République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud Mammeri de Tizi Ouzou

Faculté de Génie Electrique et informatique

Département d'Informatique

Mémoire

En vue d'obtention du diplôme de Master en Informatique

Option : Systèmes Informatique

Implémentation et évaluation d'une méthode de sélection de documents d'expansion basée sur le score de clarté de la requête.

Proposé et dirigé par :

M^r A.HAMMACHE

Réalisé par :

HAMOUM Amine

ABERKANE Ferhat

Promotion 2014/2015

REMERCIEMENTS

Nous tenons tous d'abord à remercier le Bon Dieu tout puissant de nous avoir donné patience, courage et volonté pour réussir notre mémoire.

Nous souhaitons également exprimer notre profonde gratitude à tous ceux qui, de près ou de loin ont participé à la réalisation du présent travail.

Nous adressons à cet effet nos remerciements à :
Monsieur HAMMACHE Arezki, notre promoteur, pour nous avoir proposé ce sujet et de nous avoir dirigé tout au long de sa réalisation.

Nous tenons également à remercier « TOUS » les enseignants du département informatique de l'université Mouloud Mammeri de Tizi-Ouzou qui ont assuré notre formation durant notre parcours universitaire, pour l'ensemble des connaissances qu'on a consenti à leur égard.

Nous remercions vont également aux membres du jury pour avoir accepté d'évaluer notre travail.

Sans oublier de remercier nos amis et nos collègues qui ont tous, d'une manière différente, ont contribué à ce que nous puissions aboutir à la réalisation de ce mémoire.

Enfin nous remercions notre famille pour leur soutien et l'encouragement qu'ils nous ont apporté tout au long de notre travail.

Sommaire :

Chapitre III: Evaluation et expérimentation de l'approche proposé

Liste des figures

Liste des tableaux

De nos jours, l'information joue un rôle primordial dans le quotidien des individus et dans l'essor des entreprises. Ces derniers ont engendré beaucoup de changements à la fois quantitatifs et qualitatifs de l'information manipulée, en effet, avec l'apparition du web, l'utilisateur se trouve confronté à une quantité de documents et corpus de plus en plus volumineux et hétérogènes.

De ce fait, la Recherche d'information (RI) a connu un développement considérable dans le but de rendre accessible l'information utile à l'utilisateur et lui permettre la restitution des documents produits en fonction de son besoin exprimé. Pour ce faire, la RI adopte des méthodes assurant la conception et la réalisation de système de recherche d'information (SRI).

L'objectif principal d'un système de recherche d'information (SRI) est de rendre accessible l'information pertinente pour l'utilisateur et donc satisfaire son besoin en information. La notion de pertinence représente dans ce cas un enjeu essentiel dans la conception de systèmes de recherche d'information.

Notre travail se situe dans le contexte de la recherche d'information, plus particulièrement dans le cadre de la reformulation de la requête.

En effet la reformulation de la requête consiste à modifier la requête utilisateur dans l'objectif d'avoir une meilleur description du besoin de l'utilisateur. Par conséquent, les informations à retourner seront plus pertinentes.

Pour réaliser cette opération (reformulation), généralement les premiers documents sont utilisés comme source de données pour extraire les termes d'expansion.

Cependant cette méthode ne convient pas à tous les types de requêtes et précisément les requêtes ambigües, d'où la nécessité d'effectuer la diversification dans la sélection des documents.

Notre travail consiste à mettre en œuvre cette approche qui consiste à choisir les documents d'expansion. Ce choix est basé sur le score de clarté de la requête.

Pour atteindre cet objectif nous avons répartis notre travail en trois chapitres comme suit :

Le premier chapitre fait un état de l'art de la Recherche d'Information. Tous d'abord, nous présentons les concepts de base de la recherche d'information. On y trouve l'architecture du SRI, les notions de besoin en information, de requête, de document et de pertinence et l'indexation dont on a présenté les différentes, étapes d'indexation automatique. Nous décrivons aussi les différents modèles de la recherche d'informations en particuliers le modèle booléen, le modèle vectoriel et le modèle probabiliste. Le dernier point traité dans ce chapitre concerne l'évaluation des systèmes de recherche d'information.

Le deuxième chapitre est consacré à l'expansion de requêtes. Nous définissons tout d'abord l'expansion de la requête. Nous présentons ensuite la classification des techniques de reformulation de requêtes, le processus d'expansion automatique de la requête et ses paramètres de performances. Enfin nous décrivons l'expansion de la requête dans le modèle de langue.

Dans le troisième chapitre nous présentons notre approche, puis nous décrivons l'environnement technique utilisé pour la mise en œuvre de notre approche. Enfin nous présentons les résultats des expérimentations réalisées sur une collection TREC AP88.

Nous concluons notre mémoire par une conclusion générale et quelques perspectives.

I.1 Introduction

La Recherche d'Information (RI) est un domaine de l'informatique qui s'intéresse à l'organisation, au stockage et à la sélection d'informations répandant aux besoins des utilisateurs **[Salton,70] [Salton, 84]**.

La recherche d'information vise à satisfaire les exigences des utilisateurs en informations en mettant en place un mécanisme d'appariement entre les requêtes des utilisateurs et les documents d'une base documentaire, pour restituer les documents pertinents. Elle s'attache à résoudre les problèmes de la granularité de l'information (document, passage, paragraphe, etc.) ainsi que le type de l'information recherchée (image, vidéo, texte, etc.)

Un système de Recherche d'Informations (SRI) est un système informatique qui permet l'accès à un ensemble de documents par leur contenu « sémantique ». Cet accès est le résultat d'une recherche documentaire. Celle-ci consiste à mettre en correspondance une représentation de besoin de l'utilisateur (requête) avec une représentation de contenu des documents (fiche ou enregistrement) au moyen d'une fonction de comparaison **[Ihadjadene, M, 04].**

Ce chapitre a pour but de présenter brièvement les concepts de base de la RI. Nous présentons en particulier, les notions de document, de requête et de pertinence. Ensuite les processus de base de la RI, à savoir l'indexation appariement document/requête et la reformulation de la requête. Nous présentons également les modèles de RI et l'évaluation des SRI.

I.2 Concepts de base de la recherche d'information

Historiquement, la gestion des documents concernait surtout des spécialistes bibliothécaires, documentalistes. Ceux-ci doivent d'une part stocker les documents et en assurer la pérennité, et d'autre part en rendre l'accès possible. Avec l'explosion de la quantité d'informations mises à disposition du grand public et des entreprises, notamment au travers Internet, l'automatisation du stockage et de la consultation de l'information est devenue un besoin. Le développement d'outils et de méthodes pour gérer ces quantités d'informations est bien plus qu'un besoin, une nécessité. Ce sont les

SRI qui assurent le stockage et permettent la consultation d'informations. Du côté du système, le processus de RI est composé de plusieurs fonctions :

- L'indexation des documents et des requêtes.
- La restitution des documents pertinents par rapport à la requête.

Du point de vue utilisateur, le processus de recherche est composé de deux fonctions principales :

- L'interrogation du système par une requête
- L'analyse des documents restitués par le système et une éventuelle reformulation de la requête.

L'architecture d'un Système de Recherche d'Information(SRI) est représentée schématiquement sur la figure suivante (figure I.1).

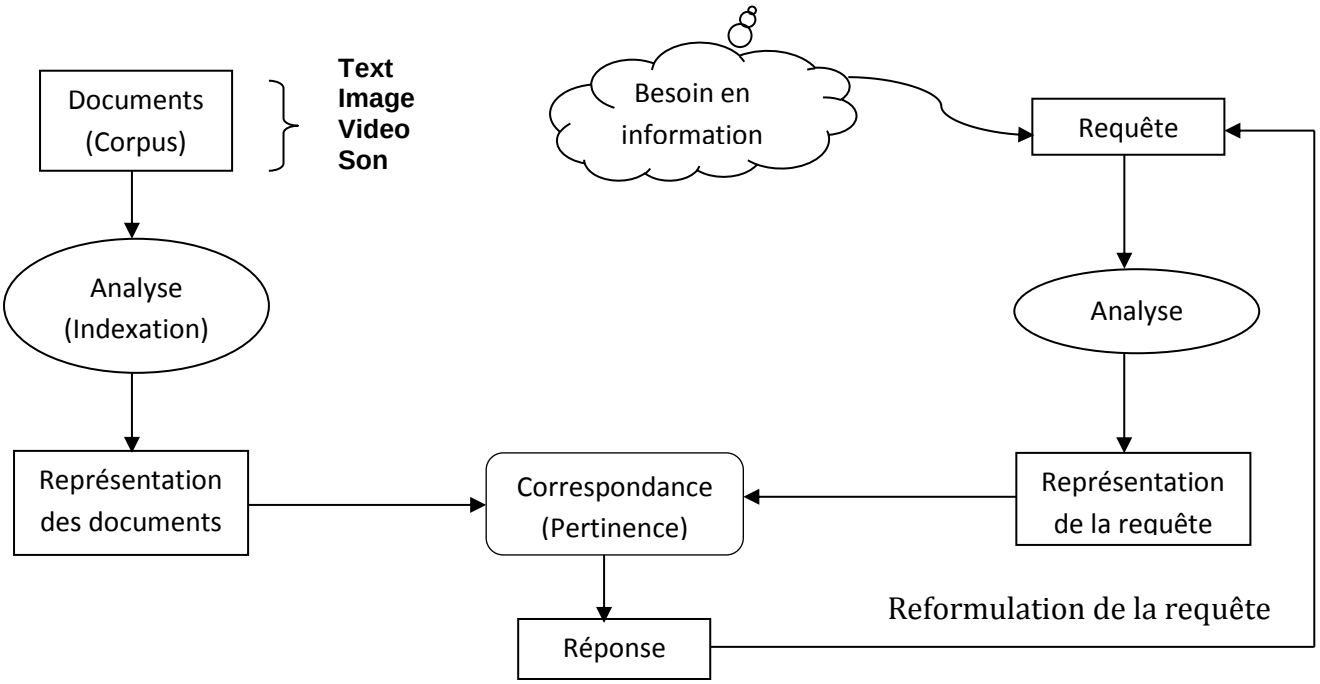

Figure I.1 Architecture du système de recherche d'information

L'analyse de l'architecture d'un SRI fait ressortir les éléments suivants :

A) Document et collection de documents

Un document est un élément essentiel dans un SRI. Dans son acceptation courante, l'une des définitions possible du terme document est de le considérer comme un support physique de l'information, qui peut être du texte, une page web, une image, un film, un morceau de musique, un schéma, ou un document multimédia.

Dans le cas d'un document texte on peut le représenter selon trois vues**[Fuhr, N, 05]**, **[Salton, G, 83]** :

- **La vue sémantique(ou contenu)** : elle se concentre sur l'information véhiculée dans le document.
- **La vue logique** : elle définit la structure logique du document (structuration en chapitre, sections).
- **La vue représentation** : elle consiste en la représentation sur un médium à deux dimensions (alignements de paragraphes, indentation, en-têtes et pieds de pages,etc.).

L'ensemble des documents manipulés par un SRI se nomme collection de document (ou base documentaire).

B) Besoin en information et requête :

Le besoin en information est l'expression mentale de ce que l'utilisateur recherche.
Trois types de besoin ont été définis dans **[Ingwersen, 92]** :

- Le besoin vérificatif : l'utilisateur cherche à vérifier une information particulière dont il sait comment y accéder (Exemple : La recherche d'un article à partir d'une adresse web connue). Ce type de besoin est stable et ne change pas au cours de la recherche.
- Le besoin thématique connu : l'utilisateur cherche à clarifier, à revoir ou à trouver de nouvelles informations dans un domaine connu (Exemple : l'utilisateur cherche les remèdes d'une maladie qu'il connait déjà). Ce type de besoin s'affine généralement au cours de la recherche.
- Le besoin thématique inconnu : l'utilisateur cherche de nouvelles connaissances dans un domaine ou un sujet qu'il ne connait pas. Un besoin de ce type est généralement exprimé de manière incomplète et imprécise.

Une requête est une expression approximative du besoin en information de l'utilisateur. Elle peut donc être considérée comme une description partielle d'un besoin d'information à un instant donné.
Les requêtes soumises au SRI par les utilisateurs peuvent ne pas refléter leur besoins en information. Cela est dû, au fait que l'utilisateur ignore le fonctionnement interne du SRI, et il n'a qu'une vision restreinte des documents disponibles dans la collection. D'autre part le SRI n'a souvent aucune connaissance a priori de ses utilisateurs (centres

d'intérêts, niveaux, etc.). Ce biais entre la requête et le besoin en information est une des difficultés majeures de tout système de recherche d'information. Afin de remédier partiellement à ce problème, un mécanisme de reformulation de requêtes peut être intégré dans les SRI.

C) La pertinence :

La pertinence est la notion centrale dans la RI, car toutes les évaluations s'articulent autour d'elle. Elle peut être définie comme un degré de correspondance entre un besoin en information de l'utilisateur et la réponse qui doit être renvoyée par le SRI.
On distingue deux types de pertinence :

- **La pertinence système :** La pertinence système est plus facile à définir. Elle représente le degré de similarité entre la requête et le document renvoyé par le SRI.
- **La pertinence utilisateur :** C'est l'appréciation de l'utilisateur sur le document retourné. Cette appréciation dépend du point de vue de l'utilisateur, c'est une mesure subjective, difficilement calculable.

I.2.1 L'indexation

I.2.1.1 Définition

L'indexation est un sujet à la fois très souvent cité, car il est situé au cœur des techniques documentaires, et largement méconnu du fait de la complexité et de la variété des notions et des traitements qu'il met en œuvre. Elle constitue un préalable à la recherche d'information.

Selon **Philipe Lefevre :**« l'indexation est la représentation formalisée et condensée du contenu thématique ou sémantique d'un document, à l'aide d'indicateur sémantique qui peuvent être des indices de classification ou des descripteurs, contrôlés ou non. Les indicateurs appartiennent au langage naturel et/ou à un langage documentaire ».**[P-Lefevre, 04]**
L'indexation peut être :

- **Indexation manuelle** :

Dans l'indexation manuelle, chaque document est analysé par un spécialiste du domaine ou par un documentaliste.

Ce type d'indexation permet une :

- ✓ Meilleure précision dans les documents que le système de RI retourne en réponse aux requête des utilisateurs.
- ✓ Meilleure correspondance entre les documents et les termes choisis pour les représenter.

Cependant, elle présente certains inconvénients, comme :

- ✓ Méthode inapplicable aux corpus de textes volumineux.
- ✓ Elle est lourde à mettre en œuvre lorsque le document est grand.
- ✓ L'effort intellectuel qu'elle exige (en temps et en nombre de personnes).

- ➢ **Indexation semi-automatique (supervisée)** :

L'indexation semi-automatique est une combinaison entre les deux méthodes (manuelle, automatique), elle utilise un vocabulaire contrôlé sous forme de thesaurus ou de base terminologique.

Les premiers termes d'indexation sont obtenu grâce à l'indexation automatique, il vient alors le rôle de l'indexeur qui corrige puis complète les informations obtenues et établi un ensemble final des termes d'indexation.

- ➢ **Indexation automatique :**

Elle est sans doute celle qui a été la plus étudiée en recherche d'information. Dans cette approche, chaque document est analysé à l'aide d'un processus entièrement automatisé. Elle permet de traiter les textes plus rapidement que l'indexation manuelle et elle est particulièrement adaptée aux corpus volumineux.

I.2.1.2 Etapes d'indexation automatique :

L'indexation automatique passe par plusieurs étapes :

a. **L'analyse lexicale** :

Il s'agit du processus de conversion de texte en une séquence de mots. Elle convertit un texte de document en une liste de termes.
Un terme est un groupe de caractères constituant un mot significatif. L'analyse lexicale permet de reconnaître les espaces de séparations des mots, les chiffres, etc.

b. **L'élimination des mots vides** :

Les mots vides sont des mots peut significatifs et porteurs de peu de sens, augmentant ainsi la taille de l'index et rendant la recherche plus lente. De ce fait, leurs élimination est impérative.

L'élimination des mots vides est importante dans la mesure où il représente un facteur qui a une grande influence sur la précision de la recherche. En effet, le fait de ne pas éliminer les mots vides provoque inévitablement du bruit (les documents non pertinents à la requête). On distingue deux techniques pour éliminer les mots vides :

-L'utilisation d'une liste de mots vides

-L'élimination des mots dépassant un certain nombre d'occurrence dans la collection

c. **La normalisation** :

La normalisation consiste à représenter les différentes variantes d'un terme par un format unique appelé lemme ou racine, Il s'agit de ramener le mot à son radical. Ce qui a pour effet de réduire la taille de l'index. Plusieurs stratégies de normalisation sont utilisées : la table de correspondance, l'élimination des affixes, la troncature.

L'inconvénient majeur de cette opération est qu'elle supprime dans certain cas la sémantique des termes originaux, c'est le cas par exemple activate/active.

d. **Le choix des descripteurs** :

Elle consiste à déterminer le type d'unités élémentaires pour représenter les documents. On parle aussi de descripteur. L'objectif est d'avoir une représentation des documents permettant une moindre perte d'information sémantique possible. On distingue plusieurs types de descripteurs :

- **Les mots simples** : les mots simples du texte de document (en éliminant les mots vides).
- **Les lemmes ou les racines** des mots extraits.
- **Les mots composés** :des groupes de mots sont repérés comme unités descriptives d'un document afin d'augmenter la précision de la réponse car un groupe de mots est souvent plus précis que les mots qui le composent pris séparément. Par exemple, le terme composé « accident de travail » est plus précis que « accident » et « travail » pris isolément.

- **Les concepts** : qui sont des expressions pris généralement d'une structure conceptuelle, tel que thésaurus ou les ontologies.

e. **Créations des structures d'indexes :**

L'index, défini comme étant la structure de stockage utilisée pour mémoriser les informations sélectionnées lors du processus d'indexation cette structure permet de sélectionner de n'importe quel terme tous les documents où il apparaît. Plusieurs solutions de stockage ont été proposées parmi lesquelles : les fichiers inverse (inverted files), les fichiers de signatures (signature files) et les tableaux de suffixes (suffixarray).

La solution la plus utilisés actuellement est celle des fichiers inverses. Ces fichiers sont composés de deux éléments principaux : le dictionnaire et le fichier posting. Le vocabulaire consiste en la liste de tous les mots distincts extraits du texte. Pour chaque mot est assigné l'ensemble des positions dans lesquelles ce dernier apparaît.

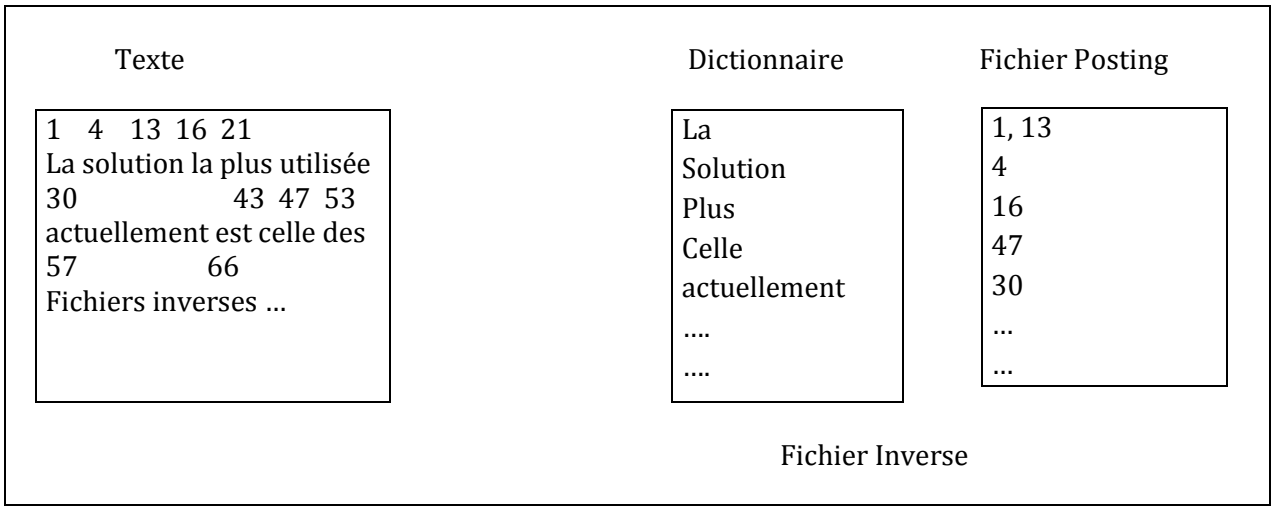

Figure I.2 Représentation en un fichier inverse d'un texte

I.2.2 La correspondance document-requêtes

La comparaison entre un document et une requête (correspondance ou appariement document-requête) revient à mesurer le degré de pertinence d'un document vis-à-vis une requête. Cela est réalisé par le SRI en représentant le document et la requête avec un même formalisme, et en comparant ces deux représentations.

Le degré de similarité (le score qui définit la probabilité de pertinence) entre un document et une requête est exprimé par une fonction nommée **R**etrieval **S**tatut **V**alue, elle est notée**RSV (d,q)**, où « d » représente un document et « q » une requête.

La valeur **RSV (d,q)** permet ensuite au SRI de classer les documents retournés à l'utilisateur.

I.2.3 La reformulation de la requête

Afin de faire correspondre au mieux la pertinence utilisateur et la pertinence système, une étape de reformulation est souvent utilisée. Cette fonction a pour objectif d'améliorer les performances du SRI, donc la précision dans les réponses du système.

Parmi les méthodes de reformulation de requête, nous pouvons citer la réinjection de pertinence ou relevance feedback. Elle consiste à exploiter l'information que le jugement de pertinence de l'utilisateur fournit sur les documents initialement restitués par le système en lien avec sa requête. La reformulation peut se faire sans expansion de la requête par simple repondération des termes (en maximisant le poids des termes apparaissant dans des documents jugés pertinent et en minimisant le poids des termes présents dans des documents reconnus non pertinent par l'utilisateur) ou en ajoutant des termes à la requête originale. L'ajout d'un terme peut se faire de plusieurs manières, soit par une interaction entre l'utilisateur et le système soit de façon automatique, on parle alors de réinjection de pertinence aveugle. Le système considère les premiers documents comme pertinents dont il extrait les termes à ajouter.

I.3 Les modèles de la Recherche d'information

Le modèle de RI à pour rôle de construire un formalisme qui permet une représentation interne des documents et des requêtes, et de définir une méthode de comparaison afin de calculer la similarité entre les documents et les requêtes ainsi représentés.

De nombreux modèles ont été proposés en RI, on distingue généralement trois grandes catégories de modèles : modèles booléens, modèles vectoriels et modèles probabilistes.

I.3.1 Le modèle booléen

Le modèle booléen est le premier modèle de la RI à avoir été mis en œuvre. Il est basé sur la théorie des ensembles **[Salton, 71]**. Dans ce modèle, les documents et les requêtes sont représentés par des ensembles de mots clés. Chaque document est représenté par une conjonction logique des termes non pondérés qui constitue l'index du document. Un exemple de représentation d'un document est comme suit : $d = t1 \wedge t2 \wedge t3 ... \wedge tn$.

Dans le modèle booléen, un document « d » est représenté par l'ensemble des termes (mots-clés) qui l'indexent. La requête « q » est représenté par plusieurs termes reliés entre eux par des opérateurs logiques : AND, OR et NOT.

La pertinence entre un document « d » et une requête « q » (noté RSV (d,q)) est déterminée de la manière suivante :

- Si la requête contient un seul terme, soit q = t (avec t : un terme)

  $RSV(d,q) = 1, si\ t \in d, 0\ sinon$ (I.1)

- Si la requête contient deux termes reliés par l'opérateur AND, soit $q = t1 \wedge t2$:

  $RSV(d,q) = 1\ si\ RSV(d,t_1) = 1\ et\ RSV(d,t_2)=1, 0\ sinon$ (I.2)

- Si la requête contient deux termes reliés par l'opérateur OR, soit $q = t1 \vee t2$:

  $RSV(d,q) = 1\ si\ RSV(d,t_1) = 1\ ou\ RSV(d,t_2)=1, 0\ sinon$ (I.3)

- Si la requête est composée de négation d'un seul terme, soit $q = \neg t$:

  $RSV(d,q) = 1\ si\ t \notin d, 0\ sinon$ (I.4)

Ce modèle présente un certains nombres d'avantages :

- Le modèle booléen est facile à implémenter.
- Il est très efficace à condition que l'utilisateur maîtrise parfaitement le langage de requêtes.

Néanmoins, ce modèle présente plusieurs inconvénients :

- Dans ce modèle les requêtes longues sont difficiles à exprimer sous forme booléenne.
- Les résultats retournés ne sont pas ordonnés selon leur pertinence et l'utilisateur préfère un classement lorsque la liste est grande.
- Si les documents ne contiennent pas tous les termes de la requête alors ils sont considérés automatiquement comme non pertinents.

La pondération des termes améliore les résultats, or que ce modèle ne prend pas cette dernière en considération : un terme à un poids égale à 1 s'il appartient au document, sinon c'est 0.

I.3.2 Le modèle vectoriel :

Le modèle vectoriel est un modèle qui a été proposé par Gerard Salton dans les années 1970 **[Salton, 71]**. C'est un modèle algébrique, il représente les documents ainsi que les requêtes par des vecteurs de termes dans un espace multidimensionnel (N dimensions, avec N : le nombre des termes d'indexation de la collection).
Les composants des vecteurs sont le poids du terme t_i d'indexation considéré dans le document ou la requête, ils sont définis de la manière suivante :

- Un vecteur document $D_j = (d_{1j}, d_{2j}, d_{3j}, \ldots, d_{Nj})$
- Un vecteur requête $Q = (q_1, q_2, q_3, \ldots, q_N)$

Où : d_{ij} est le poids du terme t_i dans le document d_j

La pondération est une fonction très importante en RI, elle permet de mesurer l'importance d'un terme dans un document.

Les technique de pondérations les plus utilisées sont basées sur les facteurs **Tf** et **Idf,** qui permettent de combiner les pondérations locale et globale d'un terme :

- **Tf (TermFrequency)** : C'est la fréquence du terme dans le document c'est-à-dire le nombre d'occurrences d'un terme dans le document (pondération locale).
- **Idf (Inverse of Document Frequency)** : C'est la fréquence du terme dans l'ensemble des documents (pondération globale). Elle doit être d'autant plus fort (faible) que le terme permet de mieux (moins bien) discriminer le document parmi les autres documents du corpus.

La mesure **Tf * Idf** donne une bonne approximation de l'importance du terme dans le document, particulièrement dans les corpus de documents de taille homogène.
Cependant, elle ne tient pas compte d'un aspect du document : sa longueur. En général, les documents les plus longs ont tendance à utiliser les mêmes termes de façon répétée, ou à utiliser plus de termes pour décrire un sujet. Par conséquence, les fréquences des termes dans les documents seront plus élevées, et les similarités à la requête seront également plus grandes.
La comparaison de la requête au document dans ce modèle est effectuée en évaluant la similarité entre leurs vecteurs respectifs. Les documents considérés comme les plus

pertinents sont ceux dont le vecteur est le plus proche de celui de la requête, en tenant compte d'une mesure de similarité définit au préalable.

Plusieurs mesures de similarité ont été définies, les principales utilisées sont les suivantes:

- *Le produit scalaire :SIM (Q, D_j) =* $\sum^{N}_{i=1} q_i * d_{ij}$ *(I.5)*
- *Mesure de Jaccard : SIM (Q, D_j) =* $\frac{\sum_{i=1}^{N} q_i * d_{ij}}{\sum_{i=1}^{N} q_i^2 + \sum_{i=1}^{N} d_{ij}^2 - \sum_{i=1}^{N} q_i * d_{ij}}$ *(I.6)*
- *Mesure cosinus : SIM (Q, D_j) =* $\frac{\sum_{i=1}^{N} q_i * d_{ij}}{(\sum_{i=1}^{N} q_i^2)^{1/2} * (\sum_{i=1}^{N} d_{ij}^2)^{1/2}}$ *(I.7)*

Le modèle vectoriel est simple à comprendre et facile à implémenter. Il permet une représentation uniforme des documents et des requêtes, il prend en considération le poids des termes dans les documents et les mesures de similarités qu'il utilise permettent de considérer un document comme pertinent même s'il ne contient pas tous les termes de la requête. De plus, ce modèle classe les résultats par ordre décroissant de pertinence.

Ce modèle présente quelques limitations comme le fait qu'il ne respecte pas l'ordre des mots, et il suppose l'indépendance entre les termes d'indexation, or que ces termes dans les documents sont sémantiquement liés.

I.3.3 Les modèles probabilistes

Les modèles probabilistes sont basés sur la théorie des probabilités. La mesure de similarité est fondée sur une estimation de probabilité.

Nous présentons ci-dessous deux types de modèles probabilistes : le modèle de base et le modèle de langue.

I.3.3.1 Le modèle probabiliste de base

Ce modèle présente le résultat de la recherche à l'utilisateur dans un ordre basé sur le rapport de la probabilité de pertinence d'un document vis-à-vis d'une requête, sur sa probabilité de non pertinence pour cette requête.

La correspondance RSV (d,q) entre un document « d » et une requête « q » est donnée par :

$$\text{RSV (d,q)} = \frac{P(R|d)}{P(NR|d)} \qquad \text{(I.8)}$$

Où :

$\boldsymbol{P}(\boldsymbol{R}|\boldsymbol{d})$(resp.$(\boldsymbol{NR}|\boldsymbol{d})$)Représente la probabilité qu'un document « d » soit pertinent (resp. non pertinent) vis-à-vis d'une requête « q ».

Selon la formule de Bayes, les deux probabilités sont obtenues de la manière suivante :

$$P(R|d) = \frac{P(d|R)P(R)}{P(d)} \qquad (I.9)$$

$$P(NR|d) = \frac{P(d|NR)P(NR)}{P(d)} \qquad (I.10)$$

Avec:

P(R) (resp. **P(NR)**) : représente la probabilité qu'un document sélectionné au hasard soit pertinent (resp. non pertinent).

P(d|R) (resp. **P(d|NR)**) : indique la probabilité que le document « d » fait partie de l'ensemble des documents pertinents **R** (resp. des documents non pertinents **NR**).

P(d) : correspond à la probabilité de choisir un document.

Après simplification, la correspondance RSV (d,q) entre un document et une requête est exprimée de la manière suivante :

$$RSV(d,q) = \frac{P(d|R)}{P(d|NR)} \qquad (I.11)$$

C'est la probabilité de pertinence de termes des documents qui fait que la probabilité de ces derniers soit pertinente. On peut calculer pour chaque terme « t_i » de la requête, la probabilité qu'un document qui contient t_i soit pertinent (**$P(t_{i=1}|R)$**) ou non pertinent (**$P(t_{i=1}|NR)$**).

Le calcul de probabilités dans ce modèle est trop complexe et il ne prend pas en considération les dépendances entre les termes, mais il a pour avantage de restituer les documents sélectionnés comme pertinents dans l'ordre de leurs pertinences.

I.3.3.2 Le modèle de langue

Ponte et Croft**[Ponte &Croft, 98]** furent les premiers à proposer un modèle de langue pour la RI.

L'intuition est qu'une requête n'est pas créée de façon aléatoire, mais l'utilisateur a une idée (un modèle) sur le document idéal qui peut apparaître dans « d » et de son modèle de langue M_d.

A partir de M_d l'utilisateur choisit (génère) les termes pour constituer sa requête. Donc la requête devrait être générée à partir d'un document pertinent ou de son modèle. La probabilité de génération de la requête à partir de modèle de document est considérée comme le score du document elle est présenté comme suit :

$$\text{Score}(q, d) = P(q|M_d) \qquad \text{(I.12)}$$

Ce modèle utilise une distribution de Bernoulli, c.-à-d. que les mots présents et ceux qui sont absents dans la requête sont pris en compte. Le score du document vis-à-vis de la requête est alors exprimé ainsi :

$$\text{Score}(q, d) = \prod_{t \in q} P(t|M_d) * \prod_{t \notin q}(1 - P(t|M_d)) \qquad \text{(I.13)}$$

Ce score est composé de deux parties : la probabilité d'observer les termes de la requête dans le document et la probabilité de ne pas observer les termes absents de la requête dans le document.

La probabilité **$P(t|M_d)$** est calculée par une méthode non paramétrique qui utilise la probabilité moyenne d'apparition du terme dans les documents qui le contiennent. **$P_{avg}(t)$** et un facteur de risque pour un terme observé dans le document R(t, d). Par contre la probabilité d'un terme dans la collection est utilisée pour les termes qui n'apparaissent pas dans le document. Le calcul de cette probabilité est exprimé ainsi :

$$P(t|M_d) = \begin{cases} P_{ML}(t|d)^{(1-R(t,d))} * P_{Avg}(t)^{R(t,d)} & \text{si } tf(t, d) > 0 \\ \frac{tf(t,C)}{|C|} & \text{sinon} \end{cases} \qquad \text{(I.14)}$$

Où : **tf (t, d)** est la fréquence de **t** dans le document **d** et tf(t, c) est la fréquence du terme **t** dans la collection **C.**

Dans le modèle proposé par Hiemstra **[Hiemstra, 98]**, la requête est considéré comme une séquence de terme, $q = (t_1, t_2, \ldots, t_n)$. Dans ce modèle on ne considère que les mots présents dans la requête, une distribution multinomial est utilisée. Le score d'un document vis- à- vis d'une requête (la probabilité de générer une requête q) sachant le modèle de document M_d est donné par la formule suivante :

$$\text{Score (q, d)} = P(d) \prod_{i=1}^{n} P(t_i|M_d) \qquad \text{(I.15)}$$

Où:$\boldsymbol{P}(\boldsymbol{d})$est la probabilité a priori d'un document.

Pour l'estimation de la probabilité $\mathbf{P(t_i|M_d)}$, Hiemstra utilise l'approche par interpolation qui combine le modèle de document $\mathbf{M_d}$ avec le modèle de langue de la collection. Le calcul de cette probabilité est exprimé comme suit :

$$P(t_i|M_d) = \lambda\, P_{ML}(t_i|M_d) + (1 - \lambda) P_{ML}(t_i|C) \qquad \text{(I.16)}$$

Où : **λ** est le poids d'interpolation qui vraie entre 0 et 1. Ce paramètre peut être considéré constant ou il peut être estimé d'une manière sophistiquée en utilisant un processus d'optimisation automatique tel que l'algorithme de maximisation d'espérance(EM).

Le calcul des deux probabilités $\mathbf{P_{ML}(t_i|M_d)}$ et $\mathbf{P_{ML}(t_i|C)}$ est réalisé selon une estimation de vraisemblance maximale, exprimé ainsi :

$$P_{ML}(t_i|M_d) = \frac{tf(t_i,d)}{|d|} P_{ML}(t_i|C) = \frac{df(t_i)}{\sum t_{j\epsilon v} df(t_j)} \qquad \text{(I.17)}$$

Où : $\mathbf{tf(t_i, d)}$ est la fréquence du terme t_i dans le document d, $\mathbf{df(t_i)}$ est le nombre de documents contenant le terme t_i, **|d|** est le taille du document et v est le vocabulaire d'index.

I.4 Evaluation des SRI

L'évaluation des méthodes et des modèles a été un centre d'intérêt important et cela dès l'apparition des premiers SRI. Et pour cela, des moyens comme les mesures de qualité, des systèmes de recherche et un ensemble de données, ont été développés dans le but d'évaluer ces systèmes.

L'évaluation des SRI peut être vue sous deux angles différents : paradigme système et paradigme usager. Le but du paradigme système consiste à mesurer les capacités du système à satisfaire les besoins en information des utilisateurs, c'est-à-dire la pertinence des résultats retournés.

Le paradigme usager par contre, vise à satisfaire l'utilisateur, et non les capacités du système.

L'approche « système » est la plus utilisée dans la recherche d'information, elle se base sur deux éléments essentiels : les mesures d'évaluation et les collections de test que nous présentons dans ce qui suit.

I.4.1 Mesures d'évaluation des SRI

L'objectif principal des SRI est de présenter à l'utilisateur tous les documents pertinents et de rejeter tous les documents non pertinents. Cet objectif est évalué avec différentes mesures d'évaluation. Les plus utilisées sont : la précision, le rappel, et les mesures alternatives.

I.4.1.1 La précision (P)

Elle mesure le pourcentage de la proportion du nombre de documents pertinents restitués parmi tous les documents restitués par le système, elle est exprimée comme suit :

$$P = \frac{Nombre\ de\ documents\ pertinents\ restitués}{Nombre\ total\ de\ documents\ restitués} \qquad (I.18)$$

La précision mesure la capacité du système à trouver seulement les documents pertinents. Elle vaut 1 si tous les documents retrouvés sont pertinents, et elle vaut 0 si aucun document retrouvé n'est pertinent.

I.4.1.2 Le rappel (R)

Le rappel mesure le pourcentage de la proportion du nombre de documents pertinents restitué parmi le nombre total de documents pertinents dans la collection il est exprimé comme suit :

$$R = \frac{Nombre\ de\ documents\ pertinents\ restitués}{Nombre\ total\ de\ documents\ pertinents} \qquad (I.19)$$

Quand le rappel vaut 1 c'est que le nombre total de documents pertinents ont tous été retrouvés par le système. Quand il vaut 0 alors aucun document pertinent n'a été retrouvé.

Des mesures complémentaires au rappel et précision ont été définies, il s'agit de bruit et de silence.

I.4.1.3 Les mesures alternatives

Plusieurs mesures alternatives ont été proposées, parmi elle, on trouve :

- **La précision exacte(ou R-Précision) :**

C'est une précision qui est calculée sur les R premiers documents retournés par le système, sachant que la requête admet R documents pertinents.

- **La précision moyenne non interpolée (MAP) :**

La précision moyenne non interpolée (AverageMeanPrecision) est une mesure de performance globale, elle est calculée en deux étapes :

Dans la première étape on calcule la précision moyenne pour une requête donnée (AP_q), et on calcule pour chaque document pertinent restitué sa précision (pr (d_i)) qui est égale au nombre de documents pertinents restitués sur le rang de ce document.
La précision des documents retrouvés non pertinents est égale à 0.
Donc calculer la précision moyenne pour une requête donnée revient à calculer la moyenne des précisions des documents pertinents , donnée comme suit :

$$AP_q = \frac{1}{N} \sum_{i=1}^{N} pr\ (d_i) \qquad (I.22)$$

Avec la valeur de **pr (d_i)** qui vaut **$r_{ni}//n_i$** si «d_i» est retrouvé ou 0 sinon.
Où :**N** est le nombre total de documents pertinents pour la requête q, **n_i** représente le rang du document d_i pertinent qui a été retrouvé et **r_{ni}** est le nombre de documents pertinents retrouvés au rang n_i.
La deuxième étape consiste à calculer la précision moyenne pour un ensemble de requêtes et cela en calculant la moyenne des précisions moyenne de chaque requête (c'est-à-dire AP_q) :

$$MAP = \frac{1}{M} \sum_{j=1}^{M} AP_{qj} \qquad (I.23)$$

Avec **M** est le nombre de requêtes considérées et **AP_{qj}** qui représente la précision moyenne pour la requête « j ».

I.4.2 Les collections de test :

Une collection de test représente le moyen d'évaluation des SRI. Elle comprend généralement :

- Un corpus de documents : c'est un ensemble de documents à indexer, sur lesquels le système sera évalué.
- Une liste de requêtes prédéfinies.
- Les jugements de pertinence : ils sont établis manuellement et ils contiennent la liste des documents pertinents pour chaque requête.

On peut juger l'efficacité et la performance d'un système de RI, en comparant les résultats retournés par ce dernier avec les jugements de pertinence.
Depuis les années 70, les collections de test se sont multipliées. Elles sont mise en place dans le cadre de compagne d'évaluation des SRI, dont la collection CACM, la collection CISI, la compagne CLEF et la compagne TREC qui constitue à ce jour la compagne de référence pour ce qui concerne l'évaluation des SRI et cela depuis l'année de son lancement 1992.
L'objectif de la compagne TREC consiste à réunir sous une plate-forme des collections de test et des protocoles d'évaluations pour des tâches spécifiques pour mesurer les différentes stratégies de recherche.
Les tâches proposées se sont diversifiés d'une compagne à une autre, parmi celles proposées dans TREC 2012 pour citer : recherche d'informations sur le web, recherche d'informations dans les micros blogs, recherche d'informations médicales, etc.

I.5 Conclusion

Dans ce chapitre nous avons présenté les principales notions et concepts de la recherche d'information. Nous avons donné la définition de la RI ainsi que le SRI et nous avons particulièrement abordé les concepts de base de la RI à savoir le document et collection de document, la requête, la pertinence et le besoin d'information. Nous avons développé aussi les processus de la RI tels que la correspondance document-requête et la reformulation de requêtes. Puis nous avons étudié les différents modèles de recherche et enfin les méthodes d'évaluation des systèmes de recherche d'information.

II.1 Introduction

Dans les systèmes de recherche d'information, souvent la requête initiale est insuffisante pour permettre de sélectionner des documents qui répondent aux besoins des utilisateurs.
De ce fait plusieurs techniques ont été proposées. Une de ces techniques est l'expansion de requêtes
Ce chapitre est consacré à cette technique qui consiste à ajouter des termes à la requête initiale pour mieux représenter le besoin en information de l'utilisateur, dans lequel nous allons d'abord définir l'expansion de requête, ensuite nous citons les ressources utilisées pour l'expansion de requêtes et les étapes du processus d'expansion automatique de requêtes, et enfin nous allons présenter les paramètres de performances de cette technique, ainsi que son utilisation dans le modèle de langue.

II.2 Définition

L'expansion de requêtes est une méthode conçue pour améliorer le processus de recherche d'information. Elle est utilisée dans le domaine de conception de SRI adaptatifs aux besoins des utilisateurs. C'est un processus qui permet de construire une nouvelle requête plus adéquate à la recherche d'information dans l'environnement du SRI, comparée à celle formulée initialement par l'utilisateur. Son principe consiste à élargir le champ de recherche pour une requête et à contenir plus de termes reliés.
Plusieurs autres définitions ont été attribuées à l'expansion de requête, Abberley**[Abberley& al, 99]** définit l'expansion de requêtes comme un moyen qui permet de reformuler les requêtes et d'améliorer le processus de recherche d'informations. Elle est considérée selon **[Salton&McGIll, 90]** comme un processus qui a pour but de préciser et d'éclaircir les résultats en permettant à l'utilisateur de modifier sa requête afin d'améliorer la pertinence de ses résultats. Efthimiadis**[Efthimiadis, 96]**, en plus de proposer des classifications des méthodes d'expansion de requêtes, il a aussi donné les définitions suivantes : « L'expansion de requêtes est un processus qui vise à compléter la requête initiale en proposant des termes supplémentaires, elle est considérée comme une amélioration de la recherche d'information ».

II.3 Classification des techniques de reformulation de requêtes

Les techniques de reformulation de requêtes peuvent être classé selon trois principaux paramètres **[Gauch & Smith, 92]**

II.3.1 Classification selon la source des termes utilisés

Différentes sources de données sont utilisées pour reformuler la requête initiale. Elles peuvent être **[Manning & al, 08]** : des ressources externes, telles que les ontologies, les thésaurus et la relation de cooccurrence entre terme dans la collection. Les méthodes basées sur ces ressources sont dites **méthodes globales,** ou des documents résultats de la première recherche ; les méthodes basées sur ces ressources sont dites **méthodes locales.** Ces méthodes sont également connues sous le nom de réinjection de pertinence. La réinjection de pertinence a montré son efficacité avec différents modèles de la RI et a affiché de meilleures résultats que les méthodes globales **[Xu & Croft,96].**

II.3.2 Classification selon la méthode de sélection des termes

Une méthode permet de sélectionner les termes à ajouter à la requête initiale, plusieurs méthodes existent, on trouve : la relation de cooccurrence, les mesures d'information, les techniques de classification, etc.

II.3.3 Classification selon le rôle de l'utilisateur

La reformulation de la requête peut être réalisée par l'utilisateur (dite manuelle), ou par le système (dite automatique) comme elle peut être réalisée conjointement par l'utilisateur et le système, dans ce cas elle est dite semi-automatique ou interactive **[Mataoui, 07]**

Cependant, le rôle de l'utilisateur dans ce processus peut être actif passif, il est actif dans la reformulation manuelle et reformulation interactive de la requête, et passif dans la reformulation automatique de la requête.

A) La reformulation Manuelle

Ce type de reformulation est associé aux systèmes de recherche booléens. On peut procéder à la reformulation de requête en utilisant un vocabulaire contrôlé (thésaurus ou classification) pour permettre à l'utilisateur de trouver les bons termes et ainsi compléter sa requête.

B) La reformulation automatique

Dans ce type de reformulation, l'utilisateur n'intervient pas. La reformulation de la requête peut être effectuée à partir d'un thesaurus, qui définit les relations entre les différents termes de l'index et permet de sélectionner de nouveaux termes à ajouter

à la requête initiale. Le thesaurus regroupe plusieurs informations de type linguistique (équivalence, association, hiérarchie) et statistique (pondération des termes). La construction du thesaurus se fait généralement pendant le processus d'indexation, et peut être automatique ou interactive. Parmi les thesaurus construits automatiquement, on peut citer un thesaurus basées sur les similarités **[Qui & Frei, 93],** un thesaurus statistique **[Grouch & Yang, 92],** ou bien des mini-thesaurus construits seulement d'après la requête et à partir de technique de clustering **[Attar & Franenkel, 77]**

Dans ce cadre de reformulation, on peut citer également la **réinjection de pertinence automatique :** c'est aussi ce qu'on appelle la **réinjection de pertinence aveugle.** Dans ce cas, on applique le même principe de la réinjection de pertinence mais en considérant les n premiers documents renvoyés par le système comme pertinents **[Croft & Harper,79],[Mitra & al, 98].**

C) La reformulation semi-automatique

La reformulation interactive de la requête est la stratégie de reformulation de la requête la plus populaire **[Rochio, 71] [Boughanem & al, 43].** On la nomme communément **réinjection de pertinence** ou **« relevance feedback »** en anglais. Dans un cycle de réinjection de pertinence on présente à l'utilisateur une liste de documents jugés pertinents par le système comme réponse à la requête initiale. Après les avoir examinés, l'utilisateur indique ceux qu'il considère pertinents. L'idée principale de la réinjection de pertinence est de sélectionner les termes importants appartenant aux documents jugés pertinents par l'utilisateur, et de renforcer l'importance de ces termes dans la nouvelle formulation de la requête.

Cette méthode a pour double avantage une simplicité d'exécution pour l'utilisateur qui s'occupe pas de détails de la reformulation, et un meilleurs contrôle du processus de recherche en augmentant le poids des termes importants et en diminuant celui des termes non importants.

II.4 Le processus d'expansion automatique de la requête

Le processus d'expansion de requêtes contient quatre principales étapes présentées dans la figure (Figure II.1) : traitement de données, génération et classement des termes, sélection des termes et reformulation de la requête. La requête initiale de l'utilisateur et

la source de données, sont considérées comme l'entrée du processus et la requête reformulée comme sa sortie.

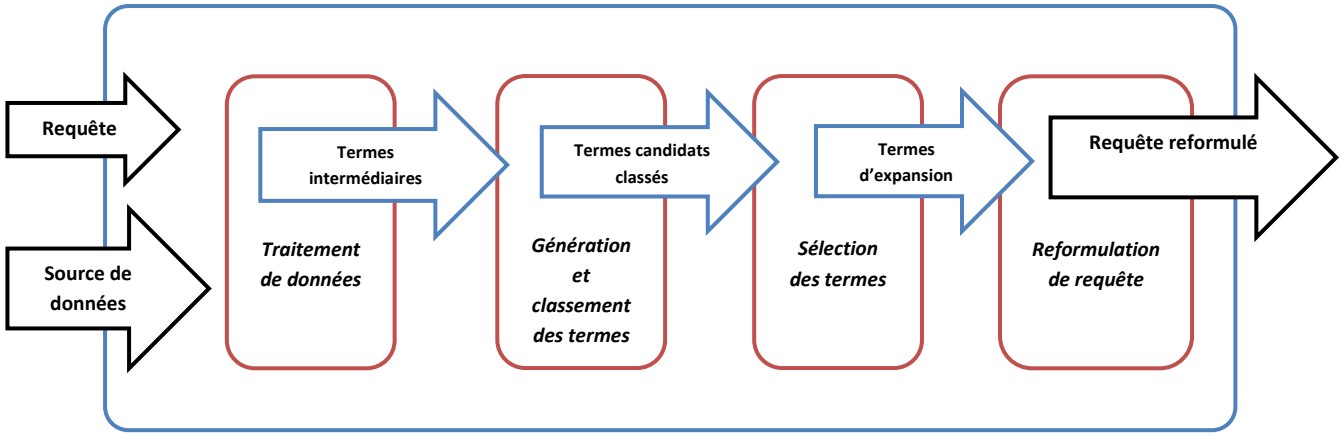

Figure II.1 Le processus d'expansion automatique de la requête

II.4.1 Traitement de données

La première étape du processus d'expansion automatique des requêtes consiste à transformer la source de données pour rendre la requête de l'utilisateur plus large dans un format qui sera traité avec plus d'efficacité par les étapes suivantes. Cette étape est souvent une phase d'extraction des termes afin de faciliter l'accès ainsi que la manipulation des fonctions de traitement.

Le prétraitement de la source de données (documents ou autres) est spécifique à son type, mais il est généralement indépendant de la requête utilisateur qui doit être étendue.

Beaucoup de techniques d'expansion se basent sur des informations extraites lors d'une réponse à une requête initiale de l'utilisateur à partir d'une collection de documents. Dans cette étape de traitement de données, il est indispensable d'indexer la collection et d'exécuter la requête.

En plus d'être représenté comme un ensemble de termes pondérés, chaque document est lui attribué un fichier complémentaire inversé de l'indice qui associe les termes du document aux termes de la requête. Le système d'indexation est aussi capable de stocker les positions des termes pour fournir la recherche basée sur la proximité. Quand un corpus externe est utilisé (des données Web pour la recherche sur intranet ou des données de bureau personnel pour la recherche sur le Web), comme dans **[Xu &Croft,**

00], **[Voorhees, 04]**, **[Diaz &Metzler, 06]** et **[Chirita& al, 07]** il est nécessaire d'utiliser un système de recherche d'information différent.
Il existe d'autres techniques d'expansion de requêtes qui se basent sur l'analyse du corpus. Elles requirent l'extraction manuelle des termes particuliers de la collection qui sont généralement différents de ceux utilisés à des fins d'indexation par un SRI classique.

II.4.2 Génération et classement des termes candidats d'expansion

La deuxième étape de l'expansion automatique de la requête consiste à générer et à classer les termes candidats d'expansion. La requête initiale et la source de données transformée servent d'entrée pour cette étape, et l'ensemble des termes d'expansions représentent sa sortie.
Le classement est important, en raison des méthodes d'expansion de requêtes qui choisissent un petit nombre de termes candidats d'expansion pour les ajouter à la requête initiale.
Les techniques utilisées pour effectuer la génération et le classement des termes candidats sont classées selon le type de relation entre les termes d'expansion générés et les termes de la requête initiale. Il existe deux types, association un-à-un et association un-à-plusieurs :

II.4.2.1 Association un à un

Dans l'association un à un entre les termes d'expansion et les termes de la requête initiale, chaque terme d'expansion est associé à un terme unique de la requête initiale. Dans la pratique, un ou plusieurs termes d'expansion sont générés et marqués pour chaque terme de la requête avec l'aide d'une variété de techniques.
L'une de ces techniques se base sur les associations linguistiques, comme l'utilisation de l'algorithme de lemmatisation qui consiste à regrouper les mots d'une même famille et les réduire en une entité appelée lemme (forme canonique).
L'approche linguistique est un moyen de calculer automatiquement la similarité terme-à-terme dans une collection de documents. Par conséquent, deux termes sont sémantiquement liés lorsqu'ils apparaissent dans les mêmes documents, tout comme deux documents qui sont considérés comme similaire ils contiennent les mêmes termes.
Deux mesures utilisées pour le calcul de similarité entre termes et qui sont le coefficient Dice (D) et l'indice de Jaccard (J).

Le coefficient Dice est définit comme suit :

$$D=\frac{2.df_{u \wedge v}}{df_u+df_v} \qquad (II.1)$$

Avec **u** et **v** deux termes, $\mathbf{df_{u \wedge v}}$ le nombre de documents contenant à la fois le terme u et le terme v et $\mathbf{df_u}$et $\mathbf{df_v}$représentent le nombre de documents qui contiennent respectivement les termes u et v.

L'indice de Jaccard est définit comme suit :

$$J=\frac{df_{u \wedge v}}{df_{u \vee v}} \qquad (II.2)$$

Avec $\mathbf{df_{u \vee v}}$ le nombre de documents contenants le terme u ou le terme v.

Une approche plus générale est définit dans ce qui suit. Si on considère une matrice A terme-document où chaque cellule $\mathbf{A_{t,d}}$ représente le poids $\mathbf{W_{t,d}}$ pour le terme t dans le document d, et si on considère une matrice C de similarité terme-à-terme alors C est calculée ainsi : $\mathbf{C=AA^t}$, où $\mathbf{C_{u,v}}$ est le degré de similarité entre le terme u et le terme v.

$\mathbf{C_{u,v}}$est donnée comme suit :

$$\mathbf{C_{u,v}}=\sum_{dj} W_{u,j}.W_{v,j} \qquad (II.3)$$

Ainsi, la corrélation entre chaque terme de la requête et chaque terme dans la collection peut être calculée à l'aide de la formule définit ci-dessus et pour introduire la notion de fréquence des termes, il est préférable de générer des facteurs de similarité normalisés, comme la mesure de cosinus donnée par la formule suivante :

$$\frac{\mathbf{C_{u,v}}}{\sqrt{\sum_{\mathbf{dj}} \mathbf{W_{u,u}^2} . \sum_{\mathbf{dj}} \mathbf{W_{v,v}^2}}} \qquad (II.4)$$

La formule (II.3) peut produire différentes méthodes de similarité terme-à-terme, en se basant sur la façon dont les documents et la fonction de pondération sont choisis.

Une technique bien connue proposée par Attar et Fraenkel dans **[Attar &Fraenkel, 77]** repose sur l'ensemble des documents retournés en réponse à la requête initiale et utilise la fréquence des termes pondérés.

La cooccurrence des termes dans l'ensemble du document est simple, mais la position des termes pose un inconvénient car elle n'est pas prise en compte, alors que deux termes qui apparaissent dans la même phrase semble plus corrélés que deux termes qui apparaissent loin l'un de l'autre dans un document. Cet aspect est généralement abordé

en considérant la proximité des termes c'est-à-dire en utilisant des documents textuels restreints tels que les documents de longueur fixe pour mesurer la similarité des termes. Cependant, la cooccurrence simple, dans un contexte grand ou petit ne signifie pas nécessairement que les termes sont similaires. L'information mutuelle est une mesure plus complète pour l'association de mot, qui intègre la dépendance entre termes **[Church &Hanks, 90]**, **[van Rijsbergen, 79]**, elle est définit ainsi :

$$I_{u,v}=\log_2\left[\frac{p(u,v)}{p(u).p(v)}+1\right] \qquad (II.5)$$

Où :**P(u,v)** est la probabilité conjointe que le terme u et le terme v apparaissent dans un certains contexte (généralement un document), et **P(u), P(v)** représentent les probabilités d'occurrence des termes u et v respectivement.

L'un des inconvénients de l'information mutuelle est sa tendance à favoriser les termes rares plus que les termes communs. Ce qui peut devenir un problème plus aigu pour les données clairsemées. Sinon, nous pourrions envisager la définition classique de la probabilité conditionnelle. Elle consiste à mesurer le degré de l'association du terme v au terme u donnée comme suit :

$$P(v,u)=\frac{p(u,v)}{p(u)} \qquad (II.6)$$

De ce fait, les règles d'associations ont été utilisées afin de trouver les termes d'expansion en corrélation avec les termes de la requête **[Latiri& al, 04]**, **[Song & al, 07]**.

II.4.2.2 Association un à plusieurs

L'association un-à-un a tendance à choisir un terme lorsqu'il est fortement lié à un des termes de la requête initiale. Dans certains cas, cela ne peut pas refléter exactement les relations des termes d'expansion à la requête dans son ensemble. Ce problème a été analysé dans **[Bai & al, 07]**. Par exemple, si le mot « programme » est fortement lié à une requête contenant le mot « ordinateur », alors l'expansion automatique pourrait fonctionner seulement pour certaines requêtes comme « programme java », « programme d'applications », mais pas pour d'autres requêtes comme « programme TV », « programme spécial ». Ici la question d'ambigüité de la langue est en partie responsable de ce problème.

L'approche de l'association un-à-plusieurs consiste à étendre l'approche un-à-un décrite dans la section précédente pour les autres termes de la requête. L'idée est d'élargir

l'association un-à-un pour les autres termes de la requête. Si un terme d'expansion est corrélé à plusieurs termes de la requête, alors 'il est corrélé à l'ensemble de la requête. La formule définit dans ce qui suit, calcule les facteurs de corrélation d'un terme d'expansion candidat **v** pour chaque terme de la requête en utilisant des corrélations terme-à-terme, puis elle combine les scores trouvés pour trouver la corrélation de la requête q globale :

$$C_{q,v}=\frac{1}{|q|}\sum_{u\epsilon q}C_{u,v} \qquad \text{(II.7)}$$

Une approche similaire a été proposée dans **[Qiu & Frei, 93]** et **[Xu &Croft, 96]**, et plusieurs autres travaux de recherche ont suivi **[Bai & al, 05]**, **[Cui & al, 03]**, **[Hu & al, 06]** et **[Sun & al, 06]**.

La formule (II.3), dans **[Qiu & Frei, 93]** est utilisée pour déterminer la similarité terme-à-terme dans toute la collection. Elle est vue comme un espace concept-terme où les documents sont utilisés pour extraire les termes d'indexation.

Dans un document, le poids du terme est donné comme étant le produit de la fréquence du terme dans le document par la fréquence inverse du terme associé à ce document, et la fréquence inverse du terme d'un document $\mathbf{d_j}$ est exprimée par $\mathbf{Log}\ \frac{T}{DT_j}$ avec **T** qui représente le nombre de terme dans la collection et $\mathbf{DT_j}$ le nombre de termes distincts dans le document d_j. L'inverse de la fréquence du terme est similaire à celle du document utilisée pour le classement du document.

la formule (II.3) est utilisée pour calculer la corrélation d'un terme-concept (plutôt qu'une corrélation terme-à-terme), où $w_{u,j}$ est la fréquence du terme de la requête dans le j-ième passage et $w_{v,j}$ est la fréquence du concept c dans le j-ième passage. Un concept représente un groupe de mots adjacents (proches) dans les documents les plus recherchés, et le passage est une portion de texte de taille fixe, utilisée pour analyser les concepts candidats. Le calcul de la valeur de corrélation terme-concept exacte est basé sur la fréquence inverse du terme et du concept dans les passages contenus dans l'ensemble de la collection.

L'approche un-à-plusieurs est aussi basée sur une combinaison de relations multiples entre les paires de termes dans un cadre de la chaine de Markov **[Collins-Thompson &Callan, 05]**. Un réseau d'expression est construit pour chaque requête, il contient des paires de mots qui sont reliés par des types de relations, comme des occurrences, des

synonymes, des radicaux et les probabilités de transitions. Ces relations peuvent être générées à partir de plusieurs sources.
Ensuite les mots avec une forte probabilité de pertinence sont sélectionnés en tant que termes d'expansions.
En tenant compte des données rares, l'approche un-à-plusieurs est plus robuste.
Cette approche prend en charge des données complexes impliquant des chaines de termes. Elle présente des limitations des relations entre les termes simples, et afin de les surmonter, on peut voir la requête comme une expression, et ensuit chercher des phrases qui lui sont liées. Les phrases sont caractérisées par un contexte riche et une moindre ambigüité que leurs mots constructifs même si une l'évaluation de similarité au niveau de la phrase ne peut pas être simple.

II.4.3 Sélection des termes :

Après avoir classé les termes candidats. Dans la deuxième étape, les principaux éléments (termes) sont sélectionnés pour l'expansion de la requête.
Les termes avec une probabilité supérieure à un certain niveau peuvent être sélectionnés seulement quand les scores des termes sont considérés comme des probabilités.
Plusieurs techniques pour la sélection des termes ont été proposées, elles utilisent différentes informations et pas seulement que les poids attribués aux termes candidats.
Une de ces techniques utilise plusieurs fonctions de classement de termes, et sélectionne pour chaque requête les termes les plus courants.
Une autre stratégie consiste à choisir une quantité variable de termes d'expansion en fonction de la difficulté de la requête.
Dans **[Cao& al, 08]** les auteurs utilisent un classificateur afin de distinguer entre la pertinence et le non pertinence du classement des termes d'expansion. Pour apprendre les paramètres du classificateur, un ensemble d'informations est créés dans lequel les termes simples sont étiquetés comme bons ou mauvais selon leurs influences sur les résultats de la recherche.
La sélection des meilleurs termes d'expansion pour une requête donnée est vue comme un problème d'optimisation. Cette étape de sélection des meilleurs termes sera discutée dans les paramètres de performance d'expansion de la requête.

II.4.4 La reformulation de la requête

C'est la dernière étape du processus d'expansion automatique de la requête. Elle décrit la requête élargie qui sera soumise au système de recherche d'information, ce qui revient généralement à affecter un poids à chaque terme qui décrit la requête étendue appelé « pondération de la requête ».

Il existe plusieurs techniques de pondération de requêtes, la plus populaire est reproduite à partir de la formule de Rocchio pour la réinjection de pertinence **[Rocchio, 71]**.

La formule générale est donnée comme suit :

$$W'_{t,q'} = (1-\lambda).w_{t,q} + \lambda.score_t \qquad (II.8)$$

Où :**q'** est la requête étendue, **q** est la requête originale, **λ** représente le paramètre de pondération, et $\mathbf{score_t}$ représente le poids attribué au terme d'expansion t.

Les poids basés sur les documents pour la requête qui n'est pas étendue et les scores en fonction de différence de distribution utilisés pour les termes d'expansion possèdent de différentes échelles, alors leurs valeurs doivent être normalisées avant de les additionner dans la formule (II.8).

De ce fait, plusieurs techniques de normalisation simples, discutées dans **[Wong & al, 08]**, ont été proposées, elles produisent des résultats comparables en général.

Afin d'optimiser les performances, la valeur de **λ** peut être ajustée si les données sont disponibles. Donner plus d'importance par exemple aux termes de la requête initiale deux fois plus que les termes d'expansion. Ou autres possibilités comme la technique suggérée dans **[Amati, 03]** et qui consiste à utiliser une formule de pondération de requêtes sans paramètres.

La formule ci-dessus peut être utilisée dans le cas où les termes sont extraits d'un thésaurus ou de WordNet.

Les pondérations peuvent être fondées sur des critères multiples comme le nombre de termes, le nombre de cooccurrence, la longueur du document et le type de la relation.

L'étape de pondération de termes est naturellement prise en charge si l'approche de modélisation du langage effectue le classement des documents.

Dans la formule suivante, le modèle de requête est estimé généralement, en ne considérant que des mots de la requête initiale :

$$\text{Sim}(q,d) \propto \sum_{t \epsilon v} P(t|\theta_q) \log \frac{P(t|\theta_q)}{P(t|\theta_d)} \qquad \text{(II.9)}$$

Tandis que le modèle de document est estimé en considérant également les mots invisibles :

$$P(t|\theta'_d) = (1-\lambda).P(t|\theta_d) + \lambda.P(t|\theta_c) \qquad \text{(II.10)}$$

Où : $\boldsymbol{\theta_d}$ est le modèle de document et $\boldsymbol{\theta_c}$ est le modèle de la collection.

Ainsi la question qui se pose est de savoir s'il est possible de créer un meilleur modèle de requête en trouvant les mots connexe avec leurs probabilités associées et ensuite d'utiliser le modèle de requête correspondant pour unir le modèle de la requête initiale de la même manière que le modèle de document est lissé avec le modèle de la collection. De ce fait, plusieurs méthodes ont été explorées. Elles sont basées non seulement sur les documents feedback **[Lavrenko&Croft, 01]**,**[Zhai&Lafferty, 01]**(présentées dans la section 6), mais également sur les relations entre les termes **[Bai & al, 05]**.

Indépendamment de méthode de génération spécifique, le modèle final étendu est donné par la formule suivante :

$$P(t|\theta'_q) = (1-\lambda).P(t|\theta_q) + \lambda.P(t|\theta_{QEM}) \qquad \text{(II.11)}$$

Où : $\boldsymbol{\theta_q}$ est le modèle de la requête initiale

Cette expression peut être considérée comme une généralisation de la formule (8).

La pondération de la requête est la même dans le processus d'expansion automatique de requête, mais elle n'est pas toujours assuré.

Une approche alternative simple consiste en l'augmentation du nombre de termes qui décrivent la requête sans effectuer la pondération de requête. Une autre approche vise à augmenter le nombre de termes de la requête, et ensuite appliquer une version modifiée de la fonction de pondération utilisée par le système de classement pour explicitement traiter avec les fonctionnalités d'expansion.

II.5 Les paramètres de performance d'expansion de la requête

Un nombre considérable d'expérimentations ont été effectuées sur les collections de documents pour l'évaluation de l'impact induit par la reformulation de requête sur le processus de recherche d'information. Une étude synthétique de ces différents travaux annonce que l'ordre des performances imputées à l'intégration de l'expansion de requêtes est variable, et elle dépend de plusieurs paramètres:

- ✓ Le modèle de recherche utilisé
- ✓ L'hypothèse de base quant à la distribution des termes dans les documents, concept, phrases et relation sémantique entre eux.
- ✓ Les caractéristiques des collections de documents : nombre, taille, etc.

En faisant abstraction des paramètres caractéristiques inhérents à chacune des techniques de reformulation de requêtes présentées, les auteurs de l'étude ont dégagés , les paramètres de performance intrinsèques suivant :

II.5.1 Nombre de termes ajoutés à la requête

La pertinence est expérimentée par **[Buckley & al, 94]** dans l'environnement TREC. Ils ont montré que le taux de performance est d'avantage corrélé avec le nombre de termes ajoutés qu'avec le nombre de documents initialement retrouvés.
Ils ont abouti à la mise au point de l'équation de variation :

$$RP(N) = A.\log(N_S) + B.\log(X) + C \qquad (II.12)$$

Où :**RP (N):** Performance du système pour N documents restitués

N_S: Nombre de documents restitués

X: Nombre de termes ajoutés à la requête

A, B, C: Constantes

Ils ont conclu que le seuil critique du nombre de terme à ajouter à la requête dépend des caractéristiques de la collection.
Par ailleurs, Harman **[Harman, 92]** a montré que la meilleure méthode de sélection des termes issus des documents pertinents devient inefficace après l'ajout de *20* à *40* termes à la requête initiale, sur des bases de tailles moyenne (CACM, Cranfield ,etc.).

II.5.2 Méthode de sélection des termes :

La méthode de sélection des termes à ajouter à la requête est aussi importante que le choix de leur seuil. Nous citerons les principales méthodes expérimentées.

- Salton et Buckley **[Salton, Buckley, 90]** ont expérimenté séparément, l'ajout de tous les nouveaux termes, tous les termes issus des documents pertinents et les termes les plus fréquents dans les documents restitués à la requête initiale.

L'expansion de la requête avec tous les nouveaux termes offre de meilleurs résultats que les autres méthodes, toutefois l'écart de performance n'est pas très considérable relativement aux exigences de temps et d'espace mémoire.

Robertson **[Robertson & al, 95]** et Haines **[Haines &Croft, 93]** adoptent une méthode de sélection de nouveaux termes sur la base d'une fonction qui consiste à attribuer pour chaque terme un nombre traduisant sa valeur de pertinence. Les termes sont alors triés puis sélectionnés sur la base d'un seuil.

Robertson propose la formule suivante pour le calcul de la valeur de sélection d'un terme :

$$SV(i) = w(P_i - U_i) \qquad \text{(II.13)}$$

Où:

$$w = log\ \frac{P_i(1-U_i)}{U_i(1-P_i)} \qquad \text{(II.14)}$$

Avec **:**
P_i : probabilité (d_i =1/D est Pertinent)
U_i : Probabilité (d_i =1/D est Non Pertinent)

- Harman **[Harman, 92]** propose les fonctions suivantes :

1) $$SV(i)=\frac{RT_j * df_i}{N} \qquad \text{(II.15)}$$

Où : $\boldsymbol{RT_j}$ **:** Nombre total de documents retrouvés par la requête

$\boldsymbol{df_i}$: Fréquence d'occurrence du terme t_i dans la collection.

$\boldsymbol{N}$ **:** Nombre total de documents dans la collection.

2) $$SV(i) = \frac{r_i}{R} - \frac{df_i}{N} \qquad \text{(II.16)}$$

Où :
$\boldsymbol{r_i}$ **:** Nombre de documents pertinents contenant $\mathbf{t_i}$
$\boldsymbol{R}$ **:** Nombre de documents pertinents.

3) $$SV(i) = \log_2 \frac{P_i(1-q_i)}{(1-p_i)} \qquad \text{(II.17)}$$

Avec : $\boldsymbol{p_i}$ **:** Probabilité que t_i appartienne aux documents pertinents

$\boldsymbol{q_i}$ **:** Probabilité que t_i appartienne aux documents non pertinents

Les expérimentations réalisées sur différentes collections standards, ont révélé que la troisième fonction est la meilleure.

II.5.3 Longueur moyenne de requête :

L'accroissement des performances est plus important lorsque les collections sont interrogées par des requêtes de longueur relativement petite **[Buckley & al, 94]**. Dans ce sens, des expérimentations intéressantes ont été réalisées sur la base TREC7 et présentées dans **[Cormack& al, 99]**. Les auteurs montrent en effet que la dérivation automatique de courtes requêtes à partir de documents jugés ou supposés pertinents à la suite d'une recherche initiale, permettent d'atteindre des résultats très performants pour différentes tâches : recherche, filtrage et routing.

II.6 L'expansion de la requête dans le modèle de langue :

La modélisation de la fonction de correspondance dans le modèle de langue part d'un principe différent des approches traditionnelles de la recherche d'information. On ne tente pas de modéliser directement la notion de pertinence dans ce modèle, mais on considère que la pertinence d'un document vis- à vis d'une requête est en rapport avec la probabilité que la requête puisse être générée par le modèle de langue du document. On suppose en effet, qu'à chaque document vis-à-vis d'une requête q la pertinence est déterminée par la probabilité de génération de la requête sachant le modèle de ce document, soit $P\ (q|M_D)$.

La plupart des modèles de langue développés pour la recherche d'information se base sur ce principe de génération de la requête par un modèle de document, il existe d'autres variantes qui sont discutées dans ce qui suit.

Approche d'exploitation des modèles de langue en RI :

En se basant sur la représentation des documents, les approches de modélisation de langue pour la RI peuvent être classés en catégories :

- ✓ Génération de document à partir de modèle de la requête (Document Likelihoodmodels) : pour cette approche se procède dans le sens inverse. Ainsi un modèle de langue est associé à la requête, les documents sont alors classés selon leurs probabilités que leur contenu soit généré par le modèle de la requête, soit $P\ (d|M_q)$

- ✓ Similarité document-requête : Dans cette catégorie, on considère qu'à chaque document et requête est associée à un modèle de langue. Les documents sont alors ordonnés selon la similarité de leur modèle avec celui de la requête.

L'expansion de requêtes est modélisée dans le modèle de langue suivant l'une de ces deux approches :

A) Génération de document à partir du modèle de la requête (Document Likelihoodmodels) :

Au lieu de modéliser la RI comme processus de génération de la requête, Lavrenko et Croft**[Lavrenko&Croft, 01]** ont proposé de modéliser explicitement le modèle de pertinence. Ils ont en effet proposée d'estimer ce modèle à partir du modèle de la requête sans utilisation de données d'entrainement, en faisant le parallèle avec la modélisation de la pertinence proposée dans le modèle probabiliste classique. Ils considèrent en effet, que pour chaque requête, il existe un modèle permettant de générer le sujet (thème) abordé par la requête, c'est ce que les auteurs appellent le modèle de pertinence ($\boldsymbol{\Theta_R}$).
Le but est alors d'estimer la probabilité $\mathbf{P(t|\Theta_R)}$, de générer un terme à partir du modèle de pertinence. Comme le modèles de pertinence n'est pas connu, les auteurs ont suggérés d'exploiter les documents retournés les mieux classés (feedback document) en assument qu'ils sont générés à partir du modèle de pertinence.
Le modèle de pertinence est exprimé comme suit :

$$P(t|\Theta_R) = \sum_{d \in R} P(d)P(t|d) \prod_{i=1}^{k} P(q_i|d) \qquad \text{(II.18)}$$

Avec :

- ✓ **R** : dénote l'ensemble de documents feedback.
- ✓ **P(d)** : La probabilité de choisir un document **d** des tops documents pertinents retournés.

Ainsi, le modèle de pertinence obtenu est une combinaison pondérée du modèle individuel de chaque document feedback **P (t|d)** avec le score de ce document vis- à- vis de la requête $\mathbf{P(q_i|d)}$.

Les résultats des expérimentations ont montré que cette approche améliore sensiblement les performances de la recherche d'information, de +10% à +29% d'amélioration de précision moyenne par rapport au modèle de langue de base.**[Hammache, 13].**

B) Comparaison des modèles de requête et du document :

Cette approche est inspirée de l'approche vectorielle de la RI, dans laquelle la requête et le document sont représentés sous forme de vecteurs, la pertinence est alors interprétée par la similarité des vecteurs associés à la requête et au document.

Lafferty et Zhai**[Lafferty&Zhai, 01]**ont proposé un cadre de minimisation de risque basé sur la théorie de décision bayésienne. Dans ce cadre la requête et les documents sont modélisés par des modèles statistiques. Le modèle de la requête modélise entre autre les préférences et les besoin des utilisateurs, le modèle de document permet de modéliser le processus de génération de document et du capturer les préférences de l'auteur.

La similarité entre document et requête est mesurée par la fonction de divergence de Kullback –Leiber (KL) entre le modèle de la requête (P(t|q)) et celui de document (P(t|d)) **[Lafferty&Zhai,01]**, exprimé comme suit :

$$\text{Score (q, d)} = -\text{KL (q||d)} = -\sum_{t \epsilon V} p(t/q) \log \frac{P(t|q)}{P(t|d)} \qquad \text{(II.19)}$$

Où : V le vocabulaire d'index.

II.7 Conclusion

Ce chapitre a porté essentiellement sur l'expansion de requêtes, d'abord nous avons présenté Le processus d'expansion automatique de la requête . Ensuite, nous avons détaillé les principales étapes du processus d'expansion automatique de la requête ainsi que ses paramètres de performance. Enfin, nous avons présenté l'expansion de la requête dans le modèle de langue. L'une des étapes les plus importantes de succès de l'expansion de la requête est la méthode de sélection des termes d'expansion qui est basée sur le choix des documents dans lesquels on extrait les termes.

Dans le chapitre suivant, nous présentons une nouvelle approche de sélection des documents pour l'expansion de requête ainsi que les résultats obtenus de l'évaluation.

III.1 Introduction

Nous avons étudié dans le deuxième chapitre les différentes approches pour l'expansion automatique de requêtes, dans ce chapitre nous allons présenter une nouvelle approche de sélection des documents dans lesquels les termes d'expansion sont extraits. Ainsi le chapitre suivant est structuré comme suit :

Dans la première section, nous décrivons l'approche proposée. Dans la seconde section, nous abordons l'expérimentation de l'approche en précisant les outils et langage utilisées pour sa mise en œuvre. Enfin nous présentons quelques résultats expérimentaux obtenus sur la collection de test TREC AP88.

III.2 Présentation de l'approche proposée

III.2.1 Principe de l'approche

Le besoin en information de l'utilisateur est exprimé par une requête qui est exécutée par le système de recherche d'information. Dans certains cas, cette requête n'est pas complète et claire, alors elle retourne des résultats qui ne sont pas satisfaisants.

Pour remédier à ce problème, nous utilisons l'expansion de requête pour améliorer la requête initiale en ajoutant des termes générés , classés et choisis pour reformuler enfin cette requête. Cependant, il existe des requêtes non améliorées avec cette méthode classique et cela est dû au fait que ces requêtes traitent de plusieurs thématiques.

Pour remédier à ce problème, nous proposons une approche qui se base sur le principe contenant les éléments suivants :

1. Une requête multithématique est une requête ambigüe. Cette requête nécessite alors de la diversification dans le choix des termes d'expansion, ce qui implique l'obligation d'effectuer un travail de regroupement (clustering) des documents pseudo-pertinents afin de capter toutes les thématiques de cette requête. Une fois que ce travail est effectué, on sélectionne les documents d'expansion à partir des clusters obtenus.
   Dans notre cas, nous avons utilisé l'algorithme de KMeans pour le clustering. Son principe consiste à diviser des données en plusieurs sous-ensembles. Ces sous-ensembles sont formés par des données qui se ressemblent au sens d'un critère de similarité (dans notre cas c'est la similarité entre documents).
2. Si au contraire la requête n'est pas multithématique, alors elle est claire, donc les premiers documents pseudo-pertinents traitent de la thématique de la requête.
3. Sur la base des deux éléments énoncés précédemment, on établit un seuil pour l'ambiguïté de la requête pour spécifier quelle source (documents d'expansion) à utiliser. Ce seuil est obtenu par expérimentation.

Enfin le calcul du score de clarté (l'ambiguïté) de la requête est obtenu par la formule suivante :

$$\text{Score de clarté} = \sum_{i=1}^{n} P(t_i|M) * \log_2 \frac{P(t_i|M)}{P(t_i|C)} \qquad \text{(III.1)}$$

Où **n** représente tous les termes des top documents « **M** » et $\mathbf{P(t_i|M)}$est la fréquence globale relative du terme « t_i » dans ces top documents. $\mathbf{P(t_i|C)}$est la fréquence globale relative du terme « t_i » dans la collection « **C** ».

III.2.2 Formalisation de l'approche

Nous présentons dans cette section en premier, l'emplacement de notre approche dans le SRI, et les algorithmes mis en œuvre pour l'implémenter.

III.2.2.1 Emplacement de notre approche dans un SRI

La figure suivante illustre l'emplacement notre approche dans un SRI.

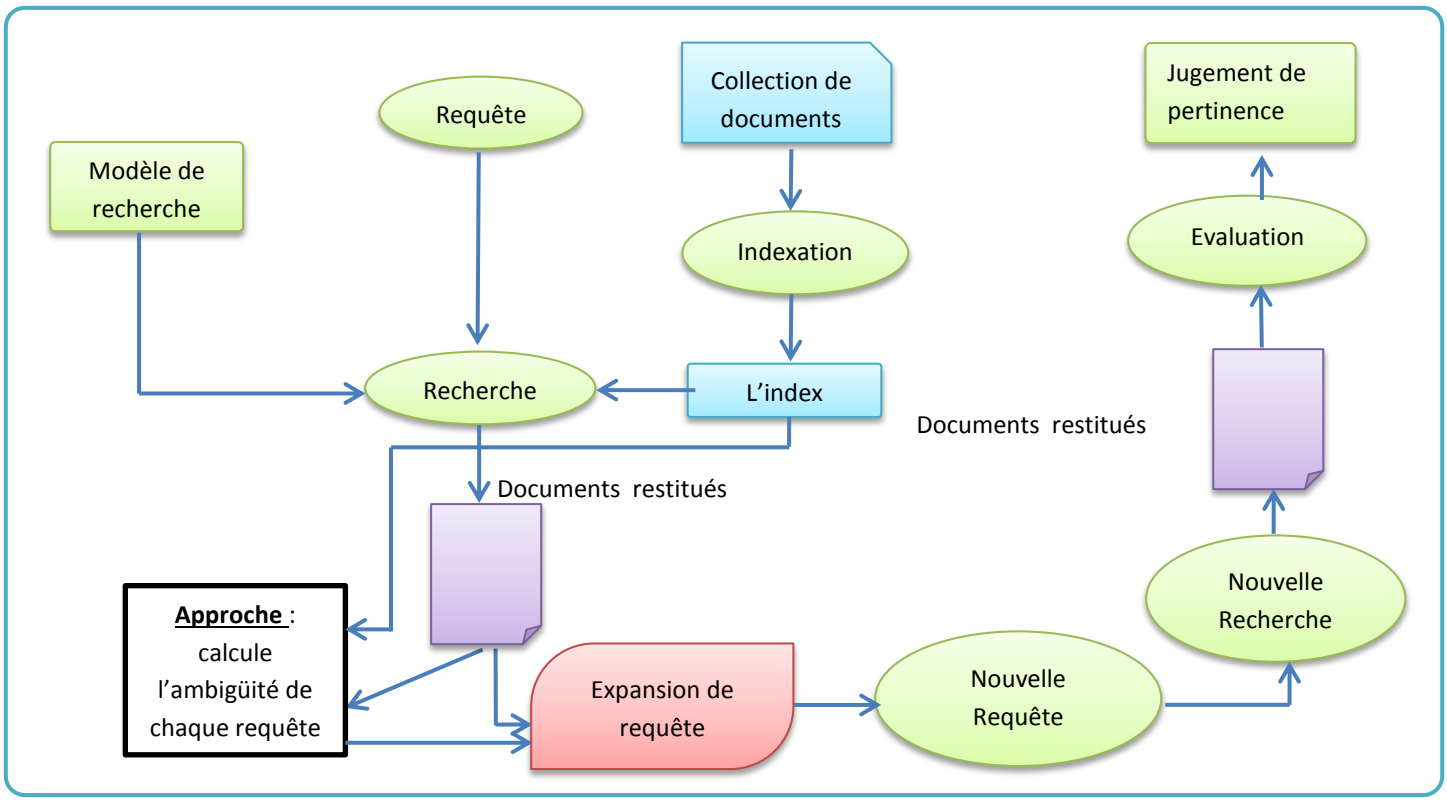

Figure III.1 Emplacement de notre approche dans un SRI

III.2.2.2 Processus d'expansion avec notre approche

Nous schématisons le processus d'expansion de requête avec notre approche dans la figure suivante :

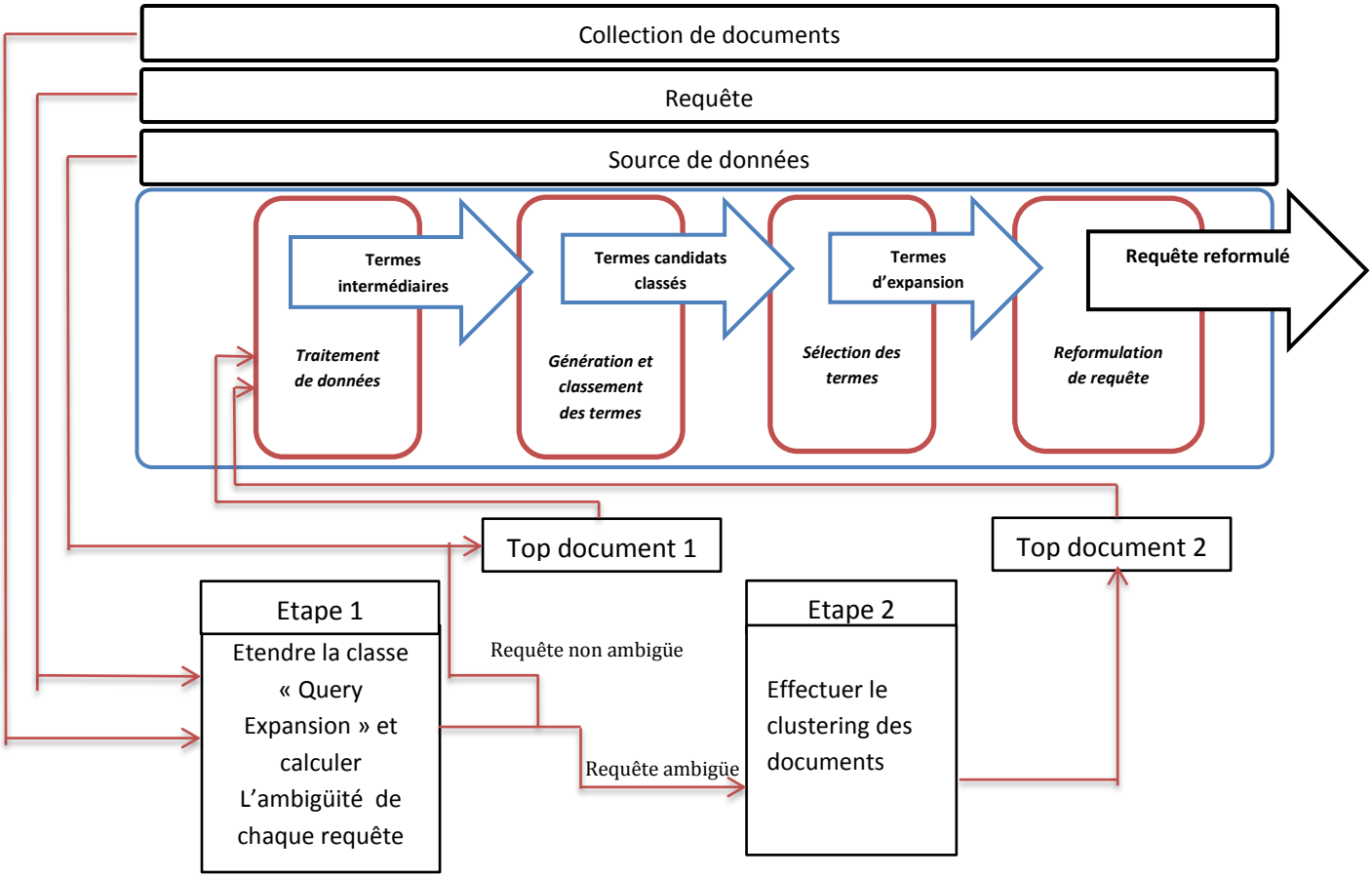

Figure III.2 Processus d'expansion avec notre approche

Afin de réaliser notre approche, nous avons utilisé deux principales étapes comme illustrer dans la figure III.2:

Etape 1 :

La première étape consiste à modifier la classe « Query Expansion » de la plate forme Terrier (présente dans la section III.3) pour calculer l'ambigüité de chaque requête et pour ensuite choisir la méthode de classement des documents à utiliser. Nous présentons dans l'algorithme ci-dessous, la manière dont il choisit la méthode de classement des documents :

Algorithme de calcul de l'ambigüité :

Entrées :Top_Document[ ] *//Vecteur des tops documents*

Sorties : score_clarté *//Le score de clarté de chaque requête*

Début

//Remplir la table

Table <clé, valeur>

Pour (i=0 jusqu'à taille de Top_Document)

Récupérer toutes les informations sur les termes du document i dans la matrice Id_terms[ ][ ]

Pour (j=0 jusqu'à taille Id_terms[0])

Mettre Id_terms[0][j], Id_terms[1][j] dansTable*//Id_terms[0][1] comme premier argument*

// et Id_terms[j][1] comme deuxième argument

Fin pour

Fin pour

// Calculer le score de clarté

Pour (i=0 jusqu'à taille de Table)

Récupérer la fréquence globale du terme i dans top_documents notée **TFdp***// de Table*

Récupérer la fréquence globale du terme i dans la collection de document notée **TFg**

// Calculer la fréquence relative du terme dans la collection notée $P(t_i|C)$

$P(t_i|C)$ = TFg/Taille de la collection *// C représente la collection*

// Calculer la fréquence relative du terme dans top document $P(t_i|M)$

$P(t_i|M)$=TFdp/Taille-totale –de-Top_Document*// M représente les premiers documents de top document*

// Calculer le score de clarté

Score_clarté += $P(t_i|M) * \log_2(P(t_i|M)/P(t_i|C))$

Fin pour

Fin

L'étape 2 est exécutée si ce score de clarté calculé est inférieur à un seuil définit par expérimentation.

Etape 2 :

La seconde étape est effectuée si la requête est ambigüe. Son rôle est de classer les tops documents dans des clusters, ensuite de sélectionner les documents d'expansion à partir de ces clusters.

Entrées :Top_Document[ ] *//Vecteur des tops documents*

Nbr_documents *// nombre de documents à utiliser dans le clustering*

Nbr_clusters *// nombre de clusters considérés*

Sorties : Top_Document2[ ] *// Vecteur de classement des tops documents obtenus après le clustering*

Début

// Calculer la similarité des documents

Pour (i=0 jusqu'à taille de Nbr_Document)

Récupérer les informations sur les termes du Top_Document [i] dans **M_terme1[ ][ ]**

Récupérer Id_term1[i] de M_terme[0]*//les identificateurs des termes pour top_document[i]*

Récupérer Tf_term1[i] de M_terme[1] noté **tf1***// les fréquences des termespour top_document[i]*

Pour (j=i+1 jusqu'à taille de Nbr_Document +1)

Récupérer les informations sur les termes du Top_Document [j]**dansM_terme2[ ][]**

Récupérer Id_term2[j] de M_terme2[0]*//les identificateurs des termes pour top_document[i]*

Récupérer Tf_term2[j] de M_terme2[0] noté **tf2***// les fréquences des termes pour top_document[i]*

Pour (i=0 jusqu'à taille de Id_term1)

Pour (i=0 jusqu'à taille de Id_term2)

Si Id_term1 est similaire à Id_term2 **alors**

Récupérer l'Idf des termes dans **idf**

// Calculer la similarité entre top_document(i) et top_document(j)

Similarité+=((tf1+tf2)*idf)/((tf1*idf)^2+(tf2*idf)^2)

Fin Si

Fin pour

Fin pour

(*) (*) Fin pour Mettre la similarité et identifiants de top_document(i) dans un Fichier nomé **similarité .txt**

```
(*)  (*)
    Fin pour

        // Classer les documents dans des clusters selon leurs similarité

    Lire le fichier et récupérer la similarité entre  documents

    Effectuer le clustering des documents

        // Parcourir la matrice des culsters obtenu (nbr_clusters * la similarité des documents) pour remplacer la similarité des

        // Documents par leurs identifiants

    Pour (i=0  jusqu'à taille de Matrice_Clusters)
        Remplacer la similarité des documents par leurs identifiants
    Fin pour


    Pour (i=0  jusqu'à taille de Matrice_Clusters)
        Transformer  Matrice_Clusters en vecteur  Top_Document2
    Fin pour
Fin
```

III.3 Expérimentation

Dans ce qui suit nous allons présenter initialement notre environnement technique et préciser les différents outils utilisés en commençant par présenter la plateforme Terrier, le langage de programmation JAVA et Netbeans. Ensuite, nous passerons à la présentation des résultats expérimentaux obtenus et à l'évaluation et la comparaison de notre approche avec le modèle de base de la RI (TF-IDF) d'une part, et le modèle de l'expansion de requête de base d'autre part.

III.3.1 Présentation de la plateforme Terrier

Terrier, TERabyteRetrlEveR : est un SRI robuste et efficace, développé par le département Informatique de l'université Glasgow de Scotland. Il est utilisé avec succès dans :

- La recherche ad-hoc
- La recherche sur le web
- La recherche multilingue

Terrier offre une plateforme idéale destinée à l'indexation de volumes importants de documents : jusqu'à 25 millions de documents. C'est un logiciel Open Source écrit en Java.

Comme tous SRI, terrier permet :

- L'indexation classique : Extraction des mots clés des documents appartenant à une collection et les stocker dans un index.
- La recherche des documents pertinents pour répondre aux requêtes formulés par l'utilisateur.
- L'évaluation des résultats de la recherche.

La Figure III.3 montre l'architecture générale de Terrier :

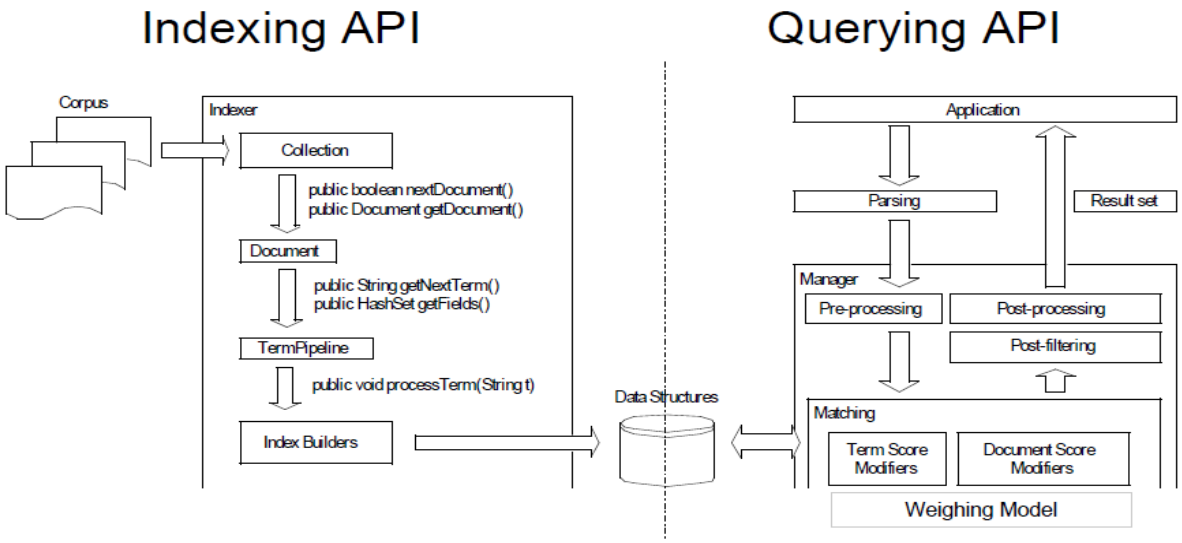

Figure III.3 Vue d'ensemble de l'architecture Terrier

L'architecture de la plate-forme Terrier distingue les deux phases classiques : l'indexation et la recherche. Un corpus documentaire est fourni en entrée au module d'indexation. Les documents de la collection passent par un ensemble de prétraitements tels que la tokenisation. Les tokens sont ensuite injectés dans une chaîne de traitementsTermePipLines, à savoir le StopWordsPipeLine pour l'élimination des mots vides de sens, ou encore les Stemming pipeline et qui dépendent de la langue en question. La phase d'indexation conduit à la construction de l'index (Data structures)

La phase de recherche comprend le Manager, un module qui interagit avec l'application, réalise la mise en correspondance à travers les calculs des pondérations (selon le schéma de pondération (Weighting Model) choisi: PL2, TF-IDF, BM25, etc.) ainsi que les scores des documents. Le résultat sont ensuit renvoyé à l'utilisateur qui sont la liste des documents jugés pertinents et leurs scores respectifs

III.3.2 Langage Java

Le langage java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au Sun World.

La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais java. Une de ses plus grandes forces est son excellente portabilité : une fois votre programme crée, il fonctionnera automatiquement sous Windows, Unix, GNU/Linux, etc. On peut faire de nombreuse sorte de programmes avec Java :

- Des applications, sous forme de fenêtre ou de console ;
- Des applications pour appareils mobiles, avec J2ME ;
- Des applets, qui sont des programmes Java incorporés à des pages web.

III.3.3 Présentation de Netbeans

Netbeans est un projet open source fondé par Sun Microsystems en juin 2000, et c'est un environnement de Développement Intégré (IDE) qui permet d'écrire, compiler, déboguer et déployer des programmes. Il est écrit en Java- mais peut supporter n'importe quel langage de programmation comme C, C++, PHP, etc.
NetBeans est gratuit et disponible sous Windows, Linux, Solaris, etc. Son installation nécessite l'installation de la JDK (Java Developpement Kit) le Kit de Développement java compatible avec la version d'IDE.
Pour la réalisation de notre système nous avons utilisés netbeans de version 8.0.2
La figure suivante illustre l'environnement de développement Netbeans :

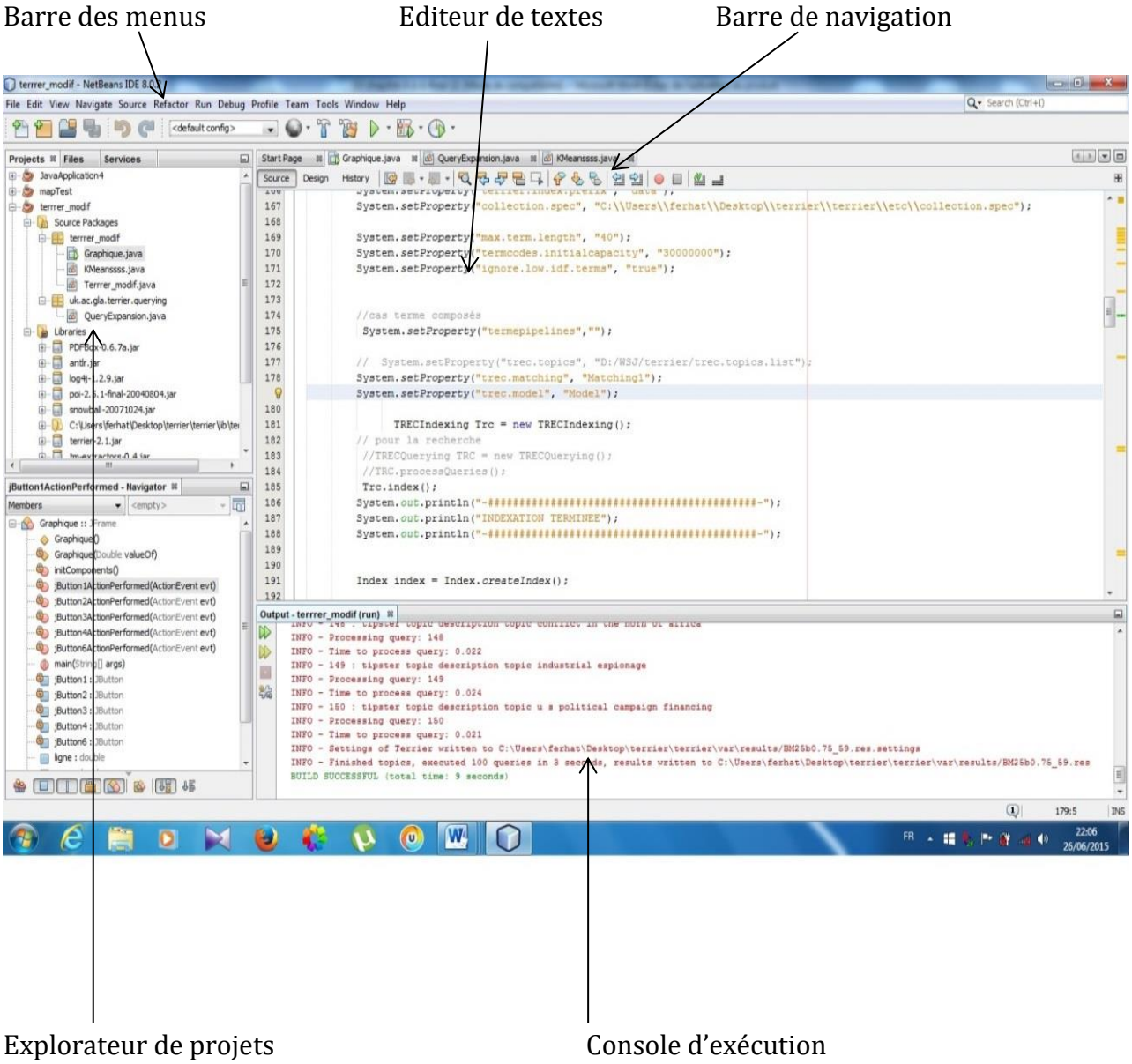

Figure III.4 Environnement de développement de Netbeans

III.3.4 Collection de test utilisée

Nous avons mené nos expérimentations en utilisant la collection de test TREC *AP88* (Associated Pressnewswire, 1988).
Pour la recherche nous avons utilisé 50 requêtes issues des topics numérotées « 51-100 » de la collection TREC.

III.3.5 Les résultats obtenus

III.3.5.1 Evaluation et résultat

Nous allons faire une comparaison entre les résultats obtenus avec la recherche simple (TF-IDF) et les résultats de notre approche. Et pour l'évaluation de ces résultats, nous avons utilisé la mesure MAP (Moyenne AveragePrecision) présentée dans le chapitre I.

III.3.5.1.1 Résultats obtenu avec la recherche simple

Dans cette étape nous avons lancé la recherche simple avec le modèle TF-IDF.

Recherche simple	MAP
Modèle avec TF-IDF	0.1188

Tableau III.1 Résultats obtenu avec la recherche simple (avec BM25, TF-IDF)

Le tableau ci-dessus résume les résultats de l'évaluation obtenus avec la recherche simple en utilisant le modèle TF-IDF.

III.3.5.1.2 Résultats obtenu avec le modèle de pertinence (de base)

Dans cette étape nous allons présenter les résultats obtenus avec le modèle de pertinence , c'est-à-dire après une recherche simple effectuée avec le modèle TF-IDF, on effectue une expansion de requête en utilisant le modèle « Bo1 » qui utilise les tops documents retournés par la recherche simple pour extraire les termes d'expansion et reformuler les requêtes.

L'expansion de requête dépend de deux paramètres, nombre de documents à utiliser pour extraire les termes d'expansion noté (DE), et le nombre de termes à ajouter à la requête initiale noté (TE).

Nous avons fait varier la valeur de « DE » de 1 à 35 avec un pas de 5, et pour chaque valeur de ce paramètre, nous avons varié la valeur de « TE » de 1 à 30 avec un pas de 5.

Et pour chaque couple de valeur des deux paramètres, nous avons effectué des tests.

Les résultats de ces tests sont illustrés dans le tableau suivant :

Nombre de documents	Nombre de termes	MAP	Nombre de documents	Nombre de termes	MAP
1	1	0.1447	20	1	0.2128
	5	0.1447		5	0.2124
	10	0.1447		10	0.2086
	15	0.1447		15	0.2060
	20	0.1447		20	0.2044
	25	0.1447		25	0.2073
	30	0.1447		30	0.2073
5	1	0.1990	25	1	0.2138
	5	0.1992		5	0.2136
	10	0.2025		10	0.2177
	15	0.2046		15	0.2176
	20	0.2081		20	0.2172
	25	0.2087		25	0.2151
	30	0.2091		30	0.2156
10	1	0.1833	**30**	1	0.2130
	5	0.1833		5	0.2123
	10	0.1918		10	0.2211
	15	0.1934		15	0.2205
	20	0.1942		20	0.2188
	25	0.1949		25	0.2188
	30	0.1965		**30**	**0.2215**
15	1	0.1977	35	1	0.2159
	5	0.1976		5	0.2158
	10	0.1950			
	15	0.1959			
	20	0.1968			
	25	0.1971			
	30	0.1968			

Tableau III.2 Résultats obtenus avec le modèle de pertinence basé sur TF_IDF

Le tableau ci-dessus résume les résultats obtenus avec le modèle de pertinence en utilisant le modèle TF-IDF, dans lequel nous constatons que la meilleure précision avec

le modèle **TF-IDF** est de**0.2215**, elle est obtenue avec un nombre de document égal à **30** et un nombre de termes est égal à **30**.

III.3.5.1.3 Résultats obtenu avec notre approche

Dans le tableau suivant, nous allons présenter les résultats obtenus avec notre approche en utilisant le modèle de pondération **TF-IDF** avec la meilleure configuration en terme de nombre de documents « DE » et nombre de termes d'expansion « TE » obtenus précédemment et qui ont les valeurs suivante : DE=30, TE=30

Notre approche à deux paramètres, le premier est nombre de clusters noté (NBC) et le second est le seuil de clarté de la requête noté (seuil)

En faisons varier des valeurs allant de **2** jusqu'à **11** avec un pas de **3**pour le premier paramètre « NBC »et des valeurs allant du minimum au maximum des score de clarté des requêtes utilisés pour le second paramètre «seuil », nous obtiendrons le tableau suivant.

Seuil	Nombre de clusters	MAP	Seuil	Nombre de clusters	MAP
1.100151104028533	2	**0.2229**	1.7280795632413444	2	0.2213
	5	**0.2356**		5	0.2093
	8	**0.2306**		8	0.1934
	11	**0.2297**		11	0.1991
1.257133218831736	2	**0.2339**	1.885061678044547	2	0.2213
	5	**0.2217**		5	0.2093
	8	0.2115		8	0.1934
	11	0.2149		11	0.1991
1.4141153336349386	2	**0.2276**	2.0420437928477497	2	0.2213
	5	0.2153		5	0.2093
	8	0.2051		8	0.1934
	11	0.2105		11	0.1991
1.5710974484381415	2	**0.2221**			
	5	0.2097			
	8	0.1983			
	11	0.2043			

Tableau III. 3 Résultats obtenus avec notre approche

Le tableau ci-dessus nous montre que le modèle de pertinence étendu avec un seuil de **« 1.100151104028533 »** et un nombre de cluster qui est égale à **5**, nous apporte une importante amélioration avec une MAP de **0.2356** par rapport au modèle de pertinence de base qui a obtenu une MAP de **0.2215**.

III.3.5.1.4 Evaluation requête par requête

Afin d'avoir une interprétation plus détaillée des résultats obtenus précédemment, nous avons effectué l'analyse des résultats requête-par-requête. Ainsi nous avons obtenus les tableaux suivants :

Requête	MAP (recherche simple)	MAP (Modèle de pertinence)	Taux d'amélioration %	Requête	MAP (recherche simple)	MAP (Modèle de pertinence)	Taux d'amélioration %
51	0.5008	0.8420	68.13	76	0.0034	0.0004	-88.24
52	0.4827	0.6338	31.30	77	0.0307	0.0172	-43.97
53	0.3855	0.6098	58.18	78	0.1154	0.1905	65.08
54	0.4199	0.7289	73.59	79	0.0000	0.0000	0
55	0.0592	0.5024	748.65	80	0.0268	0.0651	142.91
56	0.5583	0.8469	51.69	81	0.1529	0.1477	-3.40
57	0.1444	0.6519	351.45	82	0.2941	0.7233	145.94
58	0.0707	0.0216	-69.45	83	0.0048	0.0229	377.08
59	0.0018	0.0002	-88.89	84	0.0245	0.0213	-13.06
60	0.0012	0.0063	425	85	0.0384	0.1009	162.76
61	0.4764	0.7236	51.89	86	0.0082	0.0017	-79.27
62	0.0819	0.1121	36.87	87	0.0022	0.0236	972.73
63	0.0169	0.2665	1476.92	88	0.0829	0.1191	43.67
64	0.0125	0.0332	165.6	89	0.0045	0.0061	35.56
65	0.0000	0.0000	0	90	0.3332	0.4152	24.61
66	0.0007	0.0006	-14.29	91	0.0000	0.0000	0
67	0.0000	0.0000	0	92	0.0030	0.0016	-46.67
68	0.0440	0.4367	892.5	93	0.0069	0.0091	31.88
69	0.1790	0.0636	-64.47	94	0.0014	0.0007	-50
70	0.1790	0.8128	354.08	95	0.0017	0.0004	-76.47
71	0.0595	0.3984	569.58	96	0.0111	0.0093	-16.22
72	0.0018	0.0004	-77.78	97	0.5079	0.7276	43.26
73	0.0001	0.0000	-100	98	0.6218	0.5563	-10.53
74	0.0004	0.0004	0	99	0.0002	0.0004	100
75	0.0000	0.0004	0	100	0.0000	0.0000	0

Tableau III.4 Résultats obtenus avec l'analyse requête par requête avant et après l'expansion avec le modèle de pertinence

Le tableau ci-dessus montre le taux d'amélioration du modèle de pertinence par rapport à la recherche simple où nous avons obtenus les résultats suivants :

- 27 requêtes améliorées

- 16 requêtes dégradées
- 7 requêtes nulles

Requête	MAP (recherche simple)	MAP (notre approche)	Taux d'amélioration %	Requête	MAP (recherche simple)	MAP (notre approche)	Taux d'amélioration %
51	0.5008	0.8420	68.13	76	0.0034	0.0004	-88.24
52	0.4827	0.6338	31.30	77	0.0307	0.0172	-43.97
53	0.3855	0.6098	58.18	78	0.1154	0.1905	65.08
54	0.4199	0.7289	73.59	79	0.0000	0.0000	0
55	0.0592	0.5185	775.84	80	0.0268	0.0698	160.45
56	0.5583	0.8469	51.69	81	0.1529	0.1477	-3.40
57	0.1444	0.6519	351.45	82	0.2941	0.7233	145.94
58	0.0707	0.4791	577.65	83	0.0048	0.1208	2416.67
59	0.0018	0.0002	-88.89	84	0.0245	0.0213	-13.06
60	0.0012	0.0192	1500	85	0.0384	0.0795	107.03
61	0.4764	0.7236	51.89	86	0.0082	0.0017	-79.27
62	0.0819	0.0503	38.58	87	0.0022	0.0203	822.73
63	0.0169	0.2665	1476.92	88	0.0829	0.1191	43.67
64	0.0125	0.1956	1464.8	89	0.0045	0.0061	35.56
65	0.0000	0.0000	0	90	0.3332	0.4152	24.61
66	0.0007	0.0006	-14.29	91	0.0000	0.0000	0
67	0.0000	0.0000	0	92	0.0030	0.0003	-90
68	0.0440	0.4367	892.5	93	0.0069	0.0002	-97.10
69	0.1790	0.0636	-64.47	94	0.0014	0.0000	-100
70	0.1790	0.8128	354.08	95	0.0017	0.0007	-58.82
71	0.0595	0.4348	630.76	96	0.0111	0.0093	-16.22
72	0.0018	0.0004	-77.78	97	0.5079	0.7276	43.26
73	0.0001	0.0000	-100	98	0.6218	0.5563	-10.53
74	0.0004	0.0007	0	99	0.0002	0.0022	1000
75	0.0000	0.0004	0	100	0.0000	0.0000	0

Tableau III.5 Résultats obtenus avec l'analyse requête par requête avant et après l'expansion avec notre approche

Le tableau ci-dessus montre le taux d'amélioration de notre approche par rapport à la recherche simple où nous avons obtenus les résultats suivants :

- 27 requêtes améliorées
- 16 requêtes dégradées
- 7 requêtes nulles

Requête	MAP (notre approche)	MAP (Modèle de pertinence)	Taux d'amélioration %	Requête	MAP (notre approche)	MAP (Modèle de pertinence)	Taux d'amélioration %
51	0.8420	0.8420	0	76	0.0004	0.0004	0
52	0.6338	0.6338	0	77	0.0172	0.0172	0
53	0.6098	0.6098	0	78	0.1905	0.1905	0
54	0.7289	0.7289	0	79	0.0000	0.0000	0
55	0.5185	0.5024	3.20	80	0.0698	0.0651	7.22
56	0.8469	0.8469	0	81	0.1477	0.1477	0
57	0.6519	0.6519	0	82	0.7233	0.7233	0
58	0.4791	0.0216	2118.05	83	0.1208	0.0229	427.51
59	0.0002	0.0002	0	84	0.0213	0.0213	0
60	0.0192	0.0063	204.76	85	0.0795	0.1009	-21.21
61	0.7236	0.7236	0	86	0.0017	0.0017	0
62	0.0503	0.1121	-55.13	87	0.0203	0.0236	-13.98
63	0.2665	0.2665	0	88	0.1191	0.1191	0
64	0.1956	0.0332	489.16	89	0.0061	0.0061	0
65	0.0000	0.0000	0	90	0.4152	0.4152	0
66	0.0006	0.0006	0	91	0.0000	0.0000	0
67	0.0000	0.0000	0	92	0.0003	0.0016	-81.25
68	0.4367	0.4367	0	93	0.0002	0.0091	-97.80
69	0.0636	0.0636	0	94	0.0000	0.0007	-100
70	0.8128	0.8128	0	95	0.0007	0.0004	75
71	0.4348	0.3984	9.14	96	0.0093	0.0093	0
72	0.0004	0.0004	0	97	0.7276	0.7276	0
73	0.0000	0.0000	0	98	0.5563	0.5563	0
74	0.0007	0.0004	75	99	0.0022	0.0004	450
75	0.0004	0.0004	0	100	0.0000	0.0000	0

Tableau III.7 Comparaison de l'analyse requête par requête dans l'expansion avec notre approche et celle obtenue avec le modèle de pertinence

Le tableau ci-dessus montre le taux d'amélioration de notre approche par rapport au modèle de pertinence où nous avons obtenus les résultats suivants :

- 10 requêtes améliorées
- 6 requêtes dégradées
- 34 requêtes nulles

Les résultats de l'évaluation requête-par-requête, nous montre que notre approche obtient des meilleurs résultats en utilisant la meilleure configuration en terme de document et de nombre de termes qui est égal à **30,30** et un seuil de clarté de **« 1.100151104028533 »**avec le nombre de clusters qui égal à **5.**

III.4 Conclusion

Dans ce chapitre, nous avons présenté le principe de notre approche et son fonctionnement via des algorithmes, nous avons aussi présenté l'environnement d'implémentation et les tests de notre approche. Enfin, nous avons présenté et discuté les résultats de notre approche vis-à-vis le modèle de base TF-IDF et le modèle de pertinence.

Le travail réalisé rentre dans le contexte de la RI, plus précisément dans l'expansion de la requête.

En effet l'expansion de requête permet d'étendre la requête initiale avec des termes sélectionnés, généralement des premiers documents retournés par la première recherche.

Notre travail a consisté à implémenter une nouvelle technique de reformulation, elle consiste à sélectionner les documents d'expansion en se basant sur le score de clarté de la requête.

La réalisation de notre travail nous a permis d'imprégner des principaux concepts de cette nouvelle discipline qui est la recherche d'information.

Nous avons également approfondi nos connaissances sur le langage de programmation Java et l'environnement Netbeans ainsi que la plate forme de RI Terrier lors de la réalisation de notre approche, et son utilisation pour l'indexation de documents et pour la recherche.

Pour finir, d'après les résultats obtenus avec notre approche nous sommes dans la bonne voix avec la collection de test TREC AP88. Pour cela nous pensons que l'utilisation de ces modèles peut encore évoluer vers plus de performance.

[Abberley& al, 99]Abberly, D. Kirby, S. Renals, and T. Robinson. The THISL broadcast news retrieval system, In Proc. ESCA Workshop on Accessing Information In Spoken Audio, pages19 {24, Cambridge, 1999}.

[Aussenac-Gilles, 00]"Revisiting Ontology Design: a method based on corpus analysis".Proc of KAW'2000. Juan-Les-Pins (F). Oct 2000. Lecture Notes in Artificial Intelligence Vol 1937. Springer Verlag. pp. 172-188, 2000.

[Amati, 03] Probabilistic models for information retrieval based on divergence from randomness. Ph.D. thesis, Department of Computing Science, University of Glasgow, UK.

[Attar &Fraenkel, 77]Local feedback in full-text retrieval systems.J. ACM 24, 3, 397–417.

[Bai & al, 05] Query expansion using term relationships in language models for information retrieval. In Proceedings of the 14th ACM International Conference on Information and Knowledge Management.ACM Press, 688–695**[Bai & al, 07]** Extending query translation to cross-language query expansion with markov chain models.In Proceedings of the 16th Conference on Information and KnowledgeManagement (CIKM'07).ACM Press.

[Buckley & al, 94] C. Buckley, G. Salton & J. Allan : The Effect of Adding Information in a Relevance Feedback Environment, Conference on Research and Development in Information Retrieval (SIGIR), 1994.

[M. Boughanem & al, 99] Query modification based on relevance back propagation in adhoc environment .Information processing and Management, 35:pages 121-139.

[Carpineto& al, 01] An information theoretic approach to automatic query expansion.ACM Trans. Info. Syst. 19, 1, 1–27.

[Carpineto& al, 02]Improving retrieval feedback with multiple term-ranking function combination.ACM Trans. Info. Syst. 20, 3, 259–290.

[Cao& al, 08] Selecting good expansion terms for pseudorelevance feedback.In Proceedings of the 31st Annual International ACMSIGIR Conference on Researchand Development in Information Retrieval. ACM Press, 243–250.

[Church & Hanks, 90]Word association norms, mutual information and lexicography.Computat.Linguist. 16, 1, 22–29.

[Chirita& al, 07] Personalized query expansion for the web. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Developmentin Information Retrieval. ACM Press, 7–14.

[Collins-Thompson & Callan, 05] Query expansion using random walk models. In Proceedings of the 14th Conference on Information and Knowledge Management (CIKM'05). ACM Press,704–711.

[Cormack, al, 99] G. Cormack, C.R. Palme, M.V Biesbrouk& C.L.A. Clarck: Deriving Very Short Queries for High Precision and Recall, In Proceedings of the 7th Text Retrieval Conference TREC7, July 1999.

[B. Croft & D. Harper,79] Using probabilistic models of information wi-thout relevance information. Journal of documentation, 35(4):285,295.

[C.J Crouch &B. Yang,1992] Experiments in automatic statistical thesaurus construction. In proceedings of the ACM-SIGHIR Conference on Research and Development in Information Retrieval, Copenhage, Denmark, pages 77-88.

[Cui & al, 03] Query expansion by mining user logs. IEEE Trans. Knowl. Data Engin. 15, 4, 829–839.

[Diaz & Metzler, 06] Improving the estimation of relevance models using large external corpora.In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Developmentin Information Retrieval. ACM Press, 154–161.

[Efthimiadis, 96]Efthimis N. Efthimiadis.Query Expansion. Annual Review of Information Science and Technology, ARIST. 31:121,{187, 1996}
.

[Efthimiadi, 93] A user-centred evaluation of ranking algorithms for interactive query expansion.In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Developmentin Information Retrieval. ACM Press, 146–159.

[Foskett, D. J, 97]. Readings in information retrieval.Chapitre Thesaurus, pages 111 134. Morgan Kaufmann Publishers Inc, San Francisco, CA, USA.

[Fuhr, N, 05] Information Retrierval – From Information Access to Contextual Retrieval. In M. Eibl, C. Wolf, and C. Womser-Hacker, editors, Designing Information Systems. Festschrift fur jurgen Krause, pp. 47-57.UVK Verlagsgesellschaft, 2005.

[Gruber, 93] T. Gruber. "A translation approach to portable ontology specifications.Knowledge Acquisition". 5(2):199–220, 1993.

[Guarino, 95] "Ontologies and knowledge bases: Towards a terminological clarification".In Towards Very Large Knowledge Bases. N. J. I. Mars, Ed, IOS Press: 25-32, 1995.

[Haines & Croft, 93] D. Haines & W.B Croft : Relevance Feedback and Inference Networks, Conference on Research and Development in Information Retrieval (SIGIR), pp 2-11, 1993.

[Hammache,13]Hammache Arezki : la recherche d'information:un modèle de langue combinant mots simples et mots compsés, 2013.

[Harman, 92] D. Harman : Relevance Feedabck Revisited : Conference on Research and Development in Information Retrieval (SIGIR), pp 1-10, 1992.

[Hiemstra, D, 98] A linguistically motivated probabilistic model of information retrieval. In Nicolaou, C. , and Stephanides, C., editors, Research and Advanced Technology for Digital Libraries – Second European Conference, ECDL '98, Proceedings, number 1513 in Lecture Notes in Computer Science. Springer Verlag, pp. 569-584, 1998.

[Hu & al, 06]Improving retrieval performance by global analysis.In Proceedings of the 18th International Conference on Pattern Recognition.IEEE Computer Society, 703–706.

[Hudon,94] M. Hudon. Le thésaurus : construction, élaboration, gestion. ASTED, 1994.

[Ihadjadene, M, 04] les systèmes de RI, ed, HermesSience, Paris 2004.

[Ingwersen, 92] P, Ingwersen. Information retrieval interaction. London, Taylor Graham, 1992.

[Khelif, 06] M.K.Khelif. "Web sémantique et mémoire d'expériences pour l'analyse du transcriptome". Thèse de doctorat en informatique, Université de Nice-Sophia Antipolis-UFR sciences, pages 7-16, Avril 2006.

[Lafferty &Zhai, 01]Document language models, query models and risk minimization for information retrieval. In W.B Croft ,D.J.Harper , D.H Kraft, & J. Zobel (Eds). Proceedings of the 24th annual international ACM-SIGIR conference on research and development in information retrieval., pp.111-119,2001.

[Latiri& al, 04] Query expansion using fuzzy association rules between terms. In Proceedings of the 4th International Conference Journ´ees de l'InformatiqueMessine (JIM'03).

[Lavrenko& Croft, 01] Relevance based language models. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.ACMPress, 120–127.

[Lavrenko, V & Croft, W.B , 01] Relevance-based language models.In W.B Croft, D.J.

Harper, D.H. Kraft, & J. Zobel (Eds.). Proceedings of the 24th Annual International ACM-

SIGIR Conference on Research and Development in Information Retrieval New Orleans,

Louisiana, pp. 120-127, 2001.

[Lynda Tamine, 2000] Université Paul Sabatier - Toulouse III, 2000.French.

[Maning D & al, 2008] Introduction to information Retreival combridge University, Press.

[MATAOUI M'hamed,07] Specification de logiciels et traitement de l'information, << Reformulation de requête dans les systèmes de recherche d'information dans des documents XML >>

[M. Mitra & al, 98] Improving automatic query expansion. In Proceedings of the twenty-First Annual International ACM SIGIR Conference on Research and Developments in Information Retrei-val, 206-214. Melbourne.

[Neches, 91] R. Neches, R.E. Fikes, T. Finin T, T.R. Ruber, R. Patil, T. Senator, W.R. Wartout. "Enabling technology for knowledge sharing".AI Magazine, 12(3), 16- 36,1991.

[Ottens, 07] K. Ottens. « Un système multi-agent adaptatif pour la construction d'ontologies à partir de textes ». page 14, Octobre 2007.

[P, LEFEVRE, 04] la recherche d'information, du texte intégral au thésaurus, ed. HermesScience, paris 2004.

[Ponte, J.M & Croft, W.B, 1998] A language modeling approach to information retrieval.Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 275- 281, 1998.

[Qiu&Frei, 93] Concept-based query expansion.In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press,160–169.

[Robertson & al, 95] S.E Robertson, S.E. Walker & M.M Hnacock-Beaulieu : Large Test Collection Experiments on an Operational Interactive System : Okapi at TREC, in IP&M, pp 260-345, 1995.

[Rocchio, 71] Relevance feedback in information retrieval. In The SMART Retrieval System, G. Salton Ed., Prentice-Hall, Englewood Cliffs, NJ, 313–323.

[Salton, 70] G. Salton. The SMART retrieval system: Experiments in automatic document processing. Prentice Hall. 1970.

[Salton, 84] G. Salton and M. McGill. Introduction to modern information retrieval.McGraw-Hill Int. Book Co. 1984.

[Salton, G, 83] E.A.Fox, H. Wu.Extended Boolean information retrieval system.CACM 26(11), pp. 1022-1036, 1983**.**

[Salton, Buckley, 90] G.Salton&C.Buckley : Improving Retrieval Performance By Relevance Feedback, Journal of The American Society for Information Science, Vol. 41, N°4, pp 288-297, 1990.

[Salton &McGIll, 90]Improving Retrieval Performance by Relevance Feedback. Journal of the America Society for Information Science 41, 4, 288-297.

[Song & al, 07] Integration of association rules and ontologies for semantic query expansion.Data Knowl.Engin. 63, 1, 63–75.

[Sun & al, 06] Mining dependency relations for query expansion in passage retrieval.In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press, 382–389.

[vanRijsbergen, 79] Information Retrieval.Butterworths.

[Voorhees, 04]Overview of the trec 2004 robust track.In Proceedings of the 13th Text REtrieval Conference (TREC-7).NIST Special Publication 500-261.National Institute of Standards and Technology(NIST), Gaithersburg, MD.

[Wong & al, 08]Re-examining the effects of adding relevance information in a relevance feedback environment.Info.Process. Manage. 44, 3,1086–1116.

[Xu &Croft, 96] Query expansion using local and global document analysis. In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press, 4–11.

[Xu & Croft, 00]Improving the effectiveness of information retrieval with local context analysis.ACM Trans. Info. Syst. 18, 1, 79–112 la **réinjection de pertinence** .

[Zhai& Lafferty, 01] Zhai, C., Lafferty, J. Model-based feedback in the language modeling approach to information retrieval. Proceedings of the 10th International Conference on Information and Knowledge Management.ACM Press, pp.403-410, 2001.