République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Ecole Nationale Polytechnique d'Alger

Département d'Electronique

Projet de fin d'études

En vue de l'obtention du diplôme Ingénieur d'Etat en Electronique

Thème :

Modélisation de la recherche intelligente d'information par Indexation Sémantique Latente et outils de traitement d'image.

Proposé et encadré par :

- Mr. A. ALLALI
- Pr. C. LARBES

Réalisé par :

- Mr. LARABA Sohaib
- Mr. BOUMAAZA Yacoub

Devant le jury :

- Pr. M. Guerti
- Mr. L. Abdellouel

Promotion : juin 2012

Ecole Nationale Polytechnique d'Alger
10, Avenue Hassan Badi, El-Harrach, Alger

Dédicace

A mes parents, que dieu les protège

A Mes frères Abderrahmane et Oussama

A ma chère sœur Maroua

A tous mes amis

A mes collègues Microsoft Student Partners

Je dédie ce modeste travail

... Sohaib

A mes très cher parents que dieu les protège

Aucun hommage ne pourrait être à la hauteur de l'amour

Dont ils ne cessent de me combler.

A Mes frères

A mes proches

A tous mes amis

Je dédie ce modeste travail

... Yacoub

Remerciement

Nous remercions Allah, le tout clément, le tout puissant, de nous avoir donnée la force de réaliser ce travail.

Nous tenons à exprimer notre gratitude et nos vifs remerciements à Monsieur Ali ALLALI d'avoir joué pleinement son rôle de promoteur en étant à nos côtés tout au long de l'étude de notre projet. Ses conseils et orientations nous ont guidés jusqu'à l'aboutissement de ce travail.

Nous remercions notre co-promoteur Monsieur Cherif LARBES pour ses remarques pertinentes qui ont apporté une amélioration certaine à notre travail. Nous le primes de croire à notre respectueuse estime et notre sincère reconnaissance.

Nous voudrions exprimer aussi notre profond remerciement à tous nos enseignants de l'Ecole Nationale Polytechnique d'Alger, qui nous ont prodigué le savoir et nous ont permis d'arriver à ce stade, nous pensons particulièrement à Monsieur Hichem BOUSBIA-Salah, à AMMI Salah et tous nos amis qui nous ont apporté leurs soutiens pour la réalisation de ce mémoire dans d'excellentes conditions.

Enfin, nous remercions toutes nos familles et particulièrement nos parents de nous avoir soutenus et encouragés.

ملخص

الهَدَفُ الرَّئِيسِيُّ لِنِظَامِ اسْتِرْجَاعِ المَعْلُومَات هو العثور على محتوى الوثائق التي تتوافق مع استعلام معين. في هذا السياق, تمثل الوثائق بمجموعة من الكلمات الرئيسية التي تصف محتوياتها. وتقاس جودة نتائج البحوث من خلال مقارنة استجابات النظام مع الاستجابات المثالية التي يأمل المستخدم في الحصول عليها. كلما كانت استجابات النظام أكثر تطابق مع تلك التي يتوقعها المستخدم, كلما كان النظام أكثر كفاءة.

الأعمال المطروحة في هذه المذكرة, تدرس تقنية جديدة لاسترجاع المعلومات تسمى **الفهرسة الدلالية الكامنة** (Latent Semantic Indexing) و تهتم بتحسين البحث عن طريق إقتراح تقريبات جديدة تعتمد أساسا على تقنيات معالجة الصور بما في ذلك تقنية Haar Wavelet Transform و كذا تقنية Discret Cosine Transform المرتبطة بالنماذج الرياضية في بيئة تطوير متكاملة مع بنية قوية النمذجة، من خلال تصحيح الأخطاء على نحو يتسم بالكفاءة والتشخيص لإدارة النظام الأساسي للتطوير، وإدارة دورة حياة التطبيقات وأدواته و الاختبارات والرسومات.

كلمات مفتاحية:

استرجاع المعلومات, الفهرسة الدلالية الكامنة, تحويل مويجات هار, التحليل المتعدد الدقات, تحويل الجب تمام المتقطع.

Résumé

L'objectif principal d'un **S**ystème de **R**echerche d'**I**nformation (**SRI**) classique est de retrouver les documents dont le contenu est conforme à une requête donnée. Dans cette optique, les documents sont représentés par un ensemble de mots-clés décrivant leurs contenus. La qualité des résultats de la recherche se mesure 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, plus le système est jugé performant. Les travaux présentés dans ce mémoire traitent une nouvelle technique de recherche d'information qui s'appelle l'**I**ndexation **S**émantique **L**atente (en Anglais, **L**atent **S**emantic **I**ndexing ou **LSI**) et s'intéressent à l'amélioration de la recherche en proposants des nouvelles approches qui basent sur les techniques de traitement d'image, notamment la **T**ransformée en **O**ndelette de **H**aar (**HWT**) et la **T**ransformée en **C**osinus **D**iscrète (**DCT**). Associées à des modèles mathématiques dans un environnement de développement intégré puissant par son architecture et modélisation, efficace par son débogage et diagnostiques et convivial par la prise en charge de la plateforme de développement, la gestion du cycle de vie des applications et de ses outils de tests et de graphisme.

Mots clés : Recherche d'information, Indexation Sémantique Latente, Transformée en Ondelette de Haar, Analyse multirésolution, Transformée en Cosinus Discrète.

Abstract

The main objective of **I**nformation **R**etrieval **S**ystem (**IRS**) is typical to find documents whose content conforms to a given query. In this context, documents are represented by a set of keywords describing their contents. The quality of research results is measured by comparing the responses of the system with ideal responses that the user hopes to receive. More system responses match those that the user expects, the system is considered more efficient.

The works presented in this thesis deal with a new technique for information retrieval called Latent Semantic Indexing and are interested in improving research by suggesting new approaches based on image processing techniques, including the **H**aar **W**avelet **T**ransform (**HWT**) and **D**iscrete **C**osine **T**ransform (**DCT**). Associated with mathematical models in an integrated development environment with powerful architecture and modeling, through its efficient and friendly debugging and diagnostics for the management of the development platform, managing the lifecycle of applications and its tools tests and graphics.

Key Words: Information Retrieval, Latent Semantic Indexing, Haar Wavelet Transform, Multiresolution Analyses, Discrete Cosine Transform.

Table de Matière

Liste des figures

Liste des tableaux

Liste des abréviations

RI : Recherche d'information

VSM : Vector Space Model

LSI : Latent Semantic Indexing

SVD : Singular Value Decomposition

TDM : Term Document Matrix

SNR : Signal Noise Ratio

HWT : Haar Wavelets Transform

DCT : Discrete Cosine Transform

Tf-IDF : Term Frequency – Inverse Document Frequency

Introduction Générale

L'information joue un rôle vital dans la société d'information d'aujourd'hui, et la croissance exponentielle de sa volumétrie et de son sombre potentiel d'utilisateurs entrainent de nouveaux défis scientifiques dans tous les domaines dont la tâche principale est la gestion de l'information. La Recherche d'Information (RI) est, sans contexte, l'un des domaines les plus concernées.

En effet, l'objectif principal du domaine de la RI est de fournir des modèles, techniques et systèmes pour stocker et organiser des masses d'informations à partir desquelles sont sélectionnées celles qui répondent aux critères relatifs aux besoins utilisateurs.

D'énormes efforts ont été déployés pour développer des approches et des techniques permettant de retrouver l'information voulue effectivement et efficacement à partir de vastes collections de données émanant de sources de données hétérogènes ou homogènes.

Cependant, en raison de la surabondance de l'information d'une part et de sa large accessibilité à travers notamment le Web, d'autre part, leur mise en œuvre est confrontée à de nouveaux problèmes. En effet la situation est actuellement paradoxe : la masse d'informations est telle que l'accès à une information pertinente, adaptée aux besoins d'un utilisateur donné devient à la fois difficile et nécessaire. En claire, le problème n'est pas dans la disponibilité de l'information mais dans sa pertinence relativement au besoin de l'utilisateur.

Contexte de travail

Les travaux dans ce mémoire se situent dans le contexte de la recherche d'information textuelle et s'intéressent à l'amélioration de la technique d'Indexation Sémantique latente (LSI) en particulier à la phase de prétraitement en proposant des approches basées sur les techniques de traitement d'image et à celles liées à la génération de la matrice des termes et documents . Ces approches offrent le moyen d'évaluer la pertinence d'un document pour une requête. Nous rappelons que la majorité des Systèmes de Recherche d'Information (SRI) voie un document et une requête

comme une liste de mots clés pondérés. Afin de sélectionner les documents susceptibles de répondre à une requête, un SRI évalue la pertinence d'un document vis-à-vis de la requête en calculant un score de ressemblance (similarité) en fonction de ces poids.

Problématique

Les systèmes de Recherche d'Information Classiques représentent les documents et les requêtes par les mots qu'ils contiennent, et basent souvent cette comparaison sur le nombre de mots qu'ils ont en commun, c'est l'appariement lexical. Dans cette approche, des documents pertinents, se partagent pas de mots avec la requête ne sont pas retrouvés (Problème de synonymie : mots de nom différent et de même sens). Tandis que des documents non pertinents contenant des mots de la requête ne sont pas retrouvés à l'utilisateur (Problème de polysémie : mots de même nom et sens différent). Ces problèmes sont dus au fait que l'appariement lexical ne tient pas compte des sens des mots du document et de la requête.

L'indexation sémantique latente (LSI) est une technique de recherche intelligente d'information qui tente de pallier ses problèmes en offrant le moyen de distinguer ces sens, et de les utiliser lors du processus d'appariement.

Chapitre 1. Concepts de base de la recherche d'information

1.1 Introduction

Les **S**ystèmes de **R**echerche d'**I**nformation documentaire (**SRI**) sont nés de la nécessité d'automatiser la gestion et la recherche des information documentaire. Un SRI peut être défini comme étant un mécanisme de gestion qui joue l'intermédiaire entre un utilisateur et une collection d'informations. Son but est de satisfaire le besoin en information de cet utilisateur. Il est utilisé pour gérer une collection d'informations textuelles ou multimédia sous forme de documents et pour mettre à la disposition des utilisateurs un ensemble de techniques qui leur permettent de rechercher des informations et de sélectionner un sous-ensemble de documents qui répond à leurs requêtes. Le terme « Recherche d'Information » (Information Retrieval) fut donné par Calvin. Mooers en 1948 pour la première fois quand il travaillait sur son mémoire de maîtrise. La première conférence dédié à ce thème -International Conference on Scientific Information – s'est tenue en 1958 à Washington. On y comptait les pionniers du domaine, notamment, Cyril Cleverdon, Brian Campbell Vickery, Peter Luhn, etc.

Historiquement, la croissance du volume de données textuelles comme les livres et les articles dans les bibliothèques durant des siècles a imposé de définir des mécanismes efficaces pour les localiser. Les premières techniques, comme l'abstraction (abstracting), l'indexation et l'utilisation des catégories de classification ont marqué la naissance de la « Recherche d'Information » comme discipline de recherche. D'énormes efforts ont été déployés depuis, comme le montre la littérature, pour développer des approches et des techniques permettant de retrouver l'information voulue effectivement et efficacement à partir de vastes collections de données textuelles. Depuis les années 1990, notamment avec l'avènement d'Internet, la recherche d'information est devenue plus d'actualité et plus exigeante que jamais. Même si l'effort continu des chercheurs a doté le domaine d'un ensemble riche d'outils sophistiqués (protocoles de transmission efficaces, supports rapides, etc.), la sophistication des outils pour la création et la transmission de l'information est bien moindres que celle des outils qui gèrent l'information.

Nous présentons dans ce chapitre les principes fondamentaux d'un système SRI. Nous décrivons les concepts de base de ce domaine, puis nous présentons les différents types de tâches du RI et le processus global de la RI. Nous donnons l'utilité et l'importance des opérations qui composent ce processus. Nous décrivons les différents modèles de la RI et nous rappelons essentiellement la modélisation de la pertinence dans ces modèles ainsi que les grands projets dans l'histoire de la RI. Dans la dernière section, nous décrivons les techniques utilisées pour évaluer les performances des SRI.

1.2 Notions de base

La Recherche d'Information **[53]** **[12]** **[10]** **[30]** est traditionnellement définie comme l'ensemble des méthodes, procédures et techniques permettant de sélectionner à partir d'une collection de documents, ceux qui qui sont susceptibles de répondre aux besoins de l'utilisateur. Gérer des textes implique stocker, rechercher et explorer des documents ou parties de documents pertinents. La phrase « Recherche d'Information » a deux connotations en anglais, « Information Research » et « Information Retrieval ».

En anglais, la phrase « Information Research or Information Search» est un peu différente de celui de « Information Retrieval ». « Information Research » est plutôt la « Recherche d'Information » alors que « Information Retrieval » est plutôt la récupération d'information dans un stockage informatisé et une mathématique de récupération.

Plusieurs concepts clés s'articulent autour de la notion de la définition de la Recherche d'Information :

Documents : Le document constitue l'information élémentaire d'une collection de documents. L'information élémentaire, appelée aussi granule de document, peut représenter tout ou une partie d'un document.

Collection de documents : La collection de documents (ou fond documentaire, corpus) constitue l'ensemble des informations, de documents, exploitables et accessibles. C'est l'ensemble de documents que l'on recherche. La collection peut contenir des références à des documents primaires ou encore les documents mêmes.

Besoin d'information : la notion de besoin en information en recherche d'infirmation est souvent assimilée au besoin de l'utilisateur

Il convient de préciser qu'il y'a souvent une confusion largement acceptée en RI, entre besoin et requête. Le besoin en information est une expression mentale de ce que recherche l'utilisateur, or la requête est souvent une liste de mots clés en particulier en RI textuelle qui traduit le besoin.

Pertinence : La pertinence est une notion fondamentale et cruciale dans le domaine de la RI. Les travaux de recherche récents [**50**] s'accordent sur la difficulté de la définition de la pertinence et mettent en exigence deux types de pertinence. La pertinence système [**7**] et la pertinence utilisateur [**61**].

La pertinence système est déterministe, objective et de finir à travers les modèles de la RI. Elle est souvent traduite par un score cherchant à évaluer la pertinence des documents vis-à-vis d'une requête. Cette pertinence est mesurée par une similarité de représentation document-requête (modèle vectoriel), une probabilité de pertinence des documents étant donnée une requête (modèle probabiliste).

La pertinence utilisateur est liée à la perception de l'utilisateur sur l'information renvoyée par le système. Elle est subjective, deux utilisateurs peuvent juger différemment un même document renvoyé par une même requête, et évalué dans le temps d'une recherche [**61**]. Une information non pertinente pour une requête peut être jugée pertinente plus tard vers la connaissance de l'utilisateur sur le sujet à évaluer.

1.3 Différents types de tâches du RI

Le Tableau 1.1 liste un certain nombre de différents types d'activité électronique qui peut être considérée dans le domaine de la RI [**53**].

Parmi ces applications, la consultation de base de données peut être considérée comme un peu controversée, parce que la base de données et les champs de la recherche sont traditionnellement distincts. La recherche de base de données traite généralement des données fortement structurées et des questions de mise à jour, de la notation de transaction, d'autorisation d'accès et de rétablissement simultanés après échec.

Catégorie	***Description***	***Exemple de requête***
Ad hoc recherche	Retrouver les documents pertinents dans une collection fixe	Find documents which tell me about investment strategies.
Question/ Réponse	Extraire des réponses dans les documents récupérés	Who is the prime minister of Australia?
Annuaire	Navigation dans une Web page spécifique	Where is the ELSNET home page?
Diffusion sélective d'information	Contrôler un flot de documents correspondant à un profil	Send me any new information on high tech companies
Classification de documents	Regroupement automatique de documents	Find the natural groupings in this set of scientific publications
Catégorisation de documents	Affecter un document à une catégorie prédéfinie	Classify incoming books according to their Dewey decimal category
Synthèse de documents	Extraire l'information à partir des documents retrouvés	Construct a personalized travel guide for my visit to Athens in July 2000.
Recherche dans la base de données	Extraire des enregistrements à partir d'une base de données structurée	Find books where author=Salton and year=2001. (sql : langage d'extraction de données)

Tableau 1.1: Exemple de différents types d'application de RI

Les types de questions/requêtes qui peuvent être soumises à une base de données sont déterminés par le schéma de la base de données et les réponses sont précises. En revanche, la recherche d'information traite généralement des documents texte ou multimédia non structurés et souvent la nécessité de mises à jour ne sont pas considérés. Cependant, cette simplification est compensée par l'incertitude quant à ce qui constitue l'ensemble de bonnes réponses. Les systèmes de RI modernes rangent des documents par ordre décroissant de leur score de pertinence.

1.4 Clustering et classification des documents

1.4.1 But de Clustering

Le clustering (regroupement) des documents vise à mettre les documents similaires ensemble pour atteindre au moins un des buts suivants:

1) Accélération du processus de recherche ;
2) Présentation des résultats similaires ;
3) Regroupement des réponses du système.

Avec le progrès rapide enregistrés dans l'informatique, le premier objectif semble beaucoup moins important. Les deux autres restent toujours d'actualité.

1.4.2 Méthodes de clustering

Les méthodes de clustering peuvent être de deux sortes:

- Hiérarchique
- Non-hiérarchique

a) Le premier type d'algorithme hiérarchique essaie de créer une hiérarchie des clusters, les documents les plus similaires sont regroupés dans des clusters aux plus bas niveaux, tandis que les documents moins similaires sont regroupés dans des clusters aux plus hauts niveaux.

   Selon la manière de création, ce type d'algorithmes peut encore se diviser en deux: divisif ou agglomératif. En partition, on tente de diviser un grand cluster en deux plus petits (approche descendante). En regroupement, on tente de regrouper deux clusters en un plus grand (approche ascendante).

b) Pour le deuxième type d'algorithmes les clusters sont au même niveau. Parmi les algorithmes souvent utilisés, il y'a **k-means**.

1.4.3 Algorithme de k-means

Le principe est de comparer plusieurs schémas de clustering (plusieurs partitionnements) afin de retenir le schéma qui optimise un critère de qualité. Cet optimum est obtenu de façon itérative, en améliorant un schéma initial choisi. L'algorithme de k-means (ou l'algorithme des k-moyennes).

L'algorithme des k-moyennes (k-means) est la méthode de partitionnement la plus connue et la plus utilisée dans divers domaines d'application.

a) Etapes K-moyennes algorithme (k-means) :

1. Prendre l'espace de données à classifier ;

2. Prédéterminer le nombre de classes k ;

3. Initialiser k moyens pour les données $M_1, M_2, M_3, \ldots\ldots, M_K$;

4. Déterminer la distance euclidienne entre chaque point de données et la moyenne, elle est donnée par la distance d entre le point x et le point y à n composantes dans l'espace euclidien et est donnée par $\boldsymbol{d = \sqrt{\sum_{i=1}^{i=n} |x_i - y_i|^2}}$.

5. Grouper les points de données ayant une distance minimale à la moyenne correspondante.

6. Calculer la nouvelle moyenne de chaque groupe formée dans l'étape 5.

7. Répétez les étapes 4 à 6 jusqu'à ce que le nouveau moyen formé est la même que la moyenne précédente.

1.4.4 Mesures

Il est important de déterminer le cluster à découper ou les clusters à regrouper dans une approche hiérarchique, et de déterminer une fonction de similarité dans une approche non-hiérarchique.

Les mesures utilisées varient et les plus utilisées sont les suivantes :

Similarité de clusters : elle est définie comme la similarité entre :

- les centroïdes de ces clusters (le centroïde est le vecteur moyen de tous les éléments dans le cluster)
- ou bien les medoïdes de ces clusters (le medoïde est l'élément le plus au centre du cluster)

Exemple:

- $\boldsymbol{Sim\ (C1, C2) = \sum_{D1 \in C1, D2 \in C2} \frac{COS(D1,D2)}{taille(C1) * taille(C2)}}$
- $\boldsymbol{Sim(C1, C2)}$ = Similarité entre deux éléments les plus proches des deux clusters.

$\begin{cases} \textbf{C1 et C2} \text{ sont des clusters (ensemble de documents)} \\ \textbf{D1, D2} \text{ documents appartenant respectivement à C1 et C2} \\ \text{Taille } (C_i)\text{: nombre de documents du cluster } C_i \end{cases}$

Remarque : Certains des algorithmes utilisent la similarité d'un élément avec tous les éléments du cluster comme critère, ou bien une mesure de cohésion de cluster.

1.5 Processus de Recherche d'Information

Pour répondre aux besoins en information de l'utilisateur, un SRI met en œuvre un certain nombre de processus pour réaliser la mise en correspondance des informations contenues dans un fond documentaire d'une part, et des besoins en information des utilisateurs d'autre part. Ces processus supposent que la collection de documents est unique et s'appuient sur un certain nombre de modèles permettant de sélectionner des informations pertinentes en réponse à une requête utilisateur. Il s'agit principalement du processus de représentation et du processus de recherche :

Processus de représentation : Un processus de représentation a pour rôle d'extraire d'un document ou d'une requête, une représentation paramétrée qui couvre au mieux don contenu sémantique. Ce processus de conversion est appelé *indexation.* Le résultat de l'indexation constitue le descripteur du document ou de la requête, qui est une liste de termes significatifs pour l'unité textuelle correspondante, auxquels sont associés généralement des poids pour différencier leur derge de représentativité. L'ensemble des termes reconnus par le SRI est rangé dans une structure appelée dictionnaire constituant le langage d'indexation.

Processus de recherche : Il représente le processus du noyau d'un SRI. Il comprend la fonction de décision fondamentale qui permet d'associer à une requête, l'ensemble des documents pertinents à restituer. Il est utilisé pour la recherche d'information proprement dite et est étroitement lié au modèle de représentation des documents et des requêtes. Ces modèles de recherche représentent ce qui diffère le plus entre les SRI. Ils sont inspirés de concepts mathématiques afin de pouvoir évaluer certaines relations, notamment la relation **d'*appariement*** entre la requête et les documents. La problématique majeure des SRI est de retrouver les quelques dizaines ou

milliers de documents pertinents parmi des millions de documents. Cet écart de cardinalité rend cette tâche encore plus difficile.

En plus des étapes de représentation et de recherche, quelques systèmes peuvent supporter une étape supplémentaire de ***reformulation automatique de requêtes***. Cette étape à pour objectif d'améliorer les performances du SRI, donc la précision dans les réponses du système.

1.5.1 Modèles de 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 cette mesure de pertinence. Nous présentons dans la suite, de manière succincte, les principaux modèles issus de la recherche d'information.

1.5.1.1 Modèle Booléen

Le modèle booléen est basé sur la théorie des ensembles. Dans ce modèle, les documents et les requêtes sont représentés par des ensembles de mots clés. Ce modèle doit son nom à l'utilisation des opérateurs *et*, *ou* et *non* pour la représentation des documents et des requêtes. Chaque document d_j est représenté par un ensemble de termes, et c'est la conjonction de ces termes qui constitue l'index des documents. Une représentation logique des prédicats est :

$$\mathbf{d = t_1 \wedge t_2 \wedge \ldots \wedge t_n}$$

Une requête est une expression logique quelconque de termes. On peut utiliser les opérateurs et ($\wedge$), ou($\vee$) et non($\neg$). Par exemple :

$$\boldsymbol{q = (t_1 \wedge t_2) \vee (t_3 \wedge \neg t_4)}$$

Un document est représenté comme un ensemble de termes, et une requête comme une expression logique de termes.

Insuffisances du modèle

1. La correspondance entre un document et une requête est soit 1, soit 0. En conséquence, le système détermine un ensemble de documents non-ordonnés comme réponse à une requête. Il n'est pas possible de dire quel document est mieux qu'un autre. Cela crée des problèmes aux usagers, car ils doivent encore fouiller dans cet ensemble de documents non-ordonnés pour trouver des documents qui les intéressent. Ceci est particulièrement difficile dans le cas où beaucoup de documents répondent aux critères de la requête.
2. Tous les termes dans un document ou dans une requête étant pondérés de la même façon simple (0 ou 1), il est difficile d'exprimer qu'un terme est plus important qu'un autre dans leur représentation.
3. Le langage d'interrogation est une expression quelconque de la logique de propositions (un terme étant une proposition). Cela offre une très grande flexibilité aux usagers pour exprimer leurs besoins. Cependant, un problème en pratique est que les usagers manipulent très mal les opérateurs logiques.
   Il faut cependant remarquer que ce modèle booléen standard n'est utilisé que dans très peu de systèmes de nos jours.

1.5.1.2 Modèle Matching Score

L'idée de ce modèle est assez primitive et intuitive ; un document est représenté par un ensemble de termes pondérés par leur fréquence. Une requête est aussi un ensemble de termes, pondérés à 1. Le degré de correspondance est la somme des fréquences des termes de la requête dans le document :

$$\boldsymbol{R(d,q) = \sum_{i=1}^{t} f_i}$$

$$\begin{cases} \boldsymbol{q} : \text{requéte}, \\ \boldsymbol{d} : \text{ document} \\ \boldsymbol{f_i} : \text{fréquence du terme du } \boldsymbol{i} \textit{ de } \boldsymbol{q} \textit{ dans } \boldsymbol{d} \\ \boldsymbol{t} : \text{ nombre de termes différents dans } \boldsymbol{q} \end{cases}$$

La valeur R ainsi calculée est appelée le « **matching score** ». En réalité, cela est équivalent à parcourir le document, et de voir combien de fois les termes de la requête

apparaissent dans ce document. Plus ce « matching score » est élevé, plus on considère que le document correspond à la requête, et donc plus il sera classé haut dans la liste des résultats.

Insuffisance du modèle

Ce modèle est primitif car il utilise directement le résultat de l'indexation sans aucune réorganisation ou modélisation.

1.5.1.3 Modèle Vectoriel

Le modèle vectoriel [**28**] repose sur des bases mathématiques des espaces vectoriels. Dans ce modèle, un document, ainsi qu'une requête est représenté comme un vecteur de poids dans un espace Euclidien de dimension élevée. Chaque poids dans le vecteur désigne l'importance d'un terme correspondant dans ce document ou dans la requête. Pour qu'un vecteur prenne une signification, il faut d'abord définir un espace vectoriel. L'espace vectoriel est défini par l'ensemble de termes que le système a rencontré durant l'indexation. Soit l'espace vectoriel suivant :

$$< t_1, t_2, t_3, \dots, t_n >$$

Un document et une requête peuvent être représentés comme suit :

$$d =< a_1, a_2, a_3, \dots, a_n > \qquad q =< b_1, b_2, b_3, \dots., b_n >$$

Où a_i *et* b_i correspondent aux poids du terme t_i dans un document d et dans une requête q. Etant donné ces deux vecteurs, leur degré de correspondance est déterminé par leur similarité. Il y'a plusieurs façons de calculer la similarité entre deux vecteurs.

Exemple de similarité

- $Sim0 = \sum_i (a_i * b_i)$: (Produit interne) (1.1)
- $Sim1 = \frac{\sum_i (a_i * b_i)}{[\sum_i (a_i)^2 * \sum_i (b_i)^2]^{1/2}}$: (Cosinus) (1.2)
- $Sim2 = \frac{\sum_i (a_i * b_i)}{[\sum_i (a_i)^2 * \sum_i (b_i)^2]}$ (1.3)
- $Sim3 = 2\frac{\sum_i (a_i * b_i)}{[\sum_i (a_i)^2 * \sum_i (b_i)^2 - \sum_i (a_i * b_i)]}$ (1.4)

Toutes ces formules, à part la première sont normalisées, c'est-à-dire qu'elles donnent une valeur dans [0,1].

On remarque également que certaines formules peuvent être transformées en un produit interne si on manipule les poids d'une certaine façon. Le cas typique est pour la formule cosinus (1.2). Cette formule peut être transformée comme suit :

$$\boldsymbol{Sim1} = \sum_i \left[\left(\frac{a_i}{\left[\sum_j (a_j)^2\right]^{\frac{1}{2}}} \right) * \left(\frac{b_i}{\left[\sum_j (b_j)^2\right]^{\frac{1}{2}}} \right) \right] \quad (1.5)$$

Dans cette formule, on a deux éléments clairement séparés $\left(\frac{a_i}{\left[\sum_j (a_j)^2\right]^{\frac{1}{2}}} \right)$ et $\left(\frac{b_i}{\left[\sum_j (b_j)^2\right]^{\frac{1}{2}}} \right)$.

En réalité, l'ajout des dénominateurs consiste à normaliser le poids initial $\boldsymbol{a_i}$ *et* $\boldsymbol{b_i}$.

En plus, cette normalisation peut être faite indépendamment : dans le cas de document, la normalisation n'utilise que les termes du document (même chose pour la requête). Ainsi, il est possible de calculer deux autres poids :

$$\hat{a} = \left(\frac{a_i}{\left[\sum_j (a_j)^2\right]} \right)^{\frac{1}{2}} \quad et \quad \hat{b} = \left(\frac{b_i}{\left[\sum_j (b_j)^2\right]} \right)^{\frac{1}{2}} \quad (1.6)$$

au préalable et de remplacer les poids dans les vecteurs. Si on fait cela, le calcul de similarité selon la formule de cosinus sera juste un produit interne :

$Sim1 = \sum_i (a_i * b_i)$. Ainsi, le temps d'évaluation peut être raccourci. Cette approche est prise dans le système SMART [**30**].

Avantage :

L'avantage du modèle vectoriel par rapport au modèle booléen réside particulièrement dans sa simplicité de calcul de la ressemblance entre documents.

Inconvénient :

L'inconvénient majeur de l'approche vectorielle réside dans le fait que le calcul de la similarité, du centroïde des documents pertinents et des documents non pertinents fait abstraction des valeurs des composantes des vecteurs requêtes et documents. Par ailleurs, il est impossible de représenter des phrases ou des multi-termes (on considère effectivement que les termes sont indépendants) [**53**].

1.5.1.4 Modèle Probabiliste

Le modèle de recherche probabiliste utilise un modèle mathématique fondé sur la théorie des probabilités [**28**]. 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 :

$\boldsymbol{P(w_{ij}/Pert)}$: Probabilité que le terme t_i apparaisse dans le document D_j sachant que ce dernier est pertinent pour la requête.

$\boldsymbol{P(w_{ij}/Nonpert)}$: Probabilité que le terme t_i apparaisse dans le document D_j sachant que ce dernier n'est pas pertinent pour la requête.

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

Soit $D_j(t_1, t_2, \dots, t_N)$ Où $t_i = \begin{cases} 1 & si\ ce\ terme\ indexe\ D_j \\ 0 & sinon \end{cases}$:

Et $\mathbf{P(pert/D_j)} = \frac{\mathbf{P(D_j/pert)*p(pert)}}{\mathbf{p(D_j)}}$ et $\boldsymbol{P(Nopert/D_j)} = \frac{\boldsymbol{P(D_j/Nonpert)*p(Nonpert)}}{\boldsymbol{p(D_j)}}$ (1.7)

Où :

- $p(pert)$: probabilité de pertinence du document D_j sachant sa description.
- $p(D_j) = P\left(\frac{D_j}{pert}\right) * p(pert) + P\left(\frac{D_j}{Nonpert}\right) * p(Nonpert)$
- $P(D_j/pert)$ (Respectivement $P(D_j/Nonpert)$) Est la probabilité d'observer le document Dj sachant qu'il est pertinent (respectivement non pertinent).
- $P(pert/D_j) = p(t_1/pert) * p(t_1) \dots p(t_N/pert) * p(t_N)$
- $P\left(\frac{Nonpert}{D_j}\right) = p\left(\frac{t_1}{Nonpert}\right) * p(t_1) \dots p\left(\frac{t_N}{Nonpert}\right) * p(t_N)$

Où : $\begin{cases} p\left(\frac{t_1}{pert}\right) = \frac{r_1}{R} \text{ et } p\left(\frac{t_1}{Nonpert}\right) = \frac{m_i - r_i}{M-R} \\ R : nombre\ de\ documents\ pertinents\ pour\ une\ requéte \\ M : nombre\ de\ documents\ dans\ la\ collection \\ r_i : nombre\ de\ documents\ dans\ lesquels\ le\ terme\ t_i\ apparait \\ m_i : nombre\ totlal\ des\ documents\ dans\ lesquels\ le\ terme\ t_i apparait \end{cases}$

Pour caractériser l'occurrence des termes d'indexation dans les documents, on utilise une loi de distribution du type loi de Poisson. Cette occurrence est alors déduite de l'étude d'un échantillon de documents. Pour la restitution, les documents dont rangés en fonction de $P(pert/t_i)$. Il résulte du principe d'ordonnancement probabiliste que cet ordonnancement est optimal. C'est-à-dire que, quel que soit le nombre de documents pertinents restitués, le pourcentage de documents restitués qui sont effectivement pertinents est maximisé.

Avantage :

Les modèles probabilistes sont plus efficaces que les modèles booléens (appariement exact). Ils ont une base théorique saine et sont indépendants du domaine d'application.

Inconvénient :

Un obstacle majeur avec les modèles de recherche d'information probabilistes est de trouver des méthodes pour estimer les probabilités utilisées pour évaluer la pertinence qui soient théoriquement fondées et efficaces au calcul. Pour des raisons de simplicité, l'hypothèse de l'indépendance des termes est utilisée en pratique pour implémenter ces modèles.

1.5.1.5 Algorithme LSI (Latent Semantic Indexing)

a) **Introduction** : Le système LSI a été appliqué avec succès dans la recherche d'information et permet de résoudre les problèmes fondamentaux de **synonymie** et **polysémie**. L'algorithme LSI a été utilisé dans plusieurs moteurs de recherche notamment Google grâce à ces performance d'optimisation.

- **Synonymie** : Mots ayant le même sens mais de noms différents comme ***voiture*** et ***automobile***. Ainsi, les termes littéraux dans une requête utilisateur peuvent ne pas être les mêmes que ceux des documents pertinents.

Problème : Synonymie résulte un faible ***rappel***, où le ***rappel*** est défini par le rapport de nombre des documents pertinents et retrouvés sur le nombre de documents pertinents dans la base de données.

- **Polysémie** : Mots ayant le même nom mais de sens différent comme **jaguar (animal)** et **jaguar (automobile).** Ainsi, les termes dans une requête utilisateur vont correspondre littéralement à des documents pertinents.

Problème : Polysémie résulte une faible **précision**, qui est définie comme le rapport de nombre des documents pertinents et retrouvés sur le nombre total des documents retrouvés dans une requête

b) Apport du LSI

LSI essaye de résoudre ces problèmes par examiner le sens sémantique latent des termes dans un document. L'hypothèse principale de la LSI est que les mots utilisés dans un document ont une corrélation sémantique qui détermine la structure sémantique du document. En comparant les mots utilisés à travers les documents, il a été découvert que certains groupes des mots sont fréquemment partagés dans plusieurs documents et absents dans d'autres. Ces mots et les documents qui partagent sont sémantiquement proches. Pratiquement, LSI renvoie les documents qui sont similaire, même si les mots clés n'apparaissent pas dans la description du document.

Traditionnellement, LSI est implémentée en plusieurs phases

i. **Phase du prétraitement**
   - Suppression de toute ponctuation et les « stop words » (mots vides comme le, la, et pour etc…)
   - Création de la liste des mots clefs par document et dans toute la base de données.
   - Initialisation de la matrice Termes-Documents (TDM) représentant la relation entre les documents dans la base de données et les mots qui la constituent. Un algorithme de décomposition de matrices est ensuite appliqué pour décomposer la TDM en appliquant éventuellement des techniques d'optimisation de la qualité de la recherche en affectant des un poids local et global à chaque mot clef de la base donnée pour ajouter ou diminuer la valeur d'un terme.

ii. **Phase de traitement**
   - Décomposition SVD et k-approximation
   - Evaluation du système et analyse multi résolution

L'algorithme de décomposition le plus utilisé et la décomposition en valeurs singulières (ou Singular Value Decomposition en anglais - SVD), proposé par **Berry et al [43].**

La SVD décompose la TDM qui est creuse et de grande dimension pour éliminer le bruit dans la matrice en diminuant le dimensionnement de la TDM, pour trouver les relations sémantiques entre les termes et les documents tout en essayant de résoudre le problème de polysémie et synonymie. Choisir le rang optimal de réduction ($k - value$) est très important. Traditionnellement, le k optimal a été choisi en essayant plusieurs requêtes qu'on connait les documents pertinents pour différents valeurs de k. Le k qui renvoie le meilleur résultat est choisi comme le k optimal pour chaque collection. Finalement, l'ensemble de documents est comparé à la requête, et les documents qui sont proches à la requête utilisateur sont renvoyés. Plus de détails sont données au chapitre 3.

Aujourd'hui, les techniques de traitement d'image sont utilisées en corrélation avec la technique SVD comme étape de prétraitement dans le système LSI, en transformant la TDM en image en niveau de gris. Ces techniques sont généralement utilisées pour supprimer le bruit dans une image, visualisation et analyse de données.

1.6 Stratégies d'amélioration du processus de recherche

Dans le but d'accroître les performances des modèles de recherche, de nombreux stratégies sont mises en œuvre afin d'y être greffées. Ces stratégies exploitent diverses sources d'évidence : relations sémantiques définies dans le thesaurus, classes et contextes d'utilisation des concepts, résultats de recherche, jugement de pertinence des utilisateurs, éléments de la théorie de l'information, heuristiques, etc.

La requête apparaît comme le point de départ à toute recherche pour que puisse s'initier le processus de recherche d'information. Néanmoins, la question est de savoir s'il est possible d'aider l'utilisateur à formuler correctement ses choix et ainsi à lever les ambiguïtés potentielles. De nombreux travaux s'orientent désormais dans cette voie qui permet d'aider et de guider les utilisateurs pour dépasser le simple cadre de la requête.

La stratégie la plus répondue dans ce cadre est l'expansion de requêtes. On trouve par ailleurs d'autres stratégies, bien que moins répandues restent utiles. Nous décrivons dans ce qui suit quelques exemples d'entre elles

a) Expansion des requêtes

Afin de réduire la différence entre la pertinence système et la pertinence utilisateur l'exécution du processus de recherche a été rendu itérative. Cela demande à l'utilisateur non seulement de dépenser beaucoup de temps pour arriver à un résultat satisfaisant, mais lui demande surtout un gros effort cognitif pour exprimer son besoin d'information. Le temps et les efforts cognitifs qui sont demandés à l'utilisateur dépendent de son niveau d'expertise, c'est-à-dire qu'ils sont fonction de son expérience concernant, d'une part l'exécution du processus de RI, et d'autre part, du domaine d'information ciblé et contenu de la collection. C'est pour ces raisons que les études dans le domaine de RI ont comme objectifs principal d'améliorer l'efficacité de la recherche en termes de qualité du résultat, de temps et d'efforts demandés à l'utilisateur.

L'expansion des requêtes est une des solutions qui peut effectuer par des méthodes de bouclage de pertinence [**13**] [**44**] ou par l'utilisation de thésaurus [**46**] [**41**].

- ***Bouclage de pertinence***

  Le mécanisme de bouclage de pertinence suppose que la requête q_u de l'utilisateur u fournit un ensemble de documents que l'utilisateur évalueR_{qu}. De nouvelles requêtes sont ensuite générées à partir de ces jugements en ajoutant aux termes de la requête initiale des termes extraits de documents sélectionnés$R_{juge-per_u}$, on appelle cela l'expansion de requête en fonction du contexte, dans laquelle on prend en compte le jugement de l'utilisateur.

  Ce mécanisme est une approche ''individuelle'' de la fonction de bouclage de pertinence où la requête q_u et l'ensemble $R_{juge-per_u}$ des documents jugés pertinents par un seul utilisateur u sont pris en compte. On peut également remarquer que cette fonction ne prend en compte les documents que selon un seul aspect, celui de leur jugement de pertinence.

  Cependant, on pense que le fait de prendre en compte les documents sélectionnés selon d'autres aspects que la pertinence peut aussi être intéressant.

- ***Thésaurus***

Soit T un ensemble de termes et R un ensemble de relations de$T \times T$. Un thésaurus est défini par le couple (T, R). Un thésaurus est un ensemble de termes organisés suivant un nombre restreint de relations [**16**].

Les thésaurus sont principalement utilisés pour assister les documentalistes dans la tâche d'indexation manuelle de documents. Ils sont reconnus pour présenter différents avantages dans ce contexte [**17**]. Ils offrent tout d'abord une vue générale sur les termes et relations d'un domaine. Ils définissent ensuite un vocabulaire standardisé pour

l'indexation. Ils permettent d'assurer qu'un seul terme d'un ensemble de synonymes soit choisi pour l'indexation (terme dit ''à utiliser''). Ils sont également utilisés mors de la spécification d'une requête pour spécifier ou généraliser une recherche documentaire à partir des termes dits spécifiques ou plus génériques.

b) Classification interactive

La classification dont l'objet est de simplifier la recherche d'information peut également devenir interactive. La finalité des outils de classification interactive est de pouvoir définir dynamiquement une requête en fonction des thématiques successivement choisies par l'utilisateur. Ce type de classification produit une arborescence qui évolue à mesure que l'internaute comprend la nature des documents disponibles et découvre ceux qui sont les plus intéressants pour lui.

Plusieurs expérimentations ont été menées autour de ce type d'approche dite de *Scatter/Gather* pour analyser les questions posées en entrée et améliorer l'interactivité.

c) Corrélation entre documents

La corrélation (similarité) est une technique singulière qui permet à un utilisateur possédant un document électronique d'obtenir des documents similaires dans lesquels on aborde les mêmes thèmes pour compléter ou approfondir ses connaissances du sujet [**6**].

Parmi les nombreuses méthodes d'accès à l'information présentes sur Internet, la corrélation permet aux internautes d'enrichir leurs connaissances sur un document sans avoir à formuler la requête. Cette approche se détache des précédentes pour deux raisons : tout d'abord cet outil ne permet pas, à l'évidence, d'effectuer directement une recherche, il s'agit d'un outil complémentaire qui permet, une fois un document intéressant identifié, de compléter ses connaissances et d'obtenir d'autres documents sémantiquement proches. La corrélation est de ce fait une voie de recherche un peu marginale car toujours adossée à un autre outil de recherche, le plus souvent elle est directement intégrée aux moteurs de recherche eux-mêmes.

d) L'interface et visualisation du processus de recherche

Pour offrir aux utilisateurs des interfaces plus faciles à manipuler et à comprendre en répondant au double but d'économiser le temps de l'utilisateur ainsi que son niveau d'expertise requis. Aroyo propose un système AIMS (Agent-based Information Management System) qui visualise le résultat de la recherche sous forme graphique non seulement facile d'accès, mais aussi facile à sélectionner et à comprendre. Elle fournit à l'utilisateur une information complémentaire sur le contexte des résultats retrouvés. Le graphe est présenté comme une carte de concepts qui décrit explicitement la structure de domaine et tous les documents sont rattachés aux concepts. Cela permet à l'utilisateur de traiter rapidement un plus grand nombre de documents retrouvés et d'en avoir une vision synthétique. Ce type de visualisation provoque l'Imagination de

l'utilisateur en lui fournissant une connaissance complémentaire sur le contexte et la signification de l'information employée.

1.7 Grands projets dans l'histoire de la RI

Nous présentons dans cette section quelques projets qui ont marqué le domaine de la Recherche d'Information.

a) Cranfield (dirigé par Cyril Cleverdon, 1957-1967) [7]

Dans ce projet, on visait à tester l'efficacité de différentes façons d'indexer et de recherche des documents. Ces tests sont vigoureusement contrôlés. Une collection de test est constitué d'un ensemble d'articles (18,000 dans Cranfield I) est un ensemble (1,200) de requêtes. Ces requêtes sont évaluées par des experts afin de déterminer les réponses souhaitées ''les articles pertinents''. Les résultats d'une recherche automatique sont comparés avec les réponses souhaitées pour mesurer la performance en termes de précision et rappel. Le projet Cranfield a une influence marquante sur toute l'histoire de la RI. On utilise encore aujourd'hui les mêmes principes d'évaluation pour les systèmes de RI.

b) MEDLARS – MEDical Literature Analyses and Retrieval System

Comme l'indique son nom, les documents dans la collection sont dans le domaine biomédical. Ces documents sont indexés manuellement, avec un vocabulaire contrôlé. Les résultats de ce projets montrent qu'en utilisant une approche automatique, il est possible d'atteindre la même performance avec indexation manuelle et un vocabulaire contrôlé. Une analyse des résultats a aussi montré que l'utilisation d'un vocabulaire contrôlé et de l'indexation manuelle étaient largement responsable des cas d'échecs dans la recherche de documents pertinents, qui peuvent être évités par l'approche automatique.

c) SMART–System for the Mechanical Analysis and Retieval of Text [30]

Dans ce projet, une série d'expérimentations a été menée, portant sur divers sujets comme :

- La comparaison entre l'indexation manuelle et l'indexation automatique

- Le problème de recherche interactive et rétroaction de pertinence (relevance feedback)
- L'architecture de système de RI
- L'utilisation du modèle vectoriel
- Le regroupement de documents (ou clustring)
- Etc.

Le système SMART fut réécrit dans les années 1970 et 1980 par E. Fox et C. Buckley. Ce système a été, et est encore utilisé par de nombreux chercheurs pour des expérimentations en RI. Le système SMART est sans doute le système qui a eu le plus grand impact sur l'histoire de la RI.

d) STAIRS – Storage and Information Retrieval system, Blair et Maron

Les documents sons dans le domaine de droit. L'indexation automatique utilise la troncation de suffixes, et la recherche exploite une liste de synonymes. Contrairement aux expérimentations antérieures qui utilisent de petites collections, les tests de Blair et Maron portent sur une collection de taille réaliste (40,000 documents totalisant 35,000 pages). Le résultat montre que la performance de moins de 20% de rappel est insuffisante dans le domaine de droit pour lequel le rappel est très important.

e) TREC – TextREtrievalConference, D. Harman :

Cette série de conférences a pour objectifs de tester des méthodes et des systèmes de RI avec des collections de plus grande taille. Elle est organisée annuellement. Les tâches (tracks) changent d'une année sur l'autre, mais elles reflètent bien les intérêts des chercheurs et les besoins réels.

Au fil des années, il y a eu la tâche ad hoc (la tâche classique de RI – soumettre des requêtes sur une collection statiques), le filtrage de l'information, la tâche de recherche sur des documents non anglais (en espagnol, français, chinois) et translinguistique (retrouver des documents dans une langue différente de celle de la requête), les questions-réponses, la RI multimédia (vidéo et parole), etc.. .. Ces conférences attirent chaque année des chercheurs universitaires et industriels. Les conférences TREC ont grandement attribué au développement récent de la RI, en fournissant des collections de

test réalistes, et en offrant une nouvelle méthodologie d'évaluation. Elles ont grandement stimulé le domaine de RI.

Amaryllis est la version française du projet **TREC**. Il a pour objectif principal d'évaluer des logiciels de recherche d'information dans des corpus de texte en français. Des collections de textes également fournies dans ce cadre.

Chapitre 2. Spécifications fonctionnelles de l'Indexation sémantique latente (LSI)

Dans ce chapitre, après une présentation du modèle VSM, on va présenter l'algorithme d'Indexation Sémantique Latente (LSI).

2.1 Introduction au VSM (Vector Space Model)

VSM est une technologie de RI qui est basée sur le concept d'espace des vecteurs. Spécifiquement, les termes, les documents et les requêtes sont tous des vecteurs dans une espace des vecteurs. Le modèle a été présenté pour la première fois par G. Salton. Dans ce modèle [**29**], la base de données est représentée comme une matrice termes-documents (TDM) et tous les documents dans la base de données sont représentés comme des colonnes dans la matrice et tous les termes comme des lignes dans la matrice. Un document est représenté par un vecteur $d = (d_1, d_1, \dots, d_n)$ où chaqued_i est nombre indiquant le degré d'importance du termet_i dans le document d. Autrement dit, chaque document est représenté par un vecteur dans un espace des vecteurs de dimension n. De la même façon, une requête est représentée par un vecteur q dans cet espace. Dans ce modèle, les documents sont représentés par un ensemble de termes qui peuvent être pondérés (poids global et local) et manipulés. Les documents pertinents dans la base de données sont renvoyés à la requête via des opérations vectorielles simples.

VSM a été développé pour éliminer plusieurs problèmes liés aux techniques utilisant des mots-clés traditionnels, spécialement la synonymie et la polysémie. La fonction de recherche pour ce modèle est basée sur la signification sémantique et conceptuelle des documents, elle fournit un mécanisme pour comparer les termes dans une requête aux termes dans un document, aussi bien que la comparaison entre les documents dans la base de données. Ayant tous les composants de la RI dans le même espace des vecteurs, et calculant la similitude entre eux, permet d'avoir le résultat désiré. Ceci signifie que les résultats qui sont conceptuellement plus pertinents peuvent être renvoyés automatiquement aux utilisateurs.

Une représentation de l'espace des vecteurs des documents tridimensionnels est montrée dans la Fig. 2.1, où chaque document d_i, $i = 1, 2\ 3$ se compose de trois termes T_1, T_2, T_3. L'exemple tridimensionnel peut être prolongé à n dimensions où n est le nombre des différents termes représentant le document.

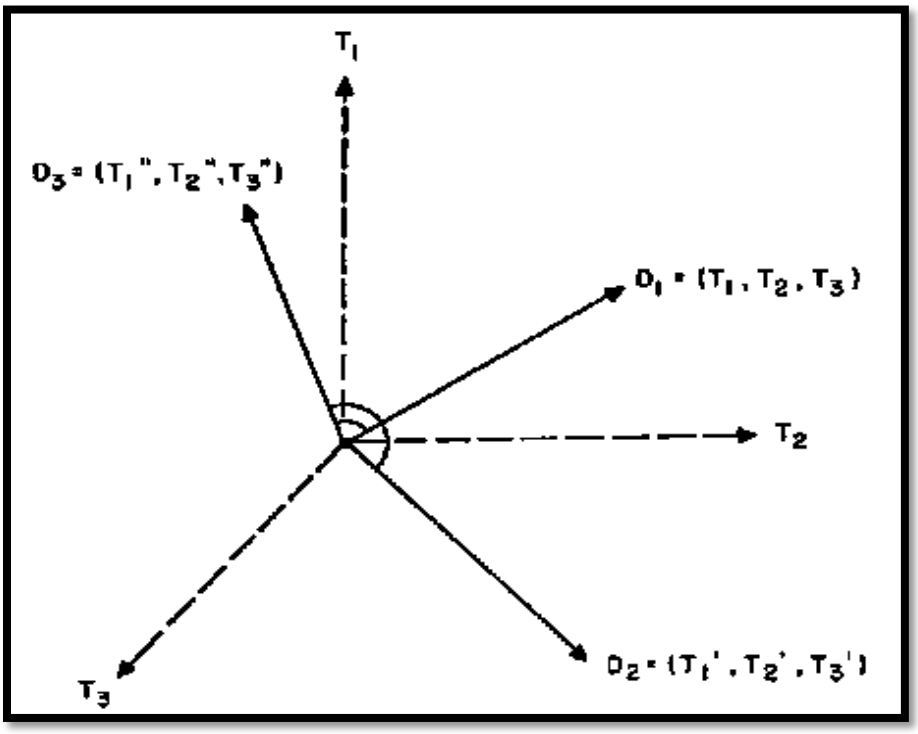


Figure 2.1 : Représentation de l'espace des vecteurs des documents tridimensionnels

Puisque la configuration de l'espace de document est en fonction des termes ou des poids des termes qui sont assignés aux divers documents de la base de données, on peut se demander si une configuration « optimum » de l'espace de document existe, c.-à-d. une configuration qui produit une performance optimale de recherche. Si rien de spécial n'est connu au sujet des documents à l'étude, on pourrait considérer qu'un espace de documents est idéal quand des documents qui sont conjointement appropriés à la requête d'utilisateur sont groupés ensemble, donc ils seraient proposés conjointement en réponse à la requête de l'utilisateur.

La configuration de documents de la Fig. 2.2 peut représenter la meilleure situation, en supposant que les documents pertinents et non pertinents en ce qui concerne les diverses requêtes sont séparables comme le montre la figure. Dans son travail de brevet [**29**], Salton clarifie cela, aucune manière pratique n'existe pour

produire réellement un tel espace, parce qu'il est difficile de produire la configuration optimale en l'absence de la connaissance des détails complets de la recherche pour la base donnée. Dans ces circonstances, le besoin d'avoir recours à LSI se fait sentir, puisque cette technique peut aider en fournissant un tel espace riche de vecteur.

Comme défini précédemment **[43]** **[57]** la LSI est un modèle de l'espace de vecteur qui a recours à la décomposition en valeur singulière (SVD). Cependant, il y a une différence importante entre la LSI et le VSM, à savoir la LSI utilise une approximation de qualité inférieure pour la représentation de l'espace de vecteur de la base de données. C'est-à-dire, la TDM originale est remplacée par une autre matrice qui est assez semblable à la TDM originale mais dont l'espace de colonne est seulement un sous-espace de l'espace de colonne de la matrice originale. L'algorithme de SVD est employé dans la LSI pour réduire l'espace de vecteur, enlever le bruit ou la redondance lexicologique (qui sont illustrés dans la prochaine section) de la TDM, afin d'essayer de résoudre le problème d'inexactitude lié à la synonymie et à la polysémie.

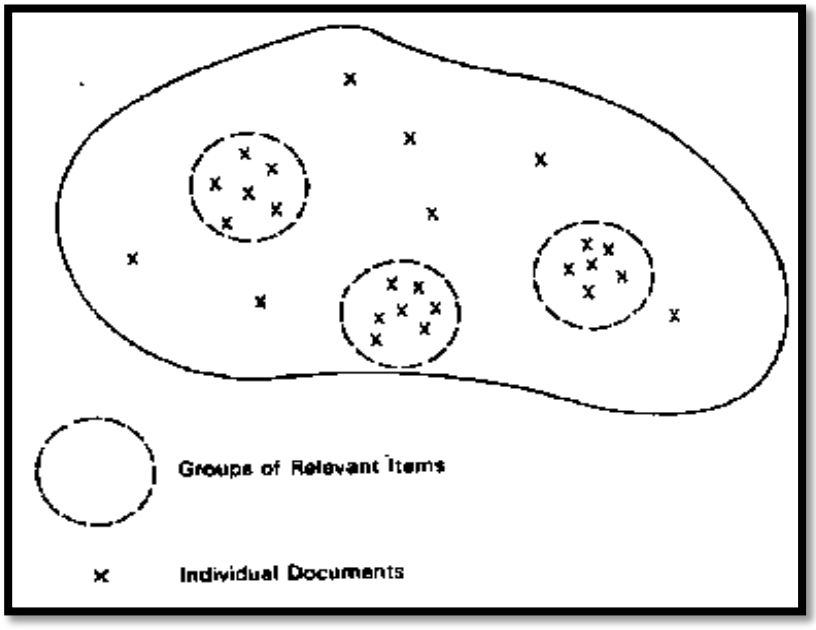


Figure 2.2 : Représentation idéal de l'espace des documents

La LSI fournit clairement un espace de vecteur riche, qui exploite les rapports sémantiques latents entre les limites et les documents. La réduction de l'espace de vecteur a l'effet d'indiquer le rapport sémantique fondamental parmi les documents, parce qu'une grande partie du bruit dans la matrice est enlevé.

2.2 Bruit lexical

Après la création de la TDM, qui est une matrice bidimensionnelle représentant le nombre fois un mot-clé apparait dans chaque document dans la base de données, la matrice résultante sera creuse, une grande partie des éléments constituant la matrice sont des zéros, car chaque mot-clé apparaîtra seulement dans quelques documents. Les zéros dans la matrice représentent le bruit lexical ou la redondance. Dans le système dans LSI, l'algorithme SVD est employé pour enlever ce bruit lexical dans la TDM originale, pour assurer la relation sémantique entre les termes et les documents en essayant de résoudre les problèmes d'inexactitude dans les modèles traditionnels de la RI.

Le bruit est généralement classé en trois catégories :

- **Bruit traditionnel**, les mots vides ou coupure des mots (stopwords) (le, la, pour…). Ce type de bruit est généralement traité et enlevé à l'étape de prétraitement.
- **Bruit généré** par une male structuration de la base de données ou du modèle de la requête utilisé. Les descriptions les plus longues augmentent le nombre des mots-clés et la distribution des valeurs différentes de zéros dans la TDM, qui ensuite aide à améliorer les relations sémantiques entres les documents. De l'autre côté, les descriptions les plus courtes représentent des structures pauvres qui ne soutiennent pas la recherche par LSI, et augmente la redondance dans la TDM.
- Autres types de bruit produit par des *spammeurs* essayant d'éviter les systèmes des filtres.

2.3 L'algorithme LSI

Comme mentionné dans le premier chapitre, la plus grande partie du travail sur la LSI concentre sur la phase de prétraitement, et sur les algorithmes de décomposition utilisés pour l'approximation de la TDM.

La figure 2.3 illustre les différents composants du système LSI :

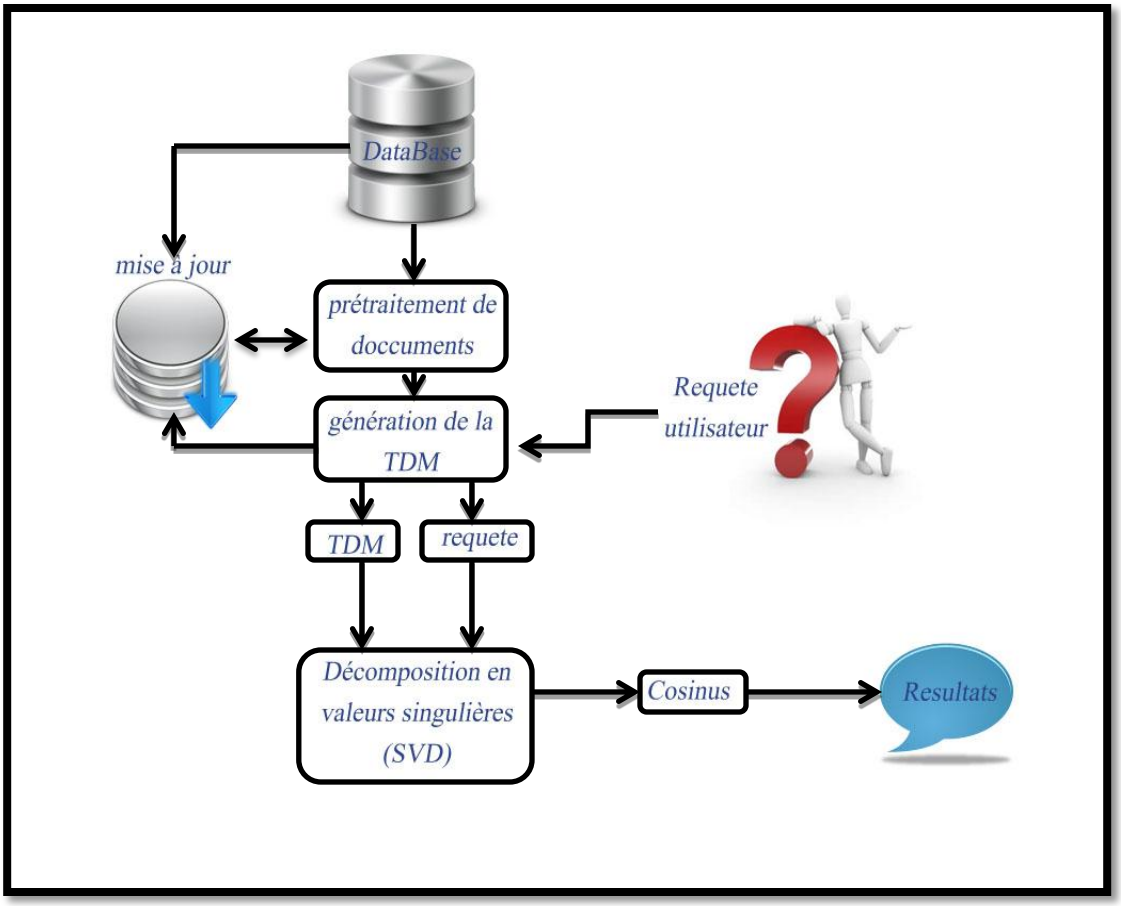


Figure 2.3 : Composants du système LSI proposé

2.3.1 Description de la base de données

La base de données contient des titres des documents sur lesquels la recherche est effectuée. Cette section décrit la structure et les contenus des bases de données utilisées dans ce travail.

- **Structure de la base de données**

  Chaque base de données utilisée est représentée comme une simple table sous Microsoft Access. Les tables ont la forme : (ID, Title), le champ ''Title'', contient les titres des documents et le champ ''ID'' contient une clé unique pour chaque entrée dans la table, simplifiant le référencement des documents.

- **Contenu de la base de données**

  Les documents utilisés dans les expériences sont tenus sous la forme de quatre bases de données. La « Memos », une simple base de données qui contient neuf

notes techniques de Bellcore, qui est très utilisée comme un exemple pratique dans plusieurs articles travaillant sur la LSI. Les cinq premiers documents concernent l'interaction Homme-Machine (HCI), et les autres concernent sont reliés aux structures des données. Nous avons inclus cette base de données pour présenter les références de base. La « Cochrane », une autre base de données de 135 documents contenant les titres des études médicales sur l'administration des médicaments, qui est aussi un exemple couramment utilisé dans les tests des systèmes LSI. Elle est disponible sur le site de Cochrane. La troisième est la « eBooks », qui est une grande collection de données contenant les titres de 658 livres électroniques conservé par '' The Science Library at Queens University''. Elle a été choisie pour sa grande taille et la bonne structure. La dernière est le sous-ensemble de la collection de catégorisation de texte (Text Categorisation Collection – TCC) « Reuteurs-21578 ». Elle est la plus utilisée dans les tests de recherches sur la catégorisation de texte. Les données ont été collectionnées par ''Carnegie Group, Inc. And Reuters, Ltd.'', un sous-ensemble de 1000 titres approximativement est utilisé. La collection de test actuelle contient un volume important d'information, mais pour simplifier, on travaille seulement sur les titres des documents.

2.3.2 Prétraitement

Dans cette section, les étapes à traiter concernent : l'identification et suppression des mots vides (ou stop words), Stemming algorithms, pondération des mots clés, contrôle pertinence et mise à jour de la base de données.

- **Mots vides ou coupure des mots (stop words)** : la recherche dans ce domaine est basée sur ce qui constitue les mots clés dans la base de données, qui décrivent la base de données et sont utilisés comme des références aux titres des documents. La règle utilisée par la plus part des chercheurs **[43]** **[57]** exige qu'un mot clés apparait dans quelque documents mais n'apparait pas dans tous les documents. Un mot qui apparait dans un seul document ou dans tous les documents doit être éliminé car il a peu ou pas de capacité pour améliorer la relation sémantique entre les documents de la TDM. Le but de cette partie est d'extraire les mots qui ont une signification et d'enlever la ponctuation, adjectives et les mots qui sont considérés sans signification comme : « and », « or », « in »...

Et donc tous les mots qui apparaissent dans plusieurs documents et qui ne sont pas des « stop words » sont inclus. La liste des « stop words » construite par FOX a été acceptée comme une norme pour identifier les mots qui n'ont pas de sens qui peuvent être éliminés de la liste des mots clés [**48**].

Exemple de la base de données Memos :

Pour illustrer l'étape de prétraitement, prenons l'exemple de la base de données Memos, les titres sont présentés dans la Table 2.1

ID	Title
B1	**human computer interface** for ABC computer applications
B2	a **survey** of **user** opinion of **computer system response time**
B3	the **EPS user interface** management **system**
B4	**system** and **human system** engineering testing of **EPS**
B5	relation of **user** perceived **response time** to **error** measurement
B6	the generation of random, binary and ordered **trees**
B7	the intersection of paths in **trees**
B8	**graph minors** IV : widths of **trees** and well-quasi ordering
B9	**graph minors** : a survey

Tableau 2.1 : La base de données Memos

Le prétraitement produit la liste des mots clés suivante : {**human, computer, interface, survey, user, system, response, time, EPS, error, trees, graph, minors**}

Term Document Matrix

Une fois le prétraitement est terminé, la TDM est construite à partir de la liste des mots clés, chaque ligne de la matrice correspond à un mot clés, et chaque colonne correspond à un document. La valeur qui de la position (i, j) de la matrice représente le nombre de fois le terme correspondant à la $i^{ème}$ ligne apparait dans le document correspondant à la $j^{ème}$ colonne. La plus part des éléments de la matrice seront nuls car un mot clé n'apparait que dans quelques documents. Il est intéressant de voir la relation entre les termes des documents. Les mots qui apparaissent dans un seul document sont enlevés car ils n'ajoutent aucune information à cette relation. De la même manière, les mots qui

apparaissent dans tous les documents sont enlevés aussi. Ça mène à éliminer les lignes qui ne contiennent qu'une seule valeur non nulle. La TDM générée pour la Memos est montrée dans le Tableau 2.2

	C1	C2	C3	C4	C5	M6	M7	M8	M9
computer	2	1	0	0	0	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
graph	0	0	0	0	0	0	0	1	1
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
minors	0	0	0	0	0	0	0	1	1
response	0	1	0	0	1	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
system	0	1	1	2	0	0	0	0	0
time	0	1	0	0	1	0	0	0	0
trees	0	0	0	0	0	1	1	1	0
user	0	1	1	0	1	0	0	0	0

Tableau 2.2 : La TDM générée pour la base de données Memos

Chaque colonne dans la base de données peut être considérée comme un vecteur décrivant un document particulier, chaque ligne peut être considérée comme un vecteur décrivant le terme qu'il représente. Il y'a sans doute beaucoup de redondance (bruit lexical) dans ce processus, comme illustré par la matrice creuse. Le processus LSI cherche à éliminer la redondance en décomposant la TDM en utilisant l'algorithme SVD.

Vecteur requête :

Les requêtes doivent être aussi représentées sous la forme vectorielle. Ceci est réalisé de la même manière utilisée pour transformer les documents en colonnes dans la TDM. La requête est représentée par un vecteur où chaque élément représente le nombre de fois un terme de la liste des mots clés apparait dans cette requête. Par exemple la requête 'response time' et transformée sous la forme (0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0), le terme 'response' correspond à la septième ligne de la TDM, et 'time' correspond à la dixième ligne, et chacun de ses termes apparait une seule fois dans la requête. Si les pairs des mots sont inclus, la requête devient donc (0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0),

'response' est la septième ligne et apparait une fois, 'response time' est la huitième ligne et apparait une fois, et 'time' est la onzième ligne et apparait une fois.

Pour améliorer les résultats de recherche, on peut utiliser plusieurs techniques, tels que :

- **Stemming Algorithm** : le premier Stemming Algorithm publié en 1968 [**64**]. Mais l'algorithme le plus cité a été introduit par Poter en 1980 [**47**]. Un Stemming Algorithm dérive les mots en radicaux, par exemple les mots clés « read », « reader » et « reading » peuvent être remplacés par le radical « read ». Ce radical peut être donc utilisé comme un mot clé plutôt que d'utiliser trois mots clés [**48**]. En conséquence, le Stemming Algorithm réduit l'espace de stockage des mots clés. L'idée principale est que les utilisateurs qui cherchent des informations sur « retrieval » seront aussi intéressés par des articles sur « retrieve », « retrieved », « retrieving », « retriever » etc. Cependant, l'utilité d'un Stemming Algorithm pour améliorer la qualité de recherche a toujours été mise en question dans la communauté de la recherche. Par conséquent, beaucoup de chercheurs ont évité l'utilisation d'un Stemming Algorithm dans la LSI [**43**].
- **Pondération des mots clés** : la pondération des mots clés est l'une des méthodes courante pour améliorer les performances de la recherche. Elle consiste à donner aux termes dans la TDM différents poids. En pratique, des poids locaux et globaux pour augmenter ou diminuer l'importance d'un mot dans les documents de la TDM. Le poids global reflète l'importance d'un terme dans toute la base de données. Le poids local reflète l'importance d'un terme dans un document donné. Quelques chercheurs ne tiennent compte d'aucune pondération des termes de la TDM et utilisent un modèle simple de TDM non-pondéré [**57**].

Dans le VSM classique proposé par Salton et autres, les poids des termes pour les documents correspondants dans la TDM est le produit des poids locaux et globaux. Le model est connu par « term frequency-inverse document frequency » (tf-idf). Le vecteur des poids pour un document d est :$V_d = \left[w_{1,d}, w_{2,d}, \dots, w_{n,d}\right]^T$, où :

$$\boldsymbol{w_{t,d} = tf_t \cdot \frac{\log|D|}{|\{t \in d\}|}} \qquad (2.1)$$

Et

- tf_t est la fréquence du terme t dans le document d (paramètre local).
- $\frac{\log|D|}{|\{t \in d\}|}$ est la fréquence inverse du document (paramètre global). $|D|$ est le nombre total des documents dans la base de données ; $|\{t \in d\}|$ est le nombre des documents contenant le terme t.

Dans le VSM simple, les poids des termes n'incluent pas le paramètre global. Les poids utilisés dans la TDM sont les fréquences des termes seulement (paramètre local) : $w_{t,d} = tf_t$.

- **Contrôle de pertinence (Relevance Feedback):** ce processus peut être identifié comme : un procédé automatique contrôlé pour la reformulation de la requête [**26**]. Souvent, l'utilisateur ne trouve pas tous les documents pertinents à la première tentative, ceci est dû à la requête pauvre en information, qui n'exprime pas exactement ce que l'utilisateur cherche. La recherche peut être améliorée en rassemblant le feedback de l'utilisateur sur la pertinence des documents renvoyés. Fondamentalement, le processus est comme suit :

- Après que les résultats préliminaire de la recherche soient présentés, permettre à l'utilisateur de donner un feedback sur la pertinence des documents renvoyés.
- Utiliser ces informations pour reformuler la requête.
- Présenter de nouveaux résultats basés sur la requête reformulée.

Cette requête reformulée est générée comme suit :

- Expansion de la requête : ajouter des nouveaux termes à la requête à partir des documents pertinents.
- Repondération des termes : augmenter les poids des termes dans les documents pertinents et diminuer les poids des termes dans les documents non pertinents [**52**].

Dans d'autres systèmes de LSI, une méthode différente pour la reformulation de la requête est adoptée. L'information de l'utilisateur pour le contrôle de pertinence est utilisée pour formuler une nouvelle requête en ajoutant les vecteurs des documents pertinents au vecteur requête, qui peut être considérée comme la somme des documents

pertinents de la première requête. Ou d'un autre côté, soustraire les vecteurs des documents non pertinents du vecteur requête.

- **Mise à jour (updating) :** il est probable que les bases de données nécessitent des modifications. L'information est continuellement ajoutée ou enlevée. Dans un système LSI, l'approche standard pour manipuler les additions est de recalculer la SVD de la nouvelle TDM, qui nécessite beaucoup de calculs, surtout pour les grandes bases de données. Pour éviter ces calculs, d'autres techniques ont été considérées, par exemple Folding-in et SVD updating. Folding-in ne demande pas beaucoup de calculs [**43**] [**57**] [**24**] et [**31**], mais résulte une représentation imprécise de la base de données. La mise à jour de la SVD demande beaucoup de calculs mais elle a l'avantage de préserver la représentation de la base de données.

2.3.3 Implémentation des algorithmes de décomposition des matrices :

Maintenant que la TDM est générée, la décomposition de matrice est appliquée pour enlever le bruit dans la TDM. Cette section illustre l'utilisation de l'algorithme de décomposition le plus utilisé, l'algorithme SVD

La décomposition en valeurs singulières – Singular Value Decomposition (SVD)

Une matrice M peur être décomposée en une forme réduite et approximative comme :

$$M = U * S * V^T$$

Où :

$$\begin{cases} M = TDM \\ U: vecteurs\ lignes\ singuliers\ de\ M \\ S: matrice\ diagonale\ contenant\ les\ valeurs\ singuliers\ de\ M\ par\ ordre\ décroissant \\ V^T: transposées\ des\ vecteurs\ colonnes\ singuliers\ de\ M \end{cases}$$

Remarque : Les éléments diagonaux de S sont représentés en ordre décroissant, les valeurs les plus grandes sont porteurs potentiels de ***''contenu sémantique''*** de M (Fig. 2.4) Les valeurs les plus petites peuvent être vues comme un ***''bruit lexical''*** [**48**].

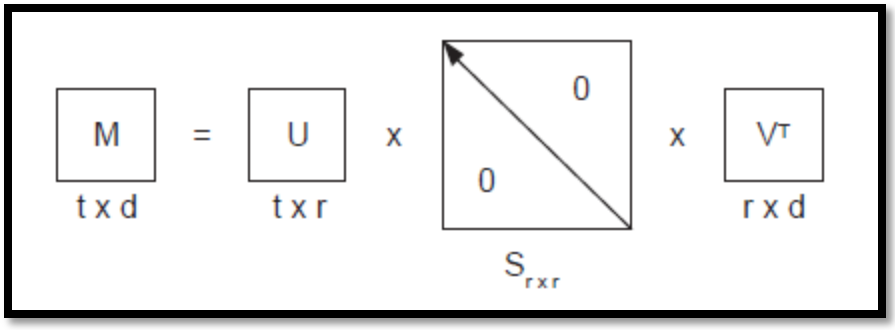


Figure 2.4 : La décomposition SVD pour une TDM $t * d$

Où :

$t = nombre\ de\ \ termes\ de\ \ la\ \ \ collection\ \ = nombre\ de\ lignes\ de\ M$
$d = nombre\ \ de\ \ documents\ \ de\ \ la\ \ collection\ \ = mbre\ \ de\ colonnes\ \ de\ M$
$r = \ \ rang\ \ de\ \ la\ \ matrice\ \ M$

Au cœur d la LSI, la structure sémantique latente de la collection des documents est identifiée par la matrice des valeurs diagonales.

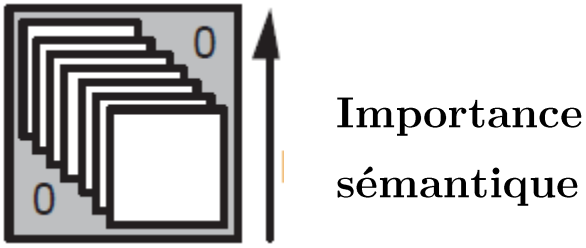


Figure 2.5 : La matrice diagonale S. Les bloques interne représentent les valeurs singulières

La TDM originale peut être approximée en multipliant les trois matrices U, S et V. Si les petites valeurs singulières de S ont été éliminées, la TDM est approximée par :

$$M_a = U_k * S_k * V_k^T$$

Où :

$$M = TDM$$
$$\begin{cases} U_k : matrice\ composée\ des\ k\ premiers\ colonnes\ de\ U \\ S_k : matrice\ diagonale\ composée\ de\ la\ recine\ carrée\ des\ valeurs\ singulières\ de\ M\ par\ ordre\ décroissant \\ V_k^T : matrice\ des\ k\ premiers\ colonnes\ de\ V^T \end{cases}$$

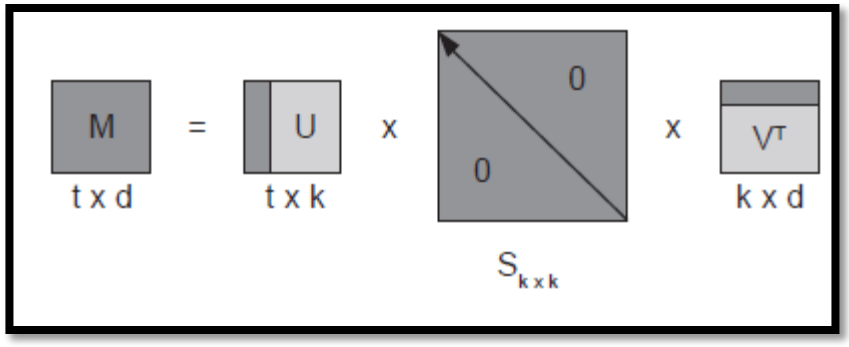


Figure 2.6 : La TDM approximée

La matrice résultante a les mêmes dimensions que la TDM originale et représente la meilleure approximation de M de rang k, où chaque document peut être approximé par : $\boldsymbol{d_k = d^T U_k S_k^{-1}}$

Et de la même manière, la requête est approximée par : $\boldsymbol{q_k = q^T U_k S_k^{-1}}$.

La requête est comparée ensuite avec tous les documents dans la nouvelle matrice. Avec un « bruit lexical » supprimé, ça va améliorer les résultats quand la requête est comparée aux documents approximés.

Pour illustrer l'utilisation de l'algorithme SVD, on prend l'exemple de la base de données Memos.

Si on calcule la similarité entre les vecteurs termes « human » et « user » et entre « human » et « minors » dans la TDM originale en utilisant la formule de cosinus (1.2) on trouve : $sim(human, user) = -0.38$ et $sim(human, minors) = -0.29$

Dans la matrice originale $human$ n'apparait pas dans le même passage avec soit $user$ ou $minors$, ils n'ont pas une cooccurrence ou ressemblance, la corrélation égale à $-0{,}38$ entre $user\ et\ human$ et un peu plus élevé -0.29 entre $human\ et\ minors$.

La décomposition linéaire est illustrée dans les Tableaux 2.3, 2.4 et 2.5, sauf pour les erreurs d'arrondi, le produit des trois matrice U, S et V reconstruit parfaitement la matrice originale comme illustré. Ensuite, on montre une reconstruction basée sur seulement deux dimensions, l'approximation de la matrice d'origine. Celui-ci

utilise seulement les deux premiers éléments vectoriels ($k = 2$) (ce qui équivaut à la mise à zéro de la matrice S sauf les deux grandes valeurs en haut).

$$U = \begin{matrix} 0.22 & -0.11 & 0.29 & -0.41 & -0.11 & -0.34 & 0.52 & -0.06 & -0.41 \\ 0.20 & -0.07 & 0.14 & -0.55 & 0.28 & 0.50 & -0.07 & -0.01 & -0.11 \\ 0.24 & 0.04 & -0.16 & -0.59 & -0.11 & -0.25 & -0.30 & 0.06 & 0.49 \\ 0.40 & 0.06 & -0.34 & 0.10 & 0.33 & 0.38 & 0.00 & 0.00 & 0.01 \\ 0.64 & -0.17 & 0.36 & 0.33 & -0.16 & -0.21 & -0.17 & 0.03 & 0.27 \\ 0.27 & 0.11 & -0.43 & 0.07 & 0.08 & -0.17 & 0.28 & -0.02 & -0.05 \\ 0.27 & 0.11 & -0.43 & 0.07 & 0.08 & -0.17 & 0.28 & -0.02 & -0.05 \\ 0.30 & -0.14 & 0.33 & 0.19 & 0.11 & 0.27 & 0.03 & -0.02 & -0.17 \\ 0.21 & 0.27 & -0.18 & -0.03 & -0.54 & 0.08 & -0.47 & -0.04 & -0.58 \\ 0.01 & 0.49 & 0.23 & 0.03 & 0.59 & -0.39 & -0.29 & 0.25 & -0.23 \\ 0.04 & 0.62 & 0.22 & 0.00 & -0.07 & 0.11 & 0.16 & -0.68 & 0.23 \\ 0.03 & 0.45 & 0.14 & -0.01 & -0.30 & 0.28 & 0.34 & 0.68 & 0.18 \end{matrix}$$

Tableau 2.3 La matrice U générée après application de l'algorithme SVD

$$S = \begin{matrix} 3.34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.54 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.35 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.64 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.50 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.31 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.85 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.56 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.36 \end{matrix}$$

Tableau 2.4 La matrice S générée après application de l'algorithme SVD

$$V = \begin{matrix} 0.20 & 0.61 & 0.46 & 046 & 0.28 & 0.00 & 0.01 & 0.02 & 0.08 \\ -0.06 & 0.17 & -0.13 & -013 & 0.11 & 0.19 & 0.44 & 0.62 & 0.53 \\ 0.11 & -0.50 & 0.21 & 0.57 & -0.51 & 0.10 & 0.19 & 0.25 & 0.08 \\ -0.96 & -0.03 & 0.04 & 0.27 & 0.15 & 0.02 & 0.02 & 0.01 & -0.03 \\ 0.05 & -0.21 & 0.38 & -0.21 & 0.33 & 0.39 & 0.35 & 0.15 & -0.60 \\ -0.08 & -0.26 & 0.72 & -0.37 & 0.03 & 0.30 & -0.21 & 0.00 & 0.36 \\ 0.18 & -0.43 & -0.24 & 0.26 & 0.67 & -0.34 & 0.15 & 0.25 & 0.04 \\ -0.01 & 0.05 & 0.01 & -0.02 & -0.06 & 0.45 & -0.76 & 0.45 & -0.07 \\ -0.06 & 0.24 & 0.02 & -0.08 & -0.26 & -0.62 & 0.02 & 0.52 & -0.45 \end{matrix}$$

Tableau 2.5 La matrice V générée après application de l'algorithme SVD

Ce qui nous donne $\widehat{M} = U_2 S_2 V_2{}'$

On calcule maintenant la similarité entre les deux vecteurs termes précédents, on trouve :

$sim(human, user) = 0.94$ et $sim(human, minors) = -0.83$

Dans la matrice reconstituée à partir de deux dimensions (rang $k = 2$) en raison de leurs relations indirectes: la corrélation entre $human - user$ est passée à 0.94, et entre $human - minors$ a diminuée à -0.83.

	C1	C2	C3	C4	C5	M6	M7	M8	M9
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
human	0.16	0.40	0.38	0.47	0.18	0.05	0.12	0.16	0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
minors	0.16	0.58	0. 38	0.42	0.28	0.06	0.13	1	1
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
trees	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19

Tableau 2.6 : La TDM approximée de la base de données Memos pour $k = 2$.

Ainsi, pour que les termes $human\ et\ user$ se trouvent dans des contextes similaires de sens, même s'ils n'ont pas apparu dans le même passage, la solution de la réduction de dimension a bien représenté la similarité des termes.

Pour examiner l'effet de la réduction de la dimension sur les relations entre les documents, nous avons calculé les corrélations entre chaque document et tous les autres, avant et après application de l'algorithme SVD ; Tables 2.7 et 2.8.

Dans les premières occurrences de données, les corrélations entre les cinq documents qui traitent l'interaction homme-machine ont été généralement faibles, même si tous les papiers étaient ostensiblement sur des sujets similaires, la moitié des r (corrélations)

étaient nulles, trois se sont révélés négatifs, deux étaient modérément positives, et la moyenne n'était que 0.02. Les corrélations entre les quatre documents de structures des données ont été mixtes, avec une moyenne de 0.44. Les corrélations entre les documents HCI et structures des données ont une moyenne de -0.30, malgré le chevauchement minimal conceptuel des deux sujets.

	C1	C2	C3	C4	C5	M1	M2	M3
C2	-0.19							
C3	0.00	0.00						
C4	0.00	0.00	0.47					
C5	-0.33	0.58	0.00	-0.31				
M1	-0.17	-0.30	-0.21	-0.16	-0.17			
M2	-0.26	-0.45	-0.32	-0.24	-0.26	0.67		
M3	-0.33	-0.58	-0.41	-0.31	-0.33	0.52	0.77	
M4	-0.33	-0.19	-0.41	-0.31	-0.33	-0.17	0.26	0.56

Tableau 2.7 : Corrélation entre les titres avant application de SVD

	C1	C2	C3	C4	C5	M1	M2	M3
C2	0.91							
C3	1.00	0.91						
C4	1.00	0.88	1.00					
C5	0.85	0.99	0.85	0.81				
M1	-0.85	-0.86	-0.85	-0.88	-0.45			
M2	-0.85	-0.86	-0.85	-0.88	-0.44	1.00		
M3	-0.85	-0.86	-0.85	-0.88	-0.44	1.00	1.00	
M4	-0.81	-0.50	-0.81	-0.84	-0.37	1.00	1.00	1.00

Tableau 2.8 : Corrélation dans un espace à deux dimensions (k=2)

Pour le cas de deux dimensions, la moyenne r entre les titres HCI augment de 0.02 à 0.92. Ce qui s'est passé, non pas que les documents sur l'interaction homme-machine étaient généralement semblables les uns aux autres, dont ils n'étaient, mais parce qu'ils contrastent avec les titres non-HCI dans les mêmes manières. De même, les corrélations entre les documents de structures des données ont été ré-estimé tous à 1.00 et ceux entre les deux sujets sont maintenant négatives.

2.4 Les application de la LSI

La LSI est évidemment employée dans la recherche d'information, mais la richesse des applications qui emploient la LSI démontre combien cette technique est efficace, et que le travail aura un impact plus large dans le futur. Cette section présente une vue d'ensemble des diverses applications dans différents secteurs qui emploient les techniques de la LSI. Les applications les plus étonnantes de la LSI ont été dans des domaines autres que la Recherche d'Information.

a) Du point de vue applications de traitement d'image, SVD a été utilisé avec des algorithmes de tatouage pour résoudre le problème de droit d'auteur, protection des documents multimédias. La réduction de rang a été utilisée dans la cryptographie et en traitement d'images. L'utilisation de la technique LSI pour la récupération basée sur le contenu du document Web a été examinée, en utilisant les deux mots-clés et les caractéristiques d'image pour représenter les documents. Les résultats expérimentaux montrent que LSI, avec des caractéristiques à la fois textuelle et visuelle, a la capacité d'identifier le concept sous-jacent sémantique des documents Web, résultant dans l'amélioration de l'exécution d'extraction.

b) D'autres chercheurs ont utilisé LSI dans le domaine de recherche d'images. Traditionnellement les images sont stockées dans des bases de données de grande taille. Il a été suggéré l'utilisation de la LSI pour extraire les images nécessaires, plutôt que de chercher dans des grandes bases de données manuellement. Les techniques de recherche d'images en général accèdent aux images sur la base de leur contenu, communément connu sous le nom de récupération d'images basée sur le contenu. Pour les travaux récents, un mot-image matricielle est créée, la matrice de poids est une représentation de grande dimension de la base de l'image dans laquelle chaque image est représentée comme un vecteur. Puis LSI est utilisé pour découvrir la relation sémantique entre les mots-clés visuels et des images, afin d'améliorer le processus de récupération.

c) Le concept de la LSI a également été utilisé dans les systèmes de récupération audio et vidéo. Dans les applications audio, le flux audio est converti en un flux de texte par un dispositif de reconnaissance de la parole. Ensuite, le texte de chaque partie du discours est représenté dans un vecteur document qui est la somme du mot dans la partie de la parole. Le travail a été axé sur la structure

pertinente sémantique de la base de données qui peut être obtenu par LSI, afin de réduire l'effet du bruit généré par la reconnaissance vocale.

d) Dans les applications vidéo les séquences sont décomposées en contenu visuel (représentant les séquences vidéo) et des mots (décrivant le contenu visuel) pour former une matrice. Puis la LSI est utilisée pour déterminer les relations entre les mots et les contenus visuels en fonction des cooccurrences de mots dans le contenu au sein de la matrice. La LSI peut aussi modéliser le contenu vidéo et réduire le bruit et renforcer la cooccurrence de l'information.

e) Dans la recherche inter-langue, en saisissant une requête dans une langue, la LSI peut être utilisée pour retourner les documents dans une autre. Ce qui est requis pour les applications inter-langues, un espace commun dans lequel les mots de plusieurs langues sont représentés. Dans quelques recherches, une méthode de recherche documentaire automatisée totalement inter-langues, dans laquelle aucune traduction de la requête n'est requise, est décrite. Les requêtes dans une langue peuvent récupérer des documents dans d'autres langues. Ceci est accompli par une méthode qui construit automatiquement un espace multi-langue sémantique en utilisant la LSI. L'analyse sémantique latente inter-langue a été utilisée pour développer une représentation de faible dimension constituée de mots et de documents dans plusieurs langues.

3.5 Inconvénients des travaux existants :

Comme on peut le voir dans les sections précédentes, il reste encore beaucoup de possibilités pour davantage de recherche dans l'amélioration de la performance du système de LSI. Les principales limites peuvent être identifiées comme suit:

- Le volume de recherches sur les étapes de prétraitement effectuées sur les bases de données, devient faible en comparaison à la recherche sur les autres phases, la plupart des travaux existants sur l'étape de prétraitement ont été largement accepté par la plupart des recherches. En outre, des outils suggérés pour l'amélioration de la recherche ont été proposés. Seulement, sur certaine bases de données, ils réalisent de petites améliorations sur les résultats de la recherche, alors que dans d'autres bases de données ils rendront la recherche pauvre. De

plus, la technique de pertinence entraîne un coût de calcul élevé avec les grandes bases de données. En outre, certains chercheurs sont enclins à tester leurs méthodes sans utiliser d'outils.

- L'amélioration des résultats pour nombreuses approches du LSI, en collaboration avec d'autres techniques, a été négligeable, et il y'a de nombreux inconvénients et faiblesses qui peuvent être identifiés. En *Telcordia LSI Engine* (moteur de recherche de *Telcordia*), le travail n'a pas abordé la précision et le rappel de LSI comme des mesures standard pour l'efficacité de LSI, et utilise seulement le temps de réponse des requêtes. Un tel système métrique n'est pas suffisant pour les questions de mesure du rendement.
- Certains travaux peuvent être considérés comme simplement un test empirique pour LSI, offrant une bonne perception des phases du système, ainsi que d'essayer de répondre à de nombreuses questions sans réponses pour LSI telles que la détermination de la meilleure valeur de k. Le résultat, comme l'indiquent les chercheurs, n'était pas satisfaisant. Tous les algorithmes de décomposition alternative à SVD qui ont été suggérées ont échoué, et la norme SVD basée LSI reste le moyen le plus efficace de chercher en termes de nombre de documents retournés.

En termes d'applications de LSI, peu de lacunes peuvent être soulevées, car le succès de la technique de LSI dépend du milieu et des objectifs qu'on veut atteindre.

Toutefois, dans les applications d'apprentissage, il est clair que LSI souffre du manque d'importantes capacités cognitives que les humains possèdent et utilisent pour appliquer les connaissances expérimentées.

Chapitre 3. Traitement d'image dans la recherche intelligente d'information

3.1 Introduction :

Le but de la recherche présentée dans ce chapitre est de développer une nouvelle approche au processus de l'indexation sémantique latente (LSI) dans la recherche des documents basée sur les techniques de traitement d'image. Hoenkamp dans ses recherche, a proposée l'utilisation de la transformée en ondelette de Haar (HWT) à la place de l'algorithme de décomposition en valeurs singulière (SVD). Cependant, les résultats n'ont pas été au rendez-vous, et la HWT a échouée comme une technique alternative de l'algorithme SVD.

Nous présentons brièvement les méthodes les plus utilisées :

- **HWT** : Transformée en Ondelette de Haar.
- **DCT** : Transformée en Cosinus Discrète.

Les méthodes de traitement d'images ont montré leur efficacité surtout quand elles sont appliquées comme une étape de prétraitement.

A travers le travail proposé , nous allons focaliser sur des techniques hybrides telles que :

- HWT / SVD , Soft / HWT / SVD , Hard / HWT / SVD
- DCT / SVD , Soft / DCT / SVD , Hard / DCT / SVD

3.2 La Transformée des ondelettes

a) Description de la méthode

La transformée en ondelette est un outil mathématique pour les fonctions de décomposition. Elle peut être considérée aussi comme une approche multirésolution ou, multi-échelle pour l'analyse des signaux **[60]**. Elle permet de décrire une fonction en termes de forme générale et une variation de détails. Les ondelettes travaillent sur des

différents types de signaux, image, courbe ou surface, offrant une excellente technique pour représenter les niveaux de détails présents, fournissant ainsi une analyse multirésolution pour le signal. Il y'a de nombreuses introductions aux ondelettes, par exemple **[32]** **[55]**, qui fournissent une bonne compréhension des principes et les fondements mathématiques nécessaires à l'étude et leur utilisation, et un large aperçu des différents types d'ondelettes est présenté ainsi. Bien que la décomposition en ondelettes multirésolution et l'analyse ont leurs racines dans la théorie d'approximation et de traitement du signal, ils ont récemment été appliquée à de nombreux problèmes dans les applications graphiques informatiques, y compris la compression d'image, de reconnaissance de visage, l'image médicale débruitage et l'image d'interrogation.

Dans les applications d'image, il est difficile d'analyser l'information contenue dans image directement à partir de l'intensité de niveau de gris de ses pixels. A cet effet, l'image peut être divisée en un ensemble de détails qui apparaissent à des résolutions différentes. La représentation multirésolution est très efficace pour analyser l'information contenue dans une image **[60]**.

La base de Haar est la forme la plus simple des ondelettes, elle est une série similaire au développement de Fourier qui est souvent utilisé dans le traitement d'image **[54]**. Dans ce chapitre, nous allons focaliser sur la HWT (Transformée en ondelette de Haar).

L'Ondelette de Haar est l'Ondelette dont le support est le plus petit, cela implique que sa transformée du signal nécessitera le minimum d'espace de stockage.
Soit h la fonction, dite de base de Haar, définie sur par :

$$h(x) = \begin{cases} 1 & si\ 0 < x < \frac{1}{2} \\ 0 & sinon \end{cases}$$

Supposons que nous avons un signal définit sur l'intervalle $[0,1]$. Pour avoir une approximation discrète du signal, nous allons calculer ses valeurs dans deux points, quatre points, huit points et ainsi de suite ; le diviser un deux fonctions, de 0 à $\frac{1}{2}$ et de $\frac{1}{2}$ à 1, puis en quatre fonctions, de 0 à $\frac{1}{4}$, de $\frac{1}{4}$ à $\frac{1}{2}$, de $\frac{1}{2}$ à $\frac{3}{4}$ et de $\frac{3}{4}$ à 1 etc.

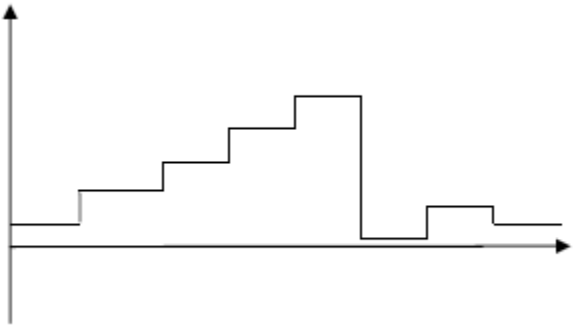
On obtient différentes résolutions et pour chacune, on peut avoir une représentation dans l'espace des fonctions à l'aide d'un système de fonctions de base, nommées fonctions de base multi-résolutions ou multi-échelles.
Les ondelettes dont des fonctions de base multi-échelles qui assurent le passage cohérent entre les différentes résolutions, la décomposition et la reconstitution de la fonction représentée. Si on utilise les ondelettes comme système de fonctions de base, à chaque niveau on dispose des approximations (moyennes) de la fonction initiale et des informations de détails.

Exemple :
Soit $f(x) = [y_1 y_2 y_3 \dots y_n]$ génère

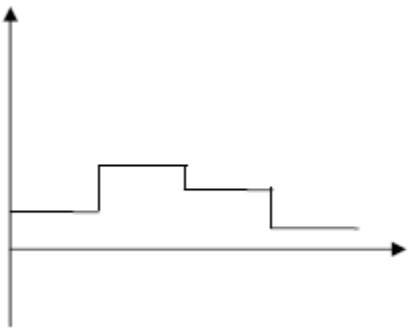
- Des approximations $[a_1 a_2 a_3 \dots a_{n/2}]$ qui sont les moyennes des valeurs initiales de la fonction prises deux par deux$a_1 = \frac{y_1+y_2}{2}$...
- Coefficients de détail ou les différences$\left[d_1 d_2 d_3 \dots d_{\frac{n}{2}}\right]$, avec$d_1 = y_1 - a_1, d_2 = y_3 - a_2$

Considérons un signal monodimensionnel avec une résolution de huit pixels

$$S = [2\ 4\ 8\ 12\ 14\ 0\ 2\ 1]$$



Pour calculer sa transformée de Haar, moyennons d'abord les paires de valeurs voisines pour obtenir [3 10 7 1.5]



Le signal peut donc être représenté par sa résolution inférieure et le signal de détail.
En appliquant ce procédé, récursivement sur le signal, on aboutit à sa transformée de Haar, à la fin, le signal est représenté par un seul coefficient de moyenne du signal et l'ensemble de coefficients des signaux de détails successifs.

Résolution	Moyenne	Détails
8	[2 4 8 12 14 0 2 1]	
4	[3 10 7 1.5]	[-1 -2 7 0.5]
2	[6.5 4.25]	[-3.5 2.75]
1	[5.375]	[1.125]

Tableau 3.1 : Transformée de Haar du signal S

On observant la transformée de Haar ainsi obtenue, en plus du coefficient de moyenne du signal, les coefficients de détails expriment les variations du signal aux différentes résolutions. A une même échelle, plus le coefficient est grand en valeur absolue, plus ces variations sont importantes, le signal original sera présenté par $[5.375\ 1.125 - 3.5\ 2.75 - 1 - 2\ 7\ 0.5]$.
Pour effectuer cette transformée, deux filtres ont été utilisés, le filtre d'échelle $[1\ 1]$ qui permet de calculer le signal de résolution et le filtre d'ondelette $[1 - 1]$ qui permet de calculer le signal de détail.
Appliquer le filtre d'ondelette au signal revient à calculer le produit du signal et de la fonction ondelette ψ.

$$\psi(x) = \begin{cases} 1 & si\ 0 \le x < \frac{1}{2} \\ -1 & si\ \frac{1}{2} \le x < 1 \\ 0 & ailleurs \end{cases}$$

Il y a plusieurs façons pour appliquer la HWT pour les structures à deux dimensions. La méthode utilisée dans ce travail est une méthode standard qui fonctionne comme suit : d'abord appliquer la décomposition sur toutes les lignes dans la structure, puis appliquer la décomposition sur toutes les colonnes de la matrice résultante.

b) Le seuillage

Un avantage de la transformée en ondelettes est que, souvent, un grand nombre des coefficients détaillés s'avèrent être de très petites amplitudes. La suppression de ces

petits coefficients de la représentation, introduit seulement des petites erreurs dans l'image reconstruite [**54**] [**58**].

Il y'avait un intérêt considérable dans l'utilisation de la transformé en ondelettes pour enlever le bruit de l'image (débruitage). Le but de débruitage est de supprimer le bruit tout en conservant la majeure partie des caractéristiques du signal important. En traitement d'image, le processus de transformation est aussi utilisé pour enlever le bruit d'une image [**33**] [**58**]. Une image est transformée en utilisant HWT, puis une fonction de seuillage est appliquée pour enlever le bruit de l'image. Typiquement, une image plus claire est ressortie après seuillage. Les fonctions de seuillage les plus courantes sont la fonction de seuillage « **soft** » et la fonction de seuillage « **hard** » [**4**], la fonction de seuillage « hard » choisit tous les coefficients d'ondelettes qui sont supérieurs au seuil donné λ et met les autres à zéro, tel que décrit dans l'équation suivante :

$$f_h(x) = \begin{cases} x & si\ |x| \geq \lambda \\ 0 & sinon \end{cases}$$

La fonction de seuillage « soft » est un peu différente, elle réduit les coefficients ondelettes par λ comme décrit dans l'équation suivante :

$$f_h(x) = \begin{cases} x - \lambda & si\ x \geq \lambda \\ 0 & si\ |x| < \lambda \\ x + \lambda si\ x \leq -\lambda \end{cases}$$

Dans certaines applications les seuillage « soft » donne une petite erreur d'estimation que le seuillage « hard ». Au contraire, dans d'autres applications, et pour une certaine classe de signaux, le seuillage « hard » donne une meilleure estimation.

c) La figure 3.1 illustre les différents composants du système LSI proposé. Elle décrit le système standard du LSI et montre la technique hybride proposée, et les étapes ajoutées pour l'analyse du bruit lexical.

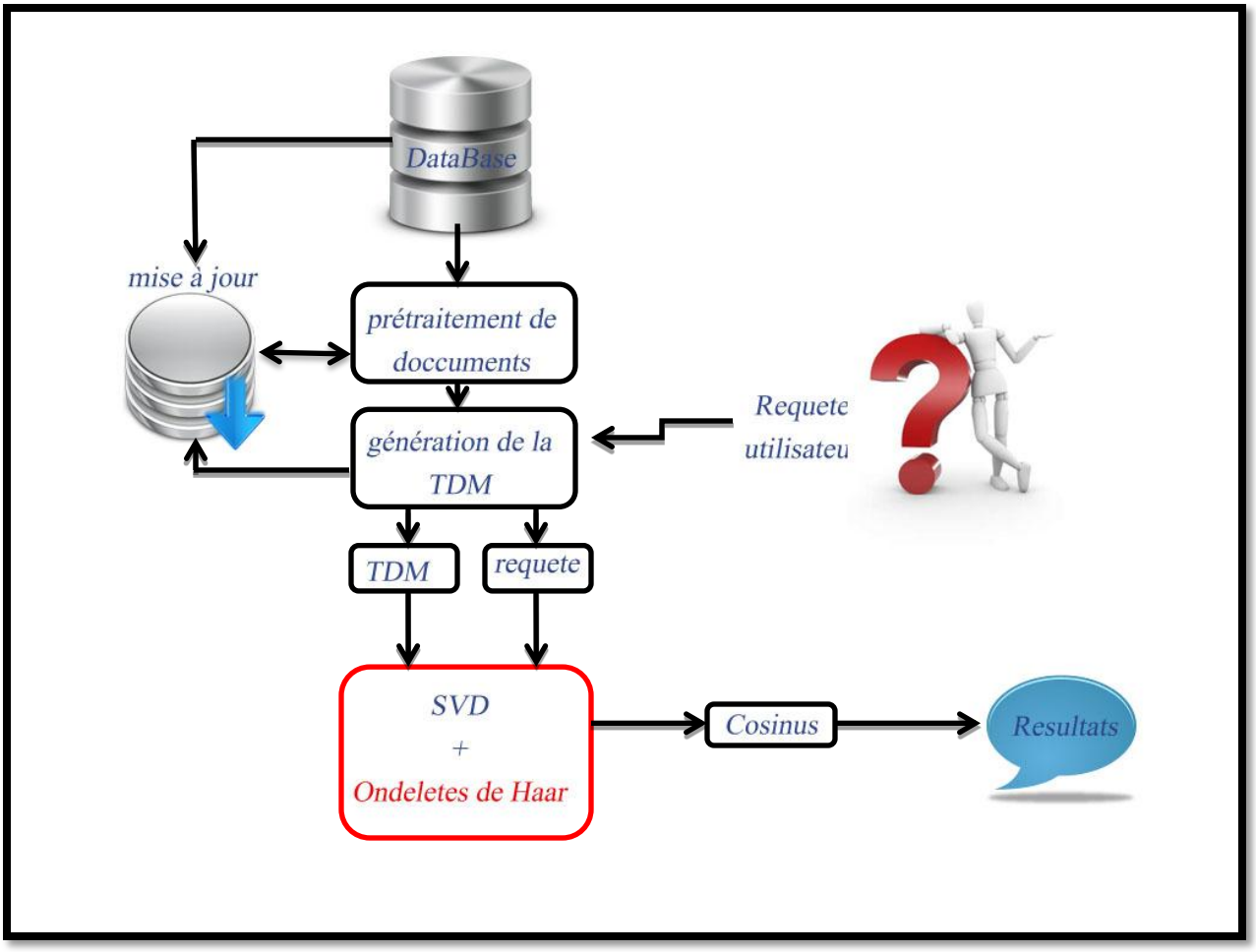


Figure 3.1 : Composants du système hybride HWT/SVD proposé

3.3 La Transformée en Cosinus Discrète (DCT) :

a) Introduction

Nous présentons l'approche par la technique de traitement d'image appelée la transformée en cosinus discrète (en anglais Discret Cosine Transform, DCT) à la phase de prétraitement avant l'application de l'algorithme SVD afin d'améliorer éventuellement les performances du système de recherche d'information basé sur l'indexation sémantique latente (LSI).

b) Système proposé

La Fig. 3.1 illustre les différents composants du système proposé basant sur la technique hybride DCT/LSI.

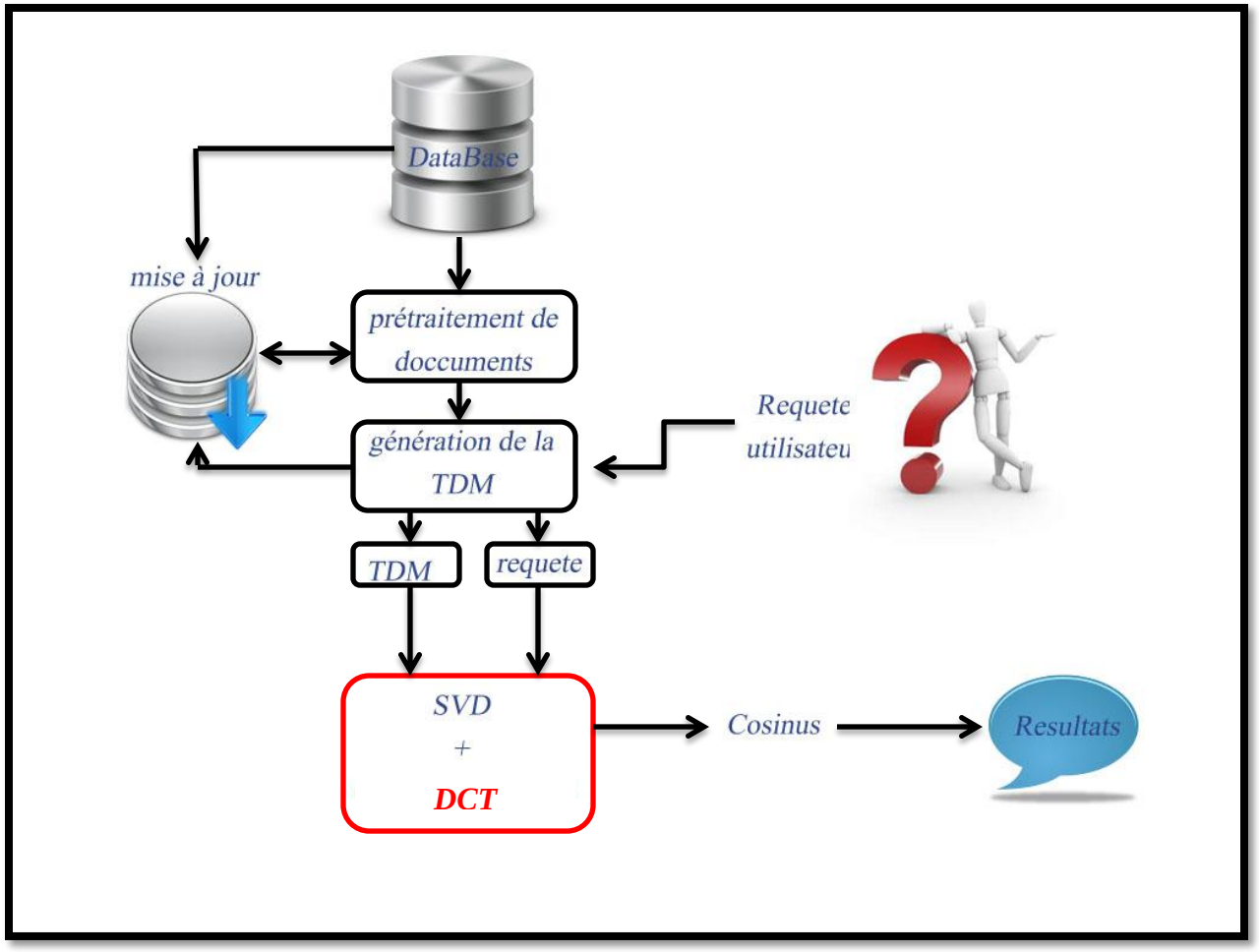


Figure 3.2 : Composants du système DCT/SVD proposé

c) Transformée en Cosinus Discrète

Le passage par la DCT a été l'idée majeure pour la compression JPEG. En effet ce processus appartient à une classe d'opérations mathématiques tout comme la transformée de Fourier Discrète (DFT). Elle permet un changement de domaine d'étude tout en gardant la même fonction étudiée, dans notre cas, une image, c'est-à-dire une fonction à trois dimensions, X et Y indiquant le pixel, et Z avec la valeur du pixel. Le noyau de projection est un cosinus et crée des coefficients réels, contrairement à la DFT,

dont le noyau est une exponentielle complexe et qui crée donc des coefficients complexes. On peut cependant exprimer la DCT en fonction de la DFT.

L'application de la DCT ou d'une Transformée de Fourier fait passer l'information de l'image du domaine spatial en une représentation identique dans le domaine fréquentiel. Comme pour la transformée en ondelettes, la DCT a la propriété que, pour une image typique, la plupart de l'information significative est concentrée dans quelques coefficients de la DCT. Pour cette raison, la DCT est utilisée dans les applications de la compression d'image.

La définition la plus courante d'un vecteur $f(x)$ de taille N est :

$$\boldsymbol{DCT(i) = C(i)\sum_{x=0}^{N-1} f(x)\cos\left[\frac{\pi(2x+1)i}{2N}\right], i = 0, 1, \ldots N-1} \qquad \textbf{(3.1)}$$

De la même manière, la transformation inverse est donnée comme suit :

$$\boldsymbol{f(x) = \sum_{i=0}^{N-1} C(i)DCT(i)\cos\left[\frac{\pi(2x+1)i}{2N}\right]} \qquad \textbf{(3.2)}$$

Pour les deux équations précédentes : $\boldsymbol{C(i) = \begin{cases} \sqrt{\frac{1}{N}} & \textbf{pour } i = 0 \\ \sqrt{\frac{2}{N}} & \textbf{pour } i \neq 0 \end{cases}}$

La transformée en deux dimensions de la DCT est équivalente à la transformée en une dimension, cette dernière est appliquée sur une dimension, soit les colonnes (verticalement), puis elle est appliquée horizontalement sur le résultat.

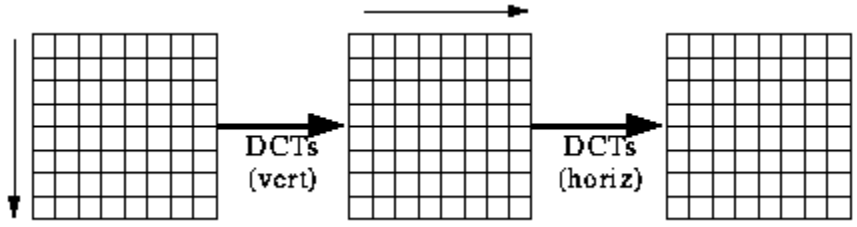


Figure 3.3 : Processus de l'application de la DCT 2D

La formule pour calculer la DCT sur une matrice$N * M$ est comme suit :

$$\boldsymbol{DCT(i,j) = \frac{1}{\sqrt{2}}C(i)C(j)\sum_{x=0}^{N-1}\sum_{y=0}^{M-1} pixel(x,y)\cos\left(\frac{(2x+1)i\pi}{2N}\right)\cos\left(\frac{(2y+1)j\pi}{2M}\right)} \qquad (3.3)$$

$$\boldsymbol{C(x) = \begin{cases} \frac{1}{\sqrt{2}} & si\ x = 0 \\ 1 & si\ x = 1 \end{cases}}$$

La transformation matricielle DCT s'accompagne d'une méthode d'inversion (IDCT) pour pouvoir revenir dans le domaine spatial. Ainsi après avoir fait des modifications dans le domaine fréquentiel, éliminer des variations de l'image quasiment invisibles pour l'œil humain en utilisant les fonctions de seuillages, on retourne à une représentation sous forme de pixels.

La formule pour calculer la IDCT sur une matrice $N * M$ est comme suit :

$$\boldsymbol{pixel(x,y) = \frac{1}{\sqrt{2N}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} C(i)C(j)\ DCT\ (i,j) \cos\left(\frac{(2x+1)i\pi}{2N}\right) \cos\left(\frac{(2y+1)j\pi}{2M}\right)} \quad (3.4)$$

$$\boldsymbol{C(x) = \begin{cases} \frac{1}{\sqrt{2}} & si\ x = 0 \\ 1 & si\ x = 1 \end{cases}}$$

Dans notre travail, comme mentionné précédemment, la TDM peut être considérée comme une image en niveau de gris. En appliquant la transformée et les fonctions de seuillages à la TDM, on peut éliminer le bruit de l'image, et comme mentionné auparavant, ce bruit représente le bruit lexical.

Chapitre 4. Evaluation des systèmes d'information

L'évaluation des SRI constitue une étape importante dans l'élaboration d'un modèle de RI. En effet, elle permet de caractériser le modèle et de fournir des éléments de comparaison entre modèles. Dans le domaine des systèmes de recherche d'information, qui sont destinés aujourd'hui à des utilisateurs non-spécialistes et qui permettent une grande richesse d'exploration facilitant la consultation directe des documents, une étude d'évaluation est nécessaire dans le sens où elle permet de contrôler et d'évaluer les opérations et la performance du système.

Cependant, la définition de la performance de manière absolue n'est pas quelque chose de possible dans la mesure où les caractéristiques des utilisateurs ne sont pas les mêmes (besoins, connaissance, objectifs, etc.). Pour ce faire, il faut fixer des repères qui permettront d'évaluer les performances relatives les unes par rapport aux autres et par rapport aux repères. En général, tout système de recherche d'information a deux objectifs principaux :

Le premier est de retrouver tous les documents pertinents pour une requête utilisateur et le deuxième est de rejeter les documents jugés non pertinents. L'évaluation dans systèmes peut être abordée selon deux angles : l'efficience qui est lié au rendement (rapidité et/ou quantité de ressources utilisées) et l'efficacité qui est lié à la qualité du résultat. Dans la suite on présente plus précisément ces angles.

Efficience : l'efficience mesure la qualité de recherche en termes de critères liés au déroulement pratique d'une session de recherche. Parmi ces critères, on cite notamment : le délai de réponse, le nombre d'entrée sortie sur disque, la taille de l'index [50].

Efficacité : l'efficacité mesure la pertinence de la recherche. Les performances liées à la l'efficacité des SRI sont mesurées en comparant les documents retrouvés par le système avec les documents que l'utilisateur souhaitent retrouver. A cet effet, on utilise une collection de test.

Une collection de test contient un corpus de documents, un ensemble de requêtes, et la liste des documents pertinents pour chaque requête. Cette liste de réponses idéales est

établie par des experts ayant une grande connaissance du corpus et du domaine des documents. L'efficacité de la recherche est alors évaluée en ''comparant'' pour chaque requête, les documents pertinents fournis par la collection de test à ceux présentés par le SRI.

4.1 Critères externes d'évaluation

Les critères principaux externes d'évaluation d'un SRI sont [**7**]:

- **Le taux de rappel** : proportion des documents pertinents restitués par le système relativement à l'ensemble des documents pertinents contenu dans la base. Le rappel mesure la capacité du système à retrouver tous les documents pertinents répondant à une requête.
- **Taux de précision** : proportion des documents pertinents restitués par le système relativement à l'ensemble des documents restitués. La précision mesure la capacité du système à rejeter tous les documents non pertinents à une requête.
- **le temps de réponse** : durée écoulée entre l'instant où l'utilisateur interroge le système et l'instant où ce dernier restitue la réponse. Le but est bien évidemment de réduire au maximum cette durée.
- **La présentation des résultats** : elle doit être conviviale, claire, simple, accessible à tous.
- **L'univers de discours de la collection** : le degré auquel le système inclut l'information pertinente.
- **La facilité d'utilisation**.

Les paramètres taux de rappel et taux de précision sont les plus importants en recherche d'informations proprement dite, cal ils sont directement représentatifs des performances du modèle de recherche et des différentes méthodes employées. Le taux de rappel et le taux de précision évaluent respectivement les notions de bruit et de silence documentaire qui constituent les deux premiers indicateurs de performance d'un SRI.

Les notions de silence et bruit présentent respectivement le taux de documents pertinents non sélectionnés et le taux de documents non pertinents sélectionnés.

Le taux de rappel et précision, silence et bruit sont donnés par les formulations suivantes :

$$\boldsymbol{Précision = \frac{R_r}{R}} \qquad\qquad \boldsymbol{Rappel = \frac{R_r}{P}}$$

Où :

$\boldsymbol{R}$: Le nombre de documents retrouvés par le système

$\boldsymbol{P}$: Le nombre de documents pertinents dans toute la collection

$\boldsymbol{R_r}$: Le nombre de documents pertinents et retrouvés par le système

$le\ silence = 1 - Rappel$ et $le\ \ Bruit = 1 - Précision.$

Dans le cas d'un système idéal, le taux de précision est égal au taux de rappel, c'est-à-dire que, tous les documents pertinents dans ce cas, et que ceux-ci, sont sélectionnés. On aurait donc une droite (Figure 4.1 – Cas idéal).

En pratique, la courbe de rappel/précision a l'allure générale de la figure 4.1 – Cas réel.

Pour ce faire on procède comme suit :

a) Pour $i = 1, 2, 3, \ldots$ de documents dans la base

Faire :

Evaluer la précision et le rappel pour les i premiers documents dans la liste des réponses du système.

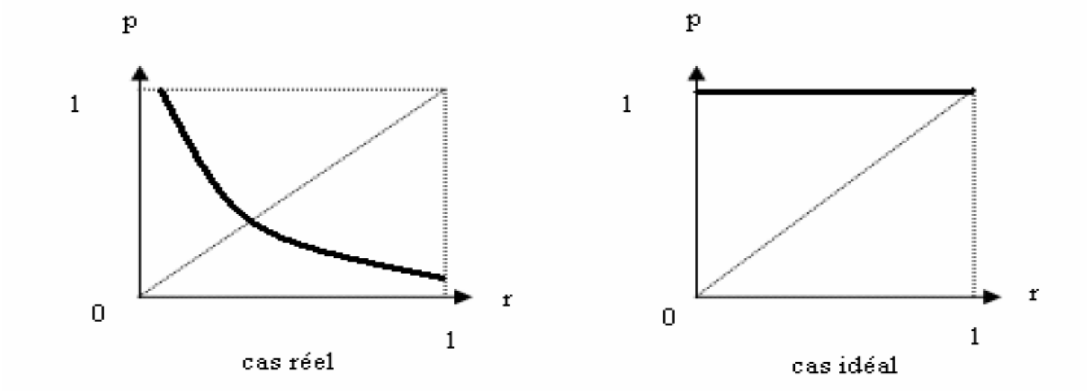


Figure 4.1 : Courbe de rappel/précision

b) On peut faire un constat simple : si le bruit et le silence sont toujours à peu près les mêmes, par exemple de 50%, un utilisateur qui reçoit dix documents en réponse à une question, on trouvera cinq pertinents. Un utilisateur qui obtiendra cent documents, on trouvera sans doute cinquante pertinents, mais aussi cinquante hors sujet. Le facteur bruit devient une gêne très réelle pour l'utilisateur dès que le volume des réponses dépasse un certain seuil tolérable.

c) Par exemple, considérons une requête pour laquelle 5 documents sont pertinents dans la base. Soit la liste des réponses du système $\{d_1, \dots, d_{15}\}$. Les documents pertinents sont marqués par la lettre P comme indiqué par la troisième colonne de la Figure 4.2.

Document	Score	Pertinent	Précision	Rappel
d1	9,92	P	1,00	0,20
d2	9,77		0,50	0,20
d3	9,76	P	0,67	0,40
d4	9,59	P	0,75	0,60
d5	8,72		0,60	0,60
d6	6,85	P	0,67	0,80
d7	6,51	P	0,57	0,80
d8	4,32		0,63	1,00
d9	4,16		0,56	1,00
d10	3,47		0,50	1,00
d11	2,69		0,45	1,00
d12	2,04		0,42	1,00
d13	1.84		0,38	1,00
d14	1,67		0,36	1,00
d15	0,07		0,33	1,00

Figure 4.2 : Exemple de valeur rappel/précision

On considère d'abord le premier document d_1 restitué par le système. A ce point, on a retrouvé un document pertinent parmi les 5 existants. Donc on a un taux de rappel de 0.2. La précision est $1/1$. Le point de la courbe est donc (0.2, 1.0).

On considère ensuite les deux premiers documents restitués. Le taux de rappel est toujours 0.2 et la précision est cette fois 0.5 (un document sur deux est pertinent). Le point est donc $0.2, 0.5)$. Ce processus est répété jusqu'à épuisement de la liste des réponses (qui peut être très longue en incluant tous les documents de la base). Les premiers points de la courbe sont alors représentés comme dans la Figure 4.3.

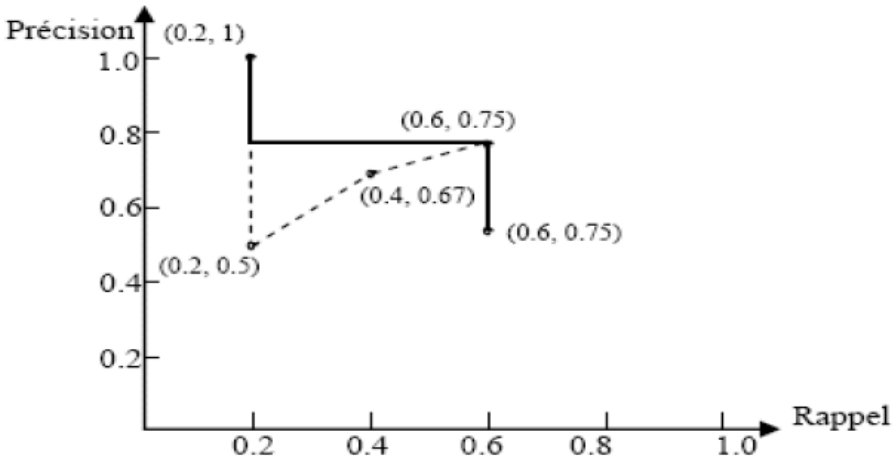


Figure 4.3 : Représentation des points de rappel/précision

Il arrive souvent qu'on applique l'interpolation sur la courbe de chaque requête. L'interpolation vis à créer une courbe descendante ayant l'allure de la forme de la Figure 4.1.

Algorithme

Soient i, j deux points de rappel avec $i < j$:

Si la précision au point i < précision au point j,

Alors on augmente la précision du point i à celle du point j.

Analyse

Concrètement, cela signifie qu'on remplit un creux de la courbe par une ligne horizontale, comme l'illustre la figure 4.4. On obtient alors une courbe en escalier. L'idée derrière l'interpolation est que les deux creux de la courbe ne représentent pas vraiment la performance du système. S'il existe un point à un rappel et une précision plus élevés, on peut toujours donner plus de documents dans la réponse pour augmenter la performance.

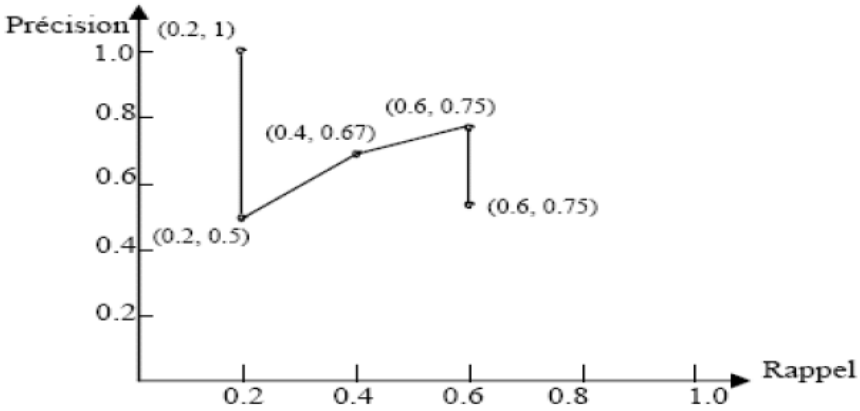


Figure 4.4 : Elimination de creux dans la courbe de rappel/précision

Un système dont la courbe dépasse (c'est-à-dire qu'elle se situe en haut à droite) celle d'un autre est considéré comme un meilleur système. Il arrive parfois que les deux courbes se croisent. Dans ce cas, il est difficile de dire quel système est le meilleur. Pour résoudre ce problème, on utilise souvent d'autres mesures d'évaluations de l'efficacité de la recherche selon le protocole TREC :

- **Précision interpolée** : on utilise pour chaque requête, la précision interpolée aux différents points de rappel $0{,}0.1, \dots, 0.9, 1$. Ceci nous permet de calculer la précision moyenne obtenue pour chaque point de rappel de chaque requête test et pour chaque collection test.
- **Précision à K documents** : on utilise particulièrement les précisions $P@5, P@10, \dots, P@100, P@1000$ représentant respectivement, la précision calculée après sélection des $5, 10, \dots, 100, 1000$ premiers documents.

$$\boldsymbol{P@K = \sum_{i=1}^{K} \frac{rel(s_i)}{k}}$$

$$\boldsymbol{rel(d) = \begin{cases} 1 \textit{ si le document d est pertinent pour le terme t} \\ 0 \textit{ sinon} \end{cases}}$$

- **Précision exacte** : c'est la précision au point où la précision vaut le rappel. Si la requête admet n documents pertinents, la précision exacte est la précision calculée à partir des n premiers documents de la liste ordonnée des documents restitués.

$$\boldsymbol{R = \sum_{d_i \in D} rel(d)}$$

$$\boldsymbol{rel(d) = \begin{cases} 1 \text{ si le document d est pertinent pour le terme t} \\ 0 \text{ sinon} \end{cases}}$$

- **Précision moyenne (MAP)** : ma moyenne des valeurs de précision à chaque document pertinent de la liste ordonnée.

$$\boldsymbol{rel(d) = \begin{cases} 1 \text{ si le document d est pertinent pour le terme t} \\ 0 \text{ sinon} \end{cases}}$$

$$\boldsymbol{MAP = \sum_{K=1}^{|S|} \frac{rel(S_k) * P@K}{R}}$$

($S = s_1, s_2, \dots, s_n$) est un sous-ensemble de documents retournés par le système.

En général, la précision moyenne décrit bien la performance d'un système. C'est la mesure, la plus utilisée en RI.

4.2 Collections de référence

Si on veut comparer deux systèmes de RI, il faut les tester avec le même corpus de test (ou plusieurs corpus de test). En pratique, des mesures telles que le temps de réponse ou la présentation des résultats ne sont pas répandues à grande échelle, à cause de la difficulté de leur mise en œuvre. Les mesures basées sur des courbes de rappel/précision demeurent largement les plus utilisées par les bancs d'essai (benchmarks) les plus connus.

De nombreux projets basés sur les corpus d'évaluation se multiplient depuis les années 70. On peut par exemple citer la collection CACM, la Collection ISI, la Collection CRANFIELD, ou encore la campagne CLEEF (Cross Language Evaluation Forum). Le projet le plus ambitieux est sans aucun doute le projet d'évaluation TREC (Text REtieval Conference) de la DARPA. La campagne d'évaluation TREC, co-organisée par le NIST et la DARPA, a commencé en 1992. Elle a pour but d'encourager la recherche documentaire basée sur de grandes collections de test, tout en fournissant l'infrastructure nécessaire pour l'évaluation des méthodologies de recherche et de filtrage d'information.

La collection-test est constituée lorsque l'on dispose, par une collection de documents donnée, d'un ensemble de questions-tests avec leurs ''réponses idéales'' associées. Les réponses idéales vont permettre d'évaluer la qualité des réponses fournies par des systèmes documentaires à évaluer. (Le temps moyen nécessaire à la reconstitution d'une collection-test est de 18 mois à deux ans, et nécessite un investissement important de la part des experts qui y collaborent). Pour évaluer un système de recherche d'informations, il suffira alors de lui soumettre les questions-tests, et de comparer les réponses qu'il fournit aux réponses types. En mesurant l'écart entre la réponse du système et la réponse-type, on obtiendra une mesure de qualité sur les performances du système documentaire.

Tous les éléments pris en compte peuvent être assimilés à des objets naturels préexistant à leur examen et non perturbés par l'observation : le système d'informations, les réponses idéales et les réponses fournies par le système. Elle permet de construire une mesure de qualité objective caractérisant la pertinence thématique des réponses fournies par le système. Toutes les mesures construites dans ce cadre sont reproductibles puisque aucun jugement individuel n'est émis par un utilisateur à posteriori.

Cependant, la mesure de qualité ainsi construite, est globale. Elle ne permettra pas de mesurer de manière distincte l'efficacité des différents constituants du système de recherche : la mesure construite prend en charge l'efficacité globale d'un processus, qui englobe à la fois, les choix sémantiques du langage adopté pour représenter le contenu des documents dans le système, le processus d'indexation qui associera certains termes du langage aux documents, et le processus d'appariement qui relie la question aux représentations des documents.

4.3 Statistique sur l'évaluation des systèmes de recherche d'information

Tague et al. **[36]** ont effectué une analyse statistique sur les résultats de TREC. L'analyse de la variance (ANOVA) a été employée pour calculer les différences significatives parmi des systèmes selon un certain nombre de mesures d'exécution. De très grandes différences d'exécution (représentant approximativement trois quarts des systèmes examinés) étaient nécessaire pour distinguer des systèmes avec la confiance de 95% ($p<0.05$). Le choix des mesures d'évaluation n'a pas eu l'impact substantiel sur les résultats.

Savoy [**35**] examine l'utilisation des méthodes classiques et bootstrap [**3**] d'exécution relative des paires de systèmes. Le bootstrap établit un modèle concret à une population hypothétique dans laquelle chaque élément de l'échantillon est replié infiniment. Cette population peut être simplifiée plusieurs fois en éliminant des éléments de l'échantillon original qui seront remplacés. Les essaies significatives ont été réalisés pour soutenir la proposition : le bootstrap rapporte une précision statistique plus élevée que des approches paramétriques, et que la médiane, par opposition au moyen, est une meilleure statistique.

Voorheers et Buckley [**20**] explorent l'effet de la longueur des requêtes, assumant également pour être un échantillon statistique. Ils mesurent la proportion des résultats discordants entre les évaluations effectués par des sous-ensembles des requêtes. Les résultats sont combinés par les différentes mesures d'évaluation entre chaque paire de systèmes. Pour chaque strate une courbe exponentielle sur deux paramètres est employée pour la proportion de paires discordantes.

Sanderson et Zobel mesurent la proportion de discordance calculée par les valeurs de précision à K documents d'un test de la différence significative entre les points de la précision moyenne. Ils observent une grande différence aux petites valeurs du K.

Plusieurs autres études ont évalué l'effet des variations des jugements de pertinence et des méthodes pooling sur l'évaluation de résultats. Ils ont montré que ma différence des jugements est non significative et n'influe pas sur le classement des systèmes qu'ils ont étudiés.

Buckley et Voorhees [**5**] considèrent l'effet d'utilisation des différents types de requêtes pour représenter le même besoin de l'information. En outre, les rapports sur les évaluations de RI incluent les tests standards tels que les *t-tests*, les tests *signed-ranked* par Wilcoxon, *sign tests ou analyse de la variance*. Les rapports incluent typiquement la signification jugements basés sur un seuil fixe deα (les précisions à K documents sont moins utilisées et les intervalles de confiance sont utilisés rarement). On suppose que la variation de topique est la seule source de produire d'erreur aléatoire.

Généralement, les analyses statistiques et en particulier celles basées sur un seuil fixeα, a relevé la critique récemment. Une telle hypothèse devrait être remplacée par une évaluation de l'importance de la différence et d'un argument de savoir si ou pas cette différence est importante.

Chapitre 5. Réalisation et tests

Dans ce chapitre, nous allons présenter les différents tests pour évaluer les systèmes hybrides proposés, et l'interface graphique réalisée.

5.1 Etude expérimentale et analyse des résultats

Cette section présente les résultats et les analyses des différents tests.

5.1.1 Méthodologie des métriques

Cette section explique la méthodologie utilisée pour générer les résultats. Chaque colonne de la TDM représente un document dans la collection originale sous une forme vectorielle et chaque ligne représente un terme, comme montré dans la table 5.1, de même pour la TDM approximée. La requête est un vecteur ligne tel que sa transposée est équivalent à un vecteur document contenant les mots qui apparaissent dans la requête.

	d1	d2	d3	d4
t1	0	0	1	1
t2	0	1	1	0
t3	1	0	0	0

Tableau 5.1 : Exemple de représentation des documents en 3 dimensions

Chaque vecteur document dans la TDM approximée peut donc être comparé à la requête en calculant le cosinus entre eux. Le cosinus est calculé à partir de l'équation suivante :

$$\boldsymbol{cos\theta = \frac{a_j^T q}{\left|\left|a_j^T\right|\right| \left|\left|q\right|\right|}}$$

Où a_j^T est la transposée de la $j^{\text{ème}}$ vecteur document dans la matrice approximée a, q est la vecteur requête, $\left|\left|a_j^T\right|\right|$ est le module de a_j^T, $||q||$ est le module de q.

Le module est équivalent à la norme euclidienne : $||q|| = \sqrt{(q_1^2 + q_2^2 + \cdots + q_{n-1}^2 + q_n^2)}$.

Une valeur du cosinus de 1 vaut dire que les deux vecteurs sont identiques dans le même espace dimensionnel. En dessous de cette valeur, les vecteurs deviennent de moins

en moins similaires. Pour déterminer quels sont les documents qui sont assez proches à la requête utilisateur, nous avons choisi un seuil de 0.75.

Dans ce travail, dans le but d'évaluer notre système par rapport au système LSI standard, nous avons évité l'utilisation de certaines techniques telles que le « stemming algorihm », contrôle de pertinence (relevance feedback) et la mise à jour.

5.1.2 Métriques utilisées

Comme mentionné dans le chapitre précédent, plusieurs techniques sont utilisées pour évaluer les performances d'un système de recherche d'information. Les techniques les plus utilisées et qui sont largement acceptées par la communauté de recherche sont *la précision et le rappel*. L'utilisation de ces deux métriques est nécessaire pour évaluer le système. Comme adressé par plusieurs chercheurs, la décomposition en valeurs singulières (SVD) est la meilleure technique en termes de nombre de documents renvoyés qui indique un niveau de rappel élevé.

Pour tester le système, nous avons utilisé un ensemble de requêtes dans lesquelles, les listes des documents pertinents sont connues.

5.1.3 Analyse du bruit lexical et les mesure en RI

Dans cette section, une nouvelle approche pour les analyses de la TDM utilisant les techniques de traitement d'image est présentée. De plus, une méthodologie pour la mesure du bruit lexical est introduite. Ces méthodes représentent les étapes initiales pour fournir une nouvelle approche pour l'implémentation de la technique LSI, dans lesquelles les performances de recherche vont s'améliorer.

5.1.3.1 Approche des analyses de TDM

Dans cette approche, les techniques de traitement d'image sont utilisées pour l'analyse de la TDM. La première étape est de représenter la TDM comme une image binaire pour visualiser le bruit. Dans cette représentation de la TDM, l'occurrence des 0, représente la présence du bruit. De la même manière, les données significatives sont représentées par des valeurs différentes de zéro. La Fig. 5.1 montre la représentation binaire de la TDM pour la base de données Memos et les figures 5.2 à 5.5 montrent les images générées en visualisant les TDMs pour les différentes bases de données.

2	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	1	1
1	0	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1
0	1	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	1
0	1	1	2	0	0	0	0	0
0	1	0	0	1	0	0	0	0
0	0	0	0	0	1	1	1	0
0	1	1	0	1	0	0	0	0

Tableau 5.2 : La TDM originale pour la base de données Memos

Si les images sont examinées, les points blancs représentent la donnée. Les points sont proches les uns aux autres formant un groupe (ou Cluster en Anglais), il est possible, en apercevant les lignes et les colonnes appropriées de dire qu'il y'a une relation entre les documents car ils contiennent les mêmes termes.

La représentation de la TDM comme une image permet d'analyser des grandes bases de données plus facilement. La distribution sur la TDM dépond de la structure, le contenu et la taille de la base de données.

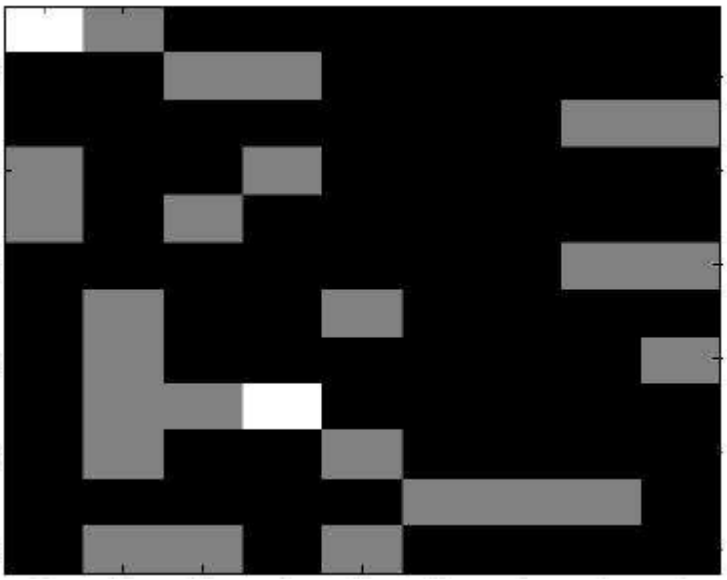


Figure 5.1 : TDM de la Memos représentée comme une image

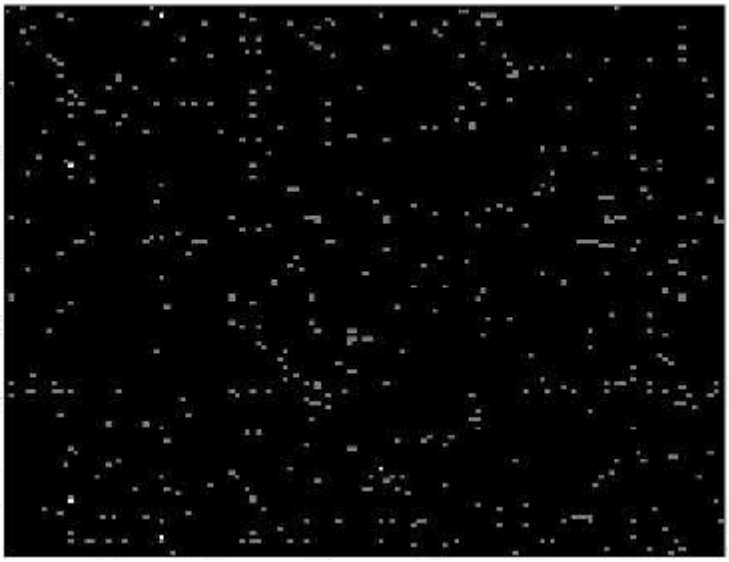


Figure 5.2 : TDM de la Cochrane représentée comme une image

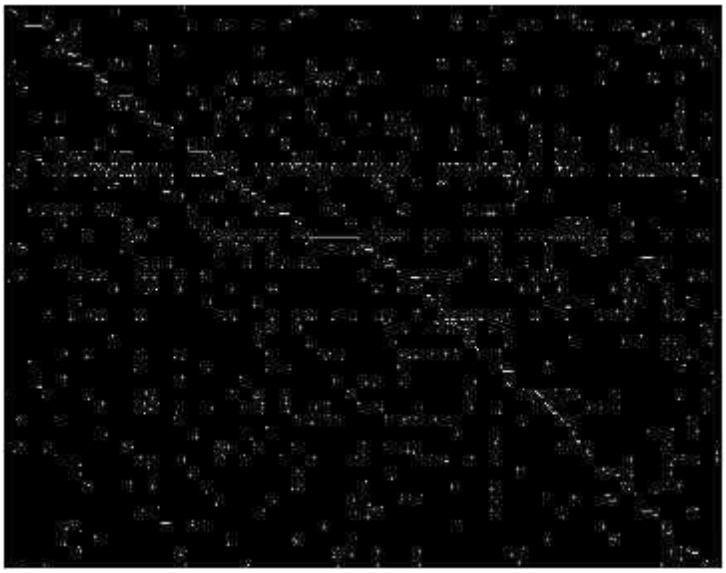


Figure 5.3 : TDM de l'eBooks représentée comme une image

5.1.3.2 Méthodologie proposée pour l'analyse du bruit lexical

Dans cette section, une méthodologie d'analyse du bruit lexical est présentée. D'abord la TDM est générée puis représentée comme une image en niveau de gris. La décomposition en valeurs singulières (SVD) est ensuite appliquée pour différentes valeurs de k, les k grandes valeurs propres sont gardées, et la matrice approximée est reconstruite. Les résultats de ce processus sont illustrés dans les figures 5.5 à 5.16.

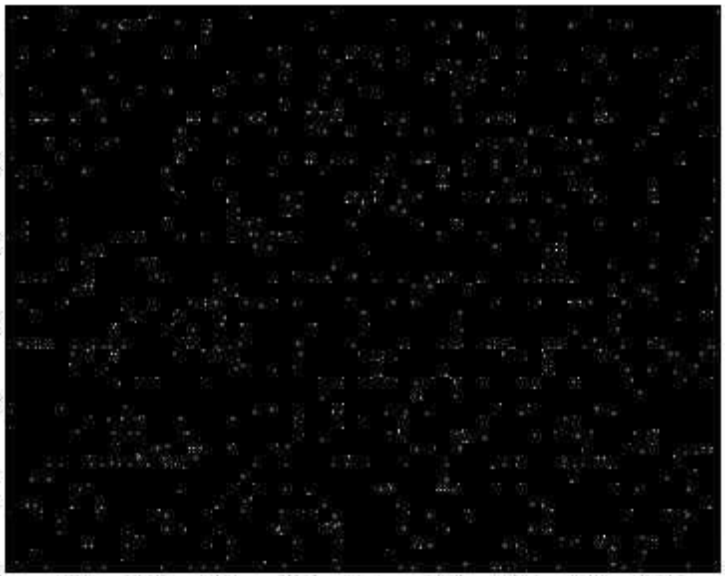


Figure 5.4 : TDM de la Reuters représentée comme une image

On peut remarquer que le bruit a été réduit après application de l'algorithme SVD (la couleur noir qui représente les valeurs nulles (0) a été réduit). Toutefois, le choix du k est une question très importante, et a un effet majeur sur la structure de la TDM qui peut être clairement remarqué sur les images.

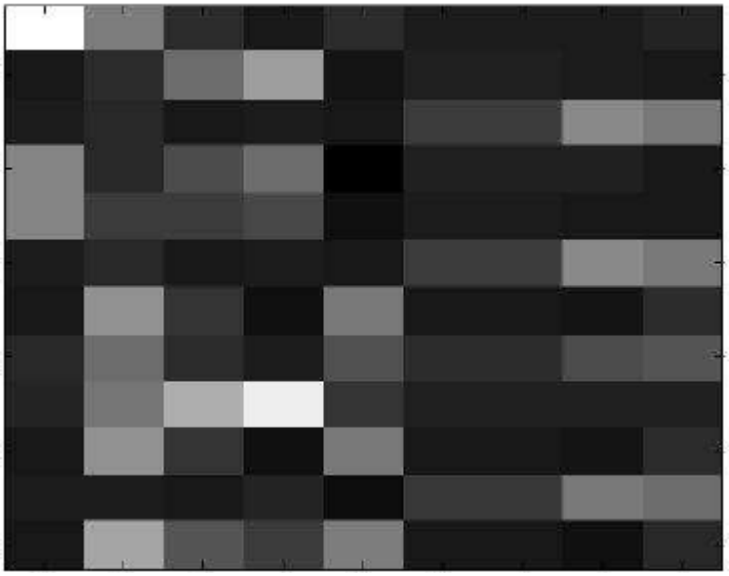


Figure 5.5 : Image de la TDM après SVD - $k = 4$ pour la Memos

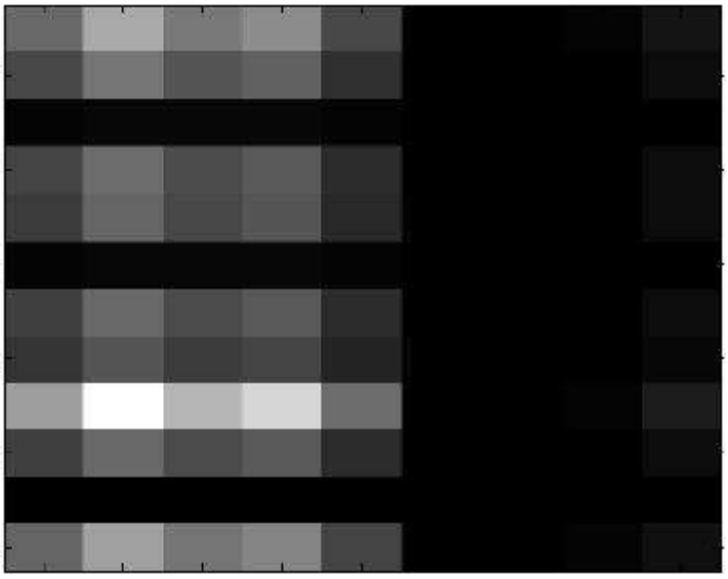


Figure 5.6 : Image de la TDM après SVD - $k = 1$ pour la Memos

La Fig.5.5 montre l'image de la TDM approximée pour $k = 4$ pour la base de données Memos. En examinant la TDM approximée, il est clair que le bruit a été réduit et que la distribution des valeurs différente de zéros a été améliorée. Dans la Fig. 5.6 l'image de la TDM approximée pour $k = 1$, apparait complétement détruite, i.e. il n'est pas possible de déterminer une information utile de l'image. Une très petite valeur de k a causé l'élimination de l'information utile, et ainsi, la destruction de la TDM approximée.

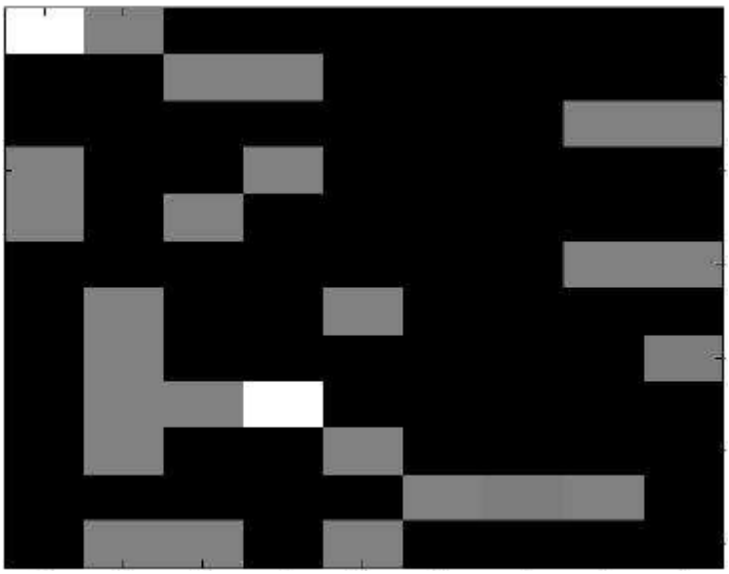


Figure 5.7 : Image de la TDM après SVD - $k = 8$ pour la Memos

Par contre, dans la Fig. 5.7, pour $k = 8$, aucun changement n'a été détecté sur la TDM. Huit dimensions ont été retenues, et donc une seule dimension a été supprimée de la matrice diagonale, qui a un effet mineur sur la TDM originale.

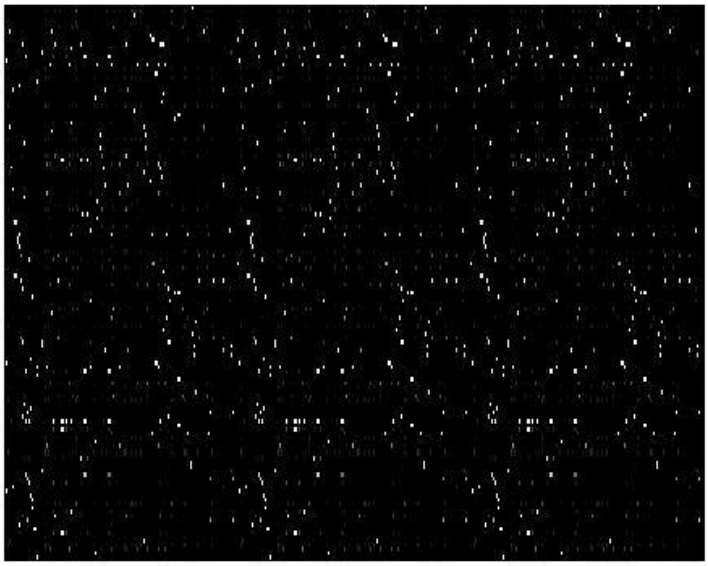


Figure 5.8 : Image de la TDM après SVD - $k = 50$ pour la Cochrane

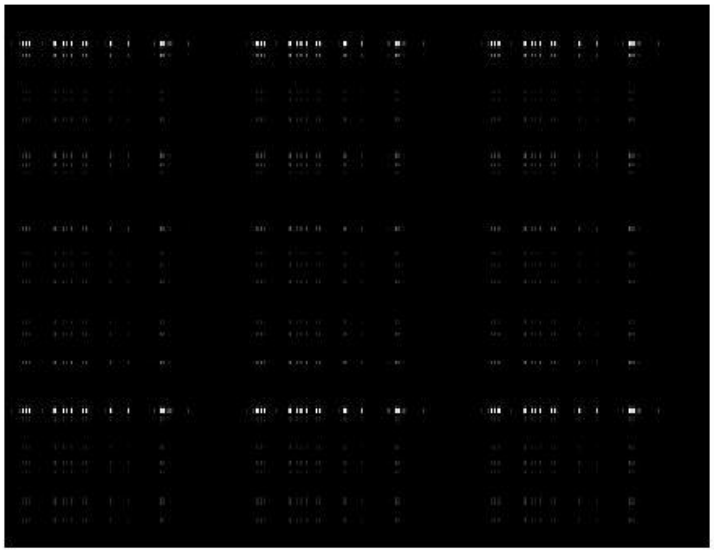


Figure 5.9 : Image de la TDM après SVD - $k = 1$ pour la Cochrane

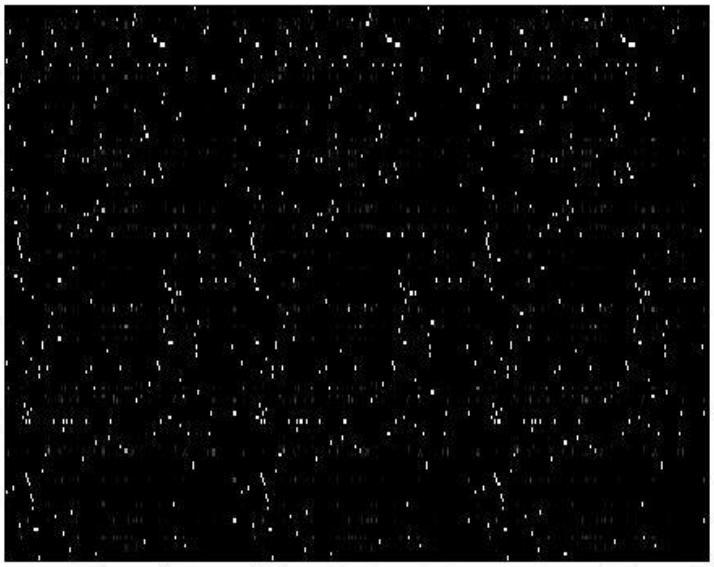


Figure 5.10 : Image de la TDM après SVD – $k = 100$ pour la Cochrane

La Fig. 5.8 montre encore une bonne structure de la TDM à $k = 50$ pour la base de données Cochrane. La répartition des valeurs non nulles a été améliorée par rapport à la TDM originale.

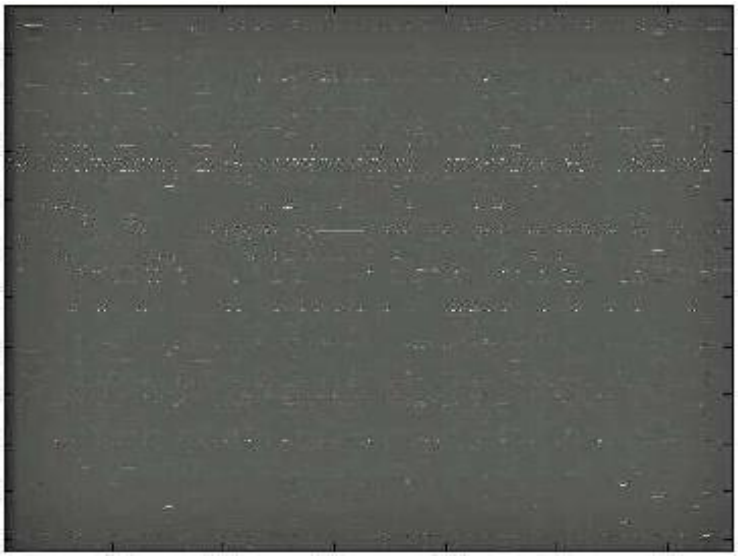


Figure 5.11 : Image de la TDM après SVD – $k = 30$ pour l'eBooks

Dans la Fig. 5.9, pour $k = 1$, l'information est complétement perdue.

Une structure similaire à la TDM originale est générée pour $k = 100$ comme montré dans la Fig. 5.10.

Encore, le même scenario est démontré dans les figures 4.11 à 4.16 pour les autres bases de données.

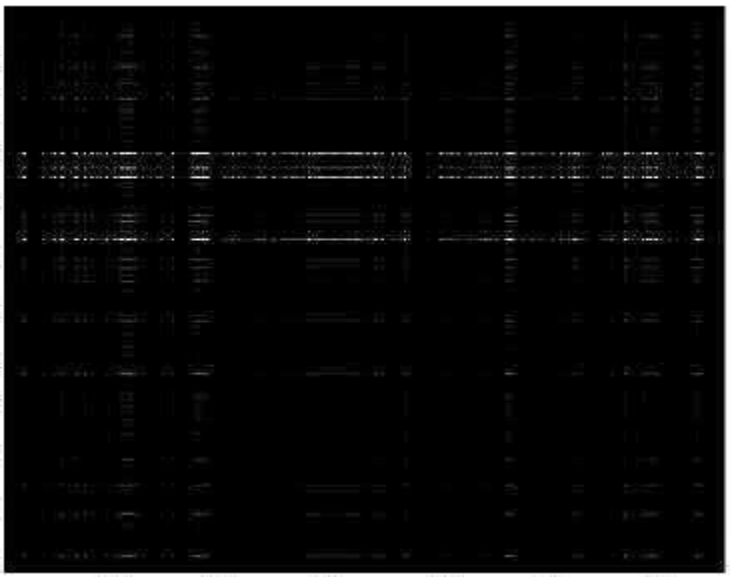


Figure 5.12 : Image de la TDM après SVD – $k = 1$ pour l'eBooks

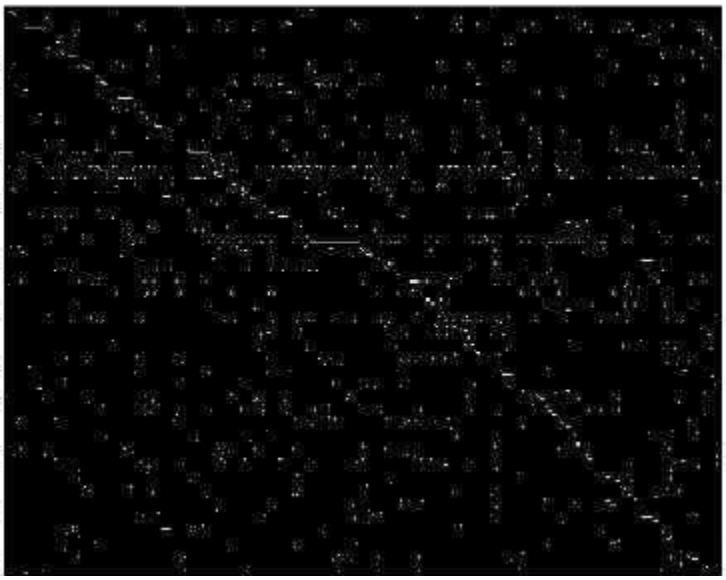


Figure 5.13 : Image de la TDM après SVD – $k = 560$ pour l'eBooks

Une dimension très petite supprime l'information importante de la TDM. De l'autre côté, une grande dimension ignore seulement peu de composants, et donc un effet négligeable sur la TDM.

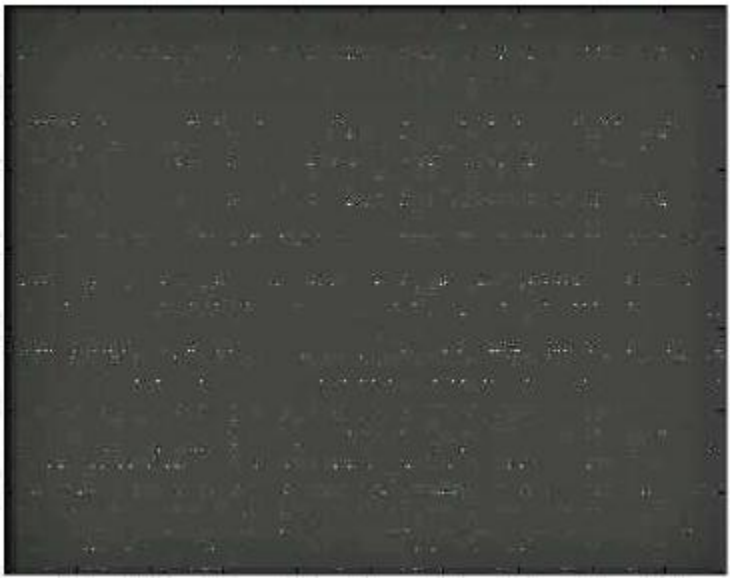


Figure 5.14 : Image de la TDM après SVD – $k = 30$ pour la Reuters

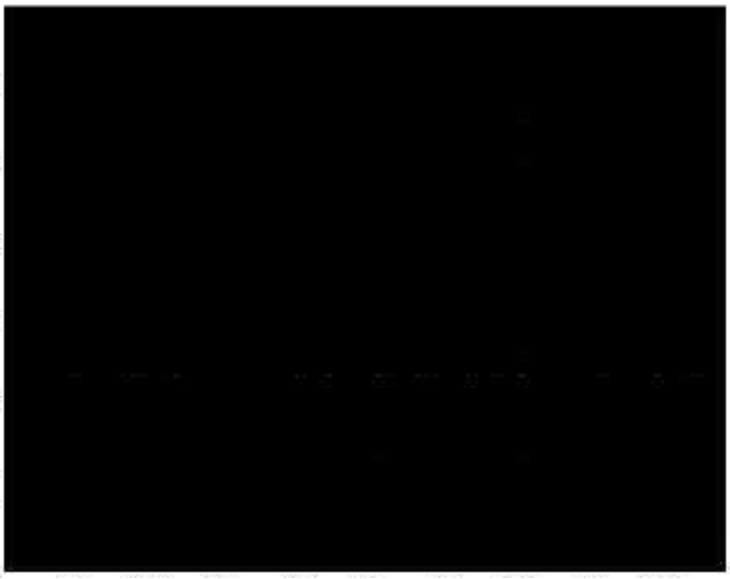


Figure 5.15 : Image de la TDM après SVD – $k = 1$ pour la Reuters

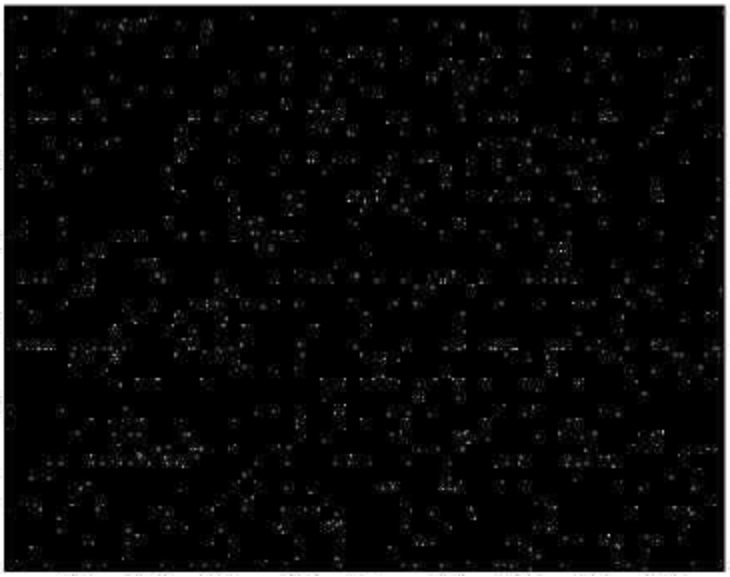


Figure 5.16 : Image de la TDM après SVD – $k = 700$ pour la Reuters

5.1.3.3 le Rapport signal sur bruit (SNR)

Parmi les critères de choix de la meilleure valeur de k, il y'a le rapport Signal sur Bruit (SNR).

Le SNR est défini comme étant le rapport de la puissance du signal utile et la puissance du bruit.

$$\boldsymbol{SNR = \frac{P_{Signal}}{P_{Bruit}}}$$

En traitement d'image, le SNR est généralement défini par le rapport de la valeur de la $\boldsymbol{moyenne}$ des pixels et la valeur de la $\boldsymbol{déviation\ standard}$ des pixels. Le SNR dans ce travail a été généré selon l'équation suivante :

$$\boldsymbol{SNR = 10.\log_{10}\left[\frac{\sum_0^{n_x-1}\sum_0^{n_y-1}[r(x,y)]^2)}{\sum_0^{n_x-1}\sum_0^{n_y-1}[r(x,y)-t(x,y)]^2}\right]} \quad (5.1)$$

Où : $\begin{cases} \boldsymbol{r(x,y)} : \ image\ \ référence \\ \boldsymbol{t(x,y)}: image\ \ test \\ Les\ \ 2\ \ images\ ont\ \ la\ \ meme\ \ taille\ \ n_x\ \ et\ n_y \end{cases}$

5.1.4 Approche empirique

Dans cette section, nous allons étudier les différents aspects de LSI. En particulier l'effet des filtres de débruitage. L'attention est accordée aussi à l'identification du k optimal (le rang de la matrice TDM approximée en appliquant l'algorithme SVD). Le but est de déterminer la meilleure structure pour une base de données qui mène aux meilleurs résultats de recherche pour la technique LSI.La Section 5.1.4.1 décrit les critères de sélection du meilleur k parmi les différentes valeurs possibles.

5.1.4.1 Critère de sélection des valeurs de k

Le travail présenté dans la section précédente peut être résumé comme suit. Une fois la TDM est générée, l'algorithme de décomposition est appliqué pour des différentes valeurs de k. Les matrices approximées sont reconstruites comme des images en niveau de gris. Le SNR est ensuite appliqué sur ces images pour mesurer le bruit après application de l'algorithme SVD pour différentes valeurs de k.

Pour choisir la meilleure valeur de k, on mesure la précision et le rappel pour les deux requête « rheumatoid arthritis » et « smoking and heart disease » de la base de données Cochrane pour différentes valeurs de k.

- Pour la requête **« rheumatoid arthritis »**

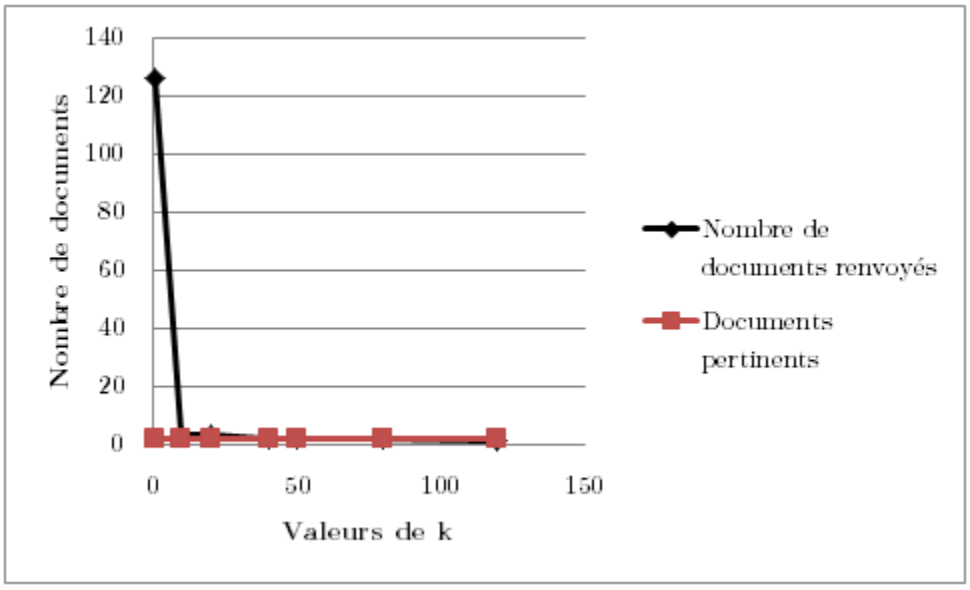


Figure 5.17 nombre de documents retrouvés pour « rheumatoid arthritis » - base de données Cochrane

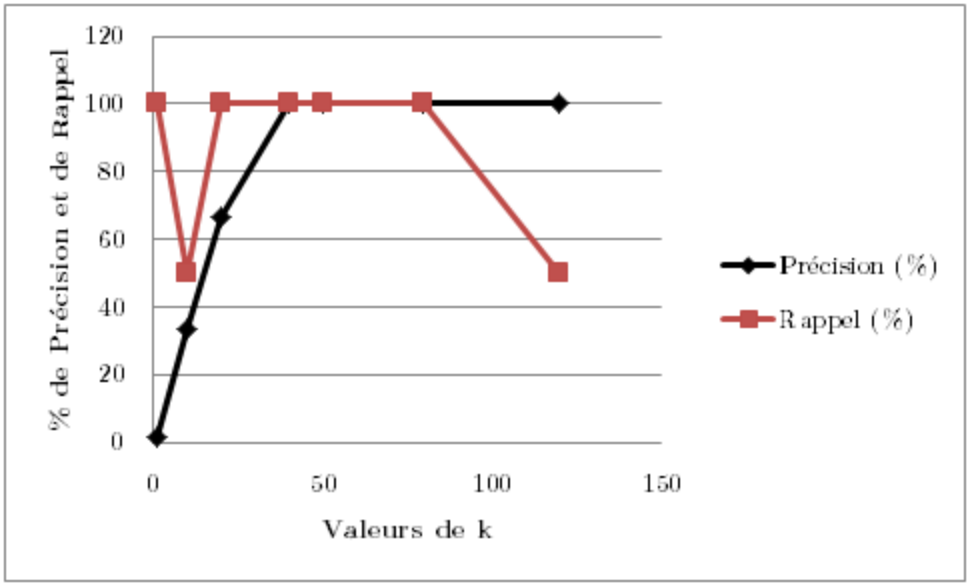


Figure 5.18 Précision et Rappel pour « rheumatoid arthritis » - base de données Cochrane

- **Smoking and heart disease**

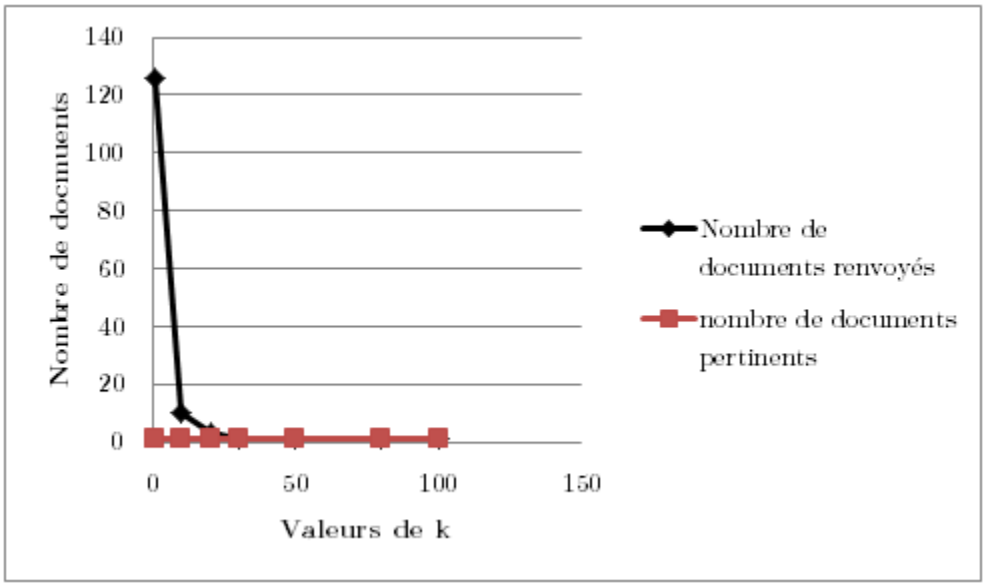


Figure 5.19 Nombre de documents retrouvés pour « smoking and heart disease » - base de données Cochrane

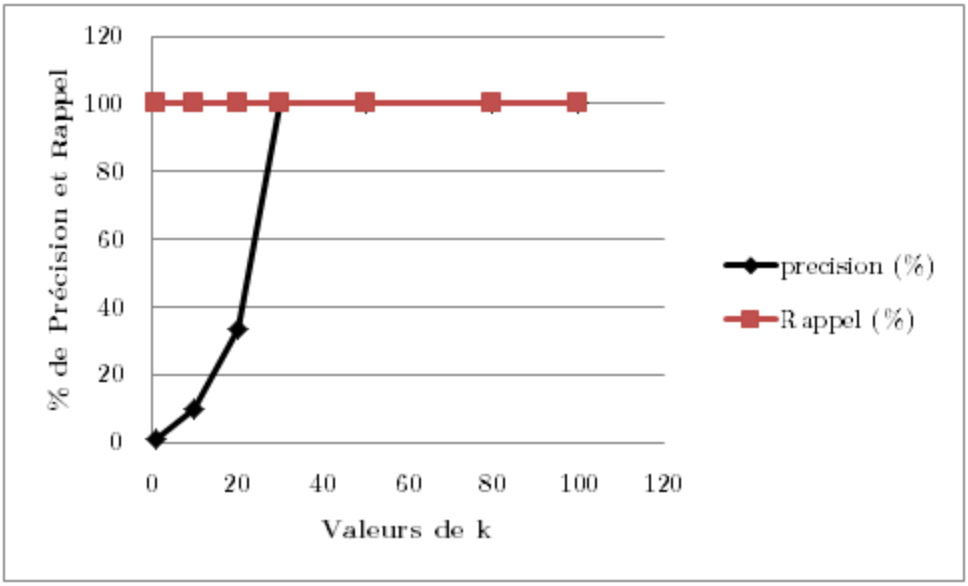


Figure 5.20 Précision et Rappel pour « smoking and heart disease »-Cochrane

Pour les deux requêtes, et pour des petites valeurs de k, le nombre de documents retrouvé est élevé qui donne donc une faible précision car le nombre de documents pertinents dans la base de données est deux pour la première requête et un pour la deuxième et ça est dû à la destruction de la TDM pour des très petites valeurs de k.

Pour k entre 50 et 80 pour la première requête on a une précision et un rappel de 100%, car le nombre de documents retrouvé est égale au nombre de documents pertinent dans la base de données pour les deux requêtes. On remarque une diminution du rappel quand k est supérieur à 80 pour la première requête.

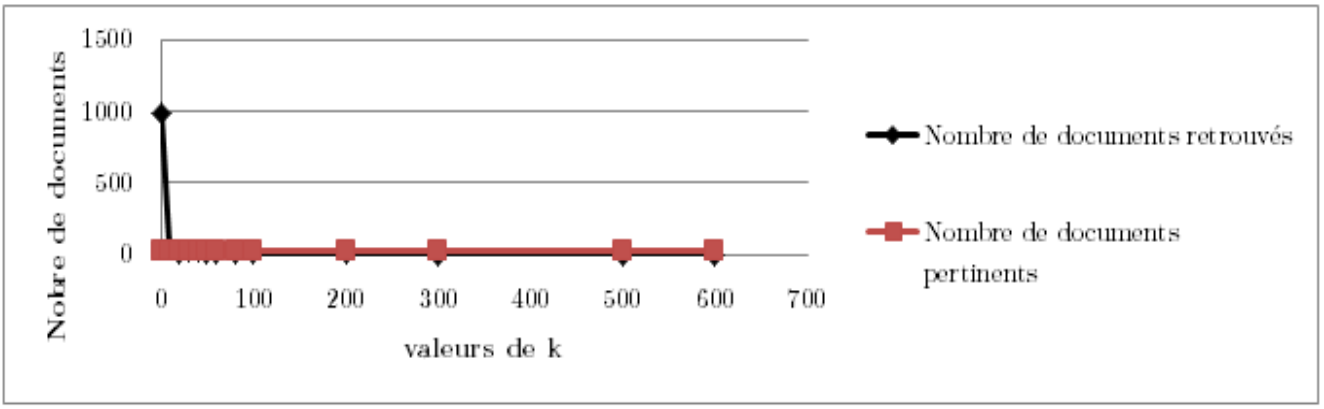


Figure 5.21 Nombre de documents retrouvés pour « japan » - base de données Reuters

Pour la deuxième requête, à partir de $k = 30$, on a une précision et un rappel de 100%, car le nombre de documents renvoyés est égale à un qui est pertinent.

Dans le cas de la base de données Reuters et pour la requête « japan » :

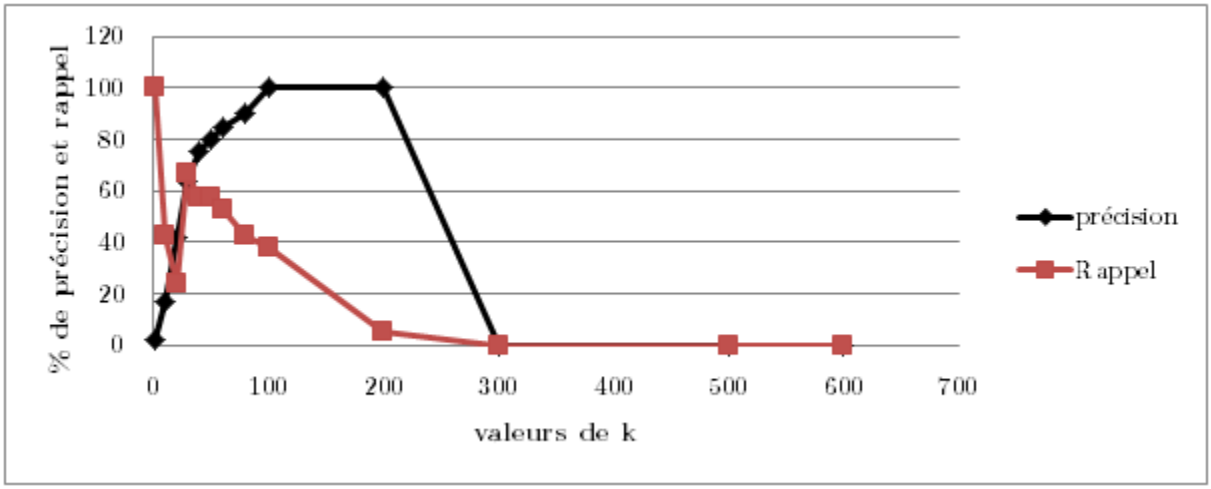


Figure 5.22 Précision et Rappel pour « japan » - base de données Reuters

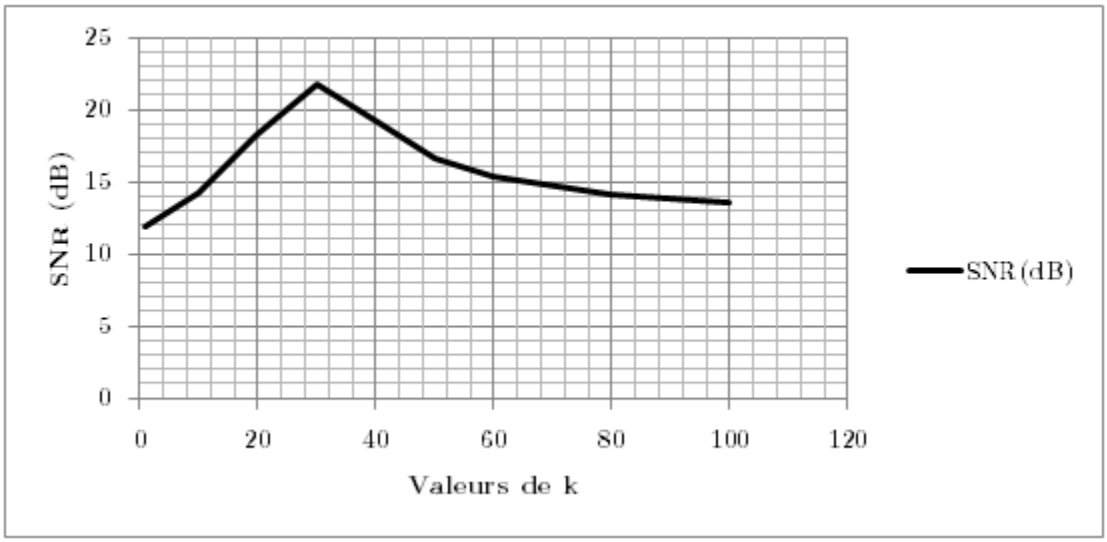


Figure 5.23 le rapport signal sur bruit (SNR) pour la base de données Cochrane

De la même manière pour la base de données Reuters, et pour des petites valeurs de k, le nombre de documents pertinents est élevé et une faible précision. Pour k entre 20 et 30, on a une précision et un rappel relativement élevés. Et pour k entre 100 et 200, on a

une précision élevée mais un rappel très faible car le nombre de documents non pertinents retrouvés est élevé.

Pour la base de données Cochrane et comme montré dans la Fig. 5.23, pour des valeurs de k entre 20 et 50, le SNR a des valeurs raisonnables, et il est claire que pour $k = 30$, on obtient la plus grande valeur du SNR, ce qui indique que le bruit dans la TDM est le mieux réduit, et la LSI pour cette valeur de k est prévue de donner les meilleures performances. Une faible valeur de SNR est obtenue pour les petites ou les plus grandes valeurs de k et il faut les éviter car elles suppriment l'information importante, ou elles ne font pas changer la TDM. Et en comparant avec la Table 5.1on remarque que le nombre de documents renvoyés pour des valeurs de k entre 30 et 50 est égale au nombre de documents pertinents dans la base de données où les meilleures valeurs du SNR sont obtenues, ce qui implique que le critère de sélection basé sur les valeurs du SNR montre une bonne prédiction pour la sélection de la meilleure valeur du k

	Recherche de « rheumatoid arthritis »
Valeurs de k	Nombre de documents renvoyés
1	126
10	3
50	2
100	2
	Recherche de « smoking and heart disease »
Valeurs de k	Nombre de documents renvoyés
1	126
10	10
30	1
100	1

Tableau 5.3 : Résultats des tests pour différentes valeurs de k

5.2 Analyse multirésolution

5.2.1 L'approche hybride Haar/SVD

Une approche généralement utilisée en traitement d'image est de combiner des différentes techniques afin d'améliorer la réduction du bruit. Dans cette section, on va combiner entre les deux techniques HWT et SVD pour examiner leur effet et la qualité des résultats en comparant avec l'approche basique LSI /SVD.

5.2.1.1 HWT – SVD LSI

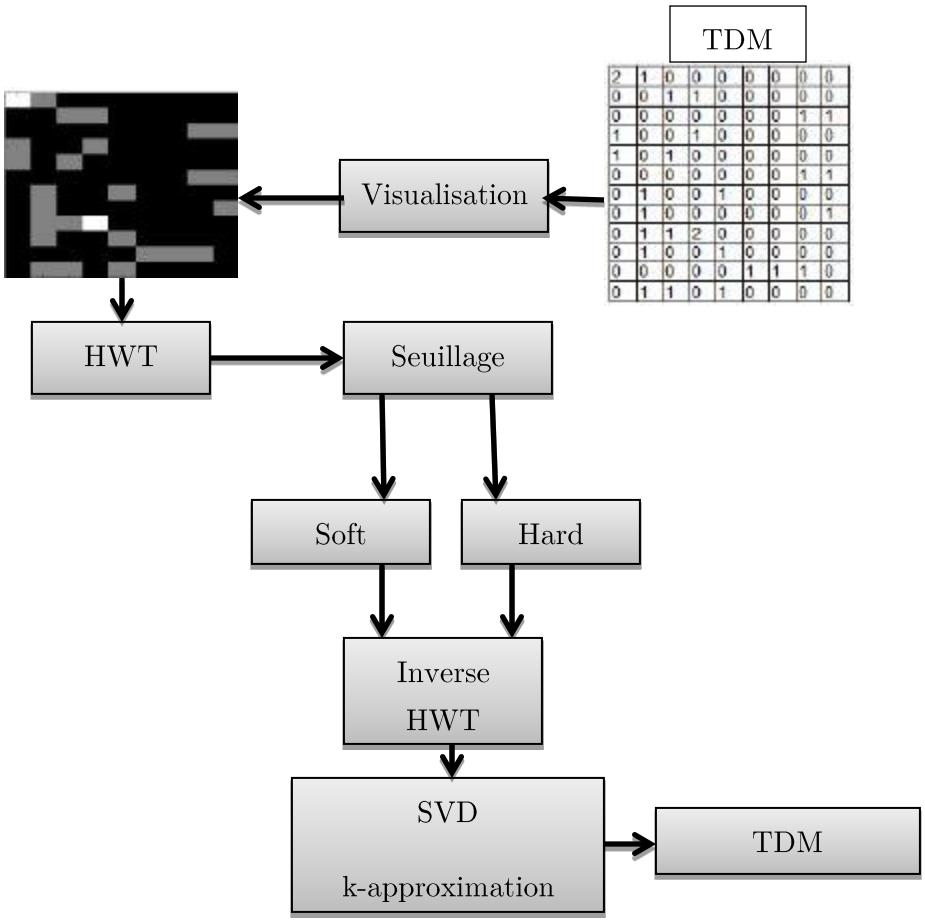


Figure 6.24 : processus de prétraitement

Base de données **Cochrane** : pour une valeur de $k = 30$.

Requête 1: « rheumatoid arthritis »

Requête 2: « smoking and heart disease »

Pour la première requête, comme montré dans la Fig. 5.25, le nombre de documents retrouvés pour l'approche standard est trois, qui donne une précision de 66.66%, par contre une précision de 100% pour l'approche hybride pour les deux fonctions de seuillage, ce qui montre la performance de l'approche hybride par rapport à l'approche standard. Pour la deuxième requête, on ne peut pas remarquer la différence car le nombre des documents retrouvés pour les deux approches est deux et ils ont pertinents, qui donne donc une précision et rappel de 100%.

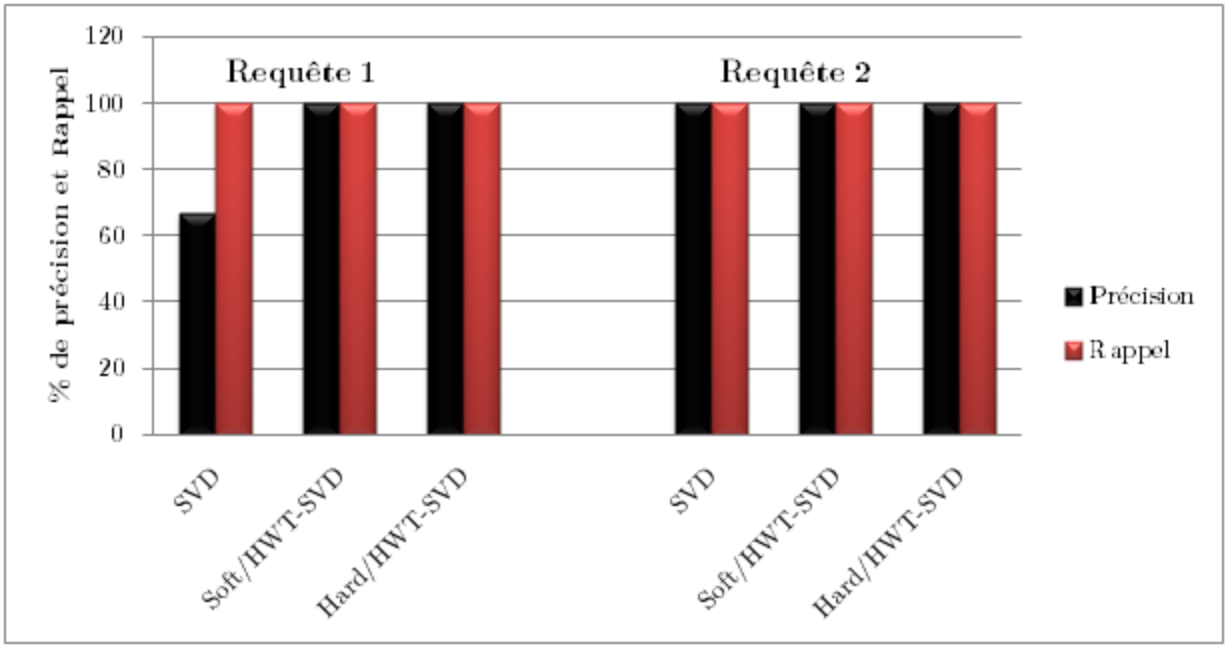


Figure 5.25 Résultats de recherche pour la base de données Cochrane

Pour la première requête, comme montré dans la Fig. 5.25, le nombre de documents retrouvés pour l'approche standard est trois, qui donne une précision de 66.66%, par contre une précision de 100% pour l'approche hybride pour les deux fonctions de seuillage, ce qui montre la performance de l'approche hybride par rapport à l'approche standard. Pour la deuxième requête, on ne peut pas remarquer la différence car le nombre des documents retrouvés pour les deux approches est deux et ils ont pertinents, qui donne donc une précision et rappel de 100%.

5.2.1.2 Analyses de Soft/HWT-SVD et Hard/HWT-SVD

D'après la section précédente, les mesures de précision et de rappel ne suffisent pas pour comparer entre les deux fonctions de seuillage « Soft » et « Hard », et donc pour se faire, on mesure le rapport signal sur bruit (SNR) des images générées après filtrage utilisant les deux fonctions et pour la même valeur de k, soit de 30 et pour le même seuil$= 0.04$.

Dans la Fig. 5.26 et pour un seuil de 0.04, on remarque que le rapport Signal sur Bruit est relativement plus élevé pour le seuillage « Soft » ce qui montre la performance de ce dernier par rapport au seuillage « Hard »

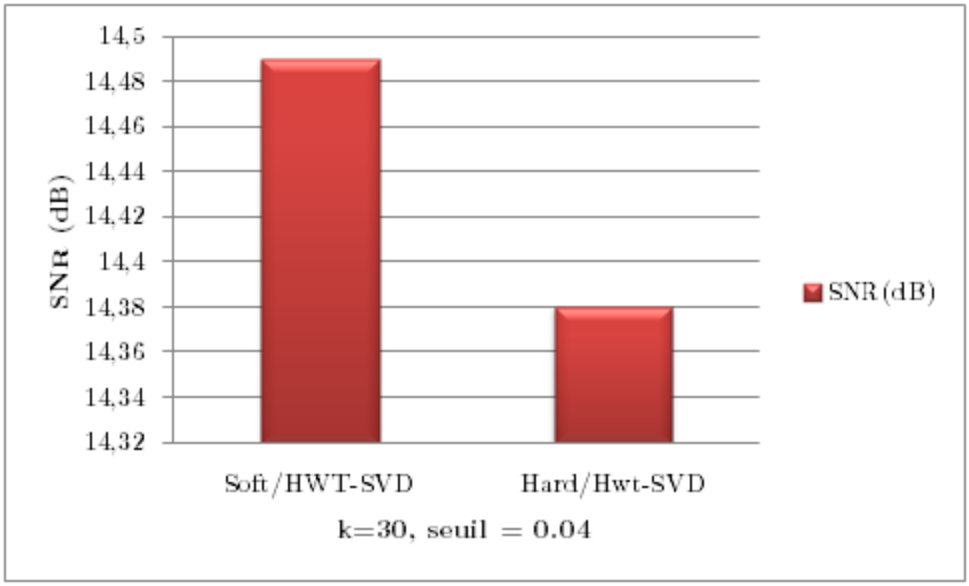


Figure 5.26 Analyse du SNR pour les fonctions Soft et Hard - Cochrane

5.3 Analyse des performances de l'approche hybride DCT/SVD

Dans cette section, nous allons étudier l'effet du technique DCT (Discret Cosine Transform) combinée avec la SVD sur les performances du système LSI en comparant avec l'approche basique LSI/SVD et l'approche hybride HWT/SVD.

La Fig. 5.27 montre les étapes du processus de prétraitement pour l'approche hybride HWT/SVD.

Dans la Fig. 5.28 et pour un seuil de 0.04, les mesurent de SNR pour différentes valeurs de k montre des résultats raisonnables pour k entre 20 et 40. En dehors de cet intervalle le SNR prend des valeurs faibles car la TDM n'a pas changé (pour k grand) ou bien l'information importante est supprimée.

La Fig. 5.29 montre encore une légère amélioration du SNR dans le cas du seuillage « Soft » par rapport au seuillage « Hard » pour un seuil =0.04, dans le cas de l'application de la Transformée en Cosinus Discrète.

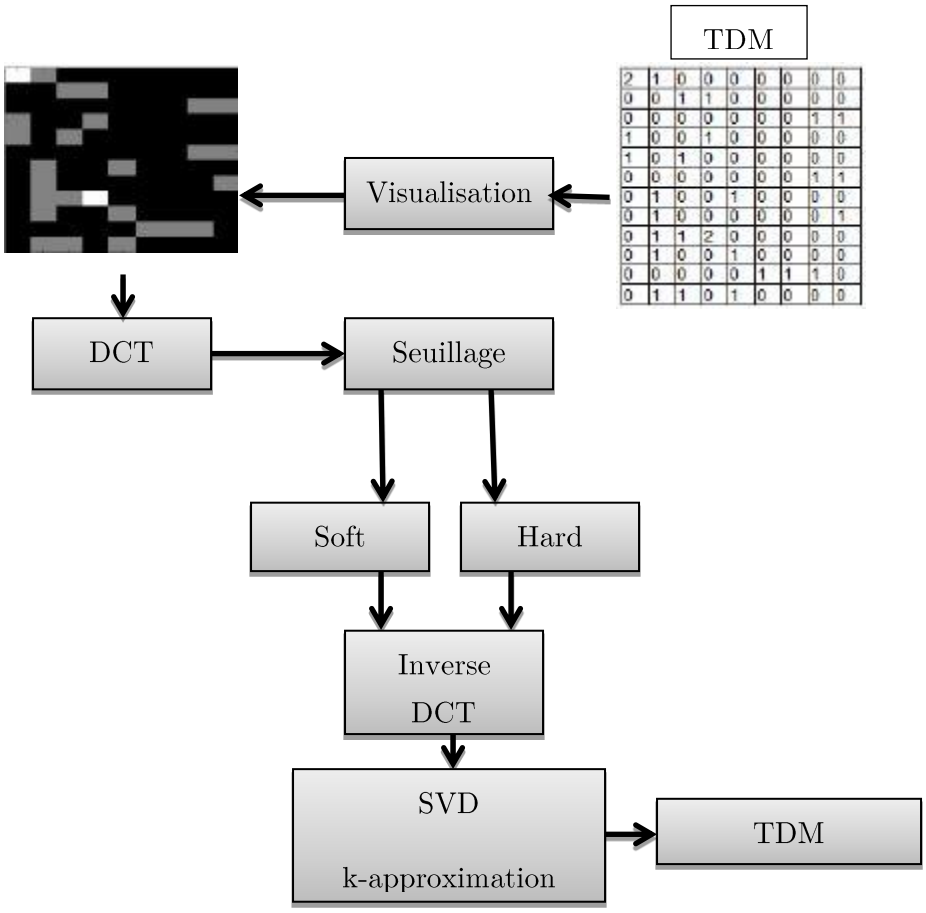


Figure 5.27 Processus de prétraitement dans le cas de l'utilisation de DCT

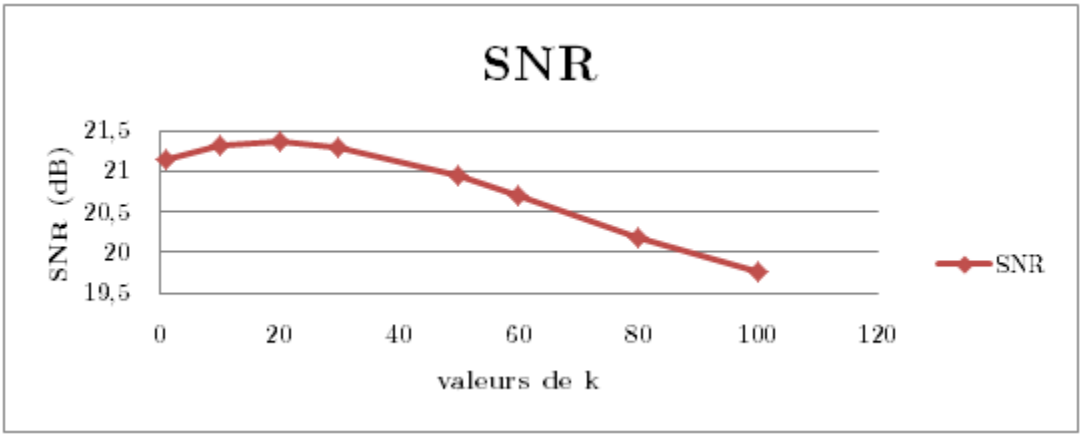


Figure 5.28 SNR pour différentes valeurs de *k*.

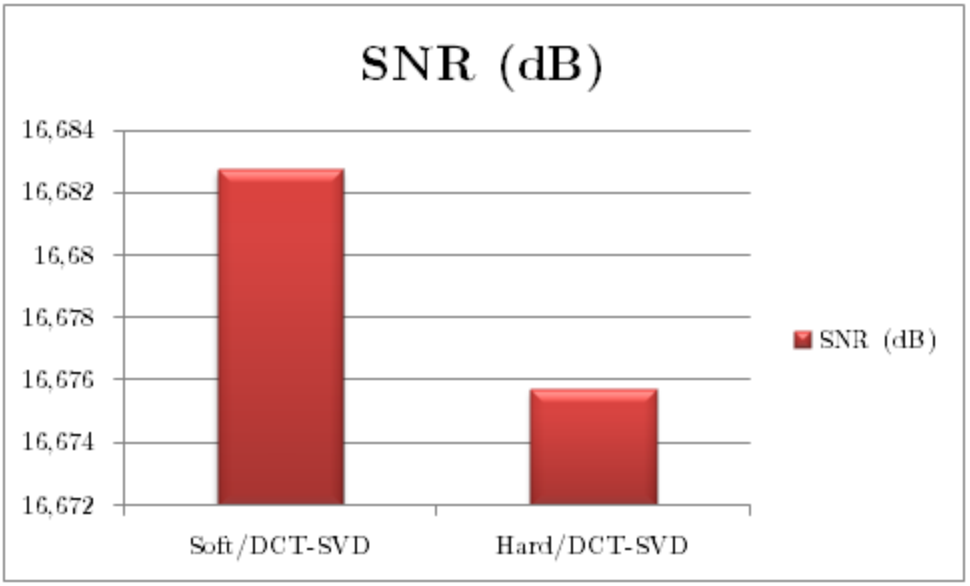


Figure **5.29 : Analyse du SNR pour Soft/DCT-SVD et Hard/DCT-SVD**

- Recherche de « rheumatoid arthritis » et « smoking and heart disease » dans la base de données Cochrane

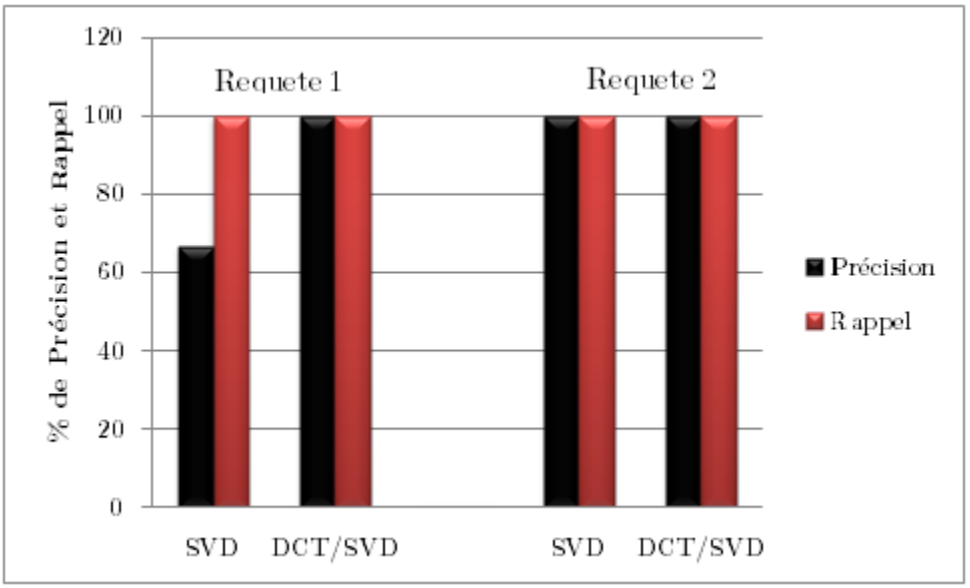


Figure 5.30 Précision et Rappel pour SVD et DCT/SVD

La Fig. 5.30, pour une valeur de $k = 30$ et un seuil=0.04, montre le même scenario que la section précédente. Pour la première requête, la méthode standard renvoie un document de plus, par contre pour l'approche hybride DCT/SVD nous avons une

précision et un rappel de 100% ce qui veut dire que tous les documents pertinents dans la base de données sont retrouvés. Pour la deuxième méthode la différence n'est pas aperçue car les deux approches montrent les mêmes résultats de précision et rappel.

Encore les résultats pour cette approche hybride montre que l'exactitude des résultats renvoyés a été améliorée en appliquant la transformée comme une étape de prétraitement.

5.4 Comparaison entre les différentes techniques

Les sections précédentes ont présentés une étude comparative entre l'application de l'algorithme SVD seul et l'application des transformées de traitement d'image comme étape de prétraitement en utilisant les métriques Précision et Rappel. Dans cette section, nous allons présenter une étude comparative entre les différentes techniques utilisées mais en calculant les SNR car la différence entre les approches hybrides n'est pas aperçue.

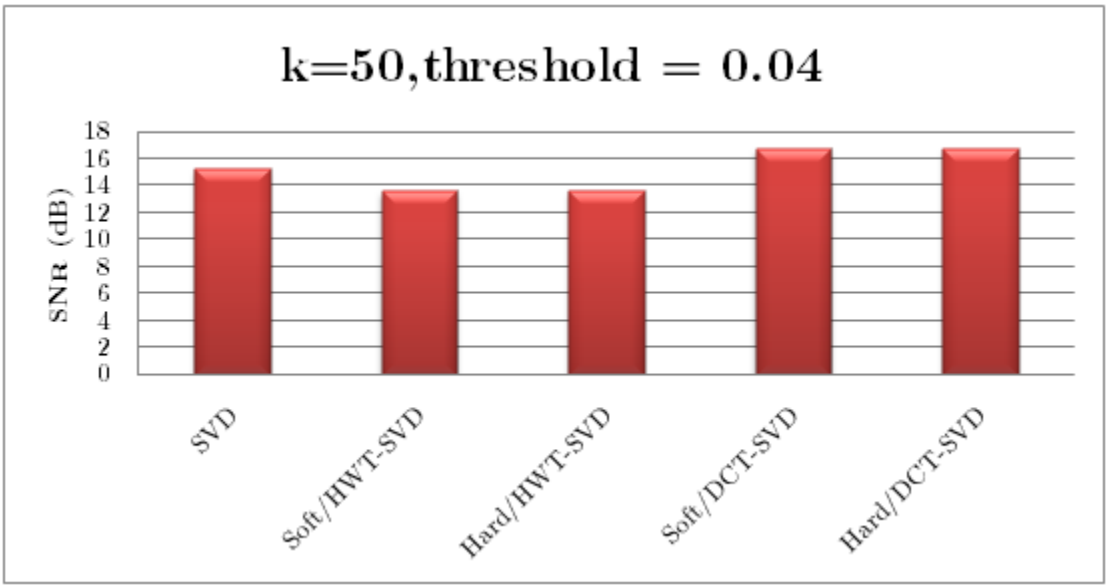


Figure 5.31 SNR pour les différentes approches

Dans la Fig. 5.31, la différence est claire, pour une valeur de $k = 50$ et pour un seuil 0.04, les résultats montre un SNR un peu plus élevé pour l'approche standard SVD/LSI que l'approche hybride HWT/SVD pour les deux cas de seuillage (Hard et Soft), par contre on remarque un SNR plus élevé pour l'approche hybride DCT/SVD et toujours pour les deux cas de seuillages, ce qui montre les performances de cette technique

comme étape de prétraitement dans un système de recherche d'information utilisant la technique LSI.

5.5 Réalisation

5.5.1 Outils utilisés

Dans cette section, nous allons présenter les différents outils utilisés pour faire les tests et l'interface graphique.

5.5.1.1 Visual Studio 2010

Microsoft Visual Studio est une suite de logiciels de développement pour Windows conçue par Microsoft. La dernière version s'appelle Visual Studio 2010. Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications WebASP.NET, des Services WebXML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# et Visual J# utilisent tous le même environnement de développement intégré (IDE, Integrated Development Environment), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du Framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications Web ASP et de Services Web XML grâce à Visual Web Developer.

Ainsi, une bibliothèque a été intégrée à Visual Studio, *Bluebit Matrix Library 6.1* pour implémenter les outils de l'algèbre tel que les matrices et les vecteurs.

5.5.1.2 Matlab

Matlab est un langage technique de haut-niveau et un environnement interactif pour le développement des algorithmes, visualisation et analyse des données et les calculs numériques.

Matlab est utilisé dans plusieurs applications telles que le traitement de signal et des images, communications, tests et mesures… etc.

5.5.1.3 Fiji

Fiji est un programme qui évalue le SNR, PSNR, RPSE and MAE d'une ou plusieurs images. Il compare une image référence $r(x,y)$ à une image test $t(x,y)$. Les deux images doivent avoir la même dimension.

5.5.2 Interface graphique

Pour une meilleure illustration des différentes étapes de l'étude, une interface graphique a été construite. Elle permet de faire une recherche dans la base de données Cochrane utilisant les trois techniques de recherche étudiées (SVD, HWT/SVD et DCT/SVD), ainsi elle présente les différentes étapes de prétraitement et les différents composants de la technique LSI.

Un aperçu de l'application est montré dans les Figures 32 et 33.

Figure 5.32 page d'accueil de l'interface

La page d'accueil, comme montré dans la Fig. 5.32 permet à l'utilisateur de faire une recherche dans la base de données Cochrane, et par défaut l'algorithme utilisé est le SVD simple avec une valeur de $k = 50$.

Dans la partie détail, l'utilisateur a la possibilité de choisir entre les quatre bases de données (Memos, Cochrane, eBooks et Reuters), il peut ensuite visualiser leurs TDMs, et les images générées.

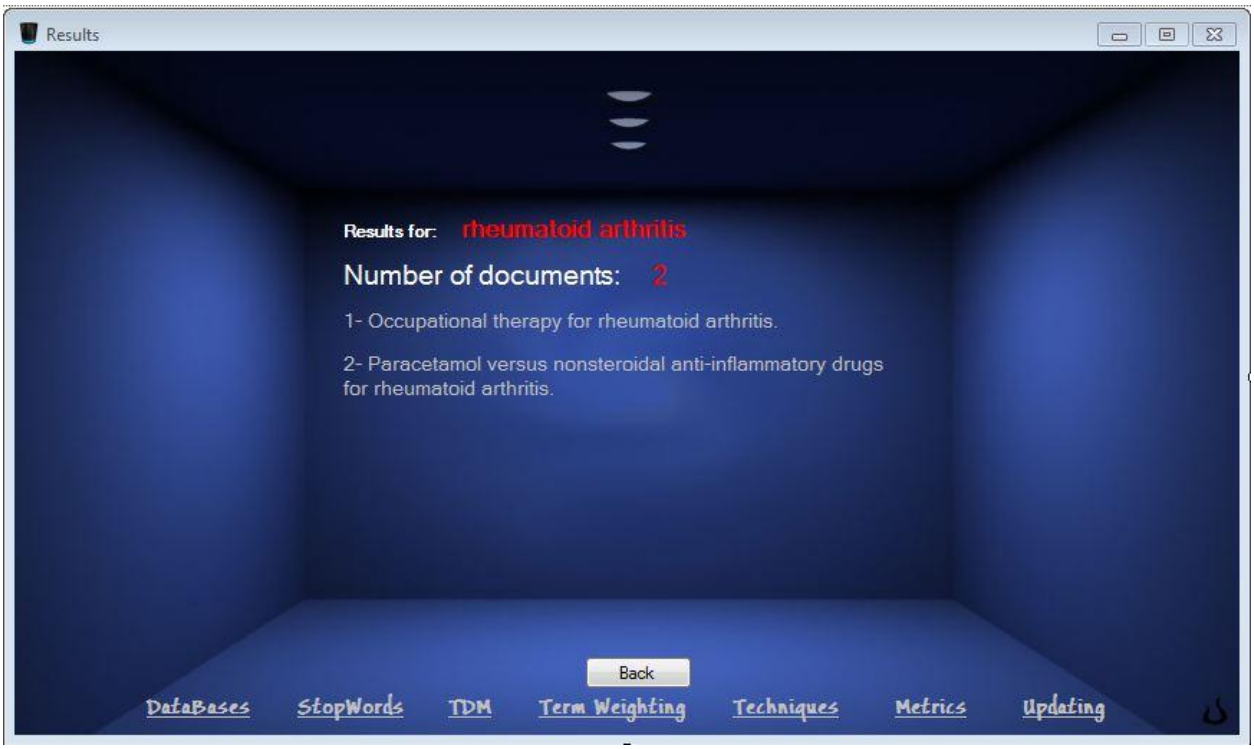


Figure 5.33 Exemple de recherche

L'utilisateur a aussi la possibilité de choisir entre les différentes techniques de recherche étudiées dans ce mémoire (LSI standard, HWT/SVD et DCT/SVD). Il peut aussi voir les résultats des tests pour les métriques utilisées de précision et rappel et de SNR.

Conclusion et perspectives

Conclusion Générale

Les travaux présentés dans ce mémoire se situent dans le contexte général des systèmes d'accès à l'information et plus précisément dans le cadre de l'Indexation Sémantique Latente.

Nous avons présenté les concepts de base de la RI en précisant les modèles essentiels dans ce domaine, ainsi que des grand projets dans l'histoire de la RI.

Nous avons ensuite présenté la technique d'Indexation Sémantique Latente (LSI) et ses composants ainsi que l'algorithme de décomposition en valeurs singulières.

Dans la partie suivante, nous avons décrit les deux transformées d'image utilisées dans les tests qui sont la transformée en ondelettes et la transformée en cosinus discrète. Ainsi que les techniques de seuillage « Soft » et « Hard ».

Le chapitre suivant, décrit l'étape d'évaluation des systèmes de recherche d'information.

Nous avons consacré la dernière partie de ce mémoire à nos propositions. Nous rappelons que notre objectif est d'étudier l'effet des techniques de traitement d'image sur l'amélioration des systèmes de recherche d'information basant sur la technique LSI. Nous nous sommes particulièrement intéressés à la comparaison entre les deux techniques HWT et DCT et aussi la technique LSI standard. Notre but est de proposer une technique qui donne les meilleurs résultats.

Afin de répondre à cela, nous avons construit les démarches suivantes :

Nous avons tout d'abord évalué le système LSI standard en appliquant l'algorithme de décomposition en valeurs singulière (SVD) seul et avons mesuré la précision et le rappel pour des requêtes prédéfinies dont on connaît leurs documents pertinents ainsi que le Rapport Signal sur Bruit (SNR) pour trouver la meilleure valeur de k. La meilleure valeur était pour $k = 30$ où le SNR donne un pique, mais généralement des résultats acceptables entre 20 et 50.

Nous avons ensuite appliqué la Transformée en Ondelettes de Haar sur l'image générée pour la TDM originale, puis appliqué les deux techniques de seuillage « Soft » et « Hard » et faire la transformée inverse pour revenir à la TDM après filtrage, ensuite appliquer l'algorithme SVD pour $k = 30$. Les résultats de précision et de rappel pour l'approche hybride HWT/SVD montre des meilleures performances que l'approche LSI standard (en appliquant le SVD seul) et donne une précision et un rappel de 100%. La mesure du SNR pour les deux techniques de seuillages donne des valeurs élevées pour le « Soft » et relativement faible pour le « Hard ».

Dans l'étape suivante, nous avons appliqué la Transformée en Cosinus Discrète (DCT) à la place de la HWT et suivi les même étapes. Les résultats ont montré encore l'exactitude de l'approche hybride par rapport à l'approche standard et montrent une meilleure performance du seuillage « Soft » par rapport au « Hard ».

La dernière étape était de comparer entre les différentes approches notamment les deux approches hybrides HWT/SVD et DCT/SVD en mesurant le SNR pour une valeur de $k = 50$ et un seuil=0.04. Les résultats obtenus sont clairs et ont montré que l'approche Hybride DCT/SVD donne le meilleur SNR, ce qui prouve les performances de cette approche par rapport aux autres.

Perspectives

Le travail présenté dans ce mémoire a ouvert un nouveau domaine d'intérêt à la recherche d'information utilisant l'Indexation Sémantique Latente (LSI). Les perspectives envisageables à nos travaux portent sur plusieurs directions, par exemple :

- Déterminer la meilleure valeur de k utilisée dans l'algorithme SVD, qui peut être bien examinée, ainsi pour les valeurs de seuillage pour les transformées.
- Appliquer les nouvelles techniques hybrides à des grandes bases de données pour confirmer les résultats.
- Etudier l'application d'autres transformées de traitement d'images, et d'autres techniques de traitement d'image en général (les bandelettes par exemple).
- Etudier l'application des techniques LSI à des bases de données distribuées et centralisées » à partir de sources de données homogènes et hétérogènes.

- Utiliser des mots clés composés (Two-Keywords par exemple)
- Utilisation de techniques de visualisations avancées pour une meilleure évaluation et analyse de la fiabilité du système proposé.

Bibliographie

[1] A. Amira and P. Farrell, "An automatic face recognition system based on wavelet transforms, " Proceedings of the IEEE International Conference on Circuits and Systems, pp. 6252-6255, 2005.

[2] A. Singhal, "Modern information retrieval: A brief overview, " IEEE Data Engineering Bulletin, vol. 24, pp. 35-43, 2001.

[3] B. Efron and R. J. Tsibirani. An Introduction to the Bootstrap. Chapman and Hall, New York, 1994.

[4] B. Yoon and P. P. Vaidyanathan, "Wavelet-based denoising by customized thresholding, " Proceedings of the IEEE International Conference on Acoustics,Speech, and Signal Processing, vol. 2, pp. 925-928, 2004.

[5] C. Buckley and E. M. Voorhees. Evaluation measures stability. In SIGIR Conference 2000, pages 33-40, Athens, Greece, 2000.

[6] C. Chateau. Corrélation sémantique entre documents : application à la recherche d'information juridique sur le web. PhD thesis, centre de recherche en informatique, Mines de Paris [ENSMP], Informatique temps réel, robotique et automatique, 2003.

[7] C. Cleverdon. Evaluation tests of info retrieval systems. Journal of documentation, 26 : 55-67, 1970.

[8] C. E. Jacobs, A. Finkelstein, , and D. H. Salesin., "Fast multiresolution image querying, " In Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, p. 277286, 1995.

[9] C. Fox, "Lexical analysis and stoplists. In information retrieval - data structures & algorithm, " Prentice-Hall, pp. 102-130, 1992.

[10] C. J. van Rijsbergen. Information retrieval. Butterworths, 1979.

[11] Cochrane, "Url: http://www.cochrane.org, " 2005.

[12] D. Grossman and O. Frieder. Information Retrieval : Algorithms and Heuristics. ISBN 0-7923-8271-4. Kluwer Academic Publishers, 1998. Second Edition 2004 by Springer Publishers.

[13] D. Haines and W.B. Croft. Relevance feedback and inference network. In 16th Annual International ACM SIGIR Conference on Research and developement in Information Retrieval, pages 2, 11, 1993.

[14] D. Harman. Relevance feedback revisited. In 15th Annual International ACM SIGIR Conference on Research and developement in Information Retrieval, pages 1,10, 1992.

[15] D. Kalman, "A singularly valuable decomposition: The svd of a matrix, " Col- ege Mathematics Journal, vol. 27, pp. 2-23, 1996.

[16] D.J. Foskett. Thesaurus. In Encyclopedia of Library and Information Science, pages 416–463. A. Kent, H. Lancour, 1980.

[17] D.J. Foskett. Thesaurus. In In Readings in Information Retrieval, pages 111–134. P. Willett, K. Sparck-Jones (Morgan Kaufmann), 1977.

[18] E. Hoenkamp, "Unitary operators on the document space source, " Journal of the American Society for Information Science and Technology, vol. 54, pp. 314- 320, 2003.

[19] E. J. Stollnitz, T. D. DeRose and D. H. Salestin. Wavelets for Computer Graphics: A Primer, Part 1.

[20] E. Voorhees and C. Buckley. The effect of topic set size on retrieval experiment error. In SIGIR Conference 2002, pages 316–323, Tampere, Finland, 2002.

[21] eBooks, "Url: http://www.library.qub.ac.uk, " 2005.

[22] F. Crestani and C. J. van Rijsbergen. A study of probability kinematics in information retrieval. In ACM Trans. Inf. Syst., volume 16(3), pages 225–255, 1998.

[23] F. Crestani. Implementation and evaluation of a relevance feedback device based on neural networks. In IWANN, pages 597–604, 1995.

[24] G. OBrien, "Information management tools for updating an ing scheme, " Master's thesis, University of Tennessee, Knoxville, TN, 1994.svd-encoded index-

[25] G. Oksa, M. Becka and M. Vajtersic. Parallel SVD computation in updating problems of Latent Semantic Indexing. Conference on Scientific Computing, pp. 113-120. Proceeding of Algorithmy 2002

[26] G. Salton and C. Buckley, "Improving retrieval performance by relevance feedback, " Journal of the American Society for Information Science, vol. 41, pp. 288-297, 1990.

[27] G. Salton and C. Buckley. Term Weighting approaches in Automatic Text Retrieval. Information Processing & Management Vol. 24 No. 5, pp. 513-523, 1988.

[28] G. Salton and M.J. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, New York, 1983.

[29] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing, " Communications of the ACM, vol. 18, p. 613620, 1975.

[30] G. Salton. The SMART retrieval system: experiments in automatic document processing. Prentice Hall, 1971

[31] H. Zha and H. Simon, "On updating problems in latent semantic indexing, "SIAM Journal on Scienti¯c Computing, vol. 21, pp. 782 - 791, 1999.

[32] I. Daubechies, "Ten lectures on wavelets, " No 61 in CBMS-NSF Series in Applied Mathematics. Philadelphia: SIAM., 1992.

[33] I. Delakis, O. Hammad, and R. I. Kitney, "Wavelet-based de-noising algorithm for images acquired with parallel magnetic resonance imaging (mri), " Physics in Medicine and Biology, vol. 52, pp. 3741-3751, 2007.

[34] J. P. Callan, W. B. Croft, and S. M. Harding. The INQUERY retrieval system. In proceeding of DEXA 92, 3rd International Conference on Database Expert Systems Applications, pages 78-83, 1992.

[35] J. Savoy. Statistical inference in retrieval effectiveness evaluation. Information Processing and Management, 33(4):495–512, 1997.

[36] J. Tague-Sutcliffe and J. Blustein. A statistical analysis of the trec-3 data. In Proceedings of TREC-3, The Third Information Retrieval Conference, pages 385–398, 1994.

[37] J.D. Harwood. Neural network implementation of a novel heuristic neural algorithm, 1990. Maryland University, College Park.

[38] L. Aroyo and D. Dicheva. Information retrieval and visualization within the context of an agent-based information management system. In Educational Multimedia, Hypermedia and Telecommunications (Ed-Media'99), pages 195-200, Washington, 1999.

[39] L. Azzopardi, M.Girolami and K. Van Rijsbergen. Investigating the Relationship between Language Model Perplexity and IR Precision-Recall Measures. In Annual ACM Conference on research and Development in Information Retrieval, July 28 – August 1 2003, pages pp. 369-370, Toronto, Canada, 2003.

[40] L. Muflikhah and B. Baharudin. Document Clustring using Concept Space and Cosine Similarity Measurement. International Conference on Computer Technology and Development 2009.

[41] M. Baziz, N. Aussenac-Gilles, and M. Boughanem. Exploitation des Liens Sémantiques pour l'Expansion de Requêtes dans un Système de Recherche d'Information. In XXIème Congrès

INFORSID 2003, Nancy, France, 03/01/03-06/06/03, pages 121-134, Inforsid, 20 rue Axel Duboul – 31000 Toulouse, Janvier 2003. INFORSID.

[42] M. Bell and N. Degani, "Latent semantic indexing, parallel svd and its appli cations, " Proceedings of ALGORITMY 2002, pp. 113-120, 2002.

[43] M. Berry, S. Dumais, and G. OBrien, "Using linear algebra for formation retrieval, " SIAM Review, vol. 37, pp. 573 - 595, 1995.intelligent in-

[44] Propagation. In 5th International Conference on Computer Assisted Information Retrieval, RIAO, pages 469-487. -, juin 1997. Dates de conference: juin 1997.

[45] M. Boughanem, C. Chrisment, and C. Soule-Dupuy. Query modification based on relevance backpropagation in adhoc environment. Information Processsing and Management, 35 : pages 121- 139, 1999

[46] M. F. Bruandet and J. P. Chevallet. Assistance intelligente à la recherche d'informations. In Ingénierie des systèmes d'Information (ISI), chapter 3, pages 85-118. Edition Hermes, Gaussier, E., Stefanini, M. H., septembre 2003.

[47] M. F. Porter, "An algorithm for suffix stripping, " Program, vol. 14, pp. 130-137,1980.

[48] M. W. Berry, Z. Drmavc, and E. R. Jessup, "Matrices, vector information retrieval, " SIAM Review, vol. 41, pp. 335-362, 1999.spaces, and

[49] O. Frieder, D. Grossman, A. Chowdhury, and G. Frieder. Efficiency considerations for very large information retrieval servers. Journal of Digital Information(British Computer Society), 1(5), April 2000.

[50] P. Borlund and P. Ingwersen. Measures of relative relevance and ranked half-life: performance indicators for interactive ir. International ACM-SIGIRN conference, pages 24-28, 1998.

[51] P. Husbands, H. Simon and C. Ding. On the Use of Singular Value Decomposition for Text Retrieval. NERSC Division. Lawrence Berkeley National Laboratory. Berkeley. CA 94720, 2001.

[52] Query operations (relevance feedback / query expansion), " PowerPoint Presentation in Information Retrieval and Web Search Course, University of Texas at Austin URL: www.cs.utexas.edu/ mooney/ir-course/, 2008.

[53] R. A. Baeza-Yates and B. A. Ribeiro-Neto. Modern Information Retrieval. ACM Press / Addison-Weseley, 1999. ISBN : 0-201-39829-X

[54] R. DeVore, B. Jawerth, and B. Lucier., "Image compression through wavelet transform coding, " IEEE Transactions on Information Theory, vol. 38, p.719746, 1992.

[55] R. T. Ogden, "Essential wavelets for statistical applications and data analysis, " Boston: Birkhauser, 1996.

[56] S. A. Khayam, "The discrete cosine transform (dct): Theory and application, " Technical Report, DCT Tutorial, 2003.

[57] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, and R. Harshman, "Indexing by latent semantic analysis, " Journal of the Society for Information Science, vol. 41, pp. 391-407, 1990.

[58] S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression, " IEEE Transactions on Image Processing, vol. 9,pp. 1532-1546, 2000.

[59] S. Laine-cruzel. Vers de nouveaux systèmes d'information prenant en compte le profil des utilisateurs. Documentaliste-Sciences de l'information, 31(3) :143–147, 1994.

[60] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation. " IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, p. 674693, 1989.

[61] S.P Harter. Psychological relevance and information science. American Society for Information Science (JASIS), 43(9):602–615, 1992

[62] Stephane G. Mallat. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. IEEE Transaction on Pattern Analysis and Machine Intelligence. Vol. II. No 7. July 1989.

[63] Susan T. Dumais, George W. Furnas and Thomas K. Landauer. Indexing by Latent Semantic Analysis.

[64] T. A. Letsche and M. W. Berry, "Large-scale information retrieval with latent semantic indexing, " Information Sciences: International Journal, vol. 100, pp. 105 - 137, 1997.

[65] T. Jaber, A. Amira and P. Milligan. A Novel Approach for Lexical Noise Analysis and Measurment.

[66] T. Jaber, A. Amira and P. Milligan. Performance Evaluation of DCT and Wavelet Transform for LSI.

[67] T. Jaber, A. Amira and P. Milligan. TDM modeling and evaluation of different transforms for LSI, 2009

[68] Y. Yang. Noise Reduction in a Statistical Approach to Text Categorization. Rochester, Minnesota 55905 USA