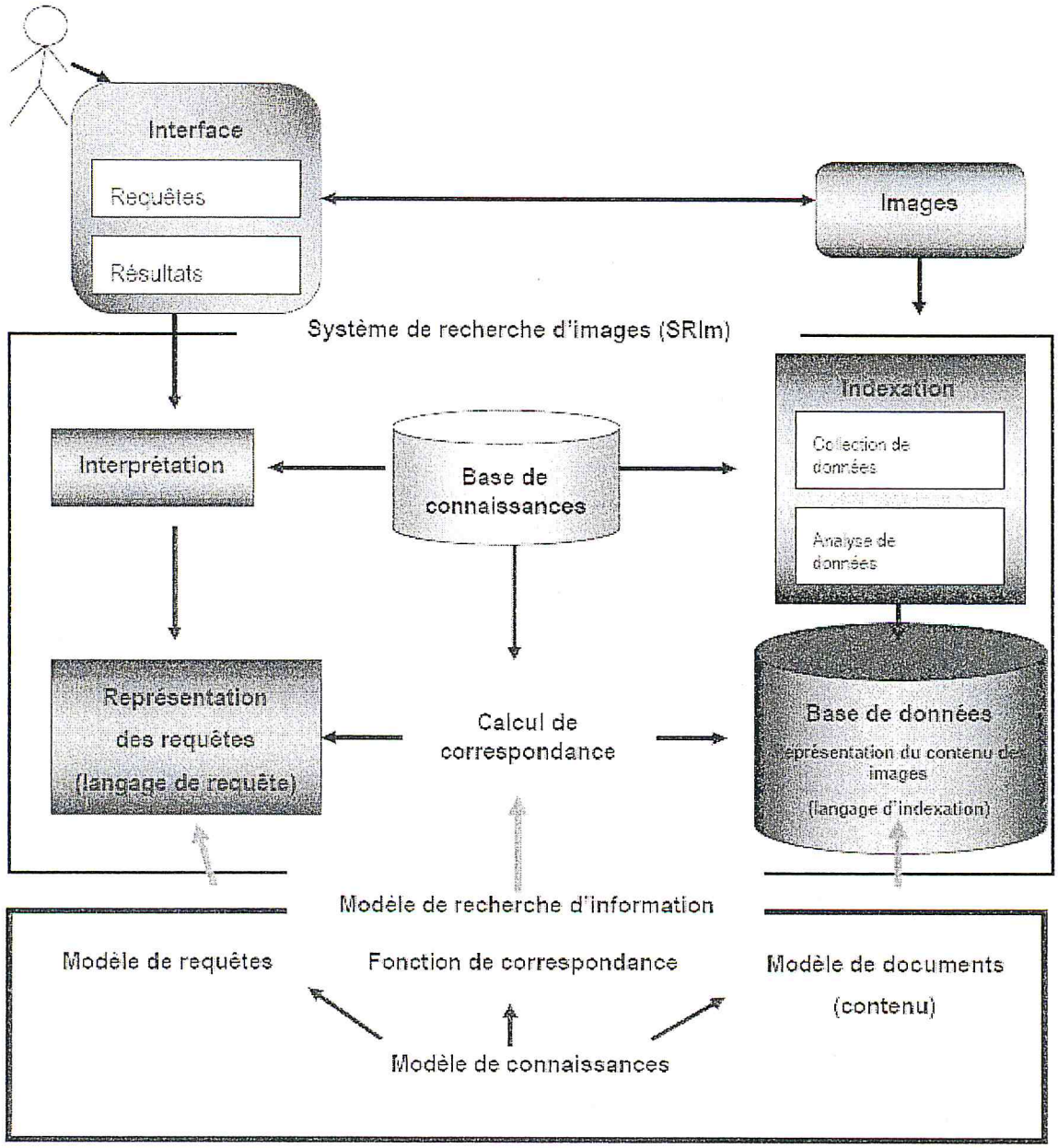


Figure 1.2 : Architecture générale d'un SRIm.

Les phases principales d'un SRIm sont l'indexation et l'interrogation, présentées ci-dessous. Chacune de ces phases a plusieurs composantes qui exécutent les tâches essentielles du système.

3.2.1. L'indexation :

Consiste à créer à partir de chaque image initiale, une instance du modèle d'image; c'est-à-dire de représenter le contenu sémantique d'une image sous la forme d'un index électronique directement manipulable par le système.

On appelle contenu sémantique d'image une représentation du sens associé au contenu brut d'image (selon le modèle de requête), qui correspond lui-même au niveau *signal* (c'est-à-dire la matrice de pixels).

De cette façon la recherche des images par rapport à un besoin particulier de l'utilisateur sera faite en fonction de ce contenu ; une image sera considérée comme pertinente si et seulement si son contenu correspond aux besoins de l'utilisateur.

Tous les index et les informations collectés sont stockés dans la base de données du système. La base est utilisée dans la phase d'interrogation pour récupérer les images stockées et ensuite déterminer leur correspondance avec la requête. Dans quelques SRIm une base de connaissances est utilisée. Cette base contient les connaissances du domaine et elle peut aussi inclure des informations supplémentaires, comme un thésaurus par exemple.

3.2.2. L'interrogation :

Est l'interaction d'un utilisateur final avec le SRIm, une fois le contenu sémantique des images représenté de manière interne sous forme d'index. L'utilisateur exprime son besoin d'information au système et ce dernier lui fourni des réponses.

La réponse à ce besoin est la liste des images qui obtiennent une valeur de correspondance élevée. Cette liste est généralement triée par ordre de valeur de correspondance décroissante – c'est-à-dire du plus pertinent au moins pertinent – et présentée à l'utilisateur.

Cette phase commence par la réception de la requête de l'utilisateur. Une fois que la requête est formulée, le système la récupère et la transmet vers une composante d'interprétation des requêtes.

Elle est traduite dans une représentation interne définie par le modèle de requête du système, pour qu'il puisse la comparer avec les index des images de la base.
Dès que la requête est traduite en représentation interne, le système calcule le degré de similarité entre chaque image et cette requête. La fonction de correspondance est un formalisme défini pour évaluer cette similarité.

Et enfin le système présente les résultats : Les images retrouvées par le système sont triés par ordre de pertinence décroissante, et présentés à l'utilisateur. La fonction de correspondance a donc pour rôle de favoriser la sélection des images les plus pertinentes pour l'utilisateur.

Éventuellement, le système peut proposer à l'utilisateur de lui indiquer parmi les résultats les documents qu'il estime pertinents et ceux qu'il estime non pertinents, afin de retourner de nouveaux résultats en fonction des indications et choix de l'utilisateur. Ce processus interactif, appelé **bouclage de pertinence**, a pour but d'affiner la recherche, d'améliorer la qualité des résultats et de diminuer le temps d'extraction des résultats. Il est aussi possible d'avoir recours à des processus d'expansion automatique des requêtes, dont nous donnons deux exemples : l'un est basé sur l'utilisation de thésaurus construits automatiquement et l'autre utilise un thésaurus construit manuellement.

La composante du système où l'usager formule son besoin d'informations et le système affiche les réponses est l'interface. Son rôle principal est de permettre l'interaction entre l'utilisateur et le système *[Dio05]*.

3.3 Le modèle de représentation d'images :

Est aussi appelé langage d'indexation. Il exprime le contenu sémantique des images dans un formalisme de représentation des connaissances. La définition de ce formalisme est critique, car elle détermine la qualité de la représentation interne des images, et donc la qualité de la recherche et des résultats (la performance du système). Si les images sont bien représentées, la recherche devient plus efficace et la pertinence du système approche mieux celle de l'utilisateur.

L'essentiel est de maintenir une représentation compacte afin d'éviter une augmentation indésirable du coût de calcul ; en même temps il faut qu'elle soit assez expressive pour décrire le contenu des images précisément.

3.4 Le modèle de requête :

Permet l'expression (contenu sémantique) du besoin de l'utilisateur. Il s'agit de la définition d'un langage formel pour la formulation des requêtes.

Ce langage doit être expressif et il doit prendre en compte le contexte d'application du système. Il est important de considérer le domaine de connaissances traitées par le système ; en même temps le niveau de connaissances des utilisateurs doit aussi imposer des contraintes sur le modèle de requêtes. Le système doit traiter des requêtes complexes et simples selon le niveau de l'utilisateur (expert/novice).

3.5 La fonction de correspondance :

Entre une requête et une image définit de manière formelle le degré de similarité entre les modèles de requête et de représentation d'images. Elle évalue la pertinence de chaque image pour une requête.

3.6 Le modèle de connaissances :

Du domaine considéré décrit un espace thématique couvert par les images considérées. Dans ce contexte se formulent les requêtes et le contenu sémantique des images est représenté. Ce modèle prend aussi en compte les connaissances externes qui peuvent enrichir le nombre de réponses du système, par exemple en incluant un thésaurus composé des termes apparaissant dans l'ensemble des documents, reliés entre eux par des liens de généricité/spécificité ou de synonymie. Par exemple, une requête portant sur "voiture" pourra retourner des documents contenant le terme "automobile".

L'utilisation d'un thésaurus permet d'augmenter le nombre de réponses du système, en incluant dans le résultat des documents contenant des mots reliés aux mots de la requête.

Partie 4 : Les différentes approches pour la représentation d'image :

La recherche d'images se heurte à deux difficultés majeures *[Chb01]:*

a) La description de l'image :

La performance d'un système de recherche d'images est fortement liée à la méthode de description de l'image. Pour la même image, deux types de description sont possibles :

- ***La description objective :*** *représente* l'image par le biais de ses caractéristiques physiques comme sa couleur, sa texture, ses formes, etc.

- ***La description subjective :*** *représente* les éléments (objets) de l'image tels qu'ils sont perçus par l'humain. En d'autres termes, il s'agit de l'interprétation humaine de l'image.

b) Les moyens proposés pour permettre à l'utilisateur d'exprimer sa requête :

Traduire les besoins exprimés par l'utilisateur n'est pas une tâche facile. Plusieurs facteurs interviennent :

- ***L'ambiguïté du besoin*** exprimé par l'utilisateur (est-ce qu'il s'agit de rechercher une image précise ou un ensemble d'images ?),

- ***L'interface Homme-Machine*** (mots-clé, iconique, à base de formulaires, etc.)

- ***Le langage d'interrogation*** (le pouvoir d'expression, la pertinence, etc.).

Notre travail de s'inscrit dans la problématique de la recherche d'images.

Comme le modèle de représentation d'image a une influence directe sur tous les composants d'un SRI (modèle d'indexation, d'interrogation, et fonction de correspondance), nous nous intéressons particulièrement aux approches de représentation d'image actuelles [2].

Il y a deux approches qui correspondent à deux vues différentes de la description de l'image.

---

[2] *Nous présentons quelques systèmes existants dans l'annexe B.*

I. *Première approche : Trois classes de représentation sont utilisés pour la description de l'image [Chb01]:*

- ***La classe orientée-contexte :*** regroupe les systèmes qui traitent l'image comme une boite noire sans aborder son contenu. Il s'intéresse à l'environnement global de l'image : son auteur, sa date d'acquisition, etc. Par exemple, "cette image a été prise par le Dr *Robert en juin 99"* est une description admise par le paradigme orienté-contexte.

- ***La classe orientée-contenu :*** concerne la manipulation du contenu physique de l'image par le biais de ses caractéristiques physiques comme la couleur, la texture, la brillance, les formes, etc. « *Rechercher les images contenant une couleur donnée et une brillance spécifique à telle position* » est un exemple typique des requêtes du paradigme orienté-contenu.

- ***La classe orientée-sémantique :*** se focalise sur l'interprétation de l'image. Il décrit les objets qui composent l'image et leurs relations comme ils sont définis et perçus par l'être humain. « *Rechercher toutes les images contenant un bateau* » est un exemple de requête orientée-sémantique.

  Les systèmes actuels sont élaborés en fonction des besoins liés à chaque domaine d'application.

II. *Deuxième approche : Deux classes de représentation pour la description de l'image [Dio05]:*

- ***Catégorie Signal (bas niveau) :*** les SRIm de cette catégorie, utilisent uniquement le contenu visuel pour indexer et rechercher une image. Le niveau de description est proche du signal (couleurs, textures, formes, positions).

- ***Catégorie sémantique automatique:*** dans ce cas le niveau de représentation est plus élevé. Une image est décrite par des termes symboliques qui expriment sont contenu sémantique.

- Dans la suite nous présentons chaque approche et leurs catégories, avant de les comparer à la fin du chapitre.

4.1 Le contexte et la classe orientée-contexte :

Définition :

La classe orientée-contexte regroupe l'ensemble des systèmes qui utilisent le contexte comme moyen de description d'image. Par définition, Le contexte concerne *l'ensemble des informations autour d'une image permettant d'orienter sa signification ou de la situer par rapport à un fait (ou à un évènement)*. On peut distinguer *le contexte de l'utilisateur* et *le contexte de l'image*. Le contexte de l'utilisateur est déterminé selon un ensemble de variables dépendant de chaque individu tels que son vécu, ses connaissances, ses souhaits, etc. Le contexte de l'utilisateur est un facteur déterminant dans la perception de l'image par chaque individu. Par exemple, la photo de la Figure 1.3 peut être interprétée par un utilisateur comme étant celle d'une montagne, alors qu'un autre utilisateur l'interprèterait comme étant celle du "Lac du Passet".

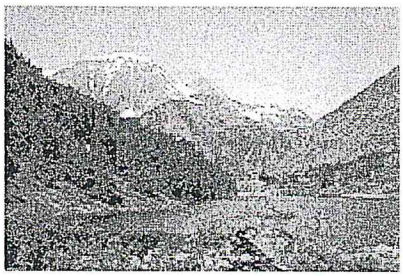


Figure 1.3 : Lac du Passet

Le contexte de l'image regroupe tout élément qui accompagne l'image et peut influencer son interprétation comme une bande-sonore ou un texte. La Figure 1.4 prise par un journaliste peut être interprétée comme une cascade dans un film si on ignore le texte qui l'accompagne.

Il s'agit en réalité d'une embuscade pendant la guerre au Liban.

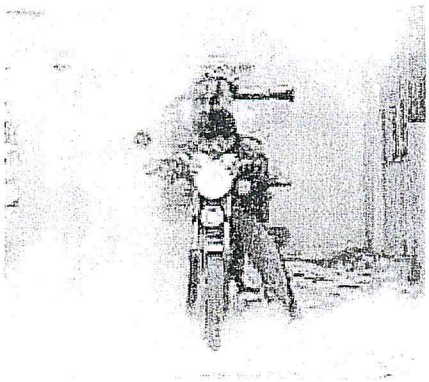


Figure 1.4 : Exemple qui montre la difficulté de décrire une image sans tenir compte de son contexte

Les systèmes qui utilisent le contexte uniquement comme moyen de recherche et d'indexation ne sont pas nombreux. Le contexte est souvent adopté par les moteurs de recherche d'images sur le WEB.

4.2 Le contenu physique et la classe orientée-contenu :

4.2.1 Définition :

La classe orientée-contenu (ou la classe par abstraction) regroupe l'ensemble des systèmes qui proposent de décrire les caractéristiques physiques de l'image. Il est basé sur un ensemble de mesures mathématiques comme les histogrammes, les textures, la distribution des couleurs, les formes, la brillance, etc.

L'utilisation de ce paradigme a permis une nouvelle souplesse et une indépendance certaine vis-à-vis du type de l'image et du domaine d'application.

Le paradigme orienté-contenu est souvent utilisé dans les domaines où la quantité et l'hétérogénéité des images acquises sont importantes (les images satellites, les images diffusées à la télévision, etc.).

Le coût de description manuelle des images a également joué un rôle déterminant en faveur de ce paradigme.

Les techniques utilisées pour calculer les caractéristiques physiques d'une image, étant basées sur des mesures mathématiques, permettent d'automatiser la procédure de recherche de l'image soit en calculant une distance de similarité entre deux images, soit entre deux portions de l'image, soit en comparant la position de deux objets dans une image. L'utilisateur peut donc comparer une image à l'ensemble des images de la base et trouver celles qui sont les plus pertinentes.

4.2.2 Méthodes de description et de recherche d'images par le contenu physique

Les recherches basées sur le contenu physique de l'image peuvent porter sur ses caractéristiques physiques globales et locales. Les caractéristiques physiques globales concernent l'image entière alors que les caractéristiques physiques locales concernent ses objets. Pour décrire les caractéristiques physiques locales, la distribution des couleurs, les histogrammes des couleurs, la texture, la transformée de Fourrier, etc. sont utilisées.

L'utilisation des caractéristiques physiques globales possède différents avantages :

- L'interrogation des caractéristiques globales de l'image est mieux adaptée aux requêtes qui touchent à la globalité de l'image. "Trouver les images dont le rouge est la couleur dominante" est un exemple type de requête basée sur les caractéristiques physiques globales de l'image.

- Les caractéristiques physiques globales de l'image sont utilisées lors de la détection des images (ou frames) dans une vidéo afin d'identifier les changements de scènes.

- L'extraction des caractéristiques physiques globales nécessite un temps de calcul faible.

L'inconvénient majeur de cette utilisation concerne le nombre important d'images non pertinentes obtenu lors d'une requête basée sur des caractéristiques physiques globales.

Les caractéristiques physiques locales de l'image portent quant à elles sur la forme des objets, leurs bordures, la disposition des couleurs etc.

4.3 *Le contenu sémantique de l'image et la classe orientée-sémantique* :

4.3.1 Définition

Le paradigme orienté-sémantique (ou le paradigme par classification) tente de donner un sens à l'image telle qu'elle est perçue par l'être humain. Il correspond à la description subjective de l'image traduite par des moyens textuels. Or, comment pouvons-nous interpréter une image de manière universelle ? Chaque image a un sens qui dépend de l'individu et qui peut varier en fonction du temps. Par exemple, l'image du président Bouteflika jeune peut être perçue et interprétée comme celle du futur président de l'Algérie ou tout simplement comme la photo d'un enfant. Cela reflète les difficultés rencontrées lors de la description subjective d'une image. Pour pallier ce problème de subjectivité, une des solutions proposées consiste à subdiviser l'image en plusieurs objets sémantiques. La description des objets est souvent effectuée manuellement (ou semi-automatiquement) lors du stockage.

4.3.2 Méthodes de description du contenu sémantique de l'image :

Les méthodes de description du contenu sémantique de l'image sont nombreuses.

Elles sont basées sur la perception et l'interprétation humaine. Elles peuvent être classées en deux catégories **: les méthodes textuelles et les méthodes visuelles.**

La méthode textuelle la plus commune est celle des mots-clés. La plupart des systèmes de recherche d'images (SRIm) s'en servent pour la description d'une base. Pour décrire une image, une personne, souvent experte dans le domaine, définit l'ensemble des mots-clé qu'elle juge pertinents.

Cependant, Le mot « *image* » seul, sera inadapté. Pour éviter ce problème, on a recours à la logique booléenne dans l'expression des requêtes. La logique booléenne fournit plusieurs connecteurs de mots-clés : ET, OU, NON, et PRÈS ou COMME.

Certains SRI permettent la recherche des images *via* des langages naturels. Cette méthode est souvent suivie par un ensemble de procédures d'analyses linguistiques servant à normaliser les termes utilisés et à extraire des mots-clés afin d'effectuer le calcul de similarité entre les descripteurs des images.

La deuxième catégorie de méthodes de description du contenu sémantique de l'image est celle de ***la description visuelle***. Pour décrire une image, l'utilisateur utilise plusieurs représentations visuelles (métaphores) comme des icônes, des images de comparaison, des outils de dessin, etc. Chaque représentation visuelle est associée à un concept dans le domaine d'application. Par une simple action (déplacement d'une icône, dessin d'une forme, choix d'une couleur, etc.) de la part de l'utilisateur, le contenu sémantique de l'image est décrit et transformé en un ensemble de descripteurs textuels.

4.4 Catégorie signal :

Les SRIm qui appartiennent à cette catégorie considèrent les caractéristiques élémentaires des images afin de décrire une image. Ces caractéristiques (comme couleurs, texture, formes) sont extraites par des techniques de vision par ordinateur et sont utilisées pour les processus d'indexation et de recherche d'images. Un des premiers SRIm de l'approche signal est le système QBIC [3]. Dans les années qui ont suivi les SRIm d'approche signal ont commencé à s'orienter vers la recherche sur le Web, on a le système Web-WISE. 10

4.5 Catégorie sémantique automatique :

L'approche sémantique automatique pour l'extraction des caractéristiques d'une image intègre une sémantique associée aux pixels des images, par l'utilisation de mots clés qui décrivent les concepts de l'image. Des formalismes plus riches que les mots clés ont été développés pour créer un réseau sémantique associé au contenu de l'image et pour mieux représenter les caractéristiques de haut niveau de l'image.

[3] *Voir l'annexe B.*

4.6 Discussion sur ces descriptions :

La classe orientée-contexte :

Le contexte de l'image peut parfois servir à décrire l'image. Il est ainsi primordial dans certains types de requêtes. Néanmoins, la description grâce au contexte présente certaines lacunes et limites:

- ✓ Les besoins des utilisateurs dans certains domaines nécessitent des modes de recherche différents. Dans la situation actuelle, le contexte seul est incapable de décrire les images de façon à satisfaire les requêtes des utilisateurs concernés.

- ✓ Le contexte n'existe pas nécessairement dans certains domaines.

la classe orientée-contenu - Catégorie signal :

Malgré le réel succès actuel du paradigme orienté-contenu, le problème de la recherche d'images reste ouvert. La recherche des images par les caractéristiques physiques demeure en effet inappropriée dans plusieurs domaines.

Le problème majeur que rencontrent les SRIm basées sur les caractéristiques physiques vient de l'absence totale de sémantique dans la représentation des images et des requêtes.

Par conséquent, l'évaluation de la similarité est uniquement d'ordre visuel. Cette similarité visuelle définit la pertinence système, et c'est donc à l'utilisateur de s'adapter au niveau de représentation du système afin d'exprimer son besoin d'information. Celui-ci exige de sa part une connaissance très précise de ce qu'il cherche, ce qui n'est pas toujours le cas. Souvent l'utilisateur ne sait pas exactement ce qu'il cherche et surtout, il ne doit pas être obligé de savoir des détails sur les attributs visuels d'une image, comme par exemple le pourcentage d'une couleur.

Imaginons cet exemple : ***un utilisateur qui recherche des images de coucher de soleil sur la mer avec le système VisualSEEk. Il doit dessiner un rectangle bleu dans la partie basse pour la mer, un rectangle orange dans la partie haute pour le ciel, et un disque rouge/orange au centre pour le soleil couchant. Il peut également préciser les textures pour chaque région dessinée, il est possible que le système lui retourne une image représentant une orange sur une table bleue sur fond orange, qui est visuellement similaire au croquis qu'il a dessiné, mais sémantiquement différente de l'image qu'il recherche.***

Cette limitation de l'expressivité des requêtes est due à la polysémie des représentations visuelles, c'est-à-dire à leur ambiguïté : deux représentations visuelles peuvent être similaires au niveau signal, tout en dénotant des sémantiques très différentes.

Alors, lorsque l'on s'intéresse au média image, la définition du modèle de représentation de document pose davantage de problèmes que pour le texte, car la sémantique n'est pas intrinsèquement exprimée dans les pixels des images. Le domaine de la Recherche d'Image (RIm) doit donc faire face à une caractéristique importante des images qui est le manque d'un langage de représentation explicite permettant d'en exprimer la sémantique. Cet écart entre les pixels et leur signification est appelé le **fossé sémantique**.

Smeulders décrit ce fossé sémantique de la manière suivante :

"Le fossé sémantique est le manque de concordance entre l'information que l'on peut extraire des données visuelles et l'interprétation des mêmes données qu'en fait un utilisateur dans une situation donnée."[Mar04].

Pour répondre aux attentes des utilisateurs vis-à-vis de systèmes de gestion d'images, les approches symboliques intègrent une sémantique dans la représentation des images et l'interrogation du système, en élevant ainsi leur niveau d'abstraction à celui des utilisateurs.

Partie 5 : Langages de recherche d'images :

Dans cette section, nous étudions les langages d'interrogation proposés afin d'identifier ceux qui sont les plus pertinents aux utilisateurs.

Rechercher une image n'est pas une tâche facile. Pour trouver les images qui répondent à ses besoins, l'utilisateur dispose de plusieurs méthodes ***[Chb01]***.

5.1 Langages textuels :

5.1.1 Langages naturels :

Ces langages permettent à l'utilisateur de formuler sa requête par une phrase dans une langue donnée.

La requête de l'utilisateur est analysée en plusieurs phases.

EMIR² [4], par exemple, propose de formuler les requêtes par le biais d'un texte dans un langage naturel. La fonction de correspondance utilisée dans EMIR² fait appel à une procédure qui fait partie de sa première version, RIME, et qui permet de transformer la requête en un ensemble de descripteurs.

5.1.2 Stuctured Query Language (SQL) :

L'intégration de l'image aux systèmes de gestion des bases de données relationnelles a donné naissance à une panoplie d'extensions de SQL : PSQL, MAPQUERY, GeoSabrina, etc. Ces extensions avaient pour objectif d'adapter le langage SQL aux caractéristiques de l'image exploitées par l'utilisateur dans ses requêtes telles que les caractéristiques physiques et les relations spatiales entre les objets. Le projet de normalisation SQL3 a démarré en août 1994 dans le but d'intégrer l'aspect spatial à la syntaxe du langage.

---

[4] *Voire l'annexe B*

5.2 Langages graphiques :

Une autre génération de langages d'interrogation est apparue : *les langages graphiques*. Souvent basés sur SQL ou une des ses extensions.

Ces langages sont classés selon trois catégories : les langages tabulaires, les langages à base de formulaires, les langages schématiques.

5.3 Langages visuels :

Ils sont appelés visuels car ils incorporent des éléments visuels dans la formulation des requêtes : icônes, dessins, etc. L'utilisateur présente au système l'image à rechercher soit en dessinant des formes (sketch), soit en choisissant des icônes (métaphores), soit en donnant une image similaire à celle(s) voulue(s). Dès lors, l'utilisateur peut formuler sa requête en créant, déplaçant, modifiant, ou supprimant les éléments visuels. Le résultat de son action est immédiat.

Des possibilités d'annuler ou d'initialiser certaines manipulations voire de reformuler la requête en fonction du résultat obtenu (relevance feedback) sont également proposées.

5.4 Hypermédia :

Introduit par Bush, l'hypermédia est un moyen d'interrogation dépourvu de langage d'interrogation. Chaque document est associé à un ensemble de liens qui le relie à d'autres documents. A l'aide des liens, appelés hypertextes ou hypermédias, l'utilisateur explore un ensemble d'informations *hétérogènes* (texte, image, vidéo, audio, etc.) organisées en structure logique hiérarchisée. Un **Système Hypermédia** est conçu pour conduire l'utilisateur *via* des chemins thématiques à trouver les informations qui lui sont appropriées. On parlera donc de la notion de *navigation* dans ce genre de système. Elle consiste en effet à détailler un sujet précis en passant par toutes sortes de médias.

Partie 6 : Visualisation et présentation des résultats :

La *présentation* est le calcul par le système de la manière dont il va proposer à la visualisation de l'utilisateur les documents retrouvés, soit directement, soit en lui donnant accès aux identifiants

L'affichage des résultats d'une requête est l'un des problèmes majeurs pour un moteur de recherche.

Lorsque l'on envisage l'accès à de grandes masses d'informations, il faut également traiter de la présentation des résultats. De manière générale, les travaux sur la visualisation de grandes masses de données ont été réalisés principalement dans le domaine des interfaces homme-machine même s'il y a des travaux spécifiques aux systèmes de recherche d'information. Les problèmes dans la taille des *corpus* dans un système de recherche d'information rendent plus difficile l'évaluation et donc la perception des bonnes réponses dans un espace devenu plus vaste. Les études liées à la présentation de grandes masses de données doivent donc s'adapter à la recherche d'information ***[Bou09].***

Depuis ces dernières années, les moteurs de recherche, ou plus exactement les méta moteurs de recherche commerciaux en ont pris conscience et proposent des interfaces de présentation des résultats de plus en plus interactives et visuelles, ne se contentant plus d'une simple liste de réponses.

Et comme le souligne Cugini dans, toute technique de visualisation requiert de l'apprentissage de la part de l'utilisateur afin qu'il puisse l'utiliser efficacement. Chaque technique peut ainsi s'adapter différemment en fonction du public visé dans l'application de recherche d'information.

Le type de donnée à afficher est prépondérant dans le choix de la représentation graphique : une visualisation adaptée au choix de quelques documents ne le sera plus forcément lorsque le nombre de documents deviendra plus important.

Et pour terminer, une visualisation doit forcément être adaptée à l'environnement et aux possibilités techniques de présentation des résultats Nous présenterons dans la suite les différents systèmes de visualisation de résultats ***[Pic04].***

6.1 Présentation par liste de réponses (1Dimension) :

La forme la plus simple et aussi la plus courante de présentation des résultats d'un moteur de recherche est la liste, en (1D), cette liste est composée de liens, ces derniers peuvent être accompagnés d'un résumé décrivant le contenu de chaque page. Cette présentation, souvent textuelle, est celle employée par tous les moteurs de recherche.

Cependant, ce type de présentation très simple oblige l'utilisateur à ne s'intéresser qu'aux tout premiers résultats comme l'ont confirmé toutes les études statistiques réalisées sur l'utilisation des plus grands moteurs de recherche.

6.2 *Visualisation graphique de résultats* :

La représentation de données de systèmes d'information sous forme graphique présente l'avantage de faire passer beaucoup d'informations très rapidement à l'utilisateur.

Ces représentations peuvent intervenir en deux, trois et même quatre dimensions en prenant en considération la notion de temps. Les travaux dans ce domaine sont nombreux et variés et semblent avoir un réel avenir prometteur dans le domaine de la recherche d'information.

Il n'existe par conséquent pas de représentation idéale de résultats pour un moteur de recherche, mais un concepteur de tels systèmes doit garder à l'esprit la simplification des informations à transmettre.

Partie 7 : Mesures d'évaluation :

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.

Alors, la qualité d'un SRI réside dans son aptitude à retrouver l'ensemble des images pertinentes pour l'utilisateur. On distingue deux types de pertinences : la *pertinence utilisateur* qui correspond aux images proposées par le système et jugés pertinentes par l'utilisateur (Relevance Feedback), et la *pertinence système* qui correspond aux images jugés pertinentes par le système ***[Chb01]***.

Le but de tout SRI est rapprocher la pertinence système de la pertinence utilisateur.

Ainsi, la *pertinence système* et la *pertinence utilisateur* peuvent différer quand une image correspond -- du point de vue du système -- parfaitement à la requête, tandis que l'utilisateur peut n'en avoir que faire (par exemple parce qu'il en connaît déjà parfaitement le contenu). L'utilité d'une image pour l'utilisateur ne peut être mesurée qu'à travers les jugements que celui-ci émet lorsque le SRI lui présente celui-là. Elle dépend naturellement du contexte, *i.e.* de facteurs aussi variés que le but poursuivi par l'utilisateur ou que le contexte socioculturel dans lequel est menée la recherche ***[Pri99].***

Pour mesurer les performances ***qualitatives*** d'un SRI, Salton distingue cinq critères ***[Chb01]*** :

- L'approbation des utilisateurs,
- Le temps de réponse : représente le temps écoulé entre la soumission de la requête et l'obtention du résultat,
- La présentation du résultat,
- La capacité du système à donner principalement de bonnes réponses à une requête,
- La capacité du système à éliminer les mauvaises réponses.

Plusieurs mesures ***quantitatives*** sont également utilisées pour mesurer les performances du SRI (figure 1.5) :

– ***Le bruit (Digression)***: Réponse non pertinente (selon l'utilisateur) fournie lors d'une recherche d'information.

– ***Le silence*** : Désigne l'ensemble des documents pertinents non retrouvés lors d'une recherche.

– ***Le rappel (Exhaustive)*** : Capacité du système à fournir en réponse **tous** les documents pertinents de la collection.

– ***La précision (Spécificité):*** Capacité du système à ne fournir **que** des documents pertinents en réponse.

Ces deux critères sont antagonistes dans la réalité … ***[Mul]***

Evaluation Précision-Rappel :

Après avoir crée notre corpus de documents nous avons établi un ensemble de requêtes dans le but de tester notre système. 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, nous avons obtenu la liste des documents répondant à chacune de nos requêtes. La méthode que nous utilisons est basée sur des calculs de précisions et de rappels, définis comme ceci: ***[Mul]***

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

– Limiter le silence

$$rappel = \frac{Nombre\ de\ documents\ pertinents\ séléctionnés}{Nombre\ total\ de\ documents\ pertinents}$$

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

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

–Limiter le bruit

$$précision = \frac{Nombre\ de\ documents\ pertinents\ \ sélectionnés}{Nombre\ total\ de\ documents\ sélectionnés}$$

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

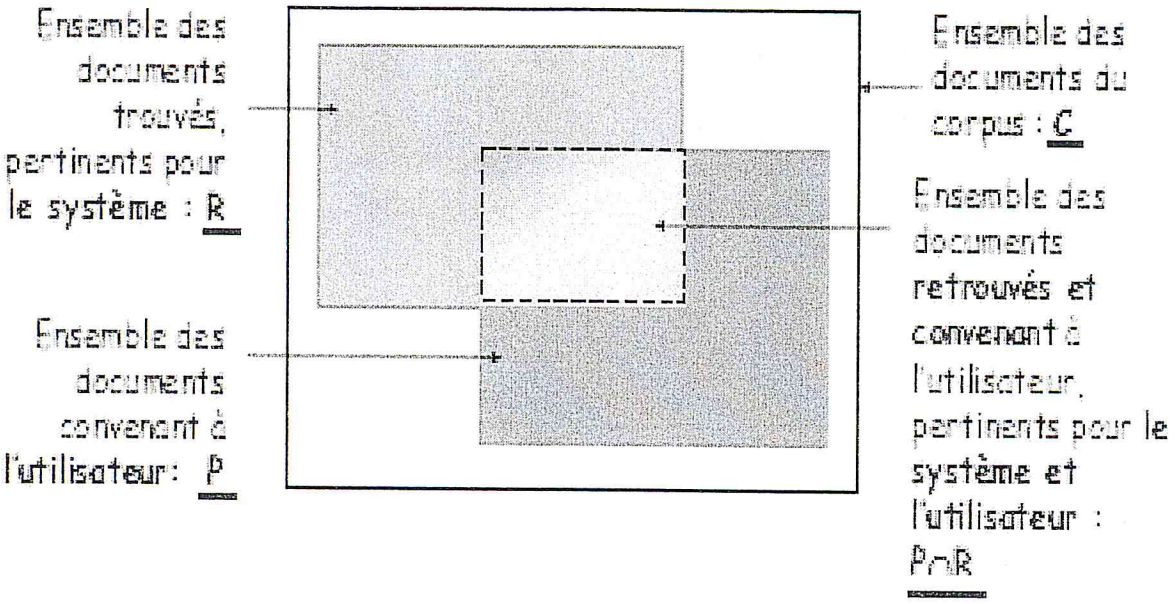


Figure 1.5 : Critères d'évaluation d'un SRI

Comment choisir le meilleur compromis lorsque la précision et le rappel sont pratiquement d'égale importance ? Une des méthodes utilisées est de maximiser la moyenne harmonique de la précision et du rappel : $\frac{r+p}{2rp}$. On appelle cette moyenne le *score F* ***[05]***.

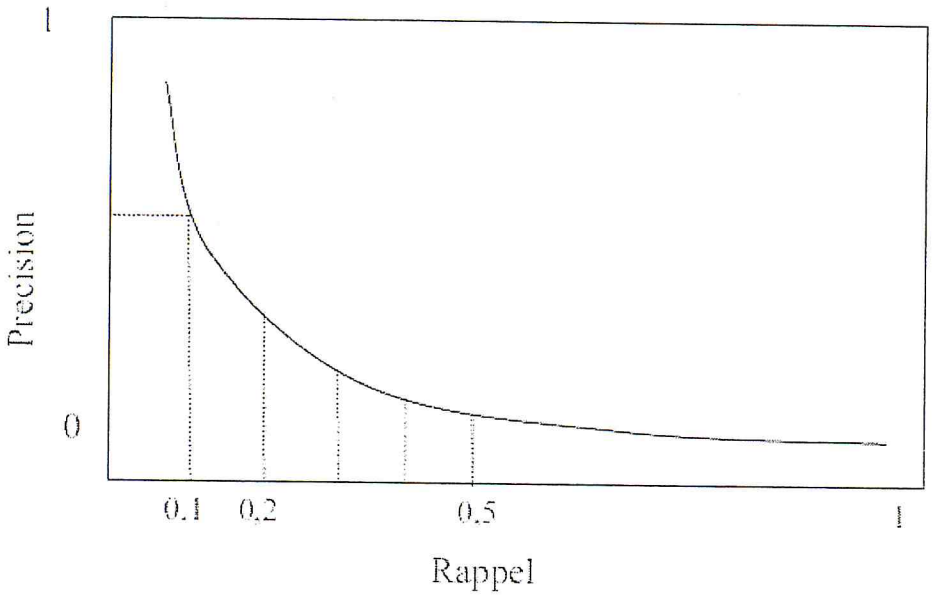
Idéalement, on voudrait qu'un système donne de bons taux de précision et de rappel en même temps. En effet, un système qui aurait 100% pour la précision et pour le rappel signifie qu'il trouve tous les documents pertinents (rappel) et rien que les documents pertinents (précision). Cela signifie que les 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 ***[Ann08]***.

En d'autre terme, 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.

7.3 Courbes de rappel/précision :

En pratique, on cherche un bon compromis entre le rappel et la pertinence. Afin d'évaluer un système, on fait souvent un graphique du rappel par rapport à la pertinence (ou vice versa).



Précision moyenne : une seule valeur reliant le rappel et précision

Figure 1.6 : Courbes de rappel/précision

– Représente l'évolution de la précision et du rappel avec des résultats triés

– Méthode :

Pour chaque document retrouvé, on calcule la précision et le rappel obtenus en considérant seulement le premier document comme réponse, puis les deux premiers, puis les trois premiers etc., jusqu'à la réponse totale du système ***[Mul]***.

Conclusion :

Nous avons évoqué dans ce chapitre la problématique générale de la recherche d'images.

Après avoir exposé les différentes approches de description d'images proposées, nous avons effectué une analyse des différents langages d'interrogation d'images.

Quatre types de langages d'interrogation d'images ont été successivement abordés : *textuels, graphiques, visuels et hypermédia.*

Ensuite, on a présenté un état de l'art sur la visualisation des résultats.

Et on a terminé cet état de l'art par la définition des mesures d'évaluation des SRI.

Problématique :

L'idée principale que l'on en retient est que les approches signal ne permettent pas aux utilisateurs de tels systèmes de rechercher des images selon la sémantique qu'elles expriment.

Les langages graphiques, visuels et hypermédia sont complexes et nécessitent une expérience de la part de l'utilisateur, ce qui n'est pas le cas pour beaucoup d'utilisateurs car ils n'ont pas exactement des caractéristiques sur les images qu'ils veulent rechercher et ils ne savent pas transformer leur besoin à une requête complexe.

Solution :

Notre solution va se baser sur l'approche sémantique automatique pour faire la recherche d'images, car nous considérons que cette approche offre une description plus riche et plus détaillée de l'image, et permettent d'interroger le système de manière sémantique. Parallèlement elle rend l'interaction avec le système plus facile pour les utilisateurs puisqu'ils expriment leur besoin d'information textuellement par des termes symboliques.

Le système que nous proposons est BMIR (Boolean Model for Image Retrieval), un modèle booléen pour la recherche des images, ayant pour but de répondre au mieux au critère de spécificité et exhaustivité demandé par l'utilisateur.

1 Introduction :

Nous avons présenté dans ce premier chapitre un état de l'art des travaux pour la recherche d'information. Et comme nous l'avons présenté, les SRIms existants adoptent plusieurs approches pour l'indexation et la recherche des images.

Afin de répondre à certaines limites des approches proposées, Notre cadre de travail va se baser sur l'approche sémantique automatique pour faire la recherche d'images, car nous considérons que cette approche offre une description plus riche et plus détaillée de l'image. Parallèlement elle rend l'interaction avec le système plus facile pour les utilisateurs puisqu'ils expriment leur besoin d'information textuellement par des termes symboliques.

Le système que nous proposons est BMIR (Boolean Model for Image Retrieval), un modèle booléen pour la recherche des images, ayant pour but de répondre au mieux au critère de spécificité et exhaustivité demandé par l'utilisateur.

2 Objectifs :

La nature des images impose de représenter le contenu des documents et des requêtes de manière précise, les objectifs fixés pour la définition du modèle, à la lumière de l'état de l'art présenté dans le chapitre précédent sont :

Le modèle de représentation des images doit pouvoir exprimer leur contenu sémantique, afin que l'utilisateur puisse les retrouver par le biais de cette représentation. Le choix du langage d'indexation doit donc offrir un niveau de représentation en accord avec cette contrainte. Les modèles de documents et de requêtes doivent permettre de manipuler des représentations aussi précises que possible du contenu des images et des requêtes.

Le modèle de correspondance doit supporter le niveau de représentation du langage d'indexation, et doit évaluer une correspondance plus ou moins exacte entre les images et les requêtes, c.à.d. que les résultats doivent être identiques aux besoin de l'utilisateur (termes de sa requête)

Enfin, l'implantation de la fonction de correspondance doit respecter les contraintes de temps de réponse d'un système interactif *[Cle66, Nie94].*

Ces objectifs ont guidé nos choix de représentation, et le système proposé dans le cadre de ce travail vise essentiellement à répondre à ces contraintes.

3 Présentation de notre système :

L'objectif principal de système textuel BMIR (Boolean Model for Image Retrieval) que nous proposons pour la recherche est d'améliorer la qualité de recherche des images. Il est basé sur l'approche sémantique automatique.

Notre système repose sur un modèle de représentation d'images et de requête et un langage d'interrogation, nos chois sont les suivants :

- L'image est annotée par mots clés qui décrivent leur contenu sémantique (on utilise une base d'images annotées en langue anglaise)

- La structure d'index utilisée est le fichier inverse pour que le temps d'exécution soit beaucoup plus rapide (le système n'a qu'à consulter le fichier inversé pour repérer tous les fichiers dans lesquels apparaît le terme de recherche).

- La requête est représentée par un ensemble de mots clés séparés par des opérateurs booléens, ce qui rend l'interaction de l'utilisateur avec le système se faite textuellement, alors plus facile et simple.

- La correspondance image-requête est déterminée en utilisant le modèle booléen, ce qui est relatif au modèle de représentation d'images et de requêtes et cela rend la recherche exacte tout en vérifiant la sémantique.

4 Architecture générale du BMIR :

La figure 2.1 illustre l'architecture générale de l'environnement de BMIR

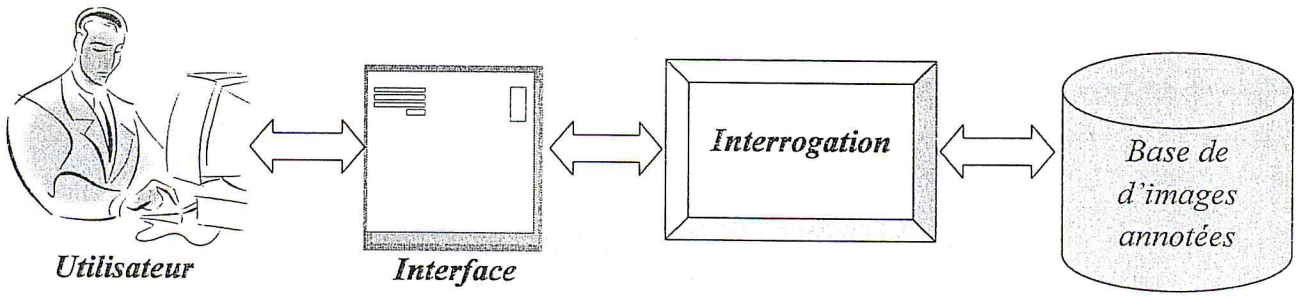


Figure 2.1 : Architecture générale de l'environnement de BMIR

✚ *Le fonctionnement du système :*

Dans notre système, on a plusieurs étapes à réaliser :

- Une étape qui s'intéresse à la génération du tableau inverse ; c.à.d. qu'il faut stocker les mots clés des index (images de la base) dans une structure appelée fichier inverse, cette phase a pour but de préparer le modèle de documents (images) à la recherche.

- Une étape qui s'intéresse à l'analyse linguistique de la requête, cette phase n'est pas simple et a pour but de préparer la requête et d'analyser son langage pour qu'elle soit significative et correspond au modèle d'images de la base afin d'améliorer la correspondance image-requête dans la recherche.

On peut noter que la première étape se fait hors ligne (figure 2.2).

- Une fois le fichier inverse est généré et la requête est analysée nous passons à la phase de la recherche, Cette étape consiste en deux pas principaux ; d'abord il faut déterminer la correspondance entre chaque mot clé de la requête et le fichier inverse et ensuite il faut implanter le modèle booléen, c.à.d. appliquer les opérateurs booléens sur les mots clés retrouvés pour obtenir les résultats pertinents.

- Enfin la dernière étape sera la visualisation, les résultats seront affichés et nous devons présenter les images pertinentes par une liste de réponses en une dimension.

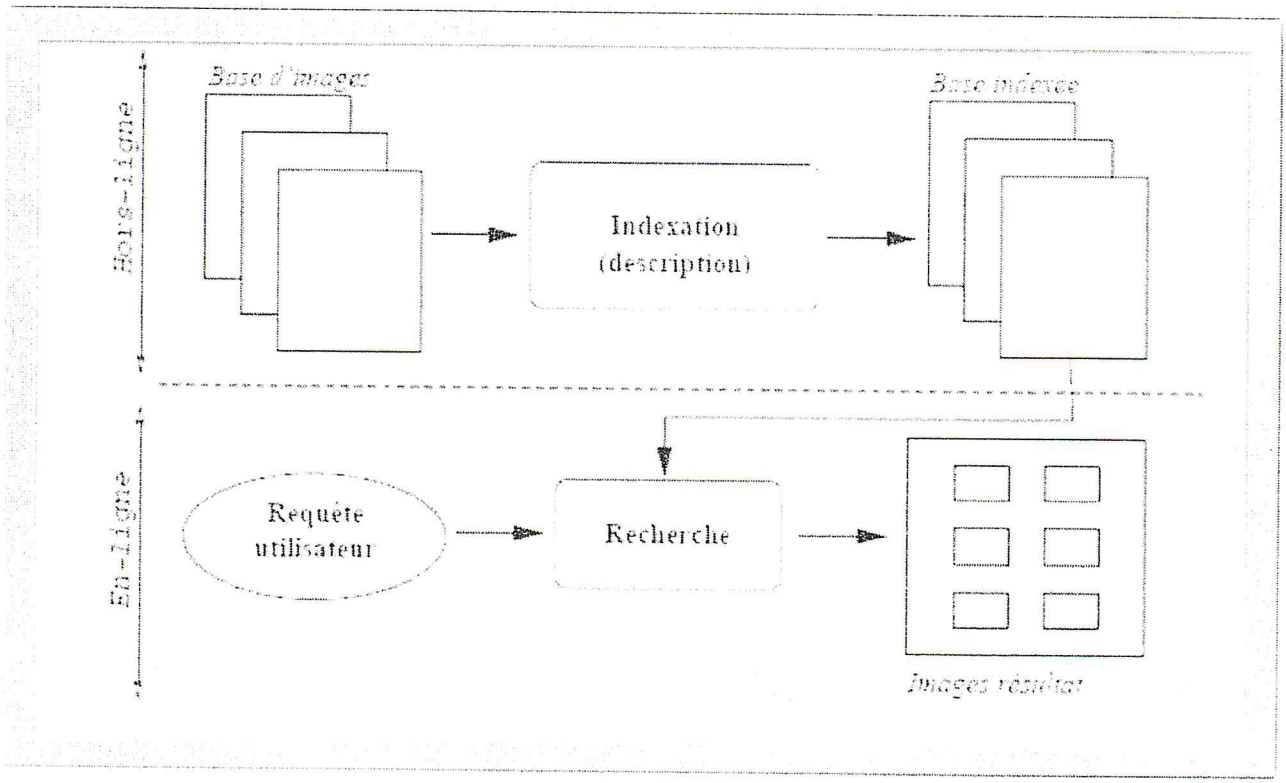


Figure 2.2 : Schéma illustrant le fonctionnement d'un système basique de recherche d'images

Et voici le fonctionnement de la phase « interrogation » ou « recherche » :

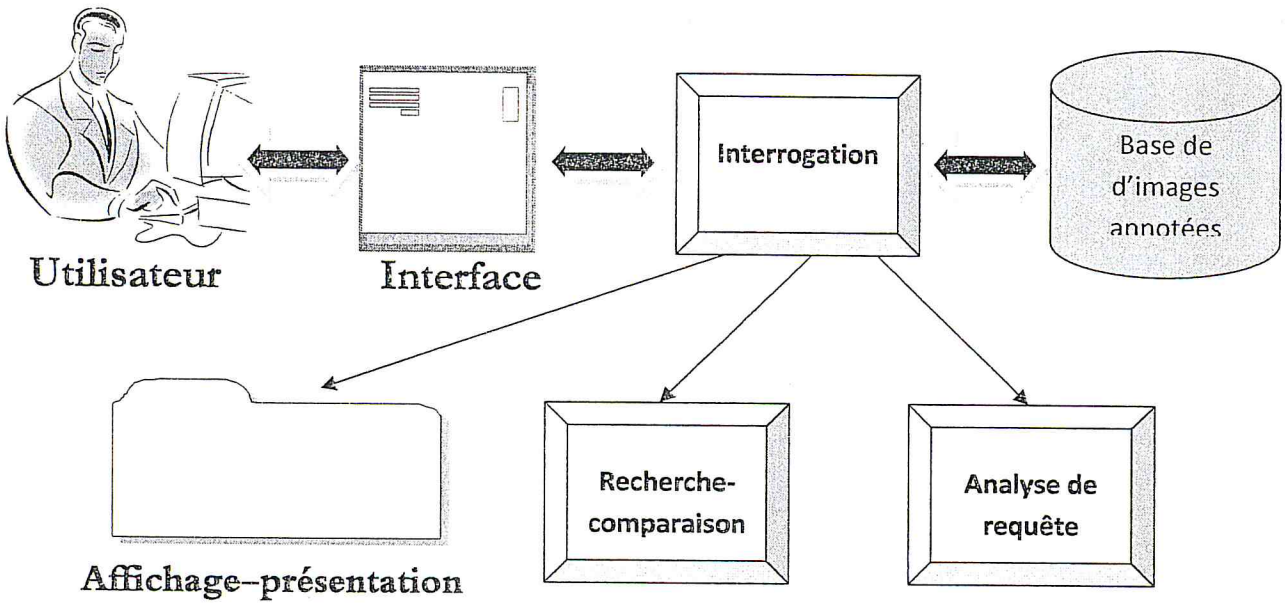


Figure 2.3 : Environnement de l'interrogation dans BMIR

5 Modélisation

5.1 Modèle de représentation des images :

Nous présentons ici la description de notre modèle de représentation des images qui est la base de notre recherche, Il est basé sur le modèle booléen de RI.

5.1.1 Modèle d'image physique :

Nous commençons par définir les éléments de base de notre modèle d'image physique, que nous allons manipuler par la suite. Ces définitions portent sur les images et les mots clés :

✓ ***Image :***

Une image est représentée comme une conjonction logique de mots clés ou termes (non pondérés).

✓ ***Mot clé :***

Un mot clé est une suite de caractères, ce mot clé permet une description d'un élément de l'image.

Plusieurs mots clés peuvent annoter une image, et plusieurs images peuvent être annotées par un mot clé.

Hypothèse (Nombre minimum et maximum de mots clés) *Soit N : le nombre de mots clés qui annotent une image de notre base :*

Une image est annotée par un mot au minimum $N \geq 1$

Une image ne peut pas être annotée par plus de 5 mots clés $N \leq 5$

Soit :

Image : $\mathbf{Im} = t_1$ *AND* ... *AND* t_N

Avec : $2 \leq N \leq 5$

Dans le cas où l'image est annotée par un seul mot clé (terme) :

$$\mathbf{Im} = t_1 \quad (N = 1)$$

Dans les deux exemples de la figure 8,

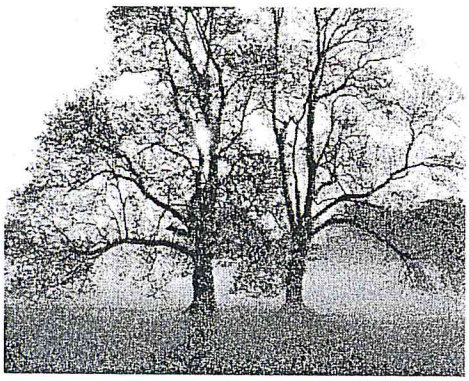


Figure 2.4 : Exemples d'images de la base

On a : N = 4

Les deux images sont représentées par :

Im_1 :

Tree *AND* **Sun** *AND* **Sand** *AND* **Sky**

(t_1 : Tree , t_2 : Sun , t_3 : Sand , t_4 : Sky) (N = 4)

Im_2 :

Tree *AND* **Sun** *AND* **Natur**

(t_1 : Tree , t_2 : Sun , t_3 = Natur) (N =3)

Nous rappelons que toutes les images de notre base sont représentées avec cette forme.

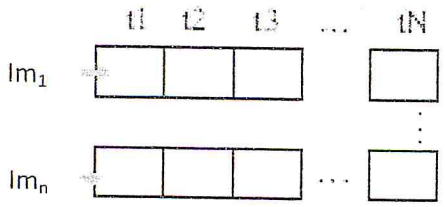


Figure 2.5 : Représentation de la base d'image

❖ ***Contraintes :***

- Chaque image de la base est caractérisée par : son code, son nom et un nombre de mots clés (1 à 5).
- Chaque mot clé est caractérisé par : son code et son libellé.
- On peut ajouter une image à la base, comme on peut la supprimer.

5.1.2 Le fichier inverse [1] :

Les termes (mots clés) des images sont stockés dans une structure appelée fichier inverse.

Dans le fichier inverse, un terme est associé à un ensemble d'images qui contiennent le terme.

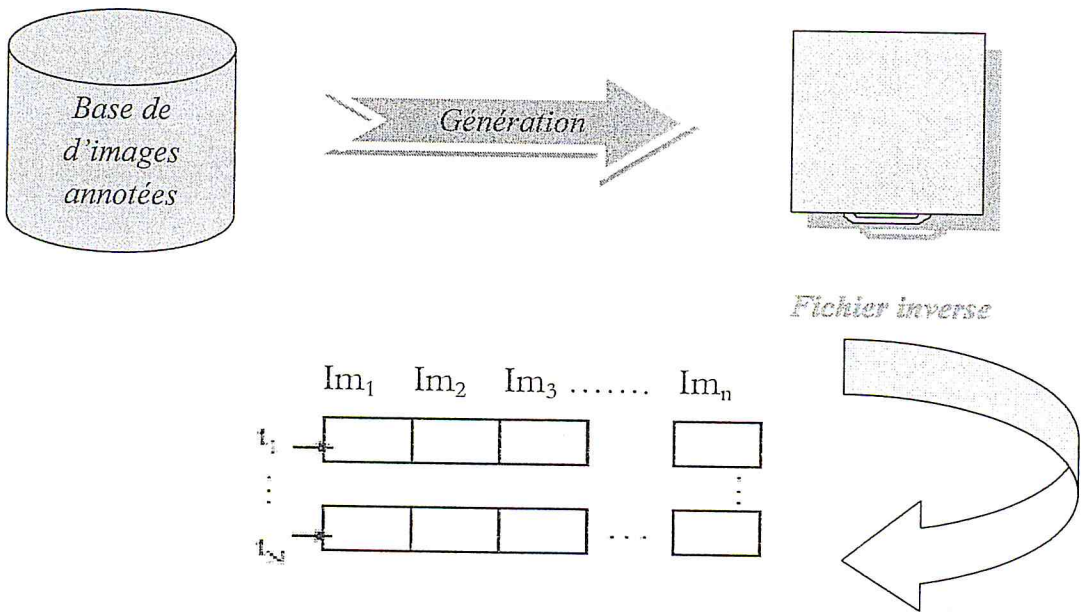


Figure 2.6 : Présentation d'un fichier inverse

[1] *Les fichiers inverses constituent le cœur de la majorité des algorithmes de recherche d'information. Lorsque le nombre de documents augmente, il en est de même pour la liste des termes uniques et les fichiers inverses. Ainsi, le temps de traitement d'une requête croît approximativement linéairement avec la taille de l'ensemble de documents (et croît approximativement linéairement avec la taille de la collection de données indexées). Dans la plupart des méthodes de recherche d'information, la première étape consiste à générer l'ensemble de tous les documents qui contiennent les termes recherchés.*

- Si on prend notre exemple, le fichier inverse qui contient les termes (mots clés) des deux images est le suivant :

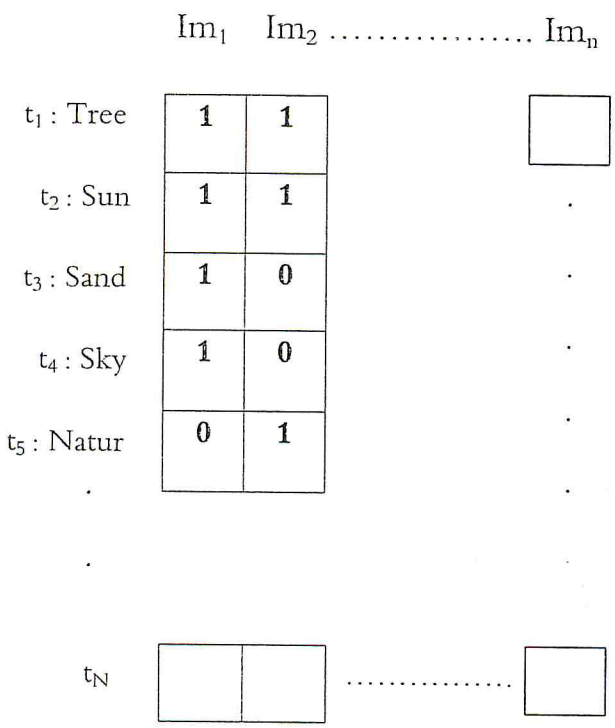


Figure 2.7: Fichier inverse correspondant à Im_1 et Im_2

Cette représentation de fichier inverse nous permet de retrouver pour un terme donné tous les images qui contiennent ce terme.

5.2 Langage d'interrogation :

Nous définissons ici le langage d'interrogation afin de permettre à l'utilisateur d'exprimer son besoin de manière plus ou moins précise, en introduisant ou non des conditions dans les requêtes.

Nous commençons par définir les éléments de base de notre modèle de requêtes :

5.2.1 Modèle de requêtes :

✓ ***Requête :***

Une requête est formulée avec de simples mots-clés reliés par des opérateurs booléens.

✓ ***Mot vide, mot utile :***

Un mot clé de la requête est soit un mot vide, soit un mot utile. Les mots clés considérés sont seulement : les mots utiles, c.à.d. qu'on va éliminer les mots vides saisis par l'utilisateur.

✓ ***Opérateur booléen :***

Un Opérateur booléen est un mot qui relie deux mots clés, et décrit une relation qui peut exister entre ces deux. Il existe trois opérateurs booléens:

"AND":
Utilisez l'opérateur "AND" pour récupérer des images qui contiennent l'ensemble des termes de recherche saisis.

"OR":
Utilisez l'opérateur "OR" pour récupérer des images qui contiennent au moins un des termes de recherche saisis.

"NOT":
Utilisez l'opérateur "NOT" pour éliminer la récupération de certains termes dans la recherche.

- *Syntaxe :* Les opérateurs booléens sont saisis en lettres capitales.

Soit :

Requete: $Q = t_1$ *AND* t_2 *NOT* t_3 .....*OR* t_{NQ} $(N_Q = N , N_{OP} = N-1)$

Dans le cas où la requête contient un seul mot clé (terme) :

$Q = t_1$ $(N_Q = 1, N_{OP} = 0)$

Exemples de requêtes :

Q1:

Tree *AND* **Sand**

On a: $(N_Q = 2, N_{OP} = 1)$
$(t_1$: Tree , t_2 : Sand , OP_1 : AND)

Q2 :

Tree *AND* **Sun** *AND* **Bird**

On a: $(N_Q = 3, N_{OP} = 2)$
$(t_1$: Tree , t_2 : Sun , t_3 : Bird , OP_1 : AND , OP_2 : AND)

Q3 :

Tree *AND* (Sun *OR* Bird)

On a : $(N_Q = 3, N_{OP} = 2)$
$(t_1$: Arbre , t_2 : Soleil , t_3 : Bird , OP_1 : AND , OP_2 : OR)

Q4 :

Tree *AND* Sun NOT (Sky *OR* Bird)

On a : $(N_Q = 4, N_{OP} = 3)$
$(t_1$: Tree , t_2 : Sun , t_3 : Sky , t_4 : Bird , OP_1 : AND , OP_2 : NOT , OP_3 : OR)

Q5 :

Horse

On a: ($N_Q = 1$, $N_{OP} = 0$)
(t_1 : Horse)

Q6 :

The Tree *AND* of **Sand** *NOT* **is Natural**

On a : ($N_Q = 6$, $N_{OP} = 2$)

(t_1 : The , t_2 : Tree , t_3 : of , t_4 :Sand , t_5 : is , t_6 : Natural , OP_1 : AND , OP_2 : NOT)

Et pour notre exemple, on a :

Im_1 correspond à Q1.
Im_1 ne correspond pas à Q2.
Im_1 correspond à Q3.
Im_1 ne correspond pas à Q4.
Im_1 ne correspond pas à Q5.
Im_1 correspond à Q6.
Im_2 ne correspond pas à Q1.
Im_2 ne correspond pas à Q2.
Im_2 correspond à Q3.
Im_2 correspond à Q4.
Im_2 ne correspond pas à Q5.
Im_2 ne correspond pas à Q6.

❖ ***Contraintes :***

- Chaque requête est caractérisée par : son numéro et un nombre de mots clés.
- Chaque mot clé de la requête est soit : un mot vide soit un mot utile.
- Chaque opérateur booléen de la requête est caractérisé par : son type.
- Une requête est refusée si :

- elle ne contient aucun mot clé ($N_Q = 0$).
- On trouve que deux mots clés ne sont pas reliés par un opérateur booléen (N_{OP} ? $N_Q - 1$).

- Chaque requête pas refusée va être analysée.
- Chaque mot vide va être éliminé.
- Chaque mot utile va être normalisé.

5.2.2 Analyse linguistique de la requête :

L'analyse lexicale est le processus qui permet de convertir la requête en un ensemble de termes. Un terme est une unité lexicale ou un radical *[Fox92].* On a trois étapes :

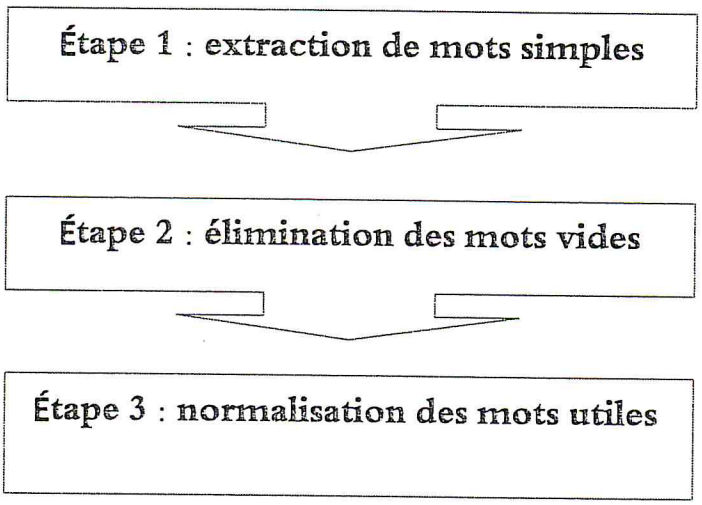


Figure 2.8 : Schéma qui illustre les 3 étapes de l'analyse linguistique

5.2.2.1 Extraction de mots (tockenisation) :

La première étape de l'analyse de requête consiste à extraire les termes significatifs (mots clés + opérateurs booléens).

Ces termes sont des suites de caractères séparées par blanc.

Nous allons mettre ces termes (mots clés) dans un tableau d'une seule ligne de taille : N_Q+N_{OP}

Prenons par exemple la requête Q6 :

The Tree *AND* of **Sand** *NOT* **is Natural**

- Après l'extraction des mots, on a le resultat suivant:

The	*Tree*	*AND*	*of*	*Sand*	*NOT*	*is*	*Natural*
1	2	3	4	5	6	7	N_Q+N_{OP} = 8

5.2.2.2 Elimination des mots vides (stoplist/ Common Wordsremoval)

Cette étape consiste à supprimer les mots « vides » (pronoms personnels, prépositions,...).

Les mots vides sont les Mots trop fréquents mais pas utiles **(selon la loi de Zipf)** [2], comme par exemple : of, the, or, a, you, I, us, …

L'élimination des mots vides a l'avantage évident de réduire le nombre de termes de la requête, elle peut aussi élever le taux de rappel[3], c'est à dire la proportion de documents pertinents renvoyés par le système par rapport à l'ensemble des documents pertinents.

Dans notre exemple de la requête Q6,

The Tree *AND* of **Sand** *NOT* **is Natural**

Les mots vides sont 3 : t_1 , t_3 , t_5 c.-à-d. les termes : **The , of , is.**

- Après l'élimination de ces mots vides, on a le résultat suivant :

Tree	*AND*	*Sand*	*NOT*	*Natural*
1	2	3	4	5

On a: ($N_Q = 3$, $N_{OP} = 2$)

(t_2 : Tree , t_4 :Sand , t_6 : Natural , OP_1 : AND , OP_2 : NOT)

- *Les éléments du nouveau tableau sont seulement : les mots utiles, et la taille du tableau va diminuer et devient :*
  $N_Q+N_{OP}-N_V$ *(N_V : Nombre de mots vides)*

Nombre maximum d'opérateurs : *Soient N_Q : Le nombre de mots clés de la requête, N_{OP} : Le nombre d'opérateurs logiques utilisés pour relier les mots clés de la requête,*
Le nombre d'opérateurs est inferieur au nombre de mots clés d'une unité)
$N_{OP} = N_Q - 1$

[2] *On trouvera cette loi dans l'annexe C.*
[3] *Voire chapitre 2.*

5.2.2.3 Normalisation « Lemmatisation » (radicalisation) / (stemming)

Cette étape consiste à un Processus morphologique permettant de regrouper les variantes d'un mot.

Un mot donné peut avoir différentes formes dans un texte, mais leur sens reste le même ou très similaire. On peut par exemple citer retrieve, retrieving, retrieval, retrieved, retrieves, etc. Il n'est pas forcément nécessaire de prendre en charge tous ces mots alors qu'un seul suffirait à représenter le concept véhiculé qui est dans cet exemple : retriev. Pour résoudre le problème, une substitution des termes par leur racine, ou lemme, est utilisée [4].

Cela va éviter à l'utilisateur de devoir entrer les formes de pluriel des noms ou les formes conjuguées des verbes lors de sa recherche.

Pour cela, on va utiliser l'algorithme de Porter ***[Por80]*** pour éliminer les affixes des mots clés.

- Algorithme basé sur la mesure de séquences voyelles-consonnes :
  - mesure m pour un «stem» est $[C](VC)^m[V]$ ou C est une séquence de consonnes et V est une séquence de voyelles [] = option
  - m=0 (tree, by), m=1 (trouble,oats, trees, ivy), m=2 (troubles, private)
- Algorithme basé sur un ensemble de conditions actions :
  - Old suffix? new suffix
  - Les règles sont divisées en étapes et sont examinées en séquence

[4] *Frakes et Baeza-yates* ***[Fra92]*** *distinguent cinq types stratégiques de lemmatisation : la table de consultation (dictionnaire), l'élimination des affixes (on peut par exemple citer l'algorithme de Porter), la troncature, les variétés de successeurs ou encore la méthode des n-gramme* ***[Ada74].***

- *Algorithme de Porter :*

• *Etape 1: Pluriels et participes passé*

– *SSES* ⟶ *SS* — *caresses* ⟶ *caress*

– *ING* ⟶ — *motoring* ⟶ *motor*

• *Etape 2: Adjectifs*

– *OUSNESS* ⟶ *OUS* — *callousness* ⟶ *callous*

– *ATIONAL* ⟶ *ATE* — *relational* ⟶ *relate*

• *Etape 3:*

– *ICATE* ⟶ *IC* — *triplicate* ⟶ *triplic*

• *Etape 4:*

– *AL* ⟶ — *revival* ⟶ *reviv*

– *ANCE* ⟶ — *allowance* ⟶ *allow*

• *Etape 5: Le « e » et les « doubles lettres »*

– *E* ⟶ — *probate* ⟶ *probat*

– **d et *l* ⟶ *1 seule lettre* — *controll* ⟶ *control*

Algo 1 : Algorithme de Lemmatiation

- Revenons à notre exemple de la requête Q6,

Tree	*AND*	*Sand*	*NOT*	*Natural*
1	2	3	4	5

On a : ($N_Q = 3$, $N_{OP} = 2$)

(t_2 : Tree , t_4 :Sand , t_6 : Natural , OP_1 : AND , OP_2 : NOT)

Si on applique l'algorithme de Porter sur ces mots clés, les mots clés qui vont etre lemmatisés sont : un mot seulement qui est **t_6 : Natural**

D'après l'étape 4 de l'algorithme de Porter :

AL ⟶

Alors : Natural ⟶ Natur

- Après la lemmatisation de ce mot clé, on a le résultat suivant :

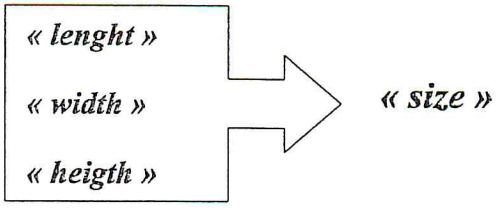
Tree	*AND*	*Sand*	*NOT*	*Natur*
1	2	3	4	5

On a : ($N_Q = 3$, $N_{OP} = 2$)

(t_2 : Tree , t_4 : Sand , t_6 : Natur , OP_1 : AND , OP_2 : NOT)

On peut noter que dans cette partie, le nombre de mots clés ne change pas (La taille du tableau ne change pas)

- ***Modèle de correspondance :*** On a construit un thésaurus qui établie des liens de généricité/spécificité ou de synonymie.
  - *Exemple de généricité/spécificité :* ***« rose »*** ⟷ ***« flower »***
  - *Exemple de généricité/spécificité :*



5.1 Modèle de correspondance :

Le modèle de correspondance doit comparer entre les deux tableaux représentant respectivement une image et une requête, ces tableaux sont les résultats des deux phases réalisées précédemment (le fichier inverse et l'analyse de requête).

La comparaison entre les index (images) et la requête revient à :

«Trouver les images ayant les mêmes mots que la requête.»

–La requête est une liste de mots clés

–Le fichier inverse est une liste de mots clés

–Comparer les mots de la requête à ceux du fichier inverse

–Sélectionner les images qui contiennent les mots de la requête.

⊣ *La fonction de correspondance (pertinence) :*

Pour qu'une image *Im* réponde à une requête *Q*, il faut que l'implication logique suivante soit valide:

$$Im \Rightarrow Q$$

La similarité (correspondance) entre une image et une requête est définie par la fonction :

R : fonction de pertinence (*Relevance*) d'un couple *(image, requête)* :

$$R(Im, Q) = \begin{cases} 1 & Si\ Im\ appartient\ à\ l'ensemble\ décrit\ par\ Q \\ 0 & Sinon \end{cases}$$

Rappelons que Le modèle booléen considère que les images sont présentes ou absentes de la base (Appariement Exact).

c-à-d. $\begin{cases} \textit{Image pertinente si : } \boldsymbol{R(Im, Q) = 1} \\ \textit{Image non pertinente si : } \boldsymbol{R(Im, Q) = 0} \end{cases}$

Si un mot clé de la requete appartient à une image :

$t \in Im$ ***alors*** $R(Im, t) = 1$

Si un mot clé de la requete n'appartient pas à une image :

$t \notin Im$ ***alors*** $R(Im, t) = 0$

En conséquence, les poids des termes dans les images ou dans la requête sont binaires (pondération binaire), c'est-à-dire $w \in \{0, 1\}$ (1 si présent et 0 si absent).

- On a aussi les formules suivantes : ***[05]***

$R(Im, t_j \text{ AND } t_k) = R(Im, t_j) \times R(Im, t_k)$

$R(Im, t_j \text{ OR } t_k) = R(Im, t_j) + R(Im, t_k) - R(Im, t_j) \times R(Im, t_k)$

$R(Im, t_j \text{ NOT } t_k) = R(Im, t_j) \times (1 - R(Im, t_k))$

Voici l'algorithme général de la comparaison (recherche) :

Algorithme Recherche :

Debut

$Im_p \leftarrow \{\}$ // Im_p : l'ensembles des images pertinentes

Pour *chaque* Im_i *de la base*

Faire

Pour *chaque* t_j , $t_k \in Q$ *(un terme de la requête)*

Si $t_j \in Im_i$ ***alors*** $R(Im_i, t_j) = 1$

Sinon $R(Im_i, t_j) = 0$

Fin Si

$R(Im_i, t_j \text{ AND } t_k) = R(Im_i, t_j) \times R(Im_i, t_k)$

$R(Im_i, t_j \text{ OR } t_k) = R(Im_i, t_j) + R(Im_i, t_k) - R(Im_i, t_j) \times R(Im_i, t_k)$

$R(Im_i, t_j \text{ NOT } t_k) = R(Im_i, t_j) \times (1 - R(Im_i, t_k))$

Fin Pour

Fin Pour

Pour *une requête* Q

Faire

Récupérer l'ensemble des images Im_i *tels que* $R(Im_i, Q) = 1$

$Im_p \leftarrow Im_p \cup Im_i$

Fin Pour

Fin

Algo 2 : Algorithme de recherche

Ainsi, le modèle booléen affirme que chaque image est soit pertinente soit non-pertinent (La fonction de comparaison est binaire : elle sépare la *base* en deux groupes – images pertinentes ou non –). Il n'y a pas de notion de réponse partielle aux conditions de la requête, Les images retournées par le système sont considérés à pertinence égale.

En effet, pour l'utilisateur, la raison pour laquelle une image a été sélectionnée par le système est claire : elle répond exactement à la requête qui a été formulée.

- Prenons nos exemples précédents (Q6, Im_1 , Im_2):

On applique l'algorithme de recherche (comparaison entre Q6 et fichier inverse contenant Im_1 et Im_2) :

Calculons R (Im_1 , Q6) :

On a: R (Im_1 , *Tree)* = *1*

R (Im_1 , *Sand)* = *1*

R (Im_1 , *Natur)* = *0*

R (Im_1 , Q6)

= R (Im_1 , *Tree AND Sand NOT Natur)*

= R (Im_1 , *Tree)* × R (Im_1 , *Sand)* × (1-(R (Im_1 , *Natur)*)

= 1 × 1 × (1- 0)

= 1

Alors Im_1 répond à Q

Calculons R (Im_2 , Q6) :

On a : R (Im_2 , *Tree)* = *1*

R (Im_2 , ***Sand)*** = ***0***

R (Im_2 , *Natur)* = *1*

R (Im_2 , Q6)

= R (Im_2 , ***Tree AND Sand NOT Natur)***

= R (Im_2 , *Tree)* × R (Im_2 , *Sand)* × (1-(R (Im_2 , *Natur)*)

= 1 × 0 × (1- 1)

= **0**

Alors Im_2 ne répond pas à Q6

- Le résultat est :

$Im_p = \{Im_1\}$

5.2 Affichage des résultats :

On a utilisé la présentation la plus simple et la plus courante qui est la liste de réponses en 1 dimension.

Les résultats sont affichés par liste sous forme d'une vignette suivie de sa notice simplifiée.

6 UML : outil de modélisation

Nous utilisons UML comme langage efficace pour modéliser notre système.

UML (en anglais **Unified Modeling Language,**) est un langage graphique de modélisation des données et des traitements.

Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard défini par l'Object Management Group (OMG).

L'OMG diffuse depuis novembre 2007 la version UML 2.1.2, et travaille à présent sur la version 2.2.

Il existe **13** diagrammes UML : *Diagramme de classes, Diagramme d'objets, Diagramme de composants, Diagramme de déploiement, Diagramme des paquetages, Diagramme de structure composite, Diagramme des cas d'utilisation, Diagramme états-transitions, Diagramme d'activité, Diagramme de séquence, Diagramme global d'interaction, Diagramme de temps* ***[06].***

- Nous intéressons à ces 4 diagrammes :

⊣ *Diagramme de classes (Class diagram):*

Le **diagramme de classe** est très utilisé pour représenter l'organisation des données dans les Systèmes d'Information.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe. Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensembles par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

✓ ***Schéma d'une classe*** : Une classe est représentée par un rectangle séparée en trois parties :

- la première partie contient le nom de la classe
- la seconde contient les attributs de la classe
- la dernière contient les méthodes de la classe

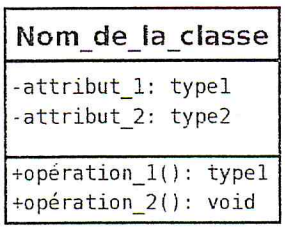


Figure 2.9 : Représentation UML d'une classe

✓ ***Relations possibles entre classes :***

Association, Classe-association, Agrégation, Composition, Dépendance, Généralisation ou Héritage.

- Multiplicité ou cardinalité :

- exactement un : 1 ou 1..1
- plusieurs : * ou 0..*
- au moins un : 1..*
- de un à six : 1..6

Diagramme de cas d'utilisation (use-cases):

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs. Les cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique.

1. ***Acteur*** **:** C'est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Il se représente par un petit bonhomme (figure 2.10) avec son nom (*i.e.* son rôle) inscrit dessous.

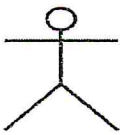


Figure 2.10 : Exemple de représentation d'un acteur

2. ***Cas d'utilisation :*** Un cas d'utilisation est une unité cohérente représentant une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un cas d'utilisation se représente par une ellipse (figure 2.11) contenant le nom du cas (un verbe à l'infinitif), et optionnellement, au-dessus du nom, un stéréotype

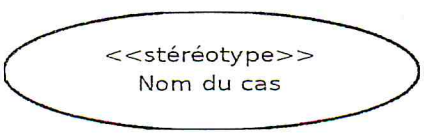


Figure 2.11 : Exemple de représentation d'un cas d'utilisation

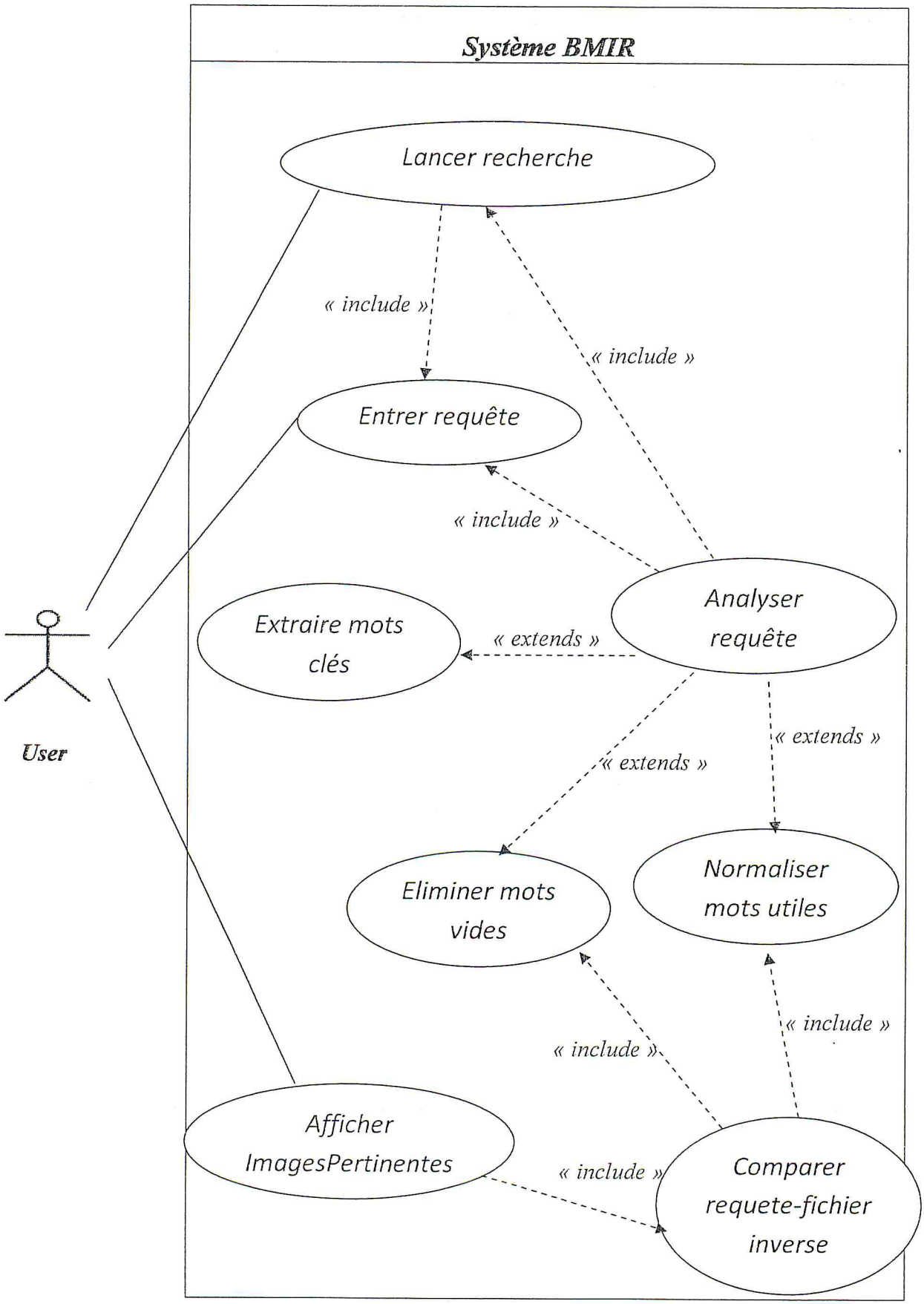
⊥ *Diagramme d'activités (Activity diagram) :*

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

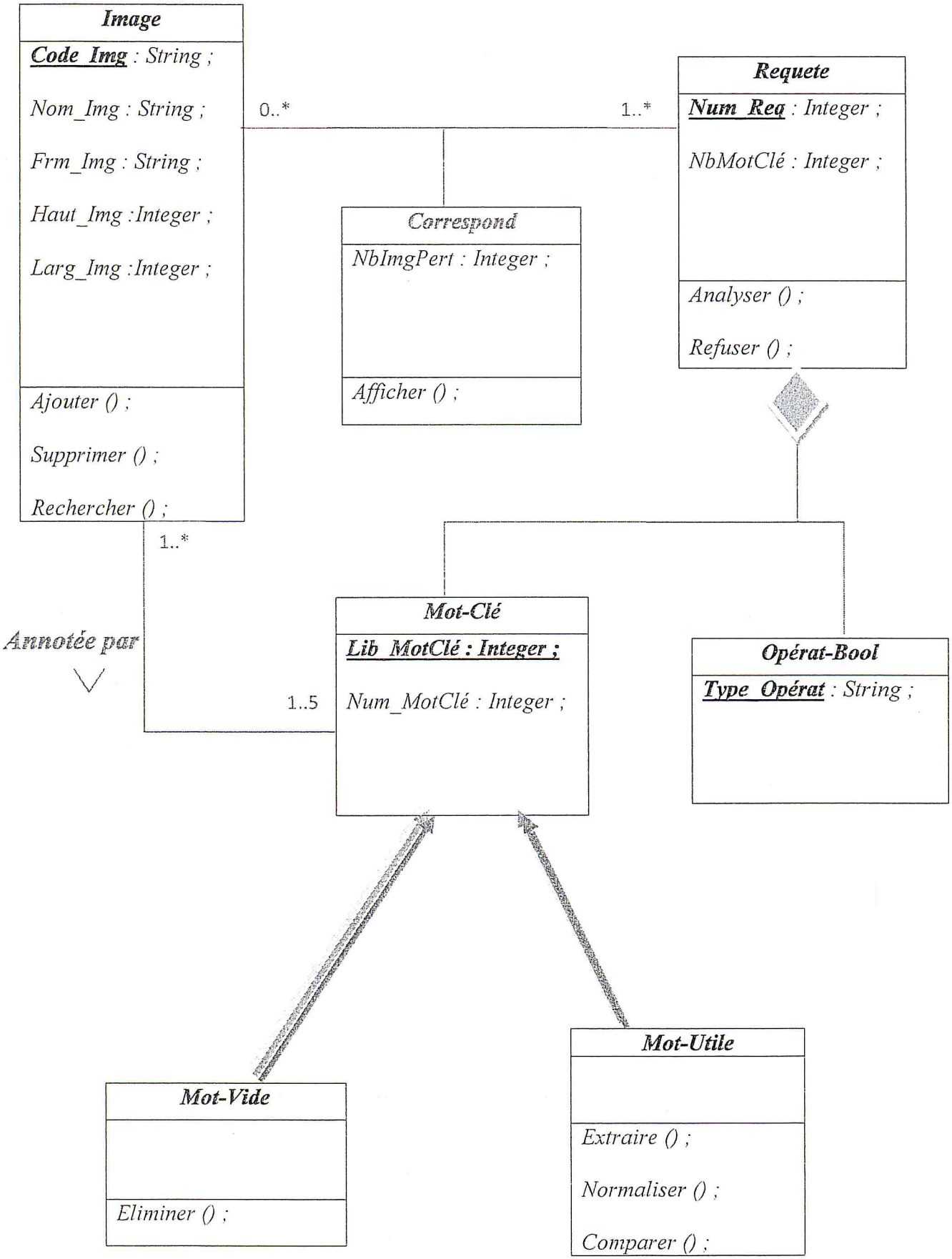
⊥ *Diagramme de séquence :*

Les diagrammes de séquence sont couramment utilisés par nombre d'acteurs d'un projet, même quelque fois à leur insu, sans savoir qu'ils utilisent là un des diagrammes UML. En effet, le diagramme de séquence est une représentation intuitive lorsque l'on souhaite concrétiser des intéractions entre deux entités (deux sous-systèmes ou deux classes d'un futur logiciel). Ils permettent à l'architecte/designer de créer au fur et à mesure sa solution

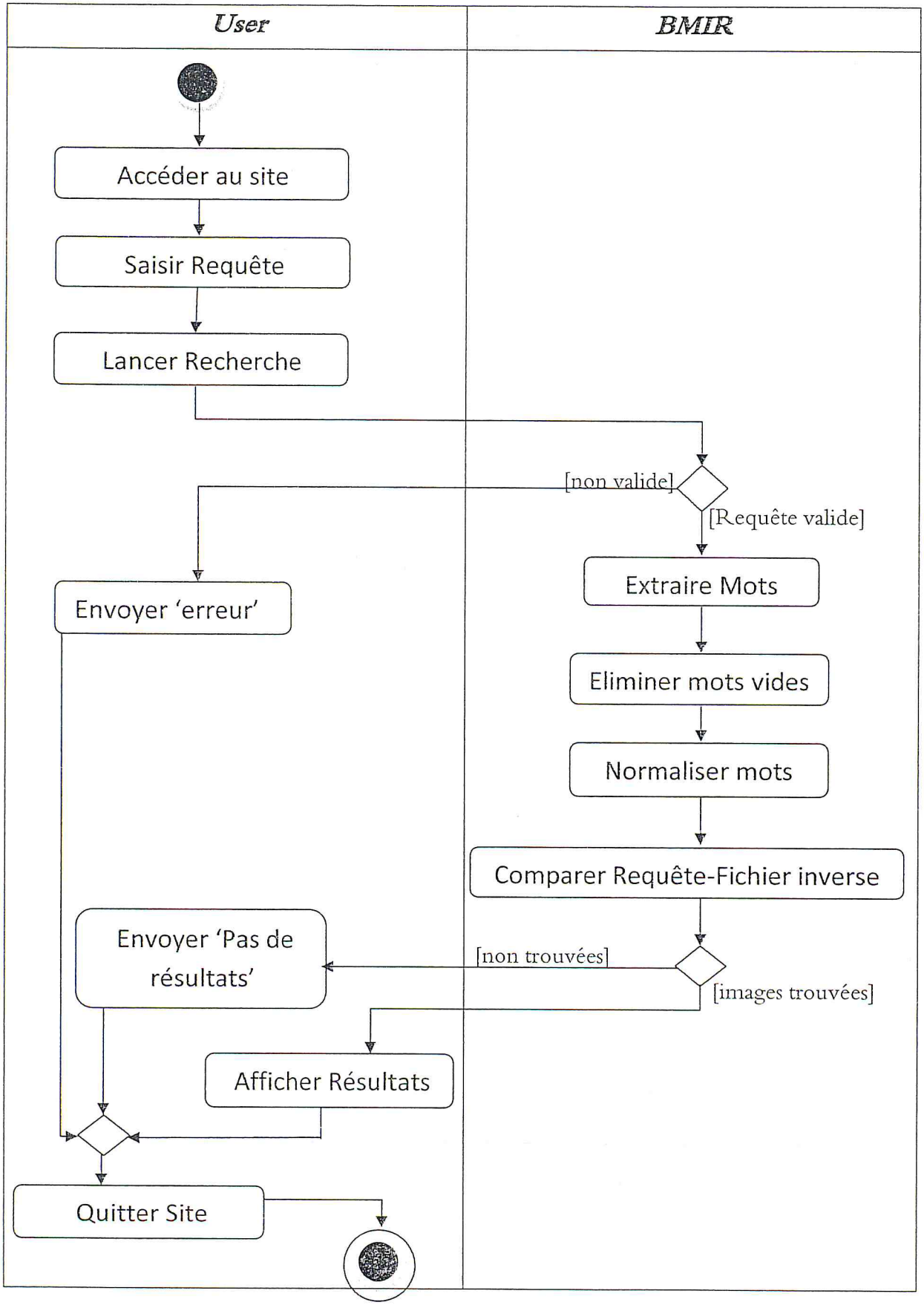
6.1 Diagramme de cas d'utilisation (use-cases) :



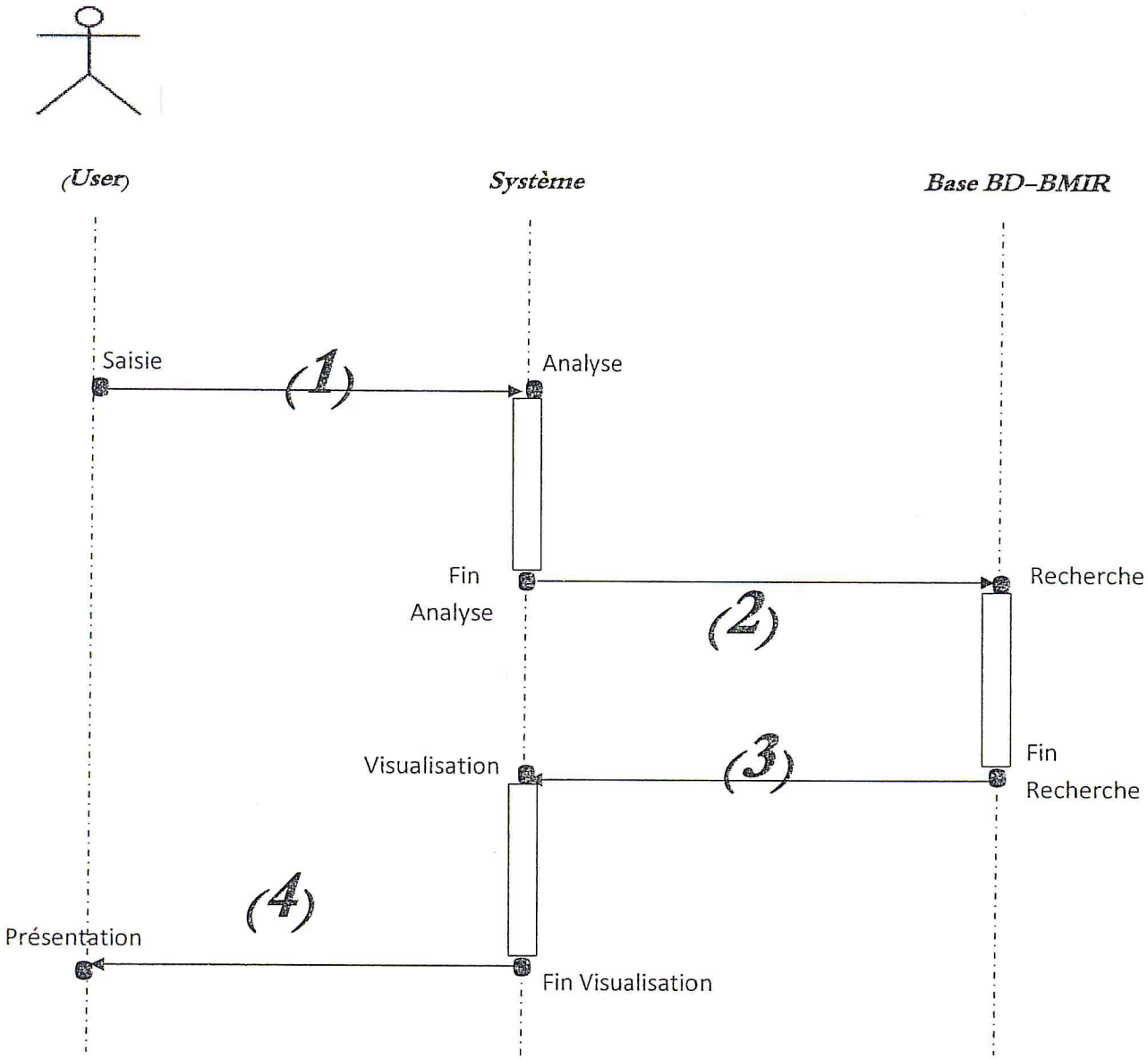
6.2 Diagramme de classes (Class diagram) :



63 6.3 Diagramme d'activités (Class diagram) :



6.4 Diagramme de séquences :



(1) Requête
(2) Mots clés utiles
(3) Images pertinentes
(4) Affichage

I. *Présentations :*

Pour mettre en œuvre notre système BMIR, nous avons créé un site web dynamique en utilisant *Dreamweaver 8 :*

Dreamweaver 8 est un logiciel édité par ***adobe*** qui permet de créer les sites web. Dreamweaver propose une partie qui permet d'éditer et de réaliser les pages dans différents formats comme le html, php, javascript, css ou bien encore le xml ***[06].***

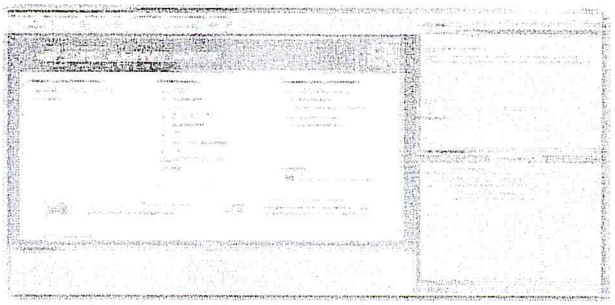


Figure 3.1 : Interface du Dreamweaver 8

On a utilisé pour :

- Le système d'exploitation : ***WINDOWS***
- Le langage : ***PHP.***
- La base de données : ***MySQL.***

Pour permettre travailler avec PHP et MySQL, il faut utiliser :

- ***Apache*** : Le serveur Web
- ***MySQL*** : Le gestionnaire de bases de données

Une plate-forme ***WAMP*** (Windows Apache MySQL PHP) s'installe généralement par le biais d'un seul logiciel qui intègre Apache, PHP, MySQL et phpMyAdmin : c'est ***EasyPHP***, on a utilisé ***EasyPHP 3_0.***

1. PHP :

PHP (sigle de *PHP: Hypertext Preprocessor* ***[07]***), est un langage interprété (de scripts libre) [1] principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP ***[07]***, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale, en exécutant les programmes en ligne de commande. PHP est un langage impératif disposant depuis la version 5 de fonctionnalités de modèle objet complètes ***[08]***.

Ses principaux atouts sont :

✓ Une grande communauté de développeurs partageant des centaines de milliers d'exemples de script PHP ;
✓ La gratuité et la disponibilité du code source (PHP est distribué sous licence GNU GPL) ;
✓ La simplicité d'écriture de scripts ;
✓ La possibilité d'inclure le script PHP au sein d'une page HTML (contrairement aux scripts CGi, pour lesquels il faut écrire des lignes de code pour afficher chaque ligne en langage HTML) ;
✓ La simplicité d'interfaçage avec des bases de données (de nombreux SGBD sont supportés, mais le plus utilisé avec ce langage est *MySQL*, un SGBD gratuit disponible sur de nombreuses plateformes : Unix, Linux, Windows, MacOs X, Solaris, etc...) ;
✓ L'intégration au sein de nombreux serveurs web (Apache, Microsoft IIS, etc.)

Le langage PHP a été mis au point au début d'automne 1994 par Rasmus Lerdorf pour son site Web, il mit en ligne en 1995 la première version de ce programme qu'il baptisa *Personal Sommaire Page Tools*, puis *Personal Home Page v1.0* (traduisez *page personnelle version 1.0*)

La version actuelle est la version 5.2.8.

Le langage PHP est utilisé principalement en tant que langage de script côté serveur, ce qui veut dire que c'est le serveur (la machine qui héberge la page Web en question) qui va interpréter le code PHP et générer du code (constitué généralement d'XHTML ou d'HTML, de CSS, et parfois de JavaScript) qui pourra être interprété par un navigateur.

Sa syntaxe et sa construction ressemblent à celles des langages Java et Perl, à la différence que le PHP peut être intégré dans du code HTML. PHP appartient par ailleurs à la grande famille des descendants du C, dont la syntaxe est très proche.

[1] *La license de PHP est reconnue libre par la Free Software Foundation* [*archive*]. Consulté le 7 novembre 2007

✦ *SGBD supportés par PHP :*

PHP permet un interfaçage simple avec de nombreux systèmes de gestion de bases de données (SGBD), parmi lesquels : Adabas, D, dBase, Empress, FilePro, Informix, Interbase, mSQL, MySQL, Oracle, PostgreSQL, Solid, Sybase, Velocis, Unix, dbm *[09].*

▪ *Exécution du code PHP :*

1. Si un client demande à voir une page PHP.
2. Le serveur n'envoie pas de suite la page au client. Il la **génère**.
   Ce que fait le serveur est simple : il va transformer la page PHP en page HTML, pour que le client puisse la lire.
3. Enfin, une fois que la page est générée, elle ne contient plus que du code HTML. Le serveur peut l'envoyer au client *[10].*

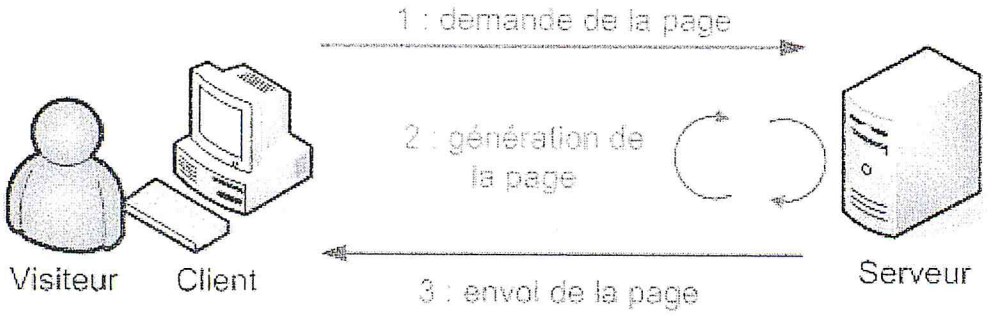


Figure 3.2 : Déroulement de l'exécution du code PHP

2. *PHP travaille avec MySQL :*

MySQL est un SGBDR (Système de Gestion de Base de Données relationnelle). Il permettra de créer et gérer des bases de données.

Sous licence GPL, ce programme est entièrement gratuit. MySQL supporte les plateformes suivantes : Windows, Solaris, Mac Os X Serveur, HP UX, AIX, SCO, SGI Irix, DEC OSF, BSDi, cette liste n'est pas exhaustive. La base de données est administrée avec *PhpMyadmin [12].*

- Lorsque le serveur a reçu une demande d'un client qui veut une page PHP :

1. Le serveur utilise toujours PHP, il lui fait donc passer le message.
2. PHP effectue les actions demandées et se rend compte qu'il a besoin de MySQL. En effet, le code PHP contient à un endroit "Va demander à MySQL d'enregistrer ce message". Il fait donc passer le travail à MySQL.
3. MySQL fait le travail que PHP lui avait soumis.
4. PHP renvoie au serveur que MySQL a bien fait ce qui lui était demandé *[11]*.

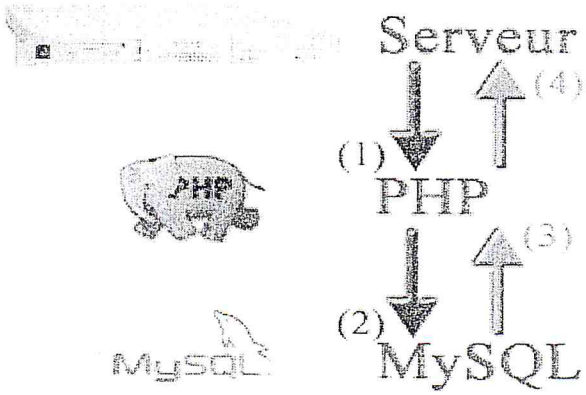


Figure 3.2 : Déroulement Serveur-PHP-MySQL

3. *EasyPHP :*

EasyPHP "kit" est un environnement de développement composé de deux serveurs : un serveur Web Apache et un serveur de base de données MySQL. De plus, il possède un interpréteur de scripts (PHP) et d'un module d'administration SQL nommé PhpMyAdmin.
Ce permet ainsi de faire fonctionner toutes sortes de scripts PHP (un langage Web) et notamment de vérifier le bon fonctionnement d'un site Internet écrit en PHP.

EasyPHP propose un ensemble d'applications permettant de faire fonctionner des scripts PHP en local (127.0.0.1), c'est-à-dire sans avoir à se connecter à un serveur externe sur Internet. Il peut tout à fait être comparé aux logiciels WAMP5 ou XAMPP, qui sont des équivalents *[12]*.

Et voici le menu de l'EasyPHP *3 :*

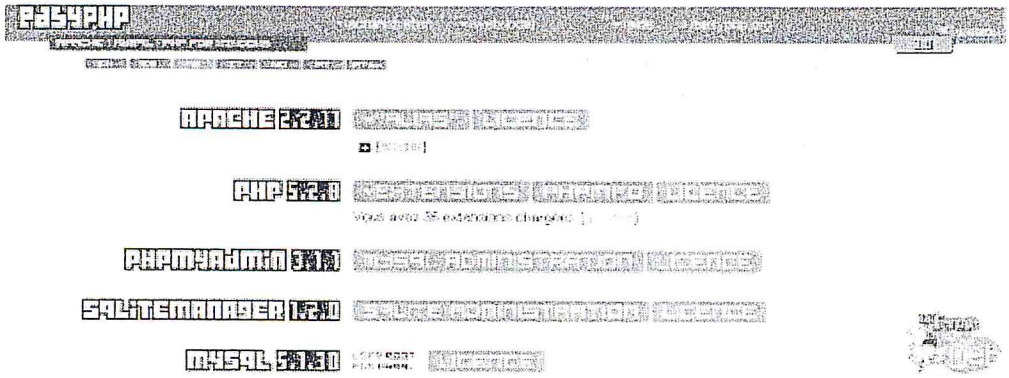


Figure 3.4 : Interface de l'EasyPHP 3.0

- Apache 2.2.11 (Dernière version d'apache à ce jour)
- PHP 5.2.8 (Dernière version de PHP à ce jour)
- PhpMyAdmin 3.1.1 (Pas la dernière. 3.1.2 sortie hier ...)
- SQLiteManager 1.2.0 (Dernière version à ce jour)
- MySQL 5.1.30 (Dernière version à ce jour)

4. PhpMyAdmin

PhpMyAdmin (prononcez « p h p ma?admine »), parfois abrégé PMA, est un ensemble de scripts écrits en PHP, c'est le meilleur outil pour administrer et faire des manipulations sur les BD mySQL par le truchement d'une interface WEB distante. Elle est livrée avec WAMP.

0.9.0 est la première version interne (9 septembre 1998).

3.2.0 est la dernière version (15 juin 2009).

La version utilisée est : ***PhpMyAdmin 3.1.1***

- Le logiciel, actuellement est disponible dans 50 langues différentes.

L'accueil de PhpMyAdmin ressemble à ceci :

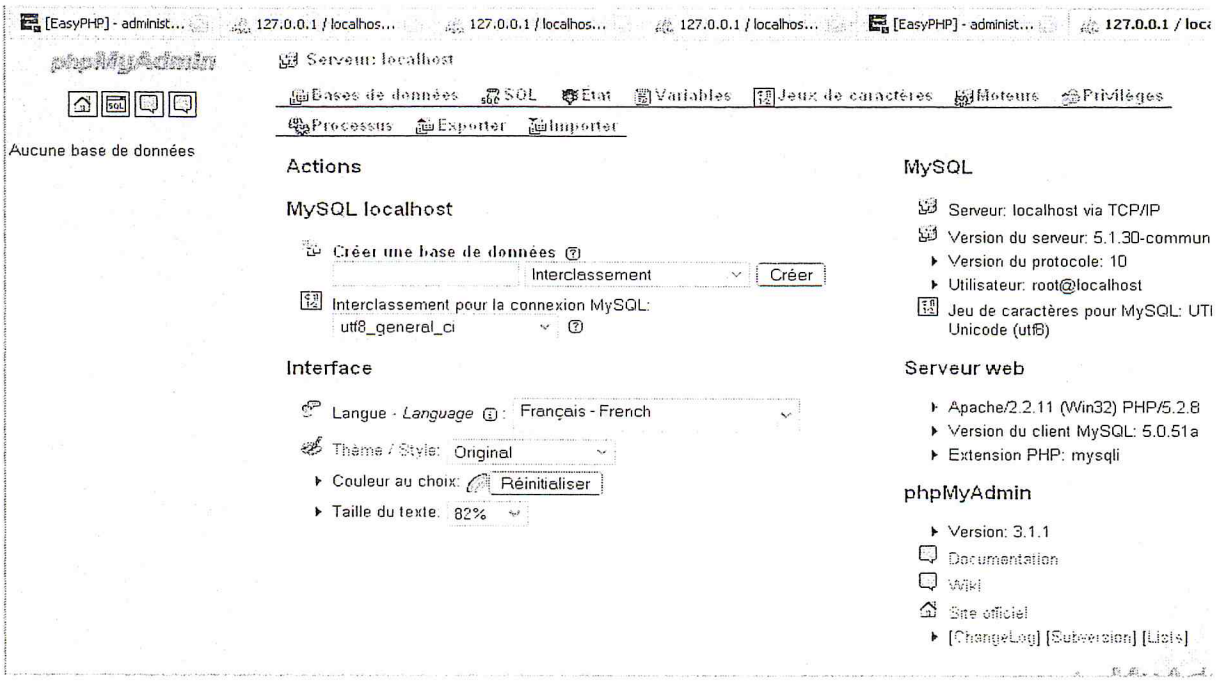


Figure 3.5 : Interface du PhpMyAdmin 3.1.1

II. Création de BD-MBIR :

Nous avons crée notre base de données qui est constituée par un ensemble d'images annotées par mots clés :

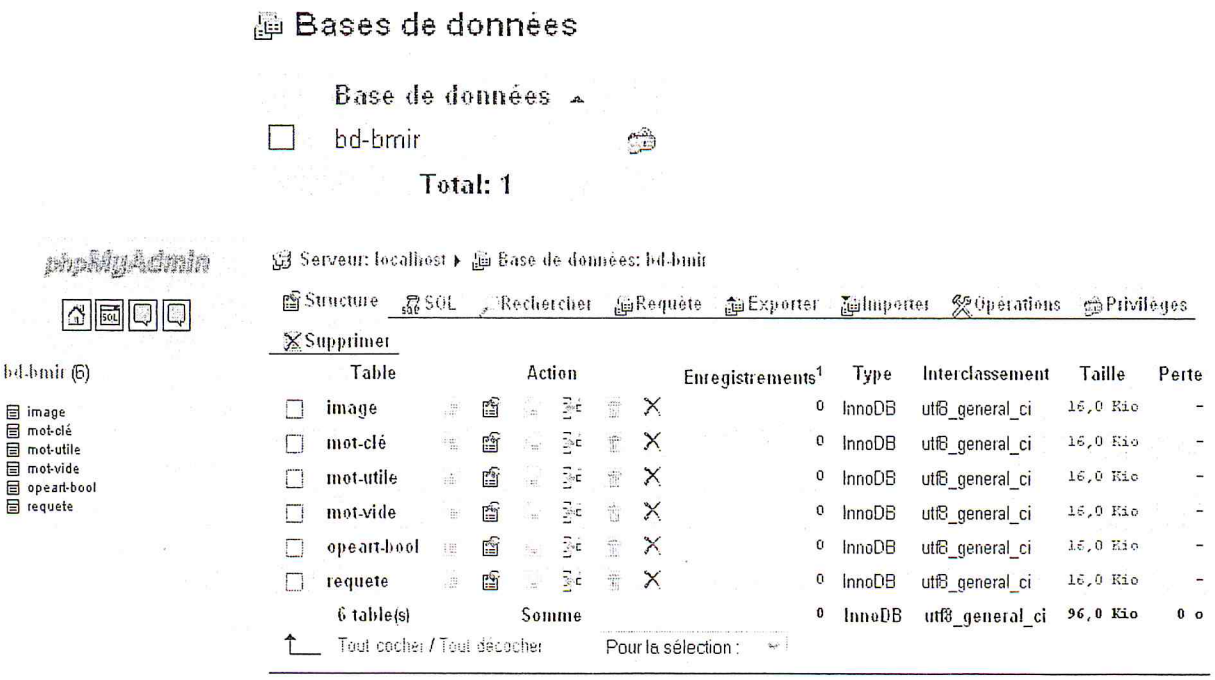


Figure 3.6 : Présentation de notre base

- Et voici la table : Image :

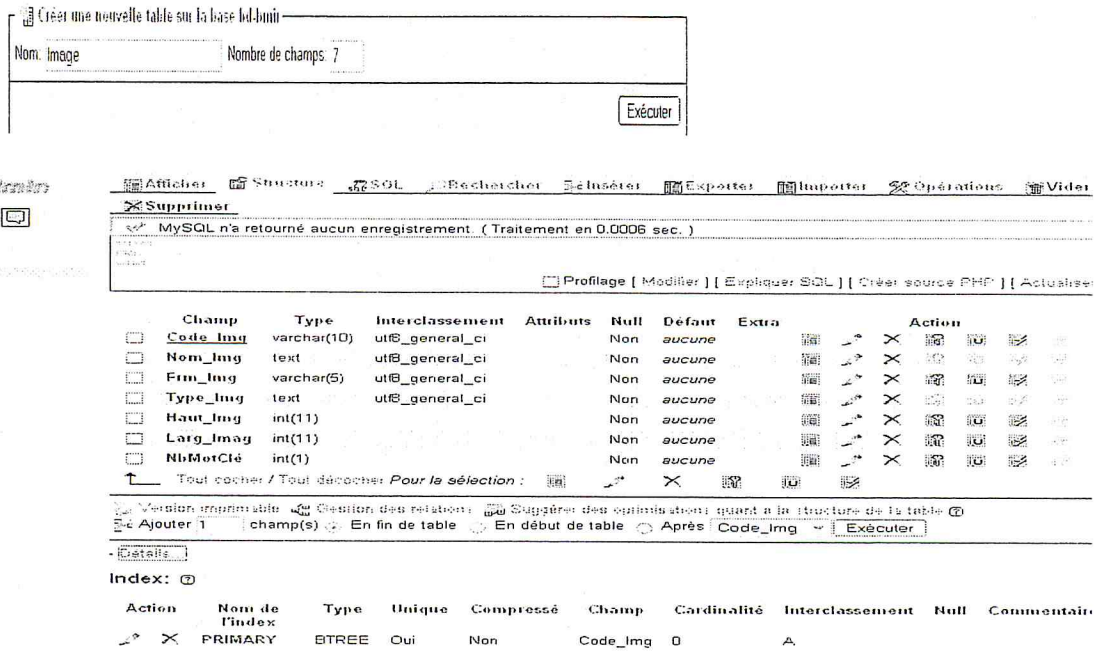


Figure 3.7 : Présentation de la table « Image »

- Et voici quelques exemples d'enregistrements dans la table « image» :

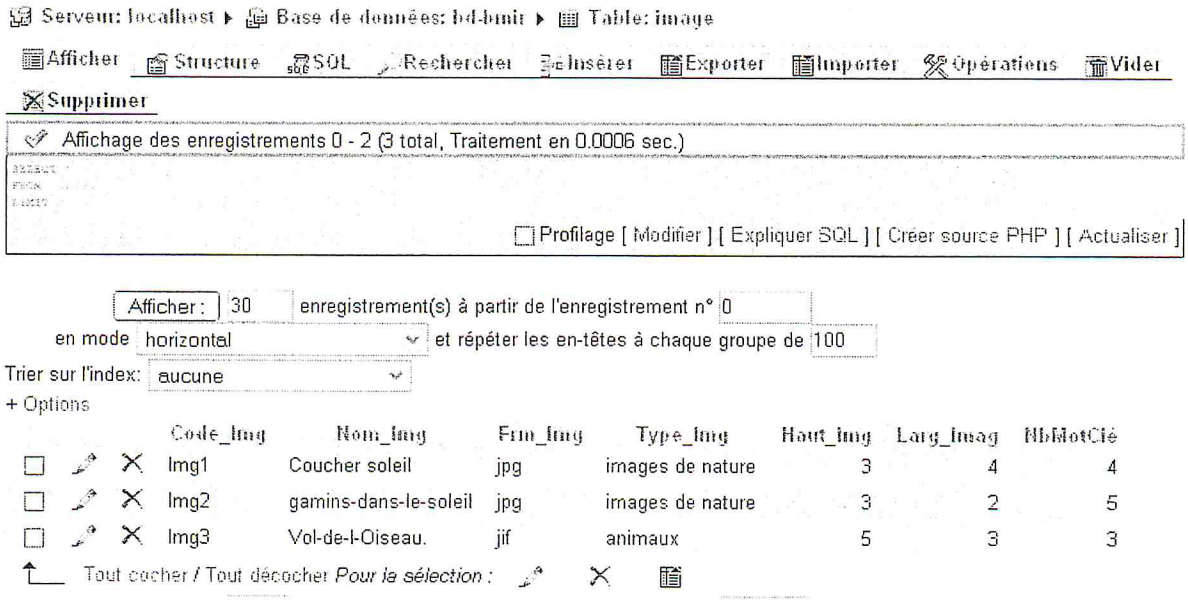


Figure 3.8 : Exemples d'images de la base

III. *Interfaces du système :*

- Voici la page d'accès au système BMIR :

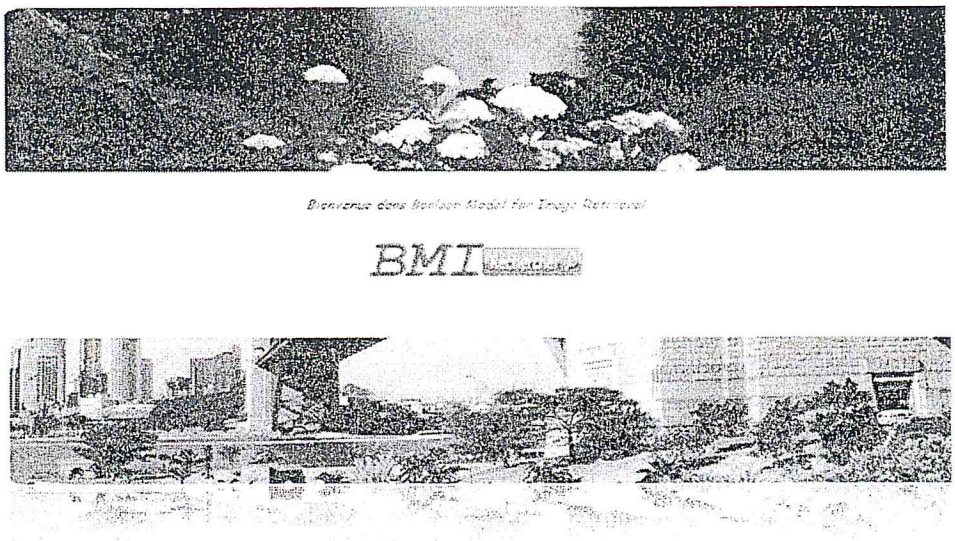


Figure 3.9 : Page d'accès du BMIR

- Et voici la page d'accueil du site : Quelques présentations et définitions(le modèle booléen, les SRI, les images et le processus de recherche d'images)

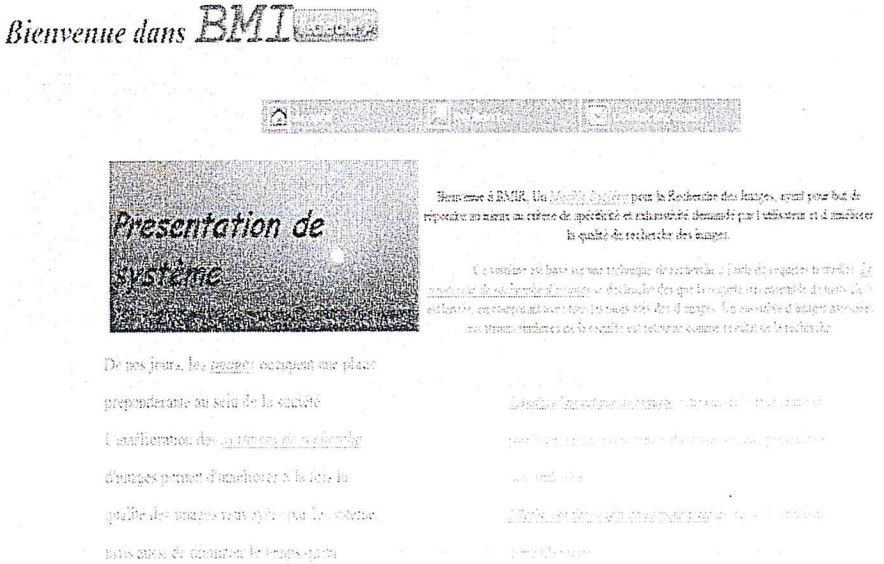


Figure 3.10: Page d'accueil du BMIR

- Et voici la page de la recherche du site : Entrer les mots clés dans la chaine, avec les boutons qui affichent les opérateurs (AND, OR et NOT). Lancer la recherche avec le bouton (GO !)

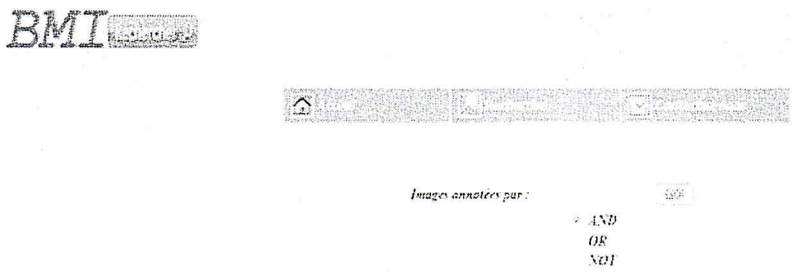


Figure 3.11 : Page de la recherche du BMIR

- Et voici quelques exemples de test : le mot clé « eiffel » donne 7 images résultats

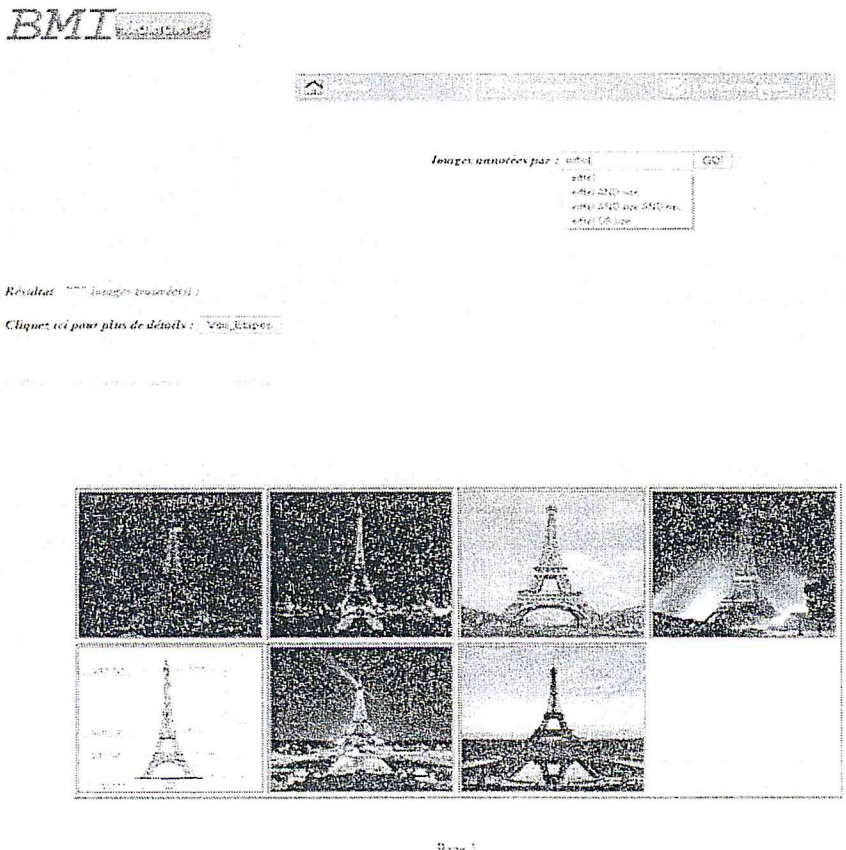


Figure 3.12 : Test 1

La requête (the sea OR natur NOT mountain) donne 25 images résultats (dans 3 pages)

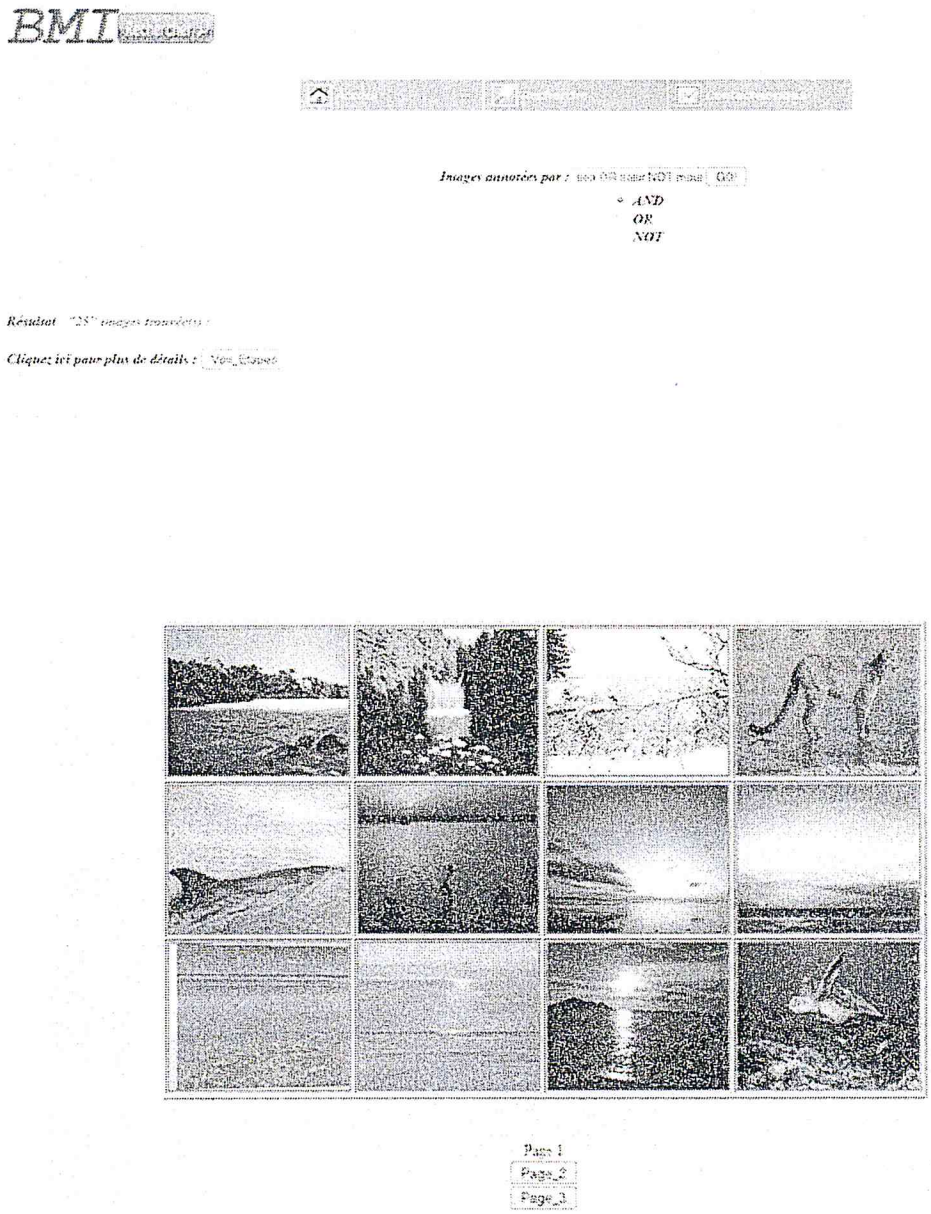


Figure 3.13 : Test 2

- Et voici la page qui présente les étapes de la recherche : En cliquant dans le bouton (Voir_Etapes) précédent, on obtient un menu vertical, on peut aussi voir le fichier inverse en cliquant dans le bouton (Fichier_Inverse)

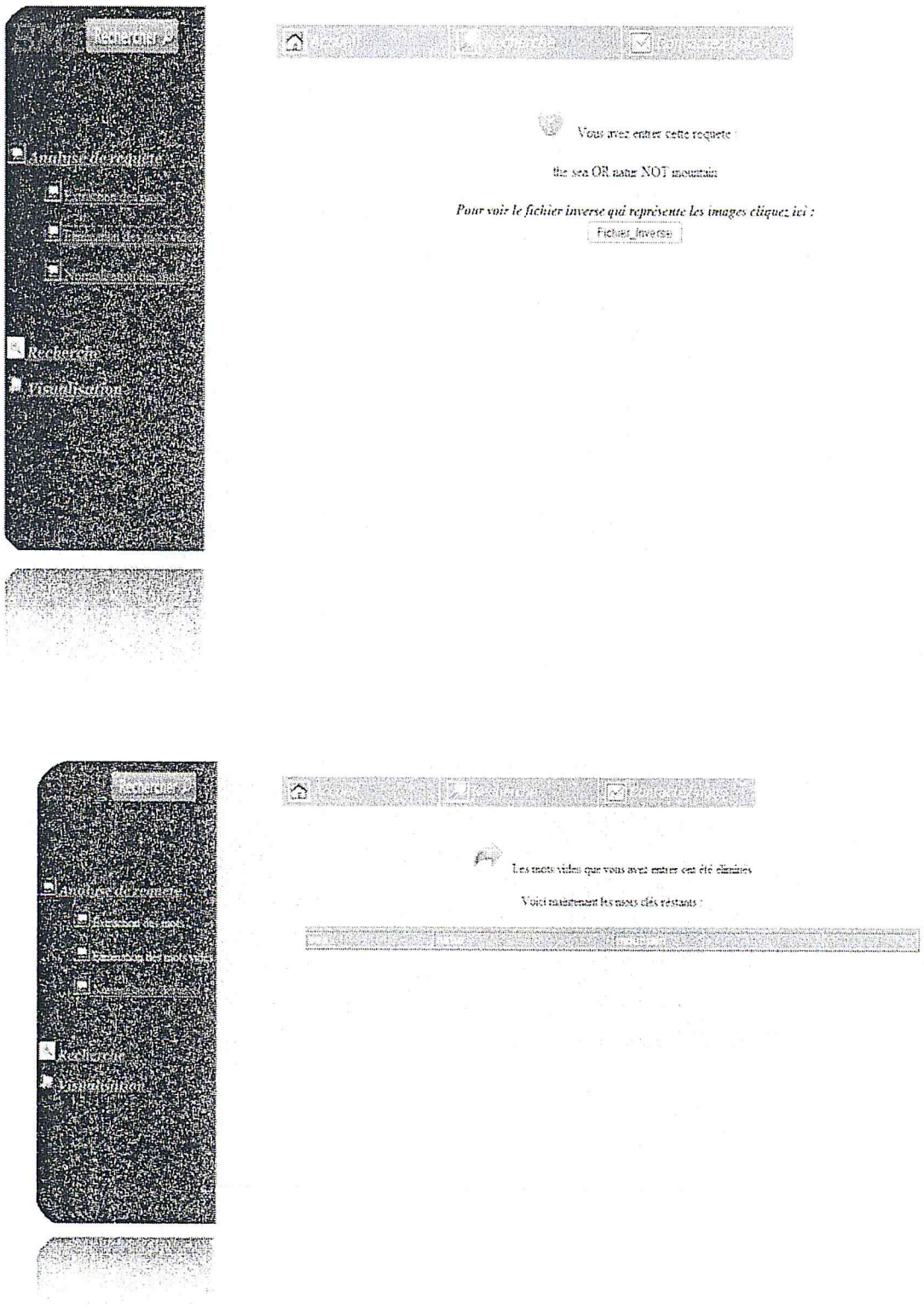


Figure 3.14 : Menu vertical du BMIR

CONCLUSION GENERALE

Le travail présenté dans le cadre de ce mémoire s'intéresse à la recherche d'information adaptée au média image, l'objectif étant de permettre la recherche de documents images par des utilisateurs. De manière similaire aux travaux de recherche d'information sur des documents textuels, définir un modèle de recherche d'information pour les images nécessite la mise en place d'un certain nombre d'éléments : modèle de documents et de requêtes, modèle de correspondance exacte.

Le cadre de ce travail se focalise sur une base d'images généraliste, sur des utilisateurs quelconques, et sur l'utilisation de descriptions textuelles du contenu des images et des requêtes tout en vérifiant la sémantique.

Au cours de l'état de l'art que nous avons réalisé, nous avons étudié la modélisation des systèmes de recherche d'information et nous avons détaillé le modèle booléen classique de recherche d'information sur des textes. Notre travail portant sur les images, nous avons également exposé des travaux existants sur la recherche d'images, en mettant en relief la difficulté d'appliquer aux images les techniques classiques de recherche d'information. Cette difficulté provient du fossé sémantique, qui sépare les pixels constituant une image de l'interprétation du sens associé à ces pixels. Nous avons en particulier distingué les approches existantes en RIm selon qu'elles considèrent ou non le sens associé aux pixels.
Nous avons cité aussi les différents langages utilisés pour la recherche d'images qui sont : les langages textuels simples et les langages complexes (graphiques, visuels et hypermedia).
Et enfin, nous avons parlé sur les différentes présentations utilisées pour la visualisation des résultats dans les SRIm ainsi que les mesures d'évaluation de ces systèmes.

Nous proposons un système qui applique le modèle booléen de recherche d'information pour les images. Ce système comprend un modèle de représentation de documents basé sur un modèle d'image physique. Notre définition repose sur le fichier inverse. Un fichier inverse est une génération logique du modèle d'images physique. Les documents – et les requêtes – sont représentés par des mots clés, les opérateurs booléens sont utilisés aussi pour la représentation des requêtes. Cela permet de vérifier la sémantique qui se trouve dans les images. Ce système comprend une fonction de correspondance basée sur la comparaison des mots clés des requêtes et ceux du fichier inverse.

Cette utilisation de mots clés comme descripteurs d'images dans le cadre d'un modèle booléen de recherche d'information donne à notre travail un avantage qui est d'un coté : la simplicité du système et d'un autre coté : la vérification de la sémantique des images qui rend les résultats plus ou moins exactes.

Perspectives :

Ce travail, comme tout travail de mémoire, pose de nombreuses questions et donne lieu à de nombreuses perspectives, parmi lesquelles :

- Il est important de modéliser un **modèle de connaissances** qui prend en compte les connaissances externes qui peuvent enrichir le nombre de réponses du système, par exemple en incluant un **thésaurus** composé des termes apparaissant dans l'ensemble des images, reliés entre eux par des liens de généricité/spécificité ou de synonymie. Par exemple, une requête portant sur "voiture" pourra retourner des images contenant le terme "automobile".
  L'utilisation d'un thésaurus permet d'augmenter le nombre de réponses du système, en incluant dans le résultat des images contenant des mots reliés aux mots de la requête.

- On peut par exemple proposer à l'utilisateur de lui indiquer parmi les résultats les images qu'il estime pertinentes et ceux qu'il estime non pertinentes, afin de retourner de nouveaux résultats en fonction des indications et choix de l'utilisateur. Ce processus interactif, appelé **bouclage de pertinence**, a pour but d'affiner la recherche, d'améliorer la qualité des résultats et de diminuer le temps d'extraction des résultats.

- Au niveau de la **normalisation** des mots clés extraits de la requête, on a utilisé **l'algorithme de Porter**. Cette phase de passage à la forme canonique présente le principal avantage de représenter par exemple le mot "cuts" et le mot "cut" de la même façon ("cut"), ce qui évite à l'utilisateur de devoir entrer les formes de pluriel des noms ou les formes conjuguées des verbes lors de sa recherche. Cependant, dans certains cas, le passage à la forme canonique supprime la sémantique originale du mot. Par exemple, "iteration" produit "iter" et "general" produit "gener". Ainsi, lorsque l'utilisateur formulera une requête avec le mot "iteration", il aura très certainement, parmi la liste des images résultats, des images non pertinentes relatives au mot "iter"... Si la lemmatisation a pour but d'augmenter le rappel, la précision (c'est-`a-dire la proportion d'images pertinentes par rapport au nombre d'images renvoyés par le système) en fait souvent les frais . . .

  Pour solutionner ce problème, C.J. Crouch [*Cro02*] propose une méthode en deux temps, dont les résultats s'avèrent encourageants :
  – Une première recherche est effectuée, en utilisant une lemmatisation des mots ;
  – Les documents sont ensuite réordonnés en fonction de la présence des termes non-lemmatisés de la requête dans leur contenu.

Annexe A

Autres modèles de recherche d'information :

Cette annexe décrit les modèles booléen étendu, vectoriel, probabiliste, logique et le *requêtes à base de questions* de RI.

Trois modèles de RI textuelle sont dits "classiques". Il s'agit du modèle **booléen**, du modèle **vectoriel**, et du modèle **probabiliste**. Un autre modèle, le modèle **logique**, est fondé sur une interprétation logique du contenu des documents.

Ces modèles ont en commun le vocabulaire d'indexation. Seule la manière de comparer les documents à la requête change. Ces modèles se basent sur le formalisme des mots clés ***[Mar04]***.

A.1 Le modèle booléen étendu :

Le modèle booléen étendu a été introduit par Salton [Sal83]. C'est une extension du modèle précédent qui vise à tenir compte d'une pondération des termes dans le corpus. Cela permet de combler le manque du modèle standard en ordonnant les documents retrouvés par le SRI.

La requête reste inchangée et est composée d'une expression booléenne classique. Par contre, la représentation d'un document se voit augmentée par l'ajout de pondération
pour chaque terme (t_i,w_i). En général ce poids est principalement basé sur le nombre d'occurrences d'un terme dans le document mais dans certains systèmes la typographie
- taille du texte, forme de la police de caractère - est également prise en compte.

La détermination de la correspondance d'un document à une requête C (d, q) peut prendre plusieurs formes. Suivant le cadre classique des ensembles flous proposé par
Zadeh ***[Zad65]***, on obtient les relations suivantes :

$$\begin{aligned} C(d,t_i) &= a_i \\ C(d,q_1 \wedge q_2) &= min(C(d,q_1),C(d,q_2)) \\ C(d,q_1 \vee q_2) &= max(C(d,q_1),C(d,q_2)) \\ C(d,\neg q) &= 1 - C(d,q) \end{aligned} \qquad (1)$$

Les opérateurs logiques $\wedge$ et $\vee$ sont évalués respectivement par min et max. Cependant, cette évaluation n'est pas parfaite. On n'obtient pas les relations $C(d, q \wedge \neg q) \equiv 0$ et $C(d, q \vee \neg q) \equiv 1$. Ce qui signifie que lorsqu'on évalue une requête sous forme de conjonction, on ne s'intéresse qu'à la partie la plus difficile et lorsqu'on évalue une requête sous forme de disjonction, c'est la partie la plus facile qui domine. C'est pour cette raison que d'autres formes de correspondances ont été proposées. Les relations les plus utilisées ont été introduites par Lukasiewicz [Lukasiewicz, 1963] sous la forme suivante :

$$\begin{aligned} C(d, t_i) &= a_i \\ C(d, q_1 \wedge q_2) &= C(d, q_1) \cdot C(d, q_2) \\ C(d, q_1 \vee q_2) &= C(d, q_1) + C(d, q_2) - C(d, q_1) \cdot C(d, q_2) \\ C(d, \neg q) &= 1 - C(d, q) \end{aligned} \qquad (2)$$

Les deux parties d'une conjonction ou d'une disjonction contribuent en même temps à l'évaluation de la correspondance du document à la requête. Cependant, ce modèle n'est pas parfait (on n'obtient toujours pas $C(d, q \wedge \neg q) \equiv 0$ et $C(d, q \vee \neg q) \equiv 1$) mais il reste convenable.

Le modèle p-norme introduit par Salton mesure les correspondances de la conjonction et de la disjonction [Sal83]. L'idée de base réside dans l'observation de la table de vérité (voir table A.1). Dans la colonne $A \wedge B$, la meilleure correspondance est atteinte dans le cas de la dernière ligne. Dans la colonne $A \vee B$, la pire correspondance correspond à la première ligne.

A	B	$A \wedge B$	$A \vee B$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Tab A.1 – Table de vérité

On peut ainsi considérer que l'évaluation d'une conjonction ou d'une disjonction consiste à calculer une sorte de distance entre le point à atteindre ou à éviter. Ce concept est illustré par la figure A.1. Dans ces schémas, les axes des abscisses correspondent à l'évaluation du document A et les axes des ordonnées à l'évaluation du document B.

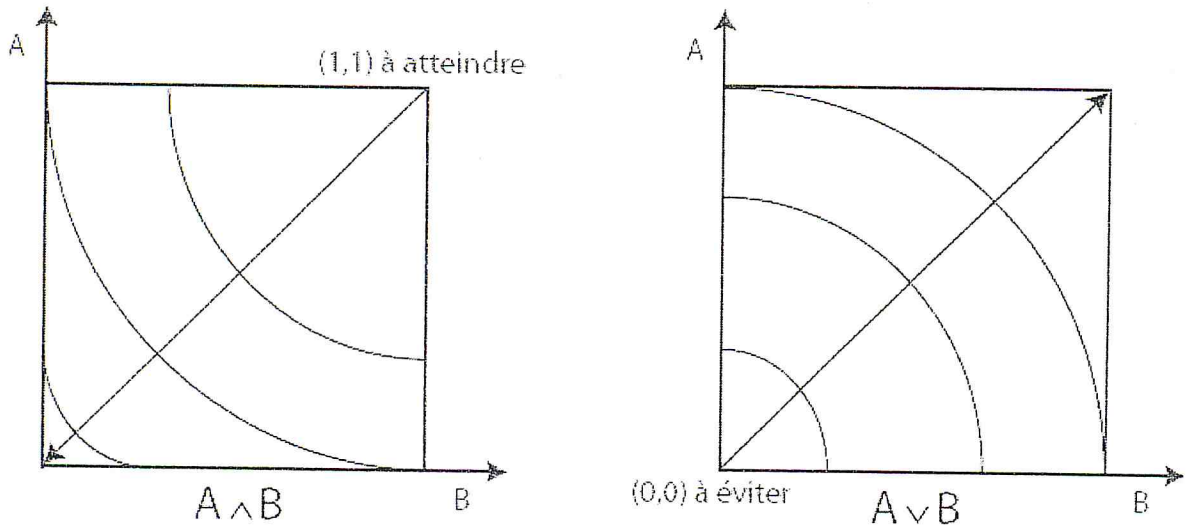


Figure A.1 : Évaluation d'une conjonction ou d'une disjonction.

Pour la conjonction, on cherche à évaluer dans quelle mesure le point c défini par l'évaluation d'un document A et d'un document B est proche de (1,1), le point à atteindre.

Ce rapprochement peut être mesuré par le complément de la distance entre le point c et le point (1,1). Plus cette distance est grande, moins $A \wedge B$ est satisfaite. Les points qui se situent sur une même courbe, ont la même distance avec (1,1) et ainsi correspondent à la même évaluation. Dans le cas de $A \vee B$, on recherche plutôt à éviter le point (0,0). Plus on est loin de (0,0), plus $A \vee B$ est satisfaite. Ainsi Salton propose les relations de correspondance normalisées suivantes :

$$\begin{aligned} C(d,t_i) &= a_i \\ C(d,q_1 \wedge q_2) &= 1 - \sqrt{\frac{(1-C(d,q_1))^2 + (1-C(d,q_2))^2}{2}} \\ C(d,q_1 \vee q_2) &= \sqrt{\frac{C(d,q_1)^2 + C(d,q_2)^2}{2}} \\ C(d,\neg q) &= 1 - C(d,q) \end{aligned} \qquad (3)$$

A.2 Le modèle vectoriel :

Le modèle vectoriel fait partie des modèles statistiques. L'utilisation des statistiques a pour but d'une part de caractériser d'un point de vue quantitatif les termes et les documents et d'autre part de mesurer le degré de pertinence d'un document vis à vis d'une requête. Le but final est d'arriver à retourner une liste ordonnée de documents selon ce degré. Un autre avantage réside dans l'expression des besoins de l'utilisateur : contrairement au modèle booléen où les termes de la requête doivent être reliés par des connecteurs logiques, l'utilisateur peut ici aussi exprimer son besoin en information en langage naturel ou sous forme d'une liste de mots clés.

Luhn ***[Luh57]*** a été le premier à proposer une approche statistique de recherche d'information à la fin des années 1950. Il suggère que l'utilisateur fournisse un document qui ressemble à son besoin en information. La mesure de similarité entre le document fourni et la représentation des documents de la collection est utilisée pour ordonner ces documents.

Le critère de similarité est ainsi défini :

Plus deux représentations contiennent les mêmes éléments, plus la probabilité qu'elles représentent la même information est élevée.

Une telle définition revient en fait à compter le nombre d'éléments que partagent la requête et la représentation du document. Pour ce faire, considérons la représentation d'un document comme un vecteur $\vec{d_j} = \{w_{1,j}, w_{2,j}, \ldots, w_{t,j}\}$, où $w_{i,j}$ est le poids (0 ou 1) des termes dans le documents, t étant le nombre total de termes de l'index, et considérons la représentation de la requête comme un vecteur $\vec{q} = \{w_{1,q}, w_{2,q}, \ldots, w_{t,q}\}$, avec les mêmes notations.
La mesure de similarité la plus simple est alors le produit scalaire :

$$RSV(\vec{d_j}, \vec{q}) = \sum_{i=1}^{t} w_{i,j} * w_{i,q} \qquad (4)$$

Comme les poids des termes sont binaires, la mesure de similarité mesure le nombre de termes partagés entre le document et la requête.

Salton ***[Sal70]*** a proposé un modèle basé sur cette mesure de similarité dans son projet SMART (Salton's Magical Automatic Retriever of Text). Le document (vecteur $\overline{d}$) et la requête (vecteur $\overline{q}$) sont représentés là encore dans un espace Euclidien de dimension élevée engendré par tous les termes de l'index. La similarité est alors le cosinus de l'angle formé par les deux vecteurs :

$$RSV(\overline{d_j}, \overline{q}) = \frac{\overline{d_j} \cdot \overline{q}}{\left|\overline{d_j}\right| * \left|\overline{q}\right|}$$

$$= \frac{\sum_{i=1}^{t} w_{i,j} * w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} * \sqrt{\sum_{i=1}^{t} w_{i,q}^2}} \qquad (5)$$

D'autres fonctions de similarité ont été proposées dans la littérature, parmi lesquelles on peut citer les mesures de Jaccard et Dice.

Les documents sont ainsi classés en fonction de la mesure de l'angle qu'ils forment avec le vecteur requête. L'aspect le plus intéressant de cette mesure est l'influence d'un terme isolé sur le score de recherche. Si un terme est présent à la fois dans la requête et le document, il contribue au score. S'il est présent uniquement dans l'un des deux, il diminue le score parce que la requête et le document se correspondent moins.

Plusieurs algorithmes de recherche d'information ont prouvés leur performance lorsque les vecteurs requête et documents étaient normalisés. L'algorithme d'apprentissage de Rocchio en est un exemple ***[Roc71]***.

Venons-en maintenant à la pondération des termes. Les travaux de Salton ***[Sal71']*** ont montré qu'il ne s'agissait pas d'un problème trivial, mais les pondérations selon TF et IDF restent les plus courantes et les plus simples.

Les avantages d'un tel modèle sont nombreux : la pondération des termes augmente les performances du système, le modèle permet de renvoyer des documents qui répondent approximativement à la requête, et la fonction d'appariement permet de trier les documents selon leur degré de similarité avec la requête.

Théoriquement, le modèle vectoriel a l'inconvénient de considérer que les termes de l'index sont tous indépendants. Cependant en pratique, la prise en compte globale de la dépendance des termes peut faire baisser la qualité des réponses d'un système (puisque les dépendances sont généralement locales).

De nombreuses méthodes d'ordonnancement des résultats ont été comparées au modèle vectoriel, et celui-ci, malgré sa simplicité, est supérieur ou au moins aussi bon que les autres alternatives. C'est pour toutes ces raisons qu'aujourd'hui le modèle vectoriel est le plus populaire en recherche d'information ***[Sau05].***

Avantages :
- utilisation facile des modèles ;
- aptitude des modèles à bien fonctionner.

Inconvénient :
- fondements purement empiriques : peu de fondements théoriques.

A.3 Le modèle probabiliste :

Le modèle probabiliste aborde le problème de la recherche d'information dans un cadre probabiliste. Le premier modèle probabiliste a été proposé par Maron et Kuhns ***[Mar60]*** au début des années 1960. Le principe de base consiste à présenter les résultats de recherche d'un SRI dans un ordre basé sur la probabilité de pertinence d'un document vis-à-vis d'une requête. Robertson ***[Rob77]*** résume ce critère d'ordre par le "principe de classement probabiliste", aussi désigné par PRP (Probability Ranking Principle).

Etant donnée une requête utilisateur, il y a un ensemble des documents qui contient exactement les documents pertinents et aucun autre. Nous appellerons cet ensemble l'ensemble de réponse idéal. Si l'on requête la description de cet ensemble idéal, on n'aura aucun problème à retrouver les documents qui le composent. Répondre à une requête revient donc à spécifier les propriétés de cet ensemble idéal.

Ce n'est bien sûr pas si simple que cela. Comme les propriétés de l'ensemble idéal ne sont pas connues au moment de la requête, il faut d'abord deviner ce qu'il pourrait être. Cette première tentative permet de générer une première description probabiliste de l'ensemble, qui est ensuite utilisée pour retrouver un premier ensemble de documents. Il faut ensuite une interaction avec l'utilisateur pour améliorer la description probabiliste de l'ensemble idéal (ou plutôt de l'échantillon représentant cet ensemble idéal) ***[Rob77].***

Le processus de recherche se traduit par calcul de proche en proche, du degré ou probabilité de pertinence d'un document relativement à une requête. Pour ce faire, le processus de décision complète le procédé d'indexation probabiliste en utilisant deux probabilités conditionnelles :

– $P\ (w_{ij}/Pert)$: probabilité que le terme t_i occurre dans le document d_j sachant que ce dernier est pertinent pour la requête.

– $P\ (w_{ij}/NonPert)$: que le terme t_i occurre dans le document d_j sachant que ce dernier n'est pas pertinent pour la requête.

Le calcul d'occurrences des termes d'indexation dans les documents est basé sur l'application d'une loi de distribution sur un échantillon représentatif de documents d'apprentissage. En posant les hypothèses suivantes :

– la distribution des termes dans les documents pertinents est la même que leur distribution par rapport à la totalité des documents.

– les variables "document pertinent", "document non pertinent" sont indépendantes.

La fonction de recherche est obtenue en calculant la probabilité de pertinence d'un document D, notée *P (Pert/D)* ***[Van79]*** **:**

$$P(Pert/D) = \sum_{i=1}^{i} log \frac{P(w_{ij}/Pert)}{P(w_{ij}/Non\,Pert)} \qquad (6)$$

On trouvera dans ***[Rob77]*** les formules utilisées pour calculer la similarité entre une requête et un document. Retenons seulement que Robertson propose aussi des formules permettant de se passer de l'intervention de l'utilisateur.
Parmi les applications du modèle probabiliste, citons le modèle 2-Poisson développé par Robertson et Walker ***[Rob94]*** ou bien encore moteur de recherche Okapi ***[Oka94, Wal97].***

Avantages :
- fondements théoriques plus sophistiqués ;
- utilisation facile des modèles.

Inconvénient :
- tendance à pousser l'utilisateur à évaluer des probabilités inconnues ou à construire des modèles.

A.4 Modèle logique :

En 1986, van Rijsbergen ***[Van86, Van86']*** a modélisé la pertinence d'un document par rapport à une requête au moyen d'un principe d'incertitude logique. Soient deux formules logiques d et q (les représentations du document et de la requête), une fonction de comparaison entre d et q mesure l'incertitude qui existe dans l'implication $\boldsymbol{d \Rightarrow q}$ relative à un ensemble de données K. Cette fonction détermine l'extension minimale à apporter à K pour établir la preuve de $\boldsymbol{d \Rightarrow q}$

Un document est considéré comme un ensemble de "phrases" interprétées dans une certaine sémantique définie. Il en est de même pour la requête, qui est généralement formée d'une seule phrase.

Ce principe, dans sa forme la plus générale, exprime la mesure de la correspondance entre le document et la requête comme suit : si une ou plusieurs phrases impliquent la requête, cette dernière est considérée comme satisfaite. Les données K correspondent à l'ensemble de la connaissance dont on dispose au moment de l'évaluation de l'implication.

Si l'implication ne peut pas être satisfaite par rapport à cet ensemble d'informations, il est nécessaire d'ajouter de nouvelles informations de façon à ce que l'implication devienne satisfaite par rapport à l'ensemble d'informations étendu. L'extension minimale signifie qu'au cours de la mesure d'incertitude, seules les informations nécessaires à la satisfaction de $\boldsymbol{d \Rightarrow q}$ sont ajoutées.

Ce modèle est théorique, et pour le mettre en œuvre il faut en définir les éléments clés : les formules logiques d et q, l'implication$\Rightarrow$, et l'ensemble de données K.

Dans le modèle de base proposé par van Rijsbergen, l'implication représentée par $\Rightarrow$ peut être toute implication définie et elle ne se limite pas à l'implication logique $\boldsymbol{d \Rightarrow q}$. On peut voir le modèle logique comme un méta-modèle de recherche d'informations, dont les instances sont opérationnelles.

Par exemple, le formalisme des graphes conceptuels ***[Sow84]*** a été proposé dans ***[Che92, Mec95, Oun98]*** comme formalisme opérationnel pour le modèle logique.

A.5 Requêtes à base de questions :

Le langage booléen est parfaitement adapté au traitement informatique de données.

Cependant, il n'offre pas l'étendue des possibilités d'expression du langage naturel. C'est pourquoi des recherches sont poursuivies dans le domaine de la reconnaissance du sens d'une phrase posée en langage naturel et de sa traduction dans une syntaxe plus compréhensible par un outil de traitement automatique.

Une première piste consiste à étendre le langage booléen à un autre langage plus développé. Un des plus connu et utilisé est certainement WebSQL. Ce langage a été introduit par Mihaila ***[Mih96].*** Il se base sur un formalisme similaire à celui développé dans le langage SQL d'interrogation de base de données.

WebSQL considère ainsi le Web comme une base de données relationnelle composée de deux tables : des documents et des liens hypertextes. La table des documents est composée d'un t-uple pour chaque document présent sur Internet - un t-uple comprend une url, un texte, un type de document, . . . - et la table des liens hypertextes d'un t-uple pour chaque lien de chaque document du réseau. Les requêtes peuvent alors être formulées sur cette base de données virtuelle d'une manière similaire à la syntaxe employée dans le langage SQL. La figure A.3 montre l'exemple d'une requête réalisée à l'aide de ce langage demandant deux documents d'URL différentes dont les titres sont les mêmes et dans lesquels figurent les mots «something interesting».

Ce langage a été implémenté dans une application réelle afin de démontrer la faisabilité d'un tel procédé ***[Aro97].***

Une autre piste, plus difficile, consiste à analyser une question posée directement en langage naturel. Par exemple, le moteur de recherche doit être capable de répondre à la question : *«Quels sont les films à l'affiche ce soir ?».*

```
SELECT d1.url, d2.url
FROM   Document d1 SUCH THAT d1 MENTIONS "something interesting",
       Document d2 SUCH THAT d2 MENTIONS "something interesting"
WHERE d.title = d2.title
AND NOT (d1.url = d2.url)
```

Figure A.2 : Exemple de requête formulée dans le langage WebSQL.

Kwok a proposé dans ***[Kwo01]*** un système se basant sur ce principe qu'il a nommé Mulder. Pour arriver à ses fins, Mulder traite tout d'abord la requête - en langage naturel - et construit un arbre en fonction de la structure de la phrase. Ce système est dépendant des règles de formation des phrases spécifiques à une langue.

Cet arbre est ensuite analysé par un classifieur qui détermine le type de réponse attendue : nominale, numérique ou temporelle demandant respectivement une réponse sous la forme d'une phrase, d'un nombre ou d'une date.

L'étape suivante consiste à formuler une ou plusieurs requêtes dans un langage compréhensible par les moteurs de recherche actuels. Ces requêtes sont composées de termes devant être contenus dans la réponse. En effet, un moteur de recherche classique est un indexeur. Lorsque l'on veut connaître la réponse à une question, il faut chercher la réponse dans une base de données et non pas la question relative à cette réponse. Ainsi, pour l'exemple donné dans le paragraphe précédent, on devrait idéalement reformuler la requête d'interrogation en «Les films à l'affiche ce soir sont».

Pour réaliser cela, le système Mulder reformule la question en utilisant des règles spécifiques à la langue comme un système d'extraction et de conjugaison de verbes.

Le système crée de cette manière plusieurs requêtes allant d'une forte spécialisation, utilisant beaucoup de termes de la question initiale, à une requête générale, contenant très peu de termes. Ces différentes requêtes permettent de s'affranchir du domaine traité et d'obtenir des réponses sans que celles-ci ne soient trop vagues - caractérisées par de très nombreuses réponses. En effet, si la spécialisation est importante et donc si le nombre de termes de la requête est trop important, il est possible qu'un moteur de recherche ne puisse fournir le moindre résultat.

Une fois les résultats des moteurs de recherche obtenus, Mulder produit un résumé en sélectionnant les zones des documents retournés dans lesquelles figurent les termes de la requête initiale. Ce système permet de laisser l'utilisateur apprécier la justesse de la réponse.

Mulder a montré son efficacité en obtenant des résultats en moyenne six fois plus rapidement que les moteurs de recherche classiques. Mais il est nécessaire de produire des règles de transformation de question pour chaque langue prise en compte, ce qui représente un travail relativement fastidieux.

Chaque moteur de recherche emploie son propre système de requête, du plus simple au plus complexe. Cependant, l'utilisateur final, n'est pas forcément apte à utiliser n'importe quel système. En effet, un moteur de recherche classique ne s'adresse pas à un public de spécialistes mais à l'ensemble des utilisateurs d'Internet, du plus novice au plus expert. Des études statistiques ont été réalisées sur l'ensemble des requêtes formulées sur des moteurs de recherche utilisés par des millions d'utilisateurs à travers le monde afin de vérifier les aptitudes des utilisateurs et leur comportement vis-à-vis de la manière dont leur sont fournis les résultats. La section suivante présente ces résultats sur deux moteurs de recherche très utilisés : Altavista et Excite.

A.6 Modèles hybrides :

Il y a des modèles (*extended boolean*, booléen flou, etc.) qui cherchent à combiner les avantages de la structure du booléen avec la pondération du vectoriel. De plus, des raffinements ont été apportés à la pondération *tf.idf*, par exemple, afin de prendre en compte la taille des documents et leur fréquence dans la collection, sur des fondements probabilistes. Finalement, le concept général de rétroaction de pertinence peut être exploité dans d'autres approches que le probabilisme.

A.7 Discussion sur ces modèles :

Le formalisme des mots clés a l'avantage d'être simple et facile à mettre en œuvre. Il pose cependant un problème d'ambiguïté des termes d'indexation. En effet la polysémie et la synonymie des mots clés limitent la précision de l'indexation. La polysémie est la propriété d'un mot qui a plusieurs sens. Le mot "pompe", par exemple, est un terme polysémique. En effet, il peut désigner un appareil (pompe à vélo) comme il peut exprimer le mot chaussure en argot. La synonymie désigne le fait que plusieurs mots peuvent avoir le même sens. Par exemple, "bicyclette" et "vélo" sont des synonymes.

L'indexation sémantique latente (ou LSI, "Latent Semantic Indexing") est une extension du modèle vectoriel. C'est une méthode d'indexation automatique dont l'objectif est de représenter le contenu sémantique (ou conceptuel) des documents.

Le LSI est fondé sur une décomposition en valeurs propres pour générer la matrice d'association terme-document, et construire un espace "sémantique" dans lequel les termes et les documents proches sont placés les uns à côté des autres. Le but est de palier les problèmes de synonymie et de polysémie, pour améliorer le rappel et la précision de la recherche. Des termes qui n'apparaissent pas dans un document peuvent quand même être proches du document, si cela est consistant avec les schémas principaux d'association dans les données. L'interrogation s'effectue en utilisant une requête pour identifier un point dans l'espace sémantique, afin de retourner les documents dans le voisinage de ce point.

A.8 Note historique :

Il est intéressant d'observer qu'il y a un lien étroit entre le modèle booléen et la théorie de l'information ! En effet, l'une des premières observations importantes dans la carrière de Shannon fut que la théorie de George Boole pouvait s'appliquer aux machines électroniques (d'où le concept de « bit » qui nous est si familier). C'est sur cette base qu'est née la théorie de l'information.

Annexe B :

Quelques Systèmes de recherche d'images :

B.1 Classe orienté-contexte :

La première génération de systèmes proposés de gestion de base d'images a adopté le paradigme orienté-contexte pour une raison évidente : les techniques de description et de gestion des informations textuelles sont étudiées depuis longtemps par la communauté scientifique.

Sur le WEB, plusieurs systèmes ont été développés afin de proposer des interfaces de recherche d'images. **AMORE** (Advanced Multimedia Oriented Retrieval Engine) est un moteur de recherche d'images sur le WEB qui a adopté les paradigmes orienté-contexte et orienté-contenu. Il permet de rechercher des images grâce à des *mots-clés, des thèmes,* et des *comparaisons d'images*. Nous détaillons ici seulement la technique qu'utilise AMORE pour décrire une image par le contexte.

Le contexte de l'image sur le WEB n'est pas facile à trouver car il n'y a pas de norme stricte de description d'image.

AMORE propose une interface classique d'interrogation (Figure B.1). La méthode de recherche consiste à utiliser des mots-clé avec la possibilité de préciser s'il y a une similarité sémantique, graphique [1], ou les deux à la fois, entre la requête de l'utilisateur et les images du système. Les images issues du résultat peuvent être triées en fonction des mots-clé utilisés ou des sites concernés par la requête. AMORE utilise plusieurs composants pour rechercher les images :

- Glimpse (http://glimpse.cs.arizona.edu/) pour sa recherche textuelle. Glimpse est un système d'indexation et d'interrogation sous UNIX permettant une recherche aisée de fichiers.

- Harvest Web Indexing (http://www.tardis.ed.ac.uk/harvest/) pour récupérer les informations sur les sites. Il s'agit d'un ensemble d'agents qui ont pour objectif de chercher automatiquement des informations sur le WEB.

- WordNet pour la partie sémantique de sa recherche. WordNet est un dictionnaire disponible sur le WEB, spécialisé dans les domaines médicaux, techniques, juridiques ainsi que dans le marketing et les affaires.

[1] *AMORE exploite également les caractéristiques physiques de l'image telles que la couleur, la texture, la brillance, etc.*

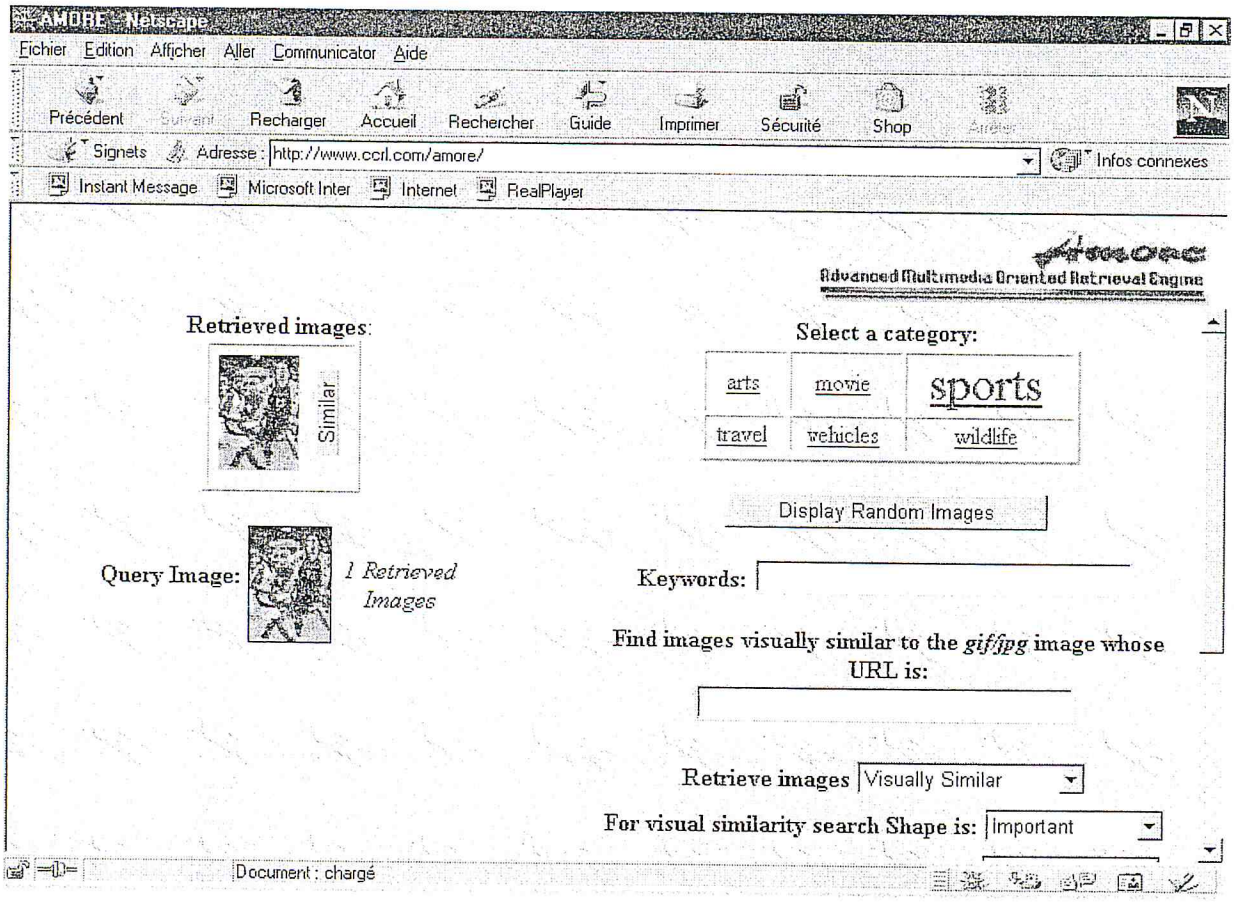


Figure B.1 : Interface graphique proposée par AMORE.

B.2 Classe orienté-contenu :

Plusieurs systèmes de recherche d'images sont maintenant disponibles sur le marché. Certains sont commercialisés avec une démonstration sur le WEB, d'autres restent en phase expérimentale. Nous avons choisi certains systèmes et prototypes parmi les plus connus :

Le **système Q.B.I.C.** (Query By Image Content) développé dans le centre de recherche IBM d'Almaden, est considéré comme le système de recherche d'images le plus connu. Il gère non seulement une collection d'images hétérogènes, mais aussi des vidéos. Q.B.I.C. existe en plusieurs versions : démonstration (http://wwwqbic.almaden.ibm.com/), standalone ou application, partie intégrante des autres produits IBM tel que DB2 Digital Library. Dans Q.B.I.C., la description du contenu de l'image est automatique. Elle est réalisée par le biais des procédures de reconnaissance de formes, d'extraction des contours des objets de l'image, des textures et de l'histogramme de couleurs. Q.B.I.C. propose également de décrire le contenu de l'image par le biais de mots-clé.

Le processus d'évaluation des requêtes calcule les mesures de similarité (en pourcentage) entre les différentes caractéristiques (croquis, couleur, texture, etc.) de l'image telle qu'elle est décrite par la requête et les caractéristiques physiques des images de la base. Pour une base de 1000 images, le temps de réponse à une requête est compris entre 2 et 40 secondes environ.

Le résultat d'une requête est restitué sous la forme d'une liste d'images classées par ordre décroissant de pertinence. La dernière version de Q.B.I.C. utilise d'autres techniques d'indexation et montre des améliorations au niveau de l'interface Homme-machine.

Le **système Art Museum** concerne un ensemble de tableaux d'art situé dans un musée japonais digitalisé. L'image est représentée par un ensemble de vecteurs contenant :

- L'ensemble des objets délimités par leurs contours.
- L'ensemble des couleurs dominantes. Ce système justifie la validité de ce choix par le fait que la vision humaine se focalise en priorité sur les couleurs dominantes d'une image.

Pour rechercher une image, le système Art Museum utilise une interface visuelle (QVE : Query by Visual Example) qui permet à l'utilisateur de dessiner le contour de l'objet à rechercher.

Le **système VIRAGE,** développé par la société Virage Incorporation, est un système commercialisé connu. Présenté sous la forme d'un ensemble de modules, Virage peut être intégré à n'importe quelle application. En particulier, Oracle, dans ses deux versions 8i et 9i, intègre le système Virage pour rechercher l'image par ses caractéristiques physiques (couleur, structure et texture). Virage existe en version d'évaluation (http://www.virage.com/online/). Il a été adopté par Altavista's AV Photo Finder (http://image.altavista.com/) pour permettre la recherche d'images par son moteur de recherche.

Le **système Excalibur** développé par la société Excalibur Technologies, offre une variété de techniques d'indexation et de correspondance basées sur la reconnaissance de formes. Une démonstration existe sur le Web (http://www.excalib.com/). Le catalogue et moteur de recherche Yahoo a intégré Excalibur dans Yahoo ! Image Surfer (http://isurf.yahoo.com/). Ce dernier permet ainsi une recherche d'images sur le WEB.

Une grande panoplie de systèmes expérimentaux ont été développés dans l'objectif de prouver la faisabilité des nouvelles techniques de recherche d'images. **Photobook** propose une interrogation par le biais de la couleur, la texture, la forme, et autres caractéristiques physiques. Une démonstration en ligne est disponible sur http://wwwwhite.media.mit.edu/vismod/demos/photobook/. Le prototype a été intégré en plusieurs applications dont FaceID de Viisage Technology (http://www.viisage.com). Une autre application considérée comme la plus connue est **VisualSeek**. Développée à l'université de Colombie, elle permet une interrogation basée sur la couleur, la forme, l'aspect spatial, et les mots-clés.

WebSeek est un système qui permet, par le biais d'agents, de rechercher des images sur le WEB à travers les caractéristiques physiques. Les images sont également indexées par mots-clé. Une démonstration en ligne existe sur http://disney.ctr.columbia.edu/WebSEEk/.

Surfimage est un prototype développé à l'INRIA (http://wwwsuntim.
inria.fr/htbin/syntim/surfimage/surfimage.cgi/). Il permet de rechercher des images en combinant leurs caractéristiques physiques. Il propose également un retour de pertinence (Relevance Feedback) pour mieux interagir avec l'utilisateur.

Les approches symboliques intègrent le contenu sémantique des images, le plus souvent le sens qu'elles dénotent. Un grand nombre de ces approches utilise des mots clés, comme c'est le cas pour les documents textuels.

B.3 Classe orienté-sémantique :

Depuis dix ans environ, l'apparition des systèmes de gestion d'images qui ont adopté le paradigme orienté-sémantique a beaucoup augmenté. La plupart de ces systèmes utilise les mots-clé comme moyen de description d'image. Nous citons : Rivage ***[Hal 89]***, QBIC, iBase (http://www.ibase.com), Index+ (http://www.ssl.co.uk),
Digital Catalogue (http://www.imageres.com), Fastfoto (http://www.picdar.co.uk/index2.html), FotoWare (http://www.fotoware.com), etc.

Le système EMIR², une extension de RIME, propose de rechercher des images par le contenu physique et sémantique. Le modèle de représentation d'images proposé dans EMIR² est inspiré de. Il utilise la notion de "vue". Deux vues de l'image sont mentionnées : physique et logique. La vue physique représente l'image brute. Elle tient compte des types d'images et elle offre, en plus de la visualisation et du stockage, une panoplie d'opérations, tels que la rotation, la translation, le changement d'échelle, etc. Quant à la vue logique, elle est subdivisée en plusieurs sous-vues :

- ***La vue structurelle*** : représente l'image comme une décomposition d'objets. Cette vue ne tient compte que d'un seul type de relation : la relation de composition *"CONTIENT"*.

  L'image est décrite sous une forme arborescente par un graphe d'objets, jusqu'aux objets élémentaires qui sont généralement des figures géométriques simples.

- ***La vue spatiale*** : permet d'affecter aux objets de l'image des descripteurs géométriques, et aux liens entre les objets des relations spatiales indiquant leur position et leur orientation dans l'image. A ce stade, chaque objet est associé à une représentation de sa forme par le biais d'une combinaison d'éléments géométriques (point, segment, polygone).

- ***La vue perceptive*** : est destinée à décrire les caractéristiques graphiques de chaque objet, comme la texture, la brillance et la couleur.

- ***La vue symbolique*** : donne une interprétation à l'image *via* des objets symboliques. A chaque objet symbolique est associé un ensemble d'attributs décrivant le contenu de l'objet en question, et un ensemble de relations. Ces dernières peuvent être de nature spatiale ou sémantique.

Un autre système très intéressant est **IM-DBMS**. Dans IM-DBMS, deux bases séparées sont utilisées pour stocker les images : d'une part, la base physique pour le stockage des images, et d'autre part la base logique pour le stockage des descripteurs des images. Notons que chaque image, dans ID-DBMS, appartient à un domaine qui définit la structure de tous les éléments de l'image. Chaque élément y figurant possède ses propres caractéristiques qui dépendent du domaine choisi. L'image est représentée de la manière suivante :

$$Image = \{E_i, \{P_j\}\}_{DK}$$

Où DK représente le domaine utilisé, E_i représente un élément ou un objet, et P_j représente une caractéristique.

Le système ELF est un système de gestion d'une base de photos d'art. La description de l'image est constituée d'un ensemble d'objets, de leurs caractéristiques et de leurs relations.

$$Image = \{\{O_I, \{R_K\}\}$$

Où O_I représente un objet et R_K représente une relation avec d'autres objets.

Chaque objet appartient à une classe se trouvant dans une hiérarchie qui lie toutes les classes par la relation de généricité. Par exemple, Les objets « Homme », « Femme » et « Enfant » sont des objets de la classe générique « Etre humain ». Les relations de généricité sont définies de la façon suivante : Etre Humain (Homme), Etre humain (Femme), etc.

Basé sur le modèle relationnel, le système ELF utilise les notions :

- **Objets** : permettent de décrire les caractéristiques d'un objet. Par exemple :
  Maison = (Image, Couleur, Position, etc.).

- **Associations** : permettent de représenter d'une part, la relation de généralisation et, d'autre part, les relations qui existent entre les objets dans une image. Par exemple la relation EST-PRES-DE (FEMME, MAISON, IMAGE1) montre que la femme est près de la maison dans l'image 1.

Le modèle relationnel utilisé dans ELF a été modifié afin de permettre l'utilisation d'un nom de relation considéré comme un attribut dans une autre relation (les relations de généralisation et de positionnement).

Annexe C :

La loi de Zipf :

• Si on classe les mots dans l'ordre décroissant de leur fréquence, et on leur donne un numéro de rang (1, 2, …), alors:

Rang.FréqRel ˜ constante
r .pr = A

–pr = n/N
-n : fréq. du terme de rang r
-N : nombre total d'occurrence
-A ˜ 0.1.

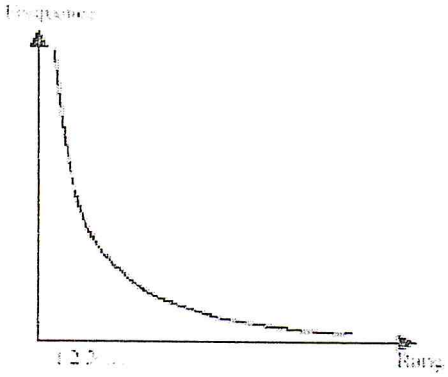
Exemple : N = 37,309,114 *[Mul]*

word	freq.	r	pr(%)	r.pr
the	2,420,778	1	6.488	0.0649
of	1,045,733	2	2.803	0.0561
to	968,882	3	2.597	0.0779
a	892,429	4	2.392	0.0957
and	865,644	5	2.32	0.116
in	847,825	6	2.272	0.1363
said	504,593	7	1.352	0.0947
for	362,865	8	0.975	0.078
that	347,072	9	0.93	0.0837
was	293,027	10	0.785	0.0785
on	291,947	11	0.783	0.0861
he	250,919	12	0.673	0.0807
is	245,843	13	0.659	0.0857
with	223,846	14	0.6	0.084
at	210,064	15	0.563	0.0845
by	209,586	16	0.562	0.0899
it	195,621	17	0.524	0.0891
from	189,451	18	0.508	0.0914
as	181,714	19	0.487	0.0925
be	157,300	20	0.422	0.0843
were	153,913	21	0.413	0.0866
an	152,576	22	0.409	0.09
have	149,749	23	0.401	0.0923
his	142,285	24	0.381	0.0915
but	140,880	25	0.378	0.0944

word	freq.	r	pr(%)	r.pr
has	136,007	26	0.365	0.0948
are	130,322	27	0.349	0.0943
not	127,493	28	0.342	0.0957
who	116,364	29	0.312	0.0904
they	111,024	30	0.298	0.0893
its	111,021	31	0.298	0.0922
had	103,943	32	0.279	0.0892
will	102,949	33	0.276	0.0911
would	99,503	34	0.267	0.0907
about	92,983	35	0.249	0.0872
i	92,005	36	0.247	0.0888
been	88,786	37	0.238	0.0881
this	87,286	38	0.234	0.0889
their	84,638	39	0.227	0.0885
new	83,449	40	0.224	0.0895
or	81,796	41	0.219	0.0899
which	80,385	42	0.215	0.0905
we	80,245	43	0.215	0.0925
more	76,388	44	0.205	0.0901
after	75,165	45	0.201	0.0907
us	72,045	46	0.193	0.0888
percent	71,956	47	0.193	0.0906
up	71,082	48	0.191	0.0915
one	70,266	49	0.188	0.0923
people	68,988	50	0.185	0.0925

Top 50 words from 84,678 Associated Press 1989 articles
(37,309,114 word occurrences, lowercased, punctuation removed, 266MB)

• La distribution de mots suit la courbe :



• On ne peut pas garder tous les mots les plus fréquents. On définit un seuil S_{MAX} sur la fréquence: si la fréquence d'occurrence d'un mot dépasse ce seuil, alors il n'est pas considéré comme important pour le document

• Hypothèse : On admet que l'utilisation de ces deux seuils S_{MIN} et S_{MAX} donne une indication sur l'informativité des mots.

• Cette notion définie non précisément en RI est censée mesurer la quantité de sens qu'un mot porte.

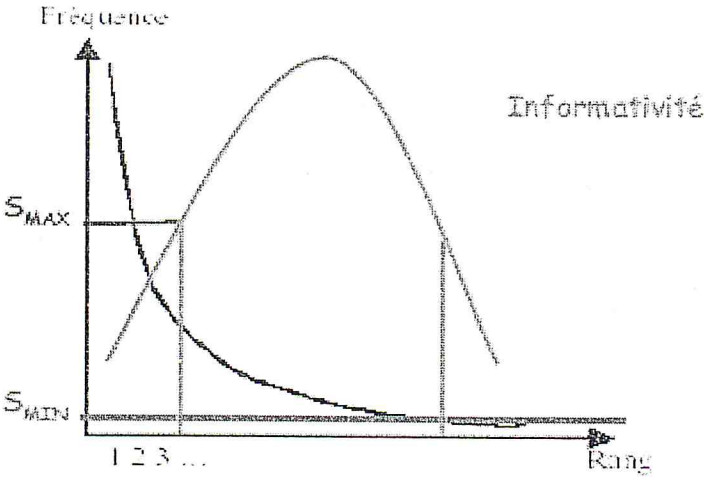


Figure C.1 : S_{MIN}, S_{MAX} et informativité

Références Bibliographiques

[Ada74] G. Adamson and J. Boreham. The use of an association measure based on character structure to identify semantically related pairs of words and document titles. Information Storage and Retrieval, 10 :pages 253–60, 1974.

[Aro97] Gustavo O. Arocena, Alberto O. Mendelzon, et George A. Mihaila. Applications of the Web query language. Computer Networks and IDSN Systems, 29(8-13): 1305-1315, 1997.

[Ann08] *G. Ann, Y. Janszen, B. Thevenet. Recherche de similarités de documents décrits par les mots-clefs, 2008.*

[Bou06] M. Boughanem. Recherche d'information. Université Paul Sabatier de Toulouse Laboratoire IRIT

[Bou09] M.Boughanem. Un nouveau passage à l'échelle en recherche d'information, publié dans « Ingénierie des Systèmes d'Information (ISI) 11, 4 (2006) 9-35 », Version 1-8 Feb 2009.

[Bri98] S.Brin et L.Page. The anatomy of a large-scale hypertextual Web search engine. In journal of Computer Networks and ISDN Systems (30), pages 107-117, Brisbane, 1998.

[Chb01] R.Chbeir. Modélisation de la description d'images : Application au domaine médical. Institut national des sciences appliquées, Lion. 14 décembre 2001.

[Cle66] C.W. Cleverdon, J. Mills, et M. Keen. Factors Determining the performance of Indexing Systems. ASLIB Cranfield Research Project, 1996.

[Cro02] C. Crouch, D. Crouch, Q. Chen, and S. Holz. Improving the retrieval effectiveness of very short queries. Information Processing and Management, 38 : pages 1–36, 2002.

[Dio05] I. Dioleti. Modélisation et expérience pour l'indexation symbolique d'images sur le web. Université Joseph Fourier – Grenoble I, 21 juin 2005.

[Fox92] C. Fox. Lexical analysis and stoplists, pages 102–130. Frakes WB, Baeza-Yates R (eds) Prentice Hall, New jersey, 1992.

[Fra92] W. B. Frakes. Stemming Algorithms, pages 131–160. Frakes W B, Baeza-Yates R (eds) Prentice Hall, New jersey, 1992.

[Kad99] C. Kaddour et S. Aissa Brahim. « Compression des Images Fixes par Fractales basée sur la Triangulation de Delaunay et la Quantification Vectorielle, **CHAPITRE I**: Généralités Sur Le Traitement D'images », Université des SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENE INSTITUT D'INFORMATIQUE ALGER. 1999

[Kwo01] Cody C. T. Kwok, Oren Etzioni, et Daniel S.Weld. Scaling question answering to the web. In World Wide Web, pages 150-161, 2001.

[Luk63] J. Lukasiewicz. Elements of Mathematical Logic. Pergamon Press, 1963.

[Luh57] H. Luhn. A statistical approach to mechanized encoding and searching of literary information. IBM, 1(4) :pages 309–317, 1957.

[Mar60] M. Maron and J. Kuhns. On relevance, probabilistic indexing and information retrieval. Journal of the Association for Computing Machinery, 7 :pages 216–244, 1960.

[Mar04] J. Martinet. Un modèle vectoriel relationnel de recherche d'information adapté aux images. Université Joseph Fourier – Grenoble I, 22 décembre 2004.

[Mec95] M. Mechkour. Un modèle étendu de représentation et de correspondance d'images pour la recherche d'informations. Thèse de Doctorat, Université Joseph Fourier, Grenoble, 1995.

[Mih96] George A. Mihaila. WebSQL – an SQL – like query language for the world wide web. PhD thesis, Université de Toronto, 1996.

[Mul] P. Mulhem. Recherche d'information (RI) – Fondements - , CLIPS-IMAG

[Nie94] J.Nielsen. Usability Engineering. Morgan Kaufmann, San Francisco, 1994.

[Oka94] S. Robertson, S. Walker, S. Jones, and M. H.-B. andM. Gatford. Okapi at TREC 3. In Proceedings of the 3rd Text REtrieval Conference (TREC-3), pages 109–126, 1994.

[Oun98] I.Ounis. Un modèle d'indexation relational pour les graphes conceptuels fondé sur une interpretation logique. Thèse de doctorat, Université Joseph Fourier, Grenoble, 1998.

[Pic04] F. Picarougne. "Recherche d'information sur Internet par algorithms évolutionnaires". Université François Rebelais Tours Ecole Doctorale : Santé, Sciences et Technologies. 19 novembre 2004.

[Por80] M. F. Porter. An algorithm for suffix stripping. Program 14, 1980.

[Pri99] Y. Prié. Modélisation de documents audiovisuels en Strates Interconnectées par les Annotations pour l'exploitation contextuelle. 1999.

[Rob77] S. Robertson. The probability ranking principle in IR. Journal of Documentation, 33(4) : pages 294–304, 1977.

[Rob94] Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Aarron Gull, et Marianna Lau. Okapi at TREC. In Text Retrieval Conference, pages 21–30, 1992.

[Roc71] J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, The SMART retrieval system : Experiments in automatic document processing, pages 313–323. Englewood Cliffs, NJ : Prentice Hall, 1971.

[Sal70] G. Salton. The SMART retrieval system : Experiments in automatic document processing. Prentice Hall, 1970.

[Sal71] G. Salton. The smart retrieval system. experiment in automatic document processing. Prentice Hall, 1971.

[Sal83] Gerard Salton, Edward A. Fox, et Harry Wu. Extended boolean informationretrieval. Commun. ACM, 26(11) :1022–1036, 1983.

[Sau05] K. Sauvagnat. Modèle flexible pour la Recherche d'Information dans des corpus de documents semi-structurés. Université Paul Sabatier de Toulouse, 30 Juin 2005.

[Tol06] S. Tollari. Indexation et recherche d'images par fusion d'informations textuelles et visuelles. Université du Sud Toulon-Var, 24 octobre 2006.

[Van79] C. J. Van Rijsbergen. Information Retrieval, 2nd edition. Dept. Of Computer Science, University of Glasgow, 1979.

[Wal97] S. Walker, S. Robertson, M. Boughanem, G. Jones, and K. S. Jones. Okapi at TREC-6 automatic and ad hoc, VLC, routing, filtering and QSDR. In Proceedings of TREC-6, pages 125–136, 1997.

[01] http://www.iro.umontreal.ca/nie/IFT6255/Introduction.pdf

[02] http://archee.qc.ca

[03] http://www.gerbeaud.com/creation/img_num/imagenum.htm

[04] aLaide.com

[05] http://benhur.teluq.uqam.ca

[06] http://www.pc-informatique.com

[07] Manuel PHP : Préface [archive] sur PHP.net [archive]. Consulté le 7 novembre 2007

[08] Nouveau modèle objet de PHP 5 [archive] sur PHP.net [archive]. Consulté le 7 novembre 2007

[09] http://www.commentcamarche.net/contents/php/phpintro.php3

[10] http://www.siteduzero.com/tutoriel-3-14425-introduction-a-php.html#ss part 1

[11] http://www.siteduzero.com/tutoriel-3-14438-presentation-de-mysql.html

[12] http://www.infos-du-net.com