Chapitre 4 Recherche d'information au sein de données complexes

Résumé. Dans ce chapitre, nous décrivons le fonctionnement général d'un système de recherche d'information (RI). Nous abordons également le cas de la recherche d'information à partir du Web. Nous traitons ensuite le cas de la recherche d'information au sein de deux médias : l'image et le texte. Nous proposons deux méthodes. La première est décrite dans le cadre de la RI visuelle mais adaptable à d'autres médias. Elle utilise les graphes de voisinage. Leur propriété de symétrie permet la mise en œuvre d'une navigation plus intuitive au sein des réponses fournies. La seconde traite des données textuelles, données de haut-niveau à caractère sémantique important. Elle est basée sur le processus d'ECT qui permet par induction et interaction de mieux prendre en compte l'aspect subjectif contenu dans la requête initiale.

Mots clefs : Recherche d'information, graphe de voisinage, navigation, interaction, apprentissage supervisé.

4.1 Problématique de la recherche d'information

Le volume des sources de données double chaque année. La croissance exponentielle des capacités de stockage et la forte diminution du coût du méga octets (TREND, 1999), voir Figure 4-1, font certainement partie des multiples causes de cet accroissement de données. La plupart d'entre elles est disponible à travers des réseaux comme les intranet et extranet pour une communication intra- ou inter-sites d'une entreprise ou l'internet pour une communication plus large. A leur échelle, ces documents sont la première source d'informations de leur groupe d'utilisateurs. Face à cette densité en ligne, il est de plus en plus difficile d'exploiter efficacement ces informations (MAES, 1994). De plus, LEFEVRE (2000) rapporte qu'une étude dans les pays développés montre que l'information se compose pour 20% de données structurées (e.g. annuaire, base de données relationnelle), et pour 80% de données non structurées (e.g. textes, images). Dès lors, la nécessité de disposer de méthodes et d'outils permettant l'exploitation des données non structurées, et par généralisation complexes, est évidente.

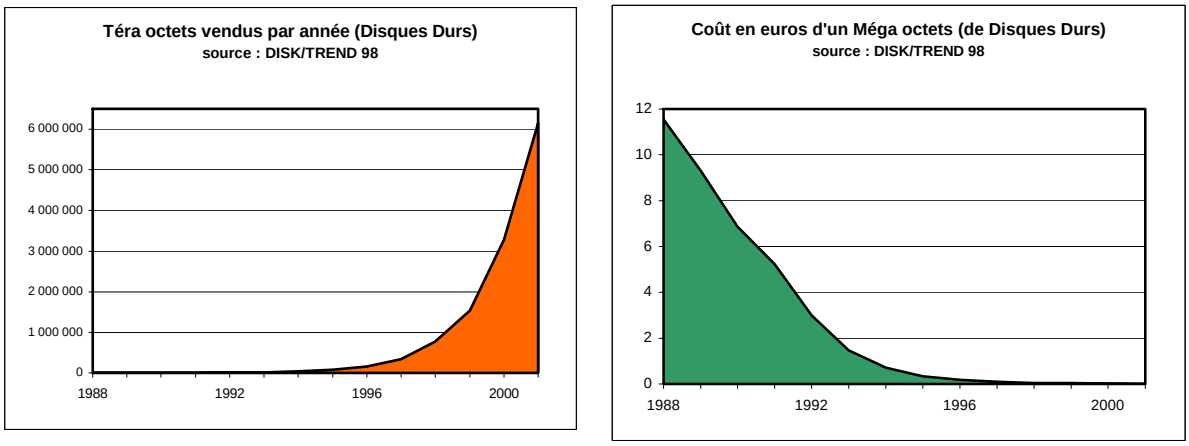


Figure 4-1 Capacité de stockage : croissance (gauche), diminution du coût (droite)

D'après HAND *et al.* (2001), dans le cadre de bases de données hébergeant les données structurées, la notion usuelle de requête est bien définie : c'est une opération visant à retourner les individus (ou enregistrements) ayant exactement les caractéristiques définies dans la requête. A contrario, dans le cadre de données complexes, les réponses nécessitent d'être plus générales et moins précises. Par exemple, dans le cadre de l'étude d'un dossier médical d'un patient, un médecin peut s'intéresser à la recherche de patients similaires pour comparer les diagnostics et les traitements afin de l'aider dans l'élaboration de son propre diagnostic. Pour de tels traitements, les méthodes doivent se baser sur des similarités et

non sur l'exactitude des relations, et la difficulté principale réside dans la définition de la similarité à partir de données de nature différente.

Par ailleurs, dans un tel processus, l'utilisateur détient un rôle prédominant puisque de simple demandeur, il devient également le juge des résultats proposés par le système. De ce constat découlent deux implications. La première est la nécessité de la visualisation et contextualisation des données pour permettre à l'utilisateur de juger les résultats. La seconde est la prise en compte du jugement de l'utilisateur afin que de manière interactive, le système propose des documents de plus en plus pertinents face à l'interprétation de la requête faite par l'utilisateur.

Dans un premier temps, dans la section 4.2, nous allons présenter des notions générales en recherche d'information : l'architecture usuelle de tels systèmes et leur évaluation. Ensuite, en section 4.3, nous aborderons le cas du web recouvrant deux aspects de la recherche d'information : l'utilisation des méta-données et l'accès aux données elles-mêmes. Dans cette section, nous y détaillerons l'utilisation de méta-données. Enfin, nous nous intéresserons au traitement de deux médias particuliers : l'image (section 4.4) puis le texte (section 4.5). Nous y détaillerons deux systèmes de recherche traitant ces médias que nous avons réalisés. Nous discuterons alors de ces systèmes et de leurs performances.

4.2 Principes élémentaires

4.2.1 Architecture d'un système de RI

Selon AMINI (2001), un système de recherche d'information (RI) textuelle se décompose en quatre principales composantes (Figure 4-2). Cependant, nous discuterons davantage de sa généralisation aux documents complexes.

- La première composante, **l'indexation**, a pour but de définir une représentation du corpus *D* permettant le calcul de similarité entre deux individus. Cet aspect ayant été abordé au Chapitre 1 dans la section 1.3, nous ne rentrerons pas dans les détails. Nous rappelons seulement que l'étape de représentation consiste à associer à un document d_i où $d_i \in D, \forall i \in [1,n]$, une description vectorielle $\vec{d_i}$ de dimension *p*, constante pour tous les documents de *D*. De même que pour d_i, à la requête *r* est associée la description vectorielle $\vec{r}$ de dimension *p*. La requête est de même nature que les documents du corpus ;

- La deuxième, **l'appariement**, consiste à évaluer à partir de ces représentations vectorielles la pertinence des documents par rapport à la requête à l'aide d'une mesure de similarité. La difficulté réside dans l'implémentation efficiente de l'algorithme de recherche puisqu'il doit pouvoir réagir en temps réel ;
- La troisième, **la décision**, doit permettre l'interprétation des résultats. Au Chapitre 3, nous avons décrit un outil de visualisation de données complexes dont l'objectif était justement d'aider l'utilisateur à interpréter un document se basant notamment sur sa contextualisation ;
- La dernière, **le jugement**, consiste à définir un module de mise à jour permettant de reformuler la requête en prenant compte des documents que l'utilisateur a jugés pertinents.

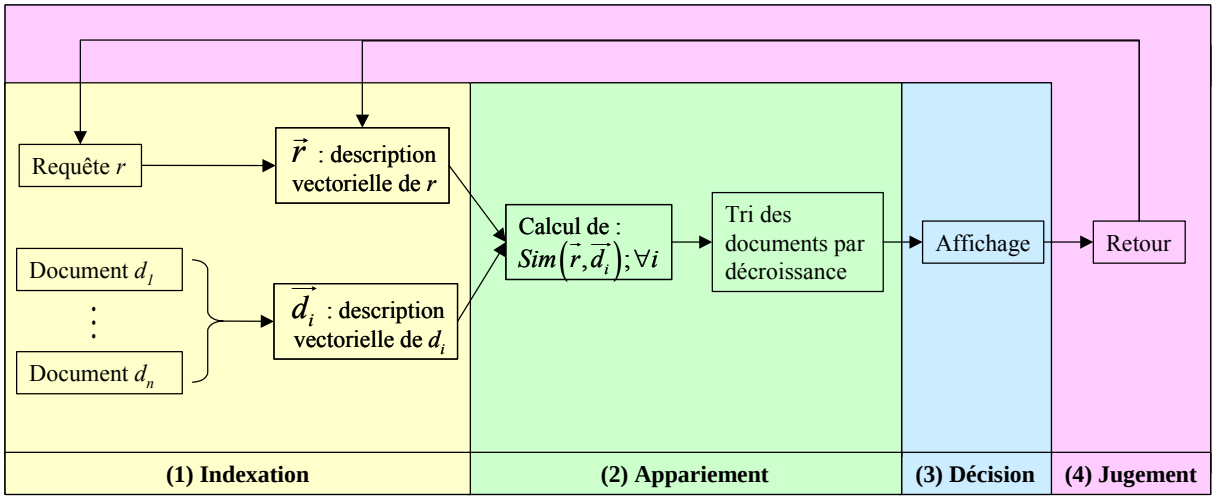


Figure 4-2 Architecture générale d'un système de RI

4.2.2 Evaluation des performances

L'évaluation des performances d'un système de RI est un problème majeur sur lequel la communauté de RI a consacré beaucoup de ressources (AMINI, 2001). L'intérêt est de pouvoir comparer des systèmes de RI entre eux à l'aide de critères objectifs. Selon VAN RIJSBERGEN (1979), la définition de ces critères doit permettre de rendre compte de la capacité du système à satisfaire l'utilisateur. Ceci revient à définir des moyens objectifs permettant la quantification d'une notion profondément subjective, ce qui est difficile. CLEVERDON *et al.* (1966) ont effectué les premières expériences et proposèrent un ensemble de quantités mesurables, dont particulièrement le temps moyen d'exécution du système pour répondre à une requête, l'effort dépensé par l'utilisateur pour l'obtention de réponses à sa recherche ainsi que le rappel et la précision du système.

Bien que partielles, les mesures de rappel et précision sont à la base de la majorité des mesures utilisées dans l'évaluation. En ce sens, le système de RI est perçu comme étant un classifieur binaire prédisant pour une requête *r* un ensemble réponse noté $R(r)$, sous-ensemble du corpus *D*, constitué de documents pertinents et non-pertinents. Nous notons par $D^{per}(r)$ et respectivement $R^{per}(r)$ l'ensemble des documents pertinents par rapport à la requête *r* au sein du corpus *D*, respectivement $R(r)$. De même, $D^{non}(r)$ et respectivement $R^{non}(r)$ désignent l'ensemble des documents non-pertinents par rapport à la requête *r* respectivement au sein du corpus *D* et au sein de l'ensemble réponse $R(r)$.

En adaptant les notations des formules de rappel et précision définies dans un contexte de catégorisation (équations (2.4.5) et (2.4.6) du Chapitre 2) à un cadre binaire, le rappel d'une requête *r*, noté $\rho(r)$, est la proportion de documents pertinents retournés par le système par rapport à l'ensemble des documents pertinents du corpus. De même, la précision $\pi(r)$ est la proportion de documents pertinents parmi les documents retournés par le système. D'où les équations (4.2.1) et (4.2.2):

$$\hat{\rho}(r) = \frac{\left|R^{per}(r)\right|}{\left|D^{per}(r)\right|} \tag{4.2.1}$$

$$\hat{\pi}(r) = \frac{\left|R^{per}(r)\right|}{\left|R(r)\right|} \tag{4.2.2}$$

Plusieurs problèmes pratiques se posent alors :

- La pertinence des documents doit être définie pour la requête utilisée et est généralement effectuée par un expert. Les performances d'un système ne pouvant être jugées par une seule requête, il faut définir un jeu de requêtes ainsi que les documents jugés pertinents pour chacune de ces requêtes ce qui implique un travail lourd et coûteux en temps et en ressource. C'est en ce sens que sont organisées les conférences internationales TREC et françaises AMARYLLIS (LESPINASSE, 1997 ; LESPINASSE *et al.*, 1999 ; TREC, 2000), mettant à disposition des corpus et des jeux de requêtes ;
- La pertinence d'un document est supposée booléenne, i.e. le document est jugé comme répondant à la requête ou hors sujet. Cependant, il arrive qu'un document soit partiellement pertinent (e.g. un paragraphe d'un texte), auquel cas la pertinence devrait être une probabilité définissant un ordre sur les documents (LEFEVRE, 2000) ;

- Sur de très grandes bases, dont le web représente l'extrême, la notion de rappel perd son sens car le besoin des utilisateurs est plutôt d'avoir une très bonne pertinence sur les dix ou vingt premiers documents (LEFEVRE, 2000). A cet effet, deux types de courbe sont utilisés : la courbe de précision en fonction du nombre de documents extraits (les cinq premiers, les dix premiers, …) et la courbe de précision et de rappel (VAN RIJSBERGEN, 1979).

Une alternative pour l'évaluation de ces systèmes est d'aborder la RI sous l'angle de la catégorisation. La requête est alors un document préalablement extrait du corpus et les réponses souhaitées sont les documents appartenant à la même catégorie que le document requête. Cet aspect ayant été décrit dans le cadre de l'apprentissage supervisé au Chapitre 2, nous n'y reviendrons pas.

4.3 Recherche d'information à partir du web

4.3.1 Introduction

Comme nous l'avons énoncé en introduction de ce chapitre, la quantité de données électroniques a formidablement augmenté ces dernières années. Le Web est devenu un support naturel et facile d'accès pour les entreprises, institutions et particuliers se traduisant par l'accroissement du nombre de sites (Figure 4-3).

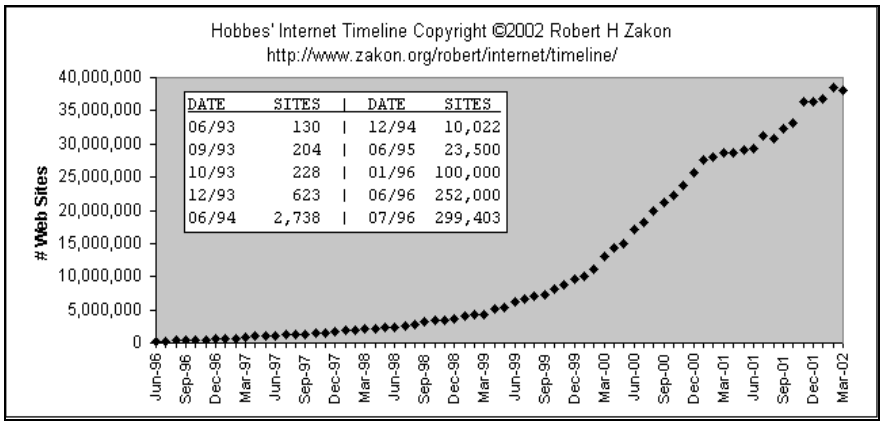


Figure 4-3 Evolution du nombre de sites web

Nous abordons ici la recherche d'information à partir du Web pour plusieurs raisons :

- Les documents web qu'il héberge sont des contenus complexes, cœur des préoccupations de nos travaux. Ces données sont structurées (e.g. chiffres dans des tableaux), non-structurées (e.g. images) mais également semi-structurées (e.g. textes décrits par des langages à balises) ;
- Les documents web évoluent au cours du temps. Les outils de traitements des données temporelles et longitudinales devraient apporter leurs lots de résultats comme par exemple dans le cadre de la veille stratégique de documents ;
- La totalité du Web peut être perçue comme l'aboutissement, en terme de contenu, d'une extraordinaire base documentaire, où les outils de RI peuvent fournir également des résultats intéressants ;
- Enfin, avec une vision systémique, le Web peut être perçu comme une formidable organisation « sociale » ou sémantique offrant quantité de méta-données traitables par les outils issus de la théorie des graphes et dont les résultats se font déjà ressentir dans l'utilisation quotidienne des dernières générations de moteurs de recherche.

De part ces raisons et bien d'autres encore, la nécessité de disposer de méthodes et d'outils avancés permettant d'y remédier a fait émerger de nouveaux domaines de recherches dont notamment celui du web mining vers le début des années 90 (COOLEY *et al.*, 1997). ETZIONI (1996) le définit comme étant l'utilisation des techniques de data mining afin d'automatiser la recherche et l'extraction de connaissances et services web.

Le web mining se décompose en trois sous-domaines (KOSALA et BLOCKEEL, 2000) :

- Le web content mining qui extrait de l'information à partir du contenu des documents stockés sur le réseau dans un but essentiellement de filtrage ;
- Le web structure mining qui extrait de l'information en utilisant la structure des hyperliens d'un réseau afin de déterminer les sites références dans une thématique ;
- Le web usage mining qui extrait de l'information à partir des habitudes de navigation stockées sur les serveurs hébergeant les sites afin de personnaliser le service en adaptant les performances en fonction de la fréquentation, en adaptant automatiquement le contenu des pages en fonction de l'utilisateur.

Le web usage mining sort du contexte de ce chapitre puisqu'il concerne la personnalisation automatique du contenu d'un site en fonction de l'utilisateur ainsi que son organisation en fonction des habitudes des utilisateurs dudit site ; pour une description d'ordre général voir BAZSCALICZA *et al.* (2001).

Par ailleurs, bien qu'ils soient plus larges, nous n'aborderons les deux autres sous-domaines que sous l'angle de la recherche d'information dans un soucis de cohérence.

Dans un premier temps, nous définissons en section 4.3.2 le terme document web et introduisons nos notations. Nous abordons ensuite dans la section 4.3.3 les particularités du contenu des documents web et leurs utilisations principales. Enfin, en section 4.3.4, nous nous intéressons en quoi la structure d'un réseau peut nous aider pour rechercher de l'information sur un tel support.

4.3.2 Définitions et notations

Document web. Un document web est une ressource électronique stockée sur un serveur web, possédant un identifiant universel nommé URL (Uniform Ressource Locator). Ce document est défini à l'aide d'un langage à balises et accessible en utilisant le protocole HTTP (Hyper Text Transfer Protocol).

Langage à balise de mise en forme. Les langages à balises de mise en forme comme le HTML (Hyper Text Markup Language) définissent la mise en forme d'un document textuel, permettent d'y inclure d'autres ressources possédant une URL comme des images, de la vidéo, du son et de définir des hyperliens vers d'autres documents web.

Hyperlien. Un hyperlien est une relation non symétrique entre deux documents web *a* et *b*. Nous noterons $a \to b$ cette relation.

Hyperlien de sortie, hyperlien d'entrée. Soit *d* un document web. Le nombre d'hyperliens contenu dans *d* est appelé le nombre d'hyperliens de sortie et noté $S(d)$. Par opposition, le nombre d'hyperliens pointant vers le document *d* est appelé nombre d'hyperliens d'entrée et noté $E(d)$. Nous ferons remarquer qu'à partir du seul document *d*, nous pouvons évaluer $S(d)$ mais pas $E(d)$ en raison de la non symétrie de la relation d'hyperlien. Nous utiliserons la notation simplifiée *E* et *S* lorsqu'il n'y aura pas d'ambiguïté.

Graphe orienté. Un graphe orienté est un graphe *G* (cf. 2.2, section 3.2.1.2) formé d'un ensemble de sommets noté Σ et où ces sommets sont reliés entre eux par un ensemble d'arcs orientés *H*. Le graphe orienté *G* est ainsi décrit par le couple (Σ, H).

Remarque : nous assimilons le Web à un graphe orienté $G = (\Sigma, H)$ où Σ est constitué de l'ensemble des documents web et *H* est constitué par les hyperliens entre les différents documents web.

4.3.3 Particularité et utilisation du contenu du Web

Le web content mining traite la problématique d'extraire de l'information à partir du contenu des documents web. Pour ce faire, il emploie les outils de recherche d'information classiques et ceux de

l'ECD. La recherche d'information dans le cadre du web se différencie de celle au sein d'une base documentaire principalement en trois points :

- Le premier concerne la temporalité du document. En effet, un site est mis à jour régulièrement tant sur le contenu que sur la forme, ce qui souvent tient plus de la « survie » du site pour ainsi essayer de conserver une « attractivité » pour les internautes ;
- Le deuxième concerne l'accès de la ressource. Cette dernière est stockée dans ce qui peut être perçu comme un large système de stockage distribué avec les aléas inhérents tels les pannes, la maintenance privant momentanément l'accès à une ressource ;
- Le dernier concerne tout simplement le format du document. En effet, le document est généralement écrit à l'aide de langages à balises comme HTML. Cela confère au document la capacité d'une part d'être multimédia (e.g. mélange de textes, d'images et de vidéos), mais surtout de contenir une riche information concernant la mise en forme du document grâce justement aux balises.

Les deux premiers aspects entraînent deux usages principaux du web : une recherche immédiate (section 4.3.3.1) et une veille documentaire (section 4.3.3.2). Le dernier entraîne une nouvelle représentation des documents (section 4.3.3.3).

4.3.3.1 Recherche immédiate

Une recherche immédiate s'effectue généralement à l'aide d'un moteur de recherche comme par exemple (ALTAVISTA) qui va retourner une liste de documents. Ce mode de consultation nécessite au fournisseur du moteur de recherche de parcourir régulièrement le web afin de mettre à jour ses index, d'indiquer la date de la dernière consultation ou de stocker le document. Ce mode de fonctionnement étant fortement similaire à la recherche d'information dans des bases documentaires (cf. 4.2.1), nous ne le détaillerons pas davantage.

4.3.3.2 Veille documentaire

La veille documentaire consiste à collecter toutes les informations liées à une thématique. La contrainte de temps est plus lâche que dans le cadre d'une recherche immédiate et peut s'étendre sur plusieurs jours. Nous parlons ici de veille documentaire et non pas de veille stratégique qui consiste à

rechercher toute nouvelle information, ce qui est une problématique bien distincte ; une large description de cette problématique est proposée dans SAMIER *et al.* (2002).

Dans le cadre de la veille documentaire, des agents de recherche intelligente (e.g. Harvest (BOWMAN *et al.*, 1995) ou FAQFinder (CHANG et HSU, 1997)) vont parcourir le Web durant une période fixée à la recherche de documents pertinents pour l'utilisateur. Nous décomposons un tel système en deux fonctions principales : la première est celle permettant l'élaboration d'un modèle prédisant si un document est intéressant pour l'utilisateur et la seconde est le parcours optimisé du Web par des agents.

Concernant la phase d'élaboration du modèle, les méthodes usuelles d'apprentissages vont bien entendu être utilisées. Cependant, cela implique à l'utilisateur de devoir constituer lui-même un corpus de documents l'intéressant et d'autres ne l'intéressant pas, et rendant de fait abscons l'intérêt d'une telle démarche. Pour pallier l'inconvénient de la constitution manuelle du corpus, nous utiliserons un ensemble de documents proposés par des moteurs de recherche répondant à des mots-clefs définis par l'utilisateur. Puis, à l'aide d'une démarche interactive dont l'interface est présentée en 4.5.3.3, l'utilisateur va indiquer des documents l'intéressant et d'autres l'intéressant moins. Un modèle sera alors construit puis appliqué sur les documents. L'utilisateur pourra affiner le modèle en évaluant les documents prédits comme intéressants par le modèle.

Grâce à ce modèle, l'agent de recherche sera capable de déterminer l'intérêt d'un document. Il nous reste donc à définir les règles de navigation de ces agents au sein du Web. Les hyperliens vont être naturellement sollicités. Lors de l'examen d'un document par un agent, les hyperliens de sorties sont stockés dans la mémoire de l'agent si les contraintes de validité d'exploration sont vérifiées. Ces contraintes, définies par l'utilisateur, ont pour objectif de limiter l'exploration du Web et sont essentiellement basées sur la notion de profondeur au sein d'un site ou sur l'obligation de rester dans un domaine particulier.

Pour optimiser le travail des agents de recherche, des agents collaboratifs peuvent être définis. Leur rôle sera de mieux gérer les agents de recherche. En effet, l'intérêt ne consiste pas à explorer la totalité du Web (ce qui prendrait du temps et des ressources considérables) mais d'explorer uniquement les zones susceptibles de contenir les informations souhaitées. De plus, une collaboration entre agents permettra également que deux agents n'analysent le même document. A partir des hyperliens de sortie retournés par les agents de recherche, les agents collaboratifs auront donc pour tâche d'ordonner aux agents de recherche de visiter des hyperliens particuliers. Ces hyperliens particuliers seront définis de

telle façon à maximiser les chances de trouver des documents intéressants. Une heuristique pour arriver à cet objectif est de privilégier la visite des hyperliens de sortie des documents jugés par le modèle comme étant pertinents.

4.3.3.3 Représentation des documents à partir de balises

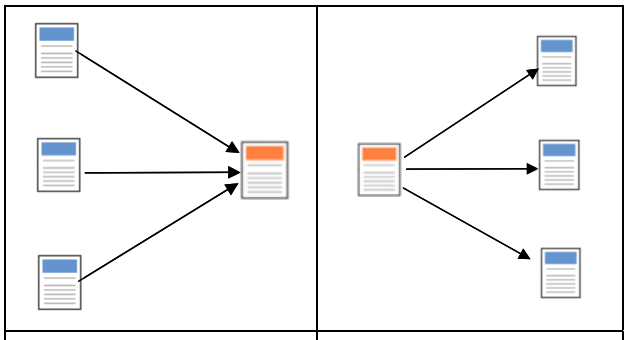
L'utilisation des balises de mise en forme peut être très riche en information et utilisable au niveau de la représentation des documents afin de permettre l'analyse du contenu de ces documents. Nous pouvons considérer ces dernières comme de la méta-information. De plus, il existe des méta-balises qui sont de véritables méta-informations (e.g. nom de l'auteur, date du document) mais ces dernières sont trop peu utilisées par les auteurs des pages HTML pour pouvoir généraliser leur utilisation. Néanmoins, l'écriture des pages à l'aide du langage à balise XML (eXtended Markup Language), visant à décrire cette fois-ci le contenu du document de façon normalisée, est en train de s'intensifier. Beaucoup de récents travaux s'orientent vers l'utilisation de cette richesse de méta-information ; citons par exemple (DUFFOUX *et al.*, 2004).

Nous avons abordé au Chapitre 1 les principales représentations des documents et notamment pour le cas des textes celles basées sur des vecteurs dans l'espace des mots. RIBEIRO *et al.* (2001) ont utilisé l'information contenue dans certaines balises pour modifier cette représentation. Ils définissent un ensemble de balises d'emphases révélant a priori de l'information importante : les balises de titres, de mise en italique, en gras… Par ailleurs, pour éviter que les documents sur-utilisant ces balises soient favorisés, la pondération du terme se fait en fonction du nombre total d'éléments mis en emphase. Ainsi, la représentation du document se base d'abord sur une pondération usuelle concernant l'ensemble du document, puis est complétée par une pondération d'emphase, concernant uniquement le vocabulaire mis en emphase.

4.3.4 Utilisation de la structure du Web

Le web structure mining s'intéresse à la structure topologique du sous-graphe étudié. Nous définissons ce sous-graphe comme étant le graphe orienté $G = (\Sigma, H)$. Un tel graphe reflète selon CHAKRABARTI *et al.* (1999) une organisation sémantique sous-jacente. De ce fait, l'hyperlien $a \to b \in H$ offre davantage qu'une relation entre les documents *a* et $b \in \Sigma$, mais une approbation du contenu de *b* de la part de l'auteur de *a*. En adoptant ce point de vue, les documents faisant autorité, ou Authority (Figure 4-4-a), sont ceux qui sont les plus souvent pointés, c'est-à-dire contenant un nombre important

d'entrées *E*. De même, un autre document atypique est le Hub (Figure 4-4-b) proposant un réseau important de liens, c'est-à-dire contenant un nombre important de sorties *S*.



(a) Authority (b) Hub

Figure 4-4 Exemples de Authority et de Hub

De nombreux travaux ont été réalisés dans ce sens, nous citerons principalement (BRIN et PAGE, 1998), (KLEINBERG, 1999), (CHAKRABARTI *et al.*, 1999) et (YANG *et al.*, 2002).

Nous allons détailler le fonctionnement de deux algorithmes particuliers PageRank et HITS utilisés dans les moteurs de recherche actuels, respectivement (GOOGLE) et (TEOMA).

4.3.4.1 PageRank

Les outils de recherche basés sur la similarité entre la requête et la représentation vectorielle des documents ne peuvent faire face à l'énorme quantité de documents hébergés sur le web puisque le critère de précision diminue par rapport au rappel (cf. 4.2.2). L'objectif de PageRank est d'augmenter la précision en utilisant la notion de document autorité afin de déterminer un score à un document au sein d'un réseau. Ainsi, lors d'une requête, les documents retournés par le système sont triés selon ce score avant d'être affichés.

Le PageRank peut être perçu comme la modélisation du comportement d'un utilisateur : un utilisateur aléatoire accède aux divers documents web en cliquant aléatoirement sur les liens, sans revenir en arrière et avec une certaine probabilité de changer de document de départ. La probabilité que ce « surfer aléatoire » visite un document *d* est appelée Page Ranking et notée PR_d (BRIN et PAGE, 1998).

Plus formellement, en reprenant les notations précédentes :

$$PR_{d_j}^{t+1} = (1 - att) + att \sum_{\substack{i=1 \\ d_i \to d_j}}^{n} \frac{PR_{d_i}^{t}}{S(d_i)} \qquad (4.3.1)$$

où *att*, appelé facteur d'atténuation, est une probabilité et généralement fixée à 0,85 et le *PR* des *n* documents est initialisé à 0. Le calcul se fait itérativement et une bonne estimation est obtenue en pratique au bout d'une vingtaine d'itérations.

Le principal inconvénient de cette méthode est que l'on peut artificiellement augmenter le *PR* d'un document. En effet, il suffit qu'un nombre restreint de sites de thématiques proches insère des hyperliens entre-eux pour augmenter leur *PR* respectif. En ce sens, plusieurs sociétés commerciales proposent d'augmenter le *PR* d'un site en appliquant ce principe.

4.3.4.2 HITS

L'un des inconvénients des Authorities, c'est qu'elles ne contiennent pas nécessairement le vocabulaire utilisé dans la requête, par exemple le vocabulaire commercial d'un site peut être fortement différent d'un vocabulaire technique d'une requête. De plus, les hyperliens n'ont pas tous la même nature : certains sont définis pour effectivement donner autorité à un document alors que d'autres sont là pour le seul besoin de la navigation entre les documents d'un site ou encore pour un besoin publicitaire. Ainsi, un document est considéré comme une « bonne » Authority (cf. Figure 4-4-a) lorsqu'il est pointé par des « bons » Hubs. De même, un « bon » Hub (cf. Figure 4-4-b) pointe vers de « bonnes » Authorities. C'est ce renforcement mutuel qui est à la base de HITS (KLEINBERG, 1999). L'algorithme se déroule en deux étapes :

- La première élabore un échantillon de documents représentatifs par rapport à la requête. Pour ce faire, un ensemble *Racine* est défini comme étant quelques centaines de documents retournés par un système standard de RI (cf. 4.2.1). Puis, une phase d'expansion est appliquée à cet ensemble *Racine* pour déterminer l'ensemble *Base* construit en ajoutant les documents désignés par les hyperliens des documents de l'ensemble *Racine* jusqu'à une profondeur choisie. C'est cette étape qui se propose de chercher les documents Authority ne contenant pas nécessairement le vocabulaire de la requête. Enfin, une phase de nettoyage est appliquée dans le but de supprimer les liens de navigations. Ainsi, tous les liens entre deux documents du même domaine sont supprimés.

- La seconde étape détermine pour chacun des documents leurs poids relatifs en tant qu'Authority et que Hub, notés respectivement x_d et y_d pour le document *d*. Le calcul s'effectue itérativement comme défini dans l'équation (4.3.2). Etant intéressé davantage par les valeurs relatives de ces poids, une pondération est effectuée à chaque itération afin de borner la somme de ces pondérations. KLEINBERG (1999) fixe comme contrainte de normalisation que la somme des carrées des poinds soient égale à 1. Enfin, comme conditions initiales, une constante est assignée à chacun des poids ne favorisant pas ainsi a priori un document plutôt qu'un autre.

$$x_{d_j}^{t+1} = \sum_{\substack{i=1 \\ d_i \to d_j}}^{n} y_{d_i}^{t} \;;\; y_{d_j}^{t+1} = \sum_{\substack{i=1 \\ d_i \to d_j}}^{n} x_{d_i}^{t} \qquad (4.3.2)$$

4.3.4.3 Bilan de PageRank et HITS

Même si ces deux algorithmes s'appuient sur le même paradigme, la perception du web comme un réseau social, leurs approches diffèrent. En effet, PageRank se focalise à retourner les Authorities, alors que HITS retourne également des Hubs, qui de facto ont un faible *PR*. Cet intérêt pour les Hubs peut se justifier lorsque l'utilisateur cherche à apprendre sur un sujet sans connaître les termes exacts. A contrario, ils sont inutiles lorsqu'il recherche précisément quelque chose (CHAKRABARTI *et al.*, 1999).

En outre, ces deux méthodes requièrent de parcourir régulièrement le réseau et de stocker les index de chacun des documents ainsi que les hyperliens. Cela implique évidemment la mise en place de structures importantes et donc coûteuses en ressources.

Enfin, un inconvénient commun aux deux méthodes de web structure mining présentées est de conférer un rang d'Authority indépendamment du contenu du document. Ainsi, lorsqu'un document traite plusieurs sujets, ces algorithmes n'indiquent pas pour lequel (lesquels) il a été défini en tant qu'Authority. Des extensions ont été proposées pour justement déterminer les thématiques qui font qu'un document soit une Authority en se basant sur son contenu, nous citerons par exemple (RAFIEI et MENDELZON, 2000) et (KUO et WONG, 2000). Ceci tend à prouver, si il en était nécessaire, que le web content mining et le web structure mining sont deux méthodes complémentaires dans le cadre de la recherche d'information sur le web.

4.4 Recherche d'information au sein d'images

4.4.1 Introduction

La recherche d'information dans de large bases de données d'images reste un challenge étant donné que les utilisateurs recherchent des images similaires d'un point de vue sémantique alors que les bases de données d'images proposent des images similaires basées sur les caractéristiques de bas-niveaux calculées à partir de la valeur des pixels. Comme pour un système de RI « classique », la RI visuelle implique l'utilisation d'index et de méthodes d'appariement usuellement basées sur les *k* plus proches voisins. Il y a actuellement deux approches principales pour réaliser l'indexation (RUI *et al.*, 1999), celle basée sur le contenu de l'image et celle basée sur les annotations textuelles décrivant l'image :

- L'indexation basée sur le contenu suppose que l'information visuelle de chaque image (extraite à partir des valeurs des pixels) peut se résumer en un vecteur de caractéristiques telles celles décrites au Chapitre 1, section 1.3.2 (e.g. caractéristiques de texture). Dès lors, comme décrit en section 4.2.1, le processus de requête se réduit par la recherche des plus proches individus dans l'espace de représentation ainsi défini. Ces plus proches voisins sont définis à l'aide d'une mesure de similarité entre deux individus. Il existe pléthore mesures, comprenant des similarités usuelles comme la distance Euclidienne ou la distance de Mahalanobis (CHANDON et PINSON, 1981), des plus spécifiques liées aux caractéristiques des images comme celles basées sur la distribution des couleurs, sur la texture, les formes… (AIGRAIN *et al.*, 1996), ou encore une combinaison de plusieurs de ces mesures. Dans ce contexte, l'utilisateur définit sa requête à l'aide d'une image exemple qui sera elle-même vectorisée dans le même espace de représentation que la base documentaire ;
- L'indexation basée sur les annotations suppose que chaque image soit annotée à l'aide de mots-clefs, d'une phrase ou plus généralement en langage naturel. Les termes utilisés constituent alors les index et des mesures de similarités sont utilisées pour déterminer la ressemblance entre deux annotations. Contrairement aux caractéristiques de bas-niveau, les annotations expriment l'aspect sémantique des images, permettant ainsi d'accéder potentiellement à des notions abstraites. Dans ce cadre, l'utilisateur soumet sa requête en langage naturel.

La section 4.5.3 traitant de la recherche d'information au sein de données textuelles, nous n'allons pas aborder ici le cas de l'indexation basée sur les annotations. Dans la section qui suit, nous allons proposer une méthode de RI basée sur les graphes de voisinage. Puis, en section 4.4.3, nous présenterons les résultats d'une expérimentation effectuée à partir d'une base composée de 259 images comparant

l'approche des *k*-PPV de celle des graphes de voisinage. Enfin, nous discuterons sur ces méthodes en section 4.4.4.

4.4.2 Proposition d'une méthode basée sur le graphe de voisinage

4.4.2.1 Motivations

Dans cette section, nous nous intéressons au concept de « voisin – voisinage » d'un document au sein d'une base documentaire, travail ayant abouti à une publication (SCUTURICI *et al.*, 2003b). Ce système de recherche par le contenu ne se distingue pas par son mode d'indexation (vectorisation des images à partir de caractéristiques bas-niveau), mais par son mode d'appariement et son mode de navigation. En effet, comme nous allons le voir dans la sous-section suivante, l'appariement s'effectue en fonction du principe de voisinage des graphes de voisinage et il utilise les capacités de navigation de la carte contextuelle décrite au Chapitre 3, section 3.2.2.3.

La plupart des algorithmes en recherche documentaire se base sur les *k* plus proches voisins (kPPV) d'un document au sens d'une mesure de similarité. Par exemple, le système de RI visuel QBIC (FALOUTSOS *et al.*, 1994), dans son développement pour le musée de l'Hermitage (HERMITAGE, 1995), renvoie les 12 images les plus proches de la requête soumise par l'utilisateur. Cependant, comme nous l'avons montré au Chapitre 2, section 2.2.3, l'algorithme des kPPV produit des résultats surprenants par rapport aux attentes des utilisateurs en raison de leur non-symétrie. De plus, fixer la valeur de *k* représente un autre inconvénient puisque rien n'indique qu'une image ait un nombre fixe de voisins. Il n'est donc pas naturel de forcer une image à avoir *k* voisins si elle en comporte moins, et inversement de la restreindre à *k* voisins si elle en comporte davantage. Nous avons vu au Chapitre 2, section 2.3.1 que ces deux propriétés étaient présentes dans les graphes de voisinage. C'est pourquoi nous proposons de les utiliser dans un tel système.

4.4.2.2 Présentation de la méthode

La méthode d'appariement à partir de graphes de voisinage s'exécute en deux temps. Le premier est la construction du graphe de voisinage de la base documentaire. Cette étape, détaillée au Chapitre 2, section 2.3.3, Algorithme 2-3, est exécutée « hors ligne ». Elle consiste à calculer les dissimilarités entre les documents à partir de leur vecteur de caractéristiques. Puis, à l'aide d'un critère géométrique,

un graphe de voisinage est élaboré. Par exemple, la Figure 4-5-a représente la projection du graphe des voisins relatifs dans le premier plan factoriel d'une base fictive composée de huit images.

Le deuxième temps s'effectue « en ligne », c'est-à-dire lorsque l'utilisateur soumet une image requête au système. Nous noterons cette image requête *I*. L'objectif de cette étape est de rechercher les images de la base documentaire les plus ressemblantes à *I*. Il conviendrait dès lors de calculer le graphe de voisinage comprenant l'ensemble des images de la base plus celle fournie en requête. Bien évidemment, en raison de la complexité en $O(n^3)$ pour la construction du graphe, cette option n'est pas envisageable. En pratique, nous allons insérer « localement » *I* au sein du graphe de voisinage de la base, amenant des modifications locales de relations de voisinage (cf. Figure 4-5-b). Cette approche nécessite que le critère de construction du graphe soit local, ce qui exclu les graphes comme par exemple l'Arbre Minimum Recouvrant.

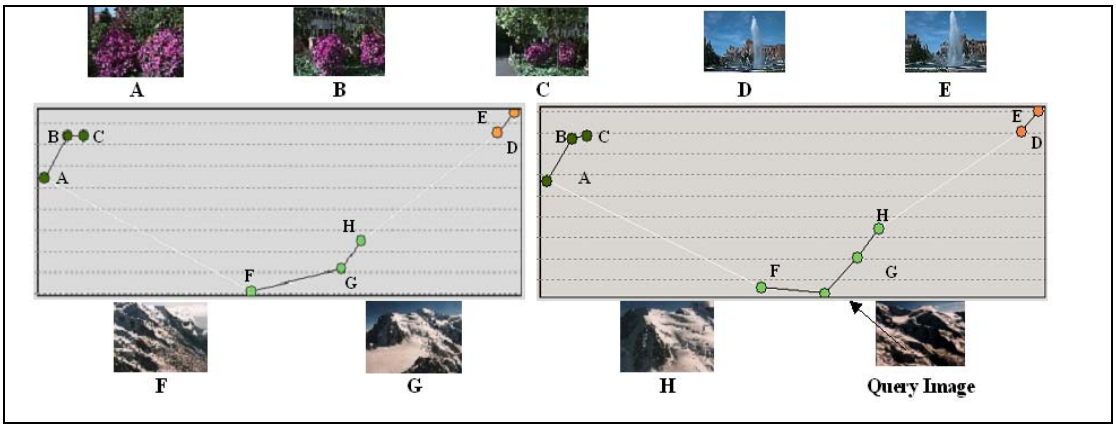


Figure 4-5 Le graphe des voisins relatifs.
La couleur des nœuds représente la catégorie sémantique d'appartenance. A gauche (a), sans l'insertion de l'image requête ; à droite avec (b)

Ainsi, lorsque *I* est inséré dans le graphe, les relations de voisinage sont modifiées localement. Il y a deux types de relations modifiées : celles établissant une relation de voisinage avec *I* et celles supprimées en raison de la présence de *I* qui empêchent la vérification du critère géométrique du graphe de voisinage. Dès lors, l'insertion locale se déroule en deux étapes, la première consistant à déterminer l'ensemble des points candidats aux modifications du voisinage local et la seconde consistant à mettre à jour les relations de voisinage concernées par les points candidats. Ces étapes sont détaillées respectivement en sous-sections 4.4.2.2.1 et 4.4.2.2.2.

L'Algorithme 4-1 décrit les principales étapes de l'opération d'insertion de *I* au sein du graphe de voisinage $G = (\Sigma, A)$: le sommet *I* est rajouté à l'ensemble Σ des sommets de *G*, et la dimension de la matrice de dissimilarité *L* est mise à jour. Puis, les dissimilarités entre *I* et l'ensemble des documents de la base sont calculées et stockées dans *L*. Enfin, la recherche du contour de *I* permet les mises à jour des relations de voisinage stockées dans *A*.

Algorithme 4-1 Mise à jour du graphe par insertion locale

```
MAJ_Locale(entrée: I, n; entrées-sorties: Σ, A, L)
Début
    a. Σ ← Σ+I; Dim(L) ← Dim(L)+1;
    b. Calcul des distances entre I et les n documents de la base et mise à
       jour de L;
    c. Recherche du contour de I;
    d. Mise à jour de A;
Fin.
```

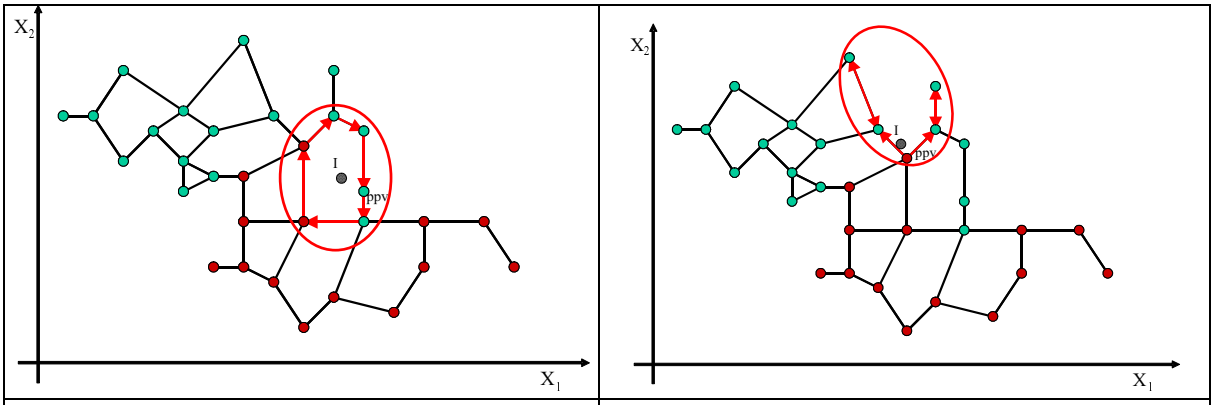
4.4.2.2.1 Recherche des points candidats

Par construction, les graphes de voisinage ne contiennent pas d'intersection entre deux arêtes (dans l'espace de construction $\mathbb{R}^p$). L'ensemble des points dont leurs relations de voisinage sont susceptibles d'être modifiées par l'insertion de *I* est donc l'ensemble des points formant le contour de *I*, noté $\mathcal{C}_I$. Il existe deux type de contours : des contours fermés (cf. Figure 4-6-a) et des contours ouverts (cf. Figure 4-6-b). Pour déterminer les points appartenant à $\mathcal{C}_I$, il suffit d'en connaître un puis de déterminer lequel de ses voisins est le plus proche de *I* et ainsi de suite. Nous rappelons qu'en accord avec nos notations, $\mathcal{V}_i$ est l'ensemble des voisins de *i*. Soit $i \in \mathcal{C}_I$ alors $j \in \mathcal{V}_i$ appartient au contour si il vérifie l'équation (4.4.1). Le critère d'arrêt est dans le cas d'un contour fermé la réalisation d'un cycle. Dans le cas d'un contour ouvert, si le point de départ est une des extrémités du contour, alors le critère est la réalisation d'un cycle, sinon de deux cycles. Pour définir $\mathcal{C}_I$, il faut appliquer l'équation (4.4.1) à l'ensemble des voisins des points formant ce contour, soit une complexité en $\mathrm{O}\left(\sum_{i \in \mathcal{C}_I} |\mathcal{V}_i|\right)$.

$$j = \underset{v \in \mathcal{V}_i}{\mathrm{Argmin}} \left[d(v, I) \right] \qquad (4.4.1)$$

Il nous faut maintenant rechercher à identifier un premier point du contour $\mathcal{C}_I$. En fait, ce point est obtenu par le « 1 Plus Proche Voisin », noté *ppv*, de *I*, qui satisfait les critères comme ceux du Graphe des Voisins Relatifs (GVR) et du Graphe de Gabriel (GG) :

- pour le GVR, l'hypersphère de rayon *d(I, ppv)* et de centre *I* est nécessairement vide (sinon ce ne serait pas le plus proche voisin de *I*), donc, par inclusion, la lunule $\mathcal{L}_{I,ppv}$ l'est également ;
- pour le GG, comme le GVR est inclus dans le GG, alors trivialement nous pouvons dire que le plus proche voisin est un des voisins de *I* au sens du GG.



(a) Points candidats dans un fermé | (b) Points candidats dans un ouvert

Figure 4-6 Exemple de recherches de points candidats
aux modifications du voisinage local

4.4.2.2.2 Mise à jour des relations de voisinage des points candidats

L'ensemble des points concernés par d'éventuelles mises à jour de relations de voisinage est celui du contour $\mathcal{C}_I$ et *I*. Il ne serait pas optimal de tester toutes les relations de voisinage possibles au sein de ce sous-ensemble. Il suffit d'une part de vérifier que *I* n'empêche pas la vérification du critère des relations existantes entre les points de $\mathcal{C}_I$, et d'autre part de définir les voisins de *I*.

Le nombre maximum d'arêtes dans un contour est au maximum égal au nombre de sommets qui le compose. Il y a donc au maximum $|\mathcal{C}_I|$ relations de voisinage à vérifier, soit une complexité de borne supérieur en $O(|\mathcal{C}_I|)$. Pour déterminer les voisins de *I*, il faut évaluer le critère sur l'ensemble des points de $\mathcal{C}_I$, à l'exception du *ppv* dont la relation de voisinage avec *I* est déjà établie, soit une complexité en $O((|\mathcal{C}_I|-1).|\mathcal{C}_I|)$. La complexité globale d'une insertion locale d'un point *I* est donc en $O((|\mathcal{C}_I|-1).|\mathcal{C}_I|)$.

4.4.2.3 Etude de la complexité de l'insertion locale

Afin d'avoir une indication sur les possibilités de passage à l'échelle de ce système, nous avons évalué le temps d'exécution d'une requête *I* au sein d'une base constituée de *n* documents sous certaines hypothèses en faisant varier différents paramètres. Nous avons supposé que le système reposait sur une machine équipée d'un processeur Intel Pentium 4 à 3Ghz. Cette dernière effectue un peu plus de 9.10^9 opérations à la seconde. Le temps total de réponse du système, hors coût de communication réseau, équivaut à la somme des étapes de l'Algorithme 4-1 :

1. L'extraction des *p* caractéristiques de l'élément requête. Cette étape dépend évidemment des caractéristiques utilisées et de leur nombre. Afin d'être générique, nous avons plutôt évalué le temps moyen pour extraire une caractéristique. En effet, le nombre de paramètres intervenant dépend de la nature du document comme le nombre de termes pour une requête textuelle et le nombre de pixels ainsi que la résolution utilisée pour les images. En outre, généralement plusieurs caractéristiques peuvent être calculées simultanément, ce qui diminue d'autant le coût. Ainsi, nous avons estimé un temps de 10^{-3} secondes par caractéristique ;
2. Le calcul des distances dans $\mathbb{R}^p$ entre la requête et les n documents de la base documentaire a une complexité en $O(pn)$;
3. La complexité de l'insertion est en $O\left(\sum_{i \in C_I} |\mathcal{V}_i|\right)$ et celle de la mise à jour locale du graphe est en $O\left(\left(|C_I|-1\right).|C_I|\right)$. Nous posons que v est le nombre moyen de voisins pour un point. De même nous posons que le contour est de taille équivalente au nombre moyen de voisins. La complexité de ces opérations est donc en $O\left(v^2 + v(v-1)\right) \approx O\left(v^2\right)$.

Nous avons ainsi fait varier le nombre *n* de documents au sein de la base, le nombre *p* de caractéristiques et le nombre moyen de voisins *v*. L'ensemble des résultats a été reporté dans le Tableau 4-1.

Principalement il en ressort que la solution semble viable car le temps d'exécution d'une requête au sein d'une base comprenant 100 millions de documents ayant une centaine de caractéristiques est de l'ordre de la seconde. Ainsi, comme pour les systèmes basés sur les *k*PPV, le goulot d'étranglement se situe au niveau du calcul des distances entre l'image requête et celles de la base. Cependant, l'emploi d'heuristiques comme l'utilisation des *K-d* Tree (BENTLEY, 1990) lors d'une recherche rapide (exacte ou approchée) du plus proche voisin pour les espaces à dimensions élevées (TSAPARAS, 1999), sont des pistes à explorer pour pallier cet inconvénient.

Les résultats de cette étude de complexité en temps sont intéressants dans le sens qu'ils montrent la capacité de passage à l'échelle de cette méthode. Néanmoins, nous nous sommes placés dans un cadre avantageux en nous soustrayant aux contraintes physiques des grandes bases de données. Ainsi, une

étude sur la complexité en espace tout comme la prise en compte des contraintes physiques s'avèrent nécessaires pour déterminer plus précisément les capacités réelles de cette méthode.

n	p	v	Calcul des distances [opérations]	Insertion et mise à jour locale [opérations]	Temps intermédiaire [s]	Temps extraction caractéristiques [s]	Temps total [s]
1 000 000	100	10	1,00E+08	1,00E+02	0,01	0,1	**0,11**
1 000 000	100	100	1,00E+08	1,00E+04	0,01	0,1	**0,11**
1 000 000	100	1 000	1,00E+08	1,00E+06	0,01	0,1	**0,11**
1 000 000	1 000	10	1,00E+09	1,00E+02	0,11	1	**1,11**
1 000 000	1 000	100	1,00E+09	1,00E+04	0,11	1	**1,11**
1 000 000	1 000	1 000	1,00E+09	1,00E+06	0,11	1	**1,11**
10 000 000	100	10	1,00E+09	1,00E+02	0,11	0,1	**0,21**
10 000 000	100	100	1,00E+09	1,00E+04	0,11	0,1	**0,21**
10 000 000	100	1 000	1,00E+09	1,00E+06	0,11	0,1	**0,21**
10 000 000	1 000	10	1,00E+10	1,00E+02	1,11	1	**2,11**
10 000 000	1 000	100	1,00E+10	1,00E+04	1,11	1	**2,11**
10 000 000	1 000	1 000	1,00E+10	1,00E+06	1,11	1	**2,11**
100 000 000	100	10	1,00E+10	1,00E+02	1,11	0,1	**1,21**
100 000 000	100	100	1,00E+10	1,00E+04	1,11	0,1	**1,21**
100 000 000	100	1 000	1,00E+10	1,00E+06	1,11	0,1	**1,21**
100 000 000	1 000	10	1,00E+11	1,00E+02	11,11	1	**12,11**
100 000 000	1 000	100	1,00E+11	1,00E+04	11,11	1	**12,11**
100 000 000	1 000	1 000	1,00E+11	1,00E+06	11,11	1	**12,11**

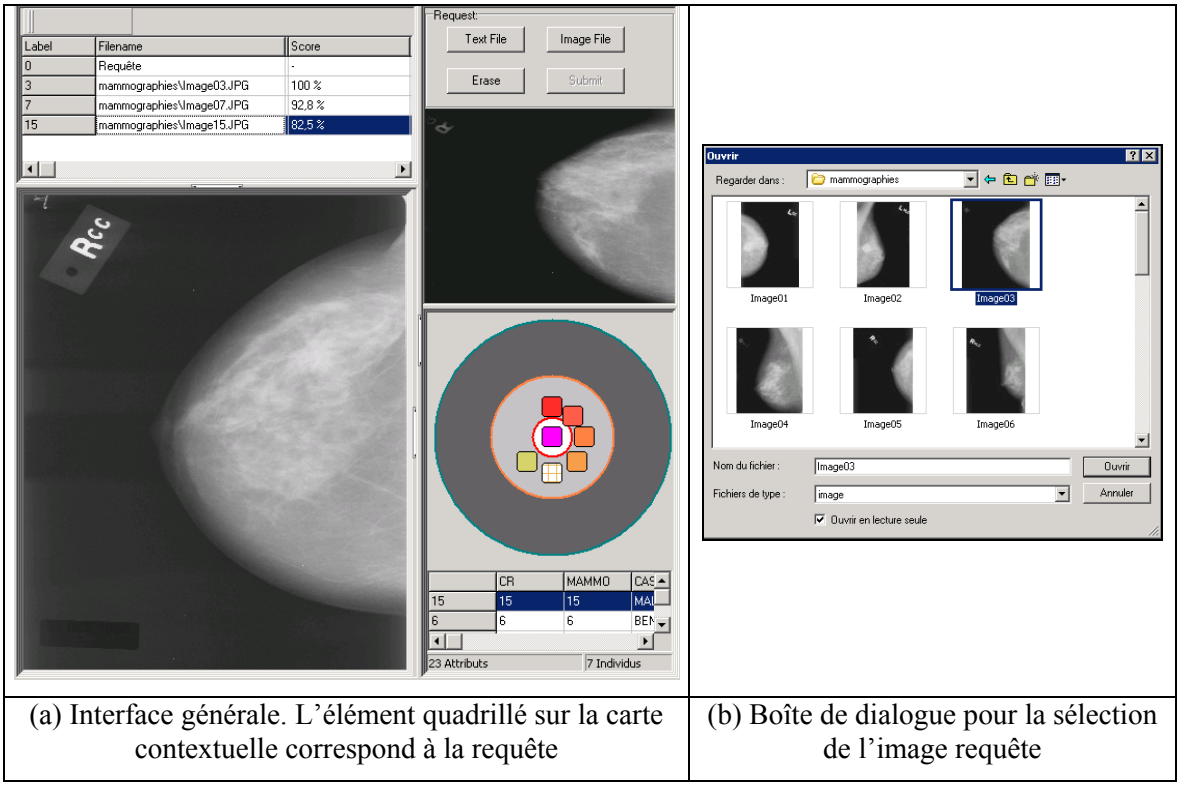
Tableau 4-1 Evaluation du temps de réponse d'un système de requête basé sur le voisinage avec un Pentium 4 à 3 Ghz

4.4.2.4 Proposition d'interface utilisateur

Une interface d'un système de RI doit répondre à deux impératifs. Le premier concerne la capacité de proposer un système de sélection de l'élément requête, en l'occurrence une image. Le second concerne l'affichage des résultats retournés par le système. Par ailleurs, nous proposons ici d'utiliser la carte contextuelle, présentée dans le cadre de la visualisation de données complexes, pour permettre une meilleure interprétation de l'individu courant et surtout une navigation contextuelle en utilisant la fonction de centrage.

La Figure 4-7 est une copie d'écran de l'interface implémentée. En reprenant notre exemple de dossier patient, le médecin souhaite rechercher des mammographies similaire à un cliché qui lui est difficile d'analyser. La boîte de dialogue permettant la sélection du cliché requête est accessible via le bouton « Image File » dans la zone supérieure droite de l'interface. Par ailleurs, sous cette zone est affichée l'image requête utilisée afin que l'utilisateur puisse la comparer avec les autres images retournées par le système. La partie supérieure gauche affiche les résultats de la requête avec le score associé. Lorsqu'un élément de cette liste est sélectionné, alors son image est affichée dessous cette zone, et la carte des voisins est centrée sur cet individu. La navigation contextuelle permet alors d'explorer les élé-

ments contenus dans la base documentaire en fonction des relations de voisinage établies. Enfin, pour mieux distinguer l'image requête au sein de cette carte, nous l'avons différenciée en quadrillant le carré la représentant.



(a) Interface générale. L'élément quadrillé sur la carte contextuelle correspond à la requête

(b) Boîte de dialogue pour la sélection de l'image requête

Figure 4-7 Interface du système de RI visuelle

4.4.3 Expérimentations

Dans notre expérience, nous avons utilisé un jeu de données de 258 images subdivisées en six catégories principales. Ces images ont été extraites de la base de données Ground Truth (WASHINGTON, 1999). Ces images sont essentiellement constituées de paysages et de monuments. Nous avons utilisé les catégories suivantes : « Arbogreens », « Australia », « Cherries », « SwissMoutains », « Greenlake » et « SpringFlowers ». Nous avons extrait à partir de ces images deux familles de caractéristiques : des caractéristiques de couleurs (L1 et L2 normalisées ; la couleur prédominante voir Chapitre 1, section 1.3.2.1) et des caractéristiques de texture (les 14 paramètres définis par HARALICK (1973)). Ici, nous avons utilisé des caractéristiques globales de bas-niveau.

Afin de comparer les résultats de notre système de recherche d'information basé sur le voisinage à un système basé sur les *k* plus proches voisins, nous avons comparé leurs taux de rappel et précision dans un contexte de catégorisation : déterminer la catégorie sémantique des images. Lorsque nous soumettons une requête au système, ce dernier retourne un ensemble de *k* réponses pour le kPPV et un ensemble de voisins pour le graphe de voisinage. Ces réponses fournies par les deux méthodes étaient pondérées par l'inverse de leur distance à la requête et constituaient un vote à la majorité permettant ainsi de définir la catégorie de la requête. Dans cette étude, nous avons utilisé la distance cosinus puisque cette dernière est invariante aux échelles des axes de l'espace de représentation (RAJMAN et LEBART, 1998). Nous avons fait varier *k* de 1 à 5 et les résultats de la prédiction ont été évalués après une 10 validation-croisée.

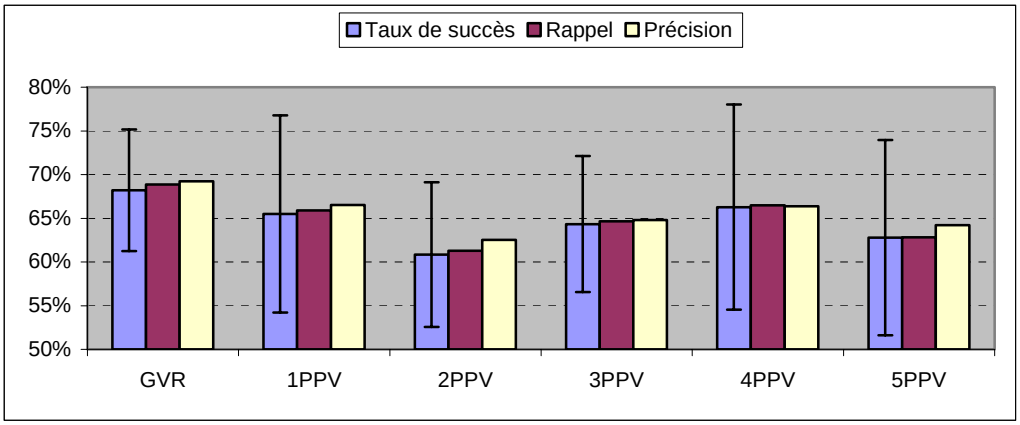


Figure 4-8 GVR et kPPV (*k* variant de 1 à 5)

La Figure 4-8 montre les résultats obtenus par nos expérimentations : taux de succès et écart-type, rappel macro-moyen et précision macro-moyenne. Nous observons que le GVR surpasse les différents modèles de plus proches voisins. Nous avons effectué les expériences jusqu'à *k=30*, mais les résultats étaient toujours inférieurs à ceux du GVR. Nous expliquons ces meilleurs résultats par l'utilisation d'un nombre de voisins dépendant de la disposition des données dans l'espace des caractéristiques qui doit être fortement bénéfique au GVR au niveau des points frontières alors que les kPPV utilisent un nombre constant de points quelle que soit la disposition des données.

4.4.4 Discussion

Les résultats montrent que sur cette base d'images, avec ces caractéristiques et en utilisant la métrique cosinus, le Graphe des Voisins Relatifs offre de meilleurs résultats que ceux des kPPV. Cependant, ils

n'indiquent aucunement que les graphes de voisinage sont « la » meilleure méthode. Plus modestement nous disons que le GVR apporte des résultats intéressants et comparables.

Un inconvénient de cette méthode est d'abord le coût calculatoire de la création du graphe de voisinage de la base documentaire. Néanmoins, nous avons montré qu'il n'était à effectuer qu'une seule fois, et nous travaillons sur un mode de construction incrémentale en utilisant le principe d'insertion locale. Puis, comme toutes les méthodes de RI visuelles basées sur le contenu, se pose le problème du choix des descripteurs et des similarités à utiliser. Ici, nous avons utilisé des caractéristiques globales, tout en sachant qu'elles ne rapportent qu'une petite part de l'information. Dans notre cas, cela est suffisant comme nous le montre cette expérimentation. Dans d'autres cas, cela est certainement insuffisant, notamment dans le cas de l'extraction d'information visuelle où l'on recherche plutôt des objets au sein d'images que l'image entière. En ce sens, nous souhaitons approfondir ces travaux en décrivant l'image (la vidéo) par les objets qui la composent et en utilisant les recommandations MPEG-7 (TELECOMITALIALAB).

Enfin, l'intérêt des graphes de voisinage est double. D'abord les relations de voisinage s'adaptent à la topologie des données offrant ainsi une représentation plus fidèle. Enfin, sa propriété de symétrie permet d'offrir un système de navigation par voisinage que nous pensons plus simple et plus intuitif.

4.5 Recherche d'information au sein de textes

4.5.1 Introduction

La recherche d'information textuelle n'est pas une discipline récente, car selon VAN RIJSBERGEN (1979) le problème du stockage de l'information et de sa recherche posait un intérêt croissant dès les années 40. Rapidement les travaux se sont concentrés sur les méthodes de sélection des documents et principalement les modèles booléens, probabilistes et vectoriels ont vu le jour :

- Le modèle booléen consiste à extraire les documents vérifiant la requête exprimée par des termes reliés par des opérateurs logiques (e.g. ET, OU, NON). L'inconvénient majeur de ces systèmes est qu'ils ne permettent pas le classement par pertinence décroissante (LEFEVRE, 2000) ;
- Le modèle probabiliste se base sur l'estimation des probabilités qu'ont les termes de la requête d'apparaître dans les documents. Pour des simplifications de calculs, ces méthodes supposent

l'indépendance des termes. Bien que cette hypothèse est en toute rigueur fausse, elle donne cependant des résultats intéressants (MARON et KUHNS, 1960 ; ROBERTSON, 1977) ;

- Le modèle vectoriel, introduit par SALTON *et al.* (1983), consiste à représenter un document par un vecteur comportant autant de composantes qu'il y a de termes informatifs. Ce système est le plus répandu actuellement.

A partir de ces modèles, les résultats sont usuellement retournés par la méthode des *k* Plus Proches Voisins. De même que dans le cadre de la RI visuelle, et pour les mêmes raisons, nous avons développé un système de RI textuelle basé sur les graphes de voisinage. La seule différence entre les deux systèmes concerne évidemment la représentation du corpus. Dans le cas du texte, elle peut se baser sur les mots, lemmes ou autres représentations présentées au Chapitre 1, section 1.3.4.1. Nous ne reviendrons pas sur ce système dans cette section puisque fonctionnant selon les mêmes principes que ceux définis en section 4.4.2.

Nos travaux se sont ensuite axés sur les procédés en amont avec notamment la mise en œuvre de méthodes de traitement de la langue naturelle faisant appel à des traitements linguistiques, et donc dépendant de la langue, comme l'utilisation de thésaurus ou de réseaux sémantiques tel WordNet (MILLER *et al.*, 1990). Leurs objectifs sont principalement une recherche plus pertinente des termes informatifs et d'éviter les écueils liés aux phénomènes linguistiques comme la synonymie ou l'homonymie. JACQUEMIN (2001) propose un outil efficace FASTR basé sur ces technologies et BOURIGAULT *et al.* (2001) font l'état de l'art des récentes avancées de la linguistique dans notamment le cadre de la RI.

Un autre axe de recherche se situe en aval avec les méthodes d'interactions avec l'utilisateur dans le but de prendre en compte le besoin non-exprimé dans la requête. Cet interaction s'effectue généralement par le jugement de l'utilisateur sur des documents proposés par le système. Les deux méthodes principales sont celle de ROBERTSON *et al.* (1976) et celle de ROCCHIO (1971) :

- La première ré-estime les poids des composantes de la requête en maximisant l'estimation de la probabilité qu'un document pertinent contienne un terme de la requête et en minimisant l'estimation de la probabilité qu'un document non-pertinent contienne ce même terme. L'inconvénient majeur de cette méthode est que le vocabulaire de la requête n'est pas modifié. Donc, si ce dernier n'est pas des plus appropriés, l'interaction n'apportera pas de gain important ;
- La seconde ré-estime également les poids des composantes de la requête, mais propose également l'expansion de la requête. Les nouveaux termes sont sélectionnés lorsqu'ils sont « suffisamment »

présents dans les documents jugés pertinents par l'utilisateur et « suffisamment » peu présents dans ceux jugés non-pertinents. Les expériences montrent que cette méthode apporte un gain en rappel et précision. L'inconvénient majeur est le choix des paramètres du modèle définissant les « suffisamment ».

Enfin, dans le courant de ces derniers travaux, des outils d'aide à la lecture ont été mis en place afin d'aider l'utilisateur dans sa recherche. Nous proposons en section 4.5.2 un tel outil.

Nous conclurons cette section par la proposition d'un nouveau système de RI textuelle en section 4.5.3, où la différence majeure consiste en la ré-interprétation de la requête en fonction du jugement de l'utilisateur. En effet, plutôt que de re-pondérer les termes de la requête ou même d'en ajouter, nous élaborons un modèle grâce aux méthodes d'apprentissage à partir de données textuelles, méthodes ayant déjà fait leurs preuves dans le domaine de la catégorisation.

4.5.2 Mise en relief de passages

Dans le cadre de la RI textuelle, les documents retournés par le système peuvent être de longs textes, pouvant être préjudiciable à l'utilisateur car entraînant un examen de beaucoup de données (SALTON *et al.*, 1996). Par ailleurs, comme AMINI (2001) le rapporte, O'CONNOR et RO (O'CONNOR, 1980 ; RO, 1988) montrent que si chaque portion de texte ou passage est triée dans l'ordre décroissant de leur pertinence par rapport à une requête, il est alors simple de mettre en place une interface Homme-Machine pour aider l'utilisateur à trouver l'information qu'il recherche. Ainsi, la mise en relief des passages (i.e. portions de texte) pertinents permet à l'utilisateur de se concentrer sur l'essentiel du document afin de lui permettre de le juger plus rapidement.

Cependant, le choix du passage reste délicat (SALTON *et al.*, 1993). Nous pensons qu'une mise en œuvre rapide est essentielle car le calcul de la pertinence des résultats doit être immédiat afin de ne pas retarder la lecture de l'utilisateur. De ce fait, nous cherchons un passage facilement détectable du point de vue informatique. LEBART *et al.* (1994, p. 35) définissent le paragraphe comme étant un ensemble de caractères situés entre deux retours-chariots, éléments informatiques facilement identifiables. Par ailleurs, VANDENDORPE (1995) affirme que pour être valide, un paragraphe doit avoir une unité sémantique : chacune de ses phrases doit se rattacher à l'idée maîtresse et contribuer à la renforcer. Dès lors, même si cet élément n'est pas le plus juste, nous nous appuierons sur le découpage en paragraphes puisqu'il satisfait aux deux contraintes pré-citées.

Enfin, il nous reste à déterminer la méthode permettant d'évaluer la pertinence de ces paragraphes par rapport à la requête. RAJMAN *et al.* (1998) rappelle que la dissimilarité *atn* peut être utilisée pour rechercher de l'information « à l'intérieur » des documents, puisqu'il est préférable d'utiliser des similarités sensibles au nombre de termes communs plutôt que sensibles à la proportion de termes communs en raison de leur faible quantité dans de tels cas.

En accord avec nos notations, nous définissons la dissimilarité *atn* entre un paragraphe d'un document $d_i \in D$, composé de *m* paragraphes, et une requête *r* comme suit :

on associe au j^{ème} paragraphe de d_i, noté d_i^j la représentation vectorielle $\overrightarrow{d_i^j} = \left(w_{i,1}^j, \cdots, w_{i,p}^j\right)$ et à la requête *r* la représentation vectorielle $\vec{r} = \left(r_1, \cdots, r_p\right)$

$$\text{avec } w_{i,k}^j = \begin{cases} \frac{1}{2}\left[1 + \frac{p_{i,k}^j}{\max_l\left(p_{i,l}^j\right)}\right]\log\left(\frac{m}{m_k}\right) & \text{si } p_{i,k}^j \neq 0 \\ 0 & \text{sinon} \end{cases} \quad \text{et} \quad r_k = \begin{cases} \frac{1}{2}\left[1 + \frac{p_k}{\max_l\left(r_l\right)}\right]\log\left(\frac{m}{m_k}\right) & \text{si } p_k \neq 0 \\ 0 & \text{sinon} \end{cases}$$

où $p_{i,k}^j$ est la fréquence relative du terme *k* dans le paragraphe *j* de d_i, p_k la fréquence relative du terme *k* dans la requête et m_k le nombre de paragraphes de d_i contenant le terme *k*.

Alors, $atn\left(d_i^j, R\right) = \overrightarrow{d_i^j} \bullet \vec{R}$, où $\bullet$ représente le produit scalaire. Le score de *atn* varie de 0 à $\log(n)$, nous le ramenons à une variation entre 0 et 1.

Afin de renseigner l'utilisateur sur la pertinence d'un paragraphe, nous modifions la couleur de la police en fonction du score obtenu. Ainsi, nous proposons ici de traduire ce score selon un dégradé allant du rouge pour la valeur 0 au gris pour la valeur 1. L'objectif étant de mettre en avant l'information la plus pertinente, un dégradé linéaire n'aurait pas été visuellement informatif puisque le système visuel humain est non linéaire (BARBA, 1981). Nous avons alors considéré que le score 0,5 servait de transition. Deux dégradés ont été définis par rapport à ce point : l'un concernant les fortes dissimilarités (de 0,5 à 1) et l'autre concernant les faibles dissimilarités (de 0 à 0,5). Trois fonctions *r*, *v* et *b* retournant respectivement les niveaux[3] de Rouge, Vert et Bleu sont définies comme suit :

[3] Nous avons utilisé une représentation de couleur sur 8 bits, soit 256 niveaux par canal.

$$r, v, b : [0,1] \mapsto [0,255]$$
$$r(s) = \begin{cases} 127(1-s) & \text{Si } s \le 0,5 \\ 382s - 127 & \text{Sinon} \end{cases} \qquad (4.5.1)$$
$$v(s) = b(s) = 127(1-s)$$

La Figure 4-9 illustre la plage de variation de la couleur dans l'espace RVB en fonction de la dissimilarité et la Figure 4-10 montre la mise en application sur un document.

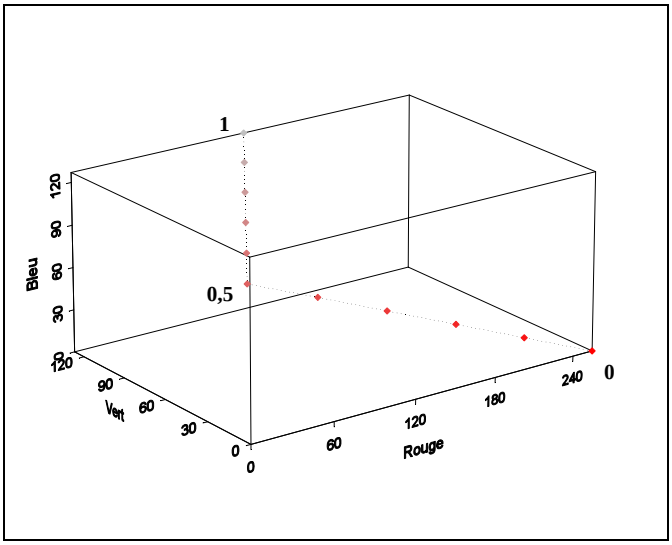
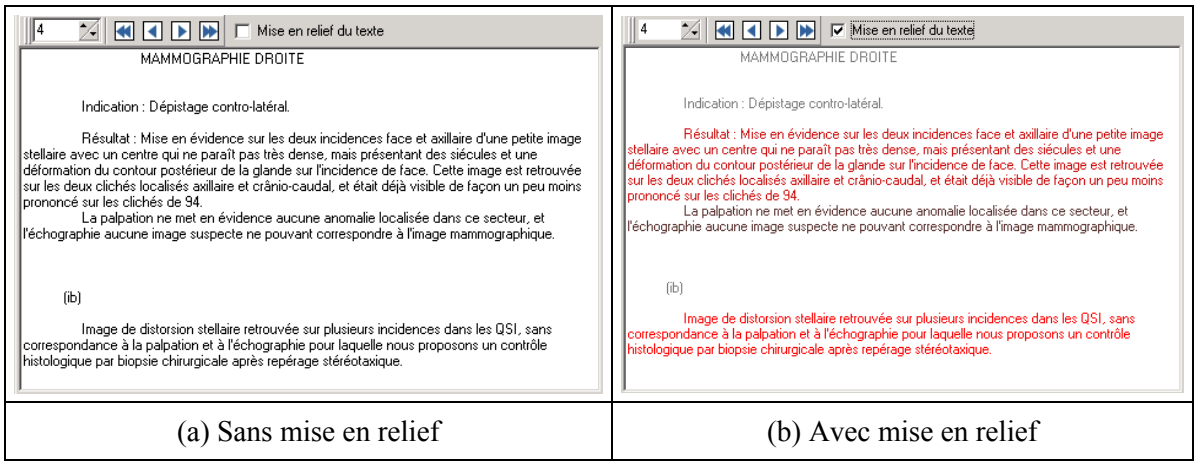


Figure 4-9 Variation de la couleur en fonction de la dissimilarité dans l'espace RVB



(a) Sans mise en relief	(b) Avec mise en relief

Figure 4-10 Exemple de mise en relief d'un document

4.5.3 Proposition d'un système de RI par des techniques d'ECT

4.5.3.1 Motivations

Le principe de la RI se base sur l'appariement entre une requête définie par l'utilisateur et les documents contenus dans la base. Originellement des mots-clefs, puis évoluant vers le langage naturel, ces requêtes restent délicates à utiliser moins par les écueils de la linguistique, dont des solutions viables existent, que par la subjectivité contenue dans ces requêtes.

Pour pallier ce problème, des systèmes interactifs, dénommés systèmes de retour de pertinence (ROCCHIO, 1971), ont été mis en place. Ils apportent un gain non-négligeable, notamment pour les méthodes d'expansion, mais nous y voyons principalement deux inconvénients :

1. La prise en compte des documents non-pertinents dans les méthodes d'expansion de requêtes, bien que nécessaire, entraîne une pondération négative de la terminologie non-pertinente en risquant de privilégier l'absence de cette terminologie. Or l'absence d'un terme n'a pas la même signification que sa présence car si un terme est absent, l'idée qui lui est associée peut elle être présente mais sous une autre forme (e.g. métaphore) alors que la présence d'un terme implique nécessairement la présence de l'idée qui lui est associée ;
2. Même après une reformulation de la requête à l'aide des systèmes interactifs, le module d'appariement reste invariant à l'utilisateur. Nous pensons que cela implique à l'utilisateur de s'adapter au système alors qu'il nous semble nécessaire que ce soit au système d'adapter le choix des documents en fonction de l'utilisateur.

Nous proposons alors un système de RI basé sur l'utilisation de méthodes inductives.

4.5.3.2 Présentation de la méthode

Notre proposition utilise un système de RI textuelle basé sur un modèle vectoriel et un système d'expansion et de re-pondération de la requête utilisant les interactions avec l'utilisateur. Le système renvoie l'ensemble des réponses $R(r)$ à la requête r de l'utilisateur. Celui-ci les évalue et détermine alors deux sous-ensembles de $R(r)$: $R^{per}(r)$ regroupant les documents jugés pertinents par l'utilisateur et $R^{non}(r)$ regroupant les documents jugés non-pertinents.

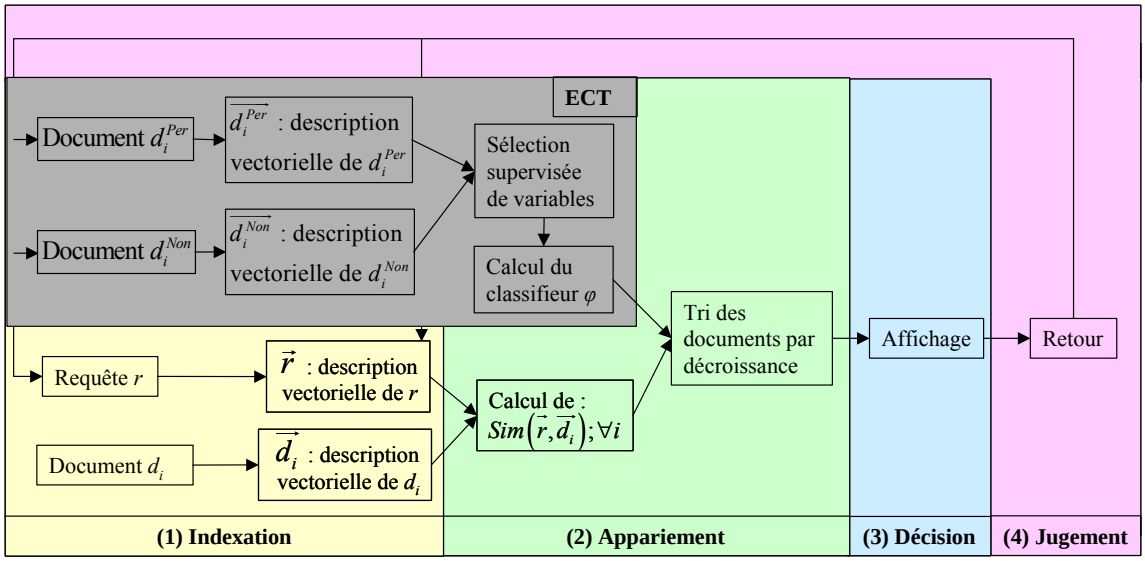


Figure 4-11 Architecture du système de RI basé sur l'analyse de contenus textuels

Lorsque les ensembles de documents pertinents et non-pertinents sont jugés suffisamment importants, le processus d'extraction de connaissances à partir de textes est appliqué :

1. Les termes sont évalués à l'aide d'une méthode de sélection de variables supervisée. L'objectif est de sélectionner les termes apportant le plus d'informations dans la séparation des ensembles $R^{per}(r)$ et $R^{non}(r)$. La section 1.6.2 du Chapitre 1 est consacrée à la description de telles méthodes. Par défaut, nous utilisons la méthode du $\chi^2_{\text{multivarié}}$;
2. A partir de ces termes sélectionnés, nous élaborons un classifieur φ afin d'induire un ensemble de règles permettant la différenciation de $R^{per}(r)$ et $R^{non}(r)$. Le chapitre 3 présente quelques unes de ces méthodes. Par défaut nous proposons le classifieur C4.5 ;
3. Les règles sont alors appliquées sur l'ensemble de la base documentaire *D*. C'est donc φ qui effectue l'appariement entre la requête et les documents de la base. A ce stade, la requête n'est plus *r* mais les ensembles d'exemples positifs $R^{per}(r)$ et d'exemples négatifs $R^{non}(r)$;
4. Les documents dont φ prédit leur appartenance à la classe des documents pertinents sont alors triés en fonction de leur confiance associée à la règle ;
5. Enfin, l'utilisateur peut à nouveau interagir avec le système en jugeant de nouveaux documents ou en modifiant ses précédents jugements.

La Figure 4-11 illustre les modifications apportées à une architecture classique de système de RI.

4.5.3.3 Proposition d'interface utilisateur

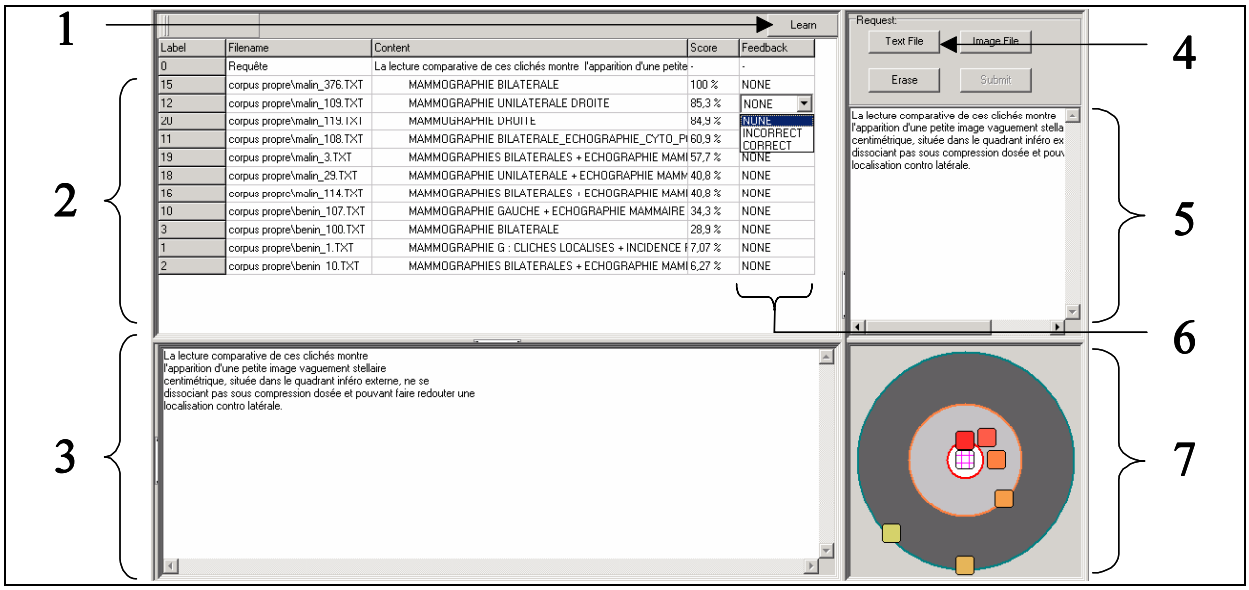


Figure 4-12 Interface du système de RI basé sur l'analyse de contenus textuels

Côté utilisateur, les fonctions élémentaires des systèmes classiques restent présentes (Figure 4-12) :

1. L'utilisateur saisit sa requête directement dans la zone appropriée (zone 5 sur la Figure), ou sélectionne un fichier (zone 4) qu'il pourra modifier au besoin dans la zone 5 et la soumet au système (zone 1) ;
2. Le système lui retourne une liste de propositions affichant un score de pertinence (zone 2) ;
3. La phase interactive permet à l'utilisateur de signifier la pertinence ou non des documents (zone 6) afin que le système prenne en compte ces choix lorsque l'utilisateur souhaite affiner sa recherche (zone 1), et retour à l'étape 2 ;
4. A la demande de l'utilisateur, le système liste tous les documents que l'utilisateur a définis comme étant pertinents.

A ces éléments, nous avons ajouté notre outil de navigation contextuelle (cf. Chapitre 3, section 3.2.2.3). Dans l'objectif d'aider la prise de décision de l'utilisateur quant à la pertinence du document, nous avons proposé la mise en relief de passages similaires à la requête lors de l'édition des documents à partir de la carte contextuelle.

Pour la carte contextuelle, nous n'utilisons pas les relations de voisinage définies par un graphe de voisinage car cela serait trop coûteux en temps de calcul. Nous nous basons sur les similarités entre les documents malgré la non symétrie des relations générées.

4.5.4 Discussion

La caractéristique majeure de notre proposition est de ne plus se centrer sur la requête en elle-même, mais sur un ensemble de documents pertinents et non pertinents. Il en découle l'utilisation des méthodes reconnues d'ECT. La difficulté majeure réside dans le besoin d'avoir un grand nombre d'éléments étiquetés pour augmenter la qualité du classifieur. Pour y remédier, nous avons mis en place un outil rapide de mise en relief du document lu par l'utilisateur et un système de navigation contextuelle.

Après la description de la proposition, la question de la qualité et des performances se pose alors. Cependant, la simple évaluation des méthodes d'ECT, déjà largement réalisée, ne peut suffire pour montrer l'efficacité d'une telle solution puisque la phase interactive visant à la construction du corpus d'apprentissage doit faire partie de l'évaluation. Or, cette phase a pour but de permettre la prise en compte de besoins non exprimés par l'utilisateur et ainsi de s'écarter, d'affiner ou de redéfinir la requête initiale. Les méthodes usuelles d'évaluation en RI ne sont pas utilisables puisqu'elles cherchent à mesurer la pertinence des résultats retournés par rapport à la requête initiale. De fait, la seule possibilité d'évaluer le système en terme de rappel et précision consisterait à soumettre à un groupe d'utilisateurs un ensemble pré-défini de requêtes. Ces derniers obtiendraient un certain nombre de documents en utilisant le système. Il faudrait alors leur demander de juger la pertinence de tous les documents de la base de tests par rapport au jeu de requêtes pour enfin évaluer le taux de rappel et précision pour chacun des utilisateurs. Mais le coût évident de sa réalisation ne nous a pas permis de la réaliser.

Le module ECT est activé seulement à partir d'un critère quantitatif du nombre de documents étiquetés. Bien que facile à mettre en œuvre, ce choix n'est pas toujours évident à déterminer. Il faut faire face au compromis entre d'une part posséder un nombre suffisant d'éléments pour pouvoir réaliser un apprentissage et d'autre part ne pas « obliger » l'utilisateur à juger un trop grand nombre de documents.

A nos yeux, la faiblesse de cette méthode réside dans cette phase transitoire où le nombre de documents n'est pas encore suffisamment élevé pour appliquer les méthodes d'apprentissage. Nous cherchons bien évidemment à traiter cet aspect et avons pour idée d'une part d'utiliser un seuil de taux de succès du modèle en tant que déclencheur d'application des règles du classifieur. D'autre part, nous pensons utiliser durant cette phase transitoire l'ordre proposé par la mesure de similarité entre la requête et les documents de la base.

Enfin, sur le plan de l'interface nous menons des travaux complémentaires afin d'améliorer l'outil de mise en relief des passages en utilisant les caractéristiques de la perception du système visuel humain.

4.6 Conclusion

Au cours de ce chapitre, nous avons décrit le fonctionnement général d'un système de recherche d'information fonctionnant principalement à l'aide d'une représentation des données à partir de descripteurs. Nous avons décrit les principales méthodes utilisées dans le cadre du Web, se basant essentiellement sur les méta-données comme les hyperliens. Nous avons ensuite proposé deux méthodes de recherche d'information. La première, décrite dans le cadre de la RI visuelle mais adaptable à d'autres médias, utilise les graphes de voisinage dont leurs propriétés de symétrie qui permettent une navigation plus intuitive au sein des réponses fournies. La seconde traitant des données textuelles, des données de haut-niveau à caractère sémantique important, est basée sur le processus d'ECT permettant par induction et interaction de mieux prendre en compte l'aspect subjectif contenu dans la requête initiale.

Outre les évaluations de ces systèmes à approfondir, nos travaux immédiats se dirigent vers l'utilisation de l'ensemble de ces méthodes afin d'être en mesure de pouvoir traiter les objets complexes composés de documents multiformes.