REPUBLIQUE ALGERRIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSTE MOULOUD MAMMERI DE TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes de Master Professionnel

Domaine : Mathématique et Informatique
Filière : Informatique
Spécialité : Ingénierie des Systèmes d'Information

Présenté par :
Melle MANSOUR Samia
Melle LAKRIB Sihem

Thème
Expansion de requête basée sur les phrases

Mémoire soutenu publiquement le 30 /09/2017 Devant le jury composé de :

Président : Md Y.YASLI
Encadreur : Mr A.HAMMACHE
Examinateur : Md L.BOUGCHICHE

Promotion 2016/2017

Remerciements

Nous tenons à témoigner notre reconnaissance à Dieu tout puissant, et miséricordieux, qui nous a donné la force et la patience durant ces langues années d'étude.

Nous tenons à exprimer toute notre reconnaissance à notre promoteur monsieur HAMMACHE. Nous le remercie de m'avoir encadré, orienté, aidé et conseillé.

A tous les enseignants de l'UMMTO qui ont contribué à notre formation.

Nos plus vifs remerciements vont aussi aux membres du jury pour avoir accepté d'honorer par leur jugement notre travail.

Nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail, en particulier nos chères familles et nos amis.

Samia et Sihem

Dédicaces

JE dédie ce modeste travail à :

A ceux que j'aime jusqu'à la frontière de l'imagination : Ma mère et mon père ; sans eux je n'aurai pas abouti à ce stade d'étude qu'ALLAH m'aide à les honorer, les servir et les combler

A ceux que j'aime et que j'adore :

Mes frères : Hakim et Toufik

Mes sœurs : Djamila, Naima, Karima et Asma

Mes neveux : Rayane, Abd el Hadi et Aylane

Mes nièces : Maria et Lina

Ma belle sœur : Sabrina

Tous mes adorables amis(e).

Toutes les personnes qui m'ont soutenues et crus en moi lors de mon parcourt et à tous ceux qui m'ont aidé de près ou de loin pour réaliser ce projet.

A toute la promotion 2017.

Samia

Dédicaces

JE dédie ce modeste travail à :

A ceux que j'aime jusqu'à la frontière de l'imagination : Ma mère et mon père ; sans eux je n'aurai pas abouti à ce stade d'étude qu'ALLAH m'aide à les honorer, les servir et les combler

A ceux que j'aime et que j'adore :

Mes frères : Abdelghani et Hamza

Mes sœurs : Chabha et Kenza

Tous mes adorables amis(e).

Toutes les personnes qui m'ont soutenues et crus en moi lors de mon parcourt et à tous ceux qui m'ont aidé de près ou de loin pour réaliser ce projet.

A toute la promotion 2017.

Sihem

La liste des figures

.

Liste des tableaux

Sommaire

Chapitre II : Expansion de la requête

Chapitre III : Evaluation et Expérimentation

Introduction Générale

Introduction générale

La recherche d'information (RI) est une branche de l'information qui s'intéresse à la représentation, l'acquisition, l'organisation, le stockage, la recherche et la sélection d'information dans une base documentaire donnée. Elle fournit les outils et les méthodes pour facilité l'accès à une collection d'informations, et retrouver l'information qui répond à un besoin informationnel de l'utilisateur exprimé sous forme de requête.

Ces outils sont appelés Systèmes de Recherche d'Information (SRI), ces systèmes consiste à construire une représentation des documents et de la requête et d'établir une comparaison entre ces deux représentations (requête, documents) pour retourner les documents pertinents. Cette comparaison est réalisée au moyen d'un modèle de recherche. Afin d'obtenir un SRI performant, il est nécessaire de construire une bonne représentation du document et de la requête et de développer un modèle de RI qui supporte ces représentations.

La plupart des SRI existants représentent les documents comme un ensemble de mots clés, ce que l'on appelle communément une représentation par sac de mots. Plusieurs méthodes en développements, parmi ou en trouve elle prenant en compte la proximité entre les termes (des termes adjacents) et utilisation d'unité de présentation plus complexe (N-gramme).

Dans les système de recherche d'information, retrouver des documents pertinents en utilisant seulement la requête initiale est une tâche presque impossible, vu le volume croissant des bases d'information. Pour cela l'expansion de requêtes est une des méthodes pour remédie le problème de la disparité des termes. L'expansion de requêtes est l'une des techniques utilisée pour résoudre ce problème. Elle consiste à étendre la requête originale avec des termes liés aux termes de celle-ci. Notre travail s'intéresse à la technique de réinjection de pertinence ; le choix de terme se base sur la relation de cooccurrence entre les termes de la requête initiale et les termes des premiers documents retournes par la première recherche. Cependant cette technique ne convient pas à tous les types de requêtes et précisément les requêtes ambigües. Pour résoudre ce problème, nous proposons une approche permettant une représentation plus précise des documents et des requêtes elle se base sur les

termes composés.il est généralement supposé que les termes composés sont moins ambigus que les termes simples, et ils représentent un sens plus précis.

L'organisation retenue pour la présentation notre mémoire, s'articule en trois chapitres :

- Chapitre 1 : porte sur des généralités sur le domaine de la recherche d'information, notamment les concepts de base de la recherche d'information, les modèles de la recherche d'information. Nous décrivons aussi l'indexation avec les mots composés et nous présenté l'évaluation des systèmes de recherche d'information.
- Chapitre 2 : Nous définissons d'abord l'expansion de la requête, nous présentons ensuite les méthodes d'expansion de requêtes et le processus d'expansion automatique de la requête. Enfin, nous décrivons l'expansion de la requête dans le modèle de langue.
- Chapitre 3 : porte sur la présentation de l'approche proposée et l'expérimentation de l'approche en précisant les outils et langage utilisés pour sa mise en œuvre. Ainsi que les résultats obtenus.

Ce travail se termine par une conclusion générale et quelques perspectives.

Chapitre I: Recherche d'information

Chapitre I: Recherche d'information

I.1.Introduction

La recherche d'information(RI) n'est pas un domaine récent, il date des années 1940, dès la naissance des ordinateurs. La recherche d'information est une branche de l'informatique qui s'intéresse à la représentation, l'acquisition, l'organisation, le stockage, la recherche et la sélection d'information dans une base documentaire donnée. Elle fournit les outils et les méthodes pour faciliter l'accès à une collection d'informations, et retrouver l'information qui répond à un besoin informationnel de l'utilisateur exprimé sous forme de requête.

Ce chapitre a pour but de présenter le domaine de RI. Dans la première section, nous avons données une petite définition de recherche d'information. Dans la seconde section, nous présentons les concepts de base de la RI. En particulier, nous décrivons les notions de document, de requête et de pertinence ; les processus d'indexation, de recherche et de reformulation de requêtes. Dans la troisième section, nous parlons des modèles de recherche d'information. Dans la dernière partie de ce chapitre est discutée l'évaluation des systèmes de recherche d'information.

I.2.Définition de la recherche d'information

La recherche d'information est une branche de l'informatique qui s'intéresse à l'acquisition, l'organisation, le stockage, la recherche et la sélection d'information **[1]**. La recherche d'information fournit donc les techniques et outils pour permettre de représenter, stocker, organiser, rechercher et retrouver, dans une masse documentaire existante, les documents contenant l'information qui répond au besoin informationnel exprimé par l'utilisateur sous forme de requête **[2]**.

I.3. Le système de recherche d'information

Un système de recherche d'information (SRI) est un ensemble de programmes informatiques qui permet de retrouver, à partir d'une collection de documents, les documents pertinents pour une requête utilisateur **[3]**. Un SRI est composé de trois fonctions principales, représentées schématiquement par le processus U de recherche d'information. Cette architecture générale est illustrée dans la figure I.1.
On distingue trois modules principaux :

- **Le module d'indexation**, il permet de construire une représentation synthétique des documents, appelé index. Lorsque l'utilisateur formule sa requête un processus similaire est effectué sur la requête. Il consiste à analyser la requête et établir une représentation interne.
- **Le module d'appariement requête-document,** consiste alors à calculer le degré de correspondance des représentations internes des documents et de la requête. Les documents qui correspondent au mieux à la requête, dits documents pertinents, sont alors retournés à l'utilisateur, dans une liste ordonnée par ordre décroissant de degré de pertinence lorsque le système le permet.
- **Le module de reformulation de la requête** : permet d'améliorer les résultats de la recherche. Ces trois modules sont détaillés ci-après.

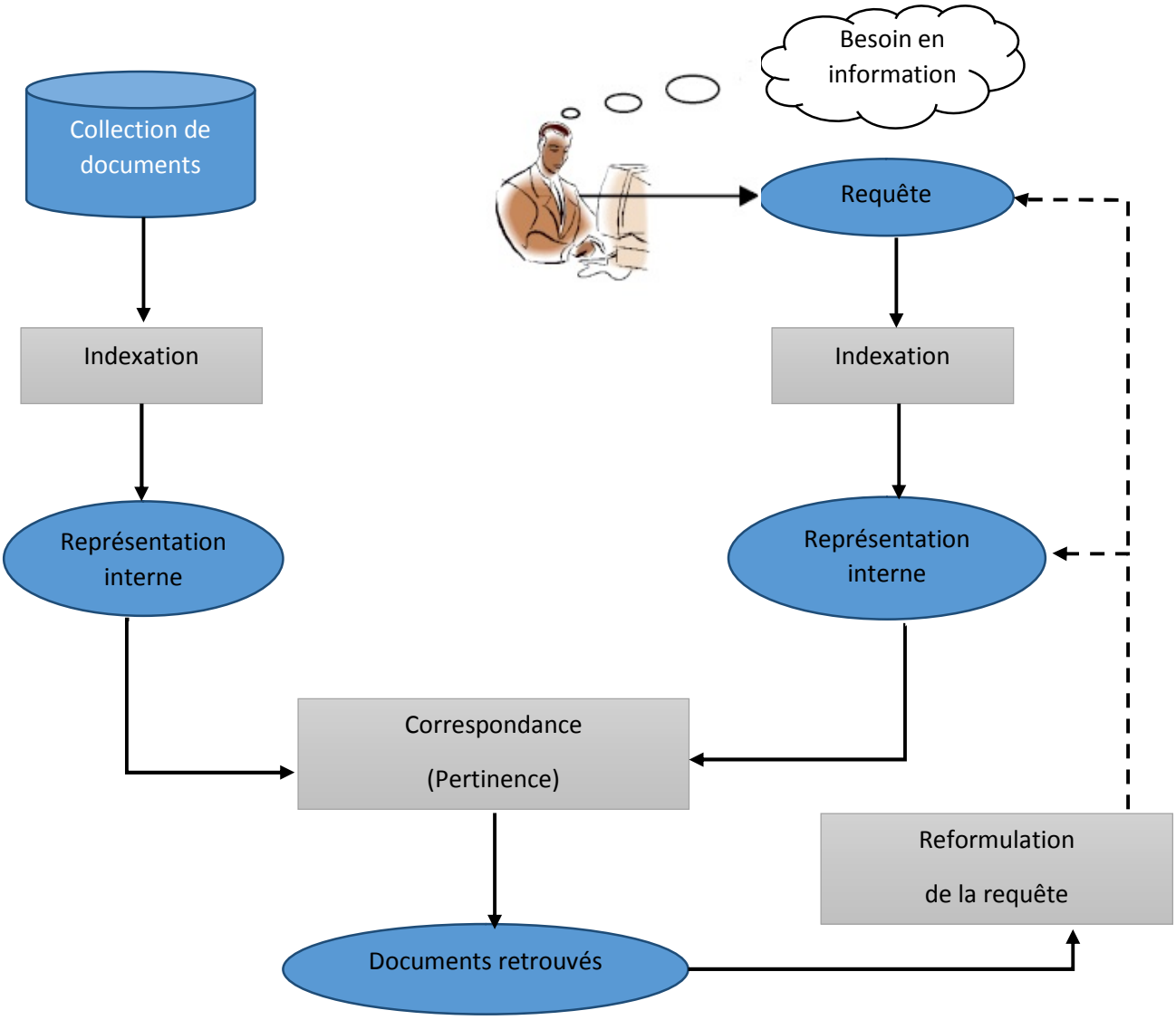

Figure I.1 : Architecture générale d'un Système de Recherche d'Information [10].

I.4. Les concepts de bases

Le Système de recherche d'information est caractérisé par trois éléments de base qui sont : document, requête et pertinence et trois processus : processus d'indexation, processus de recherche et processus de reformulation de la requête.

I.4.1. Les éléments de base

Collection de documents : la collection de documents (ou fond documentaire) constitue l'ensemble des informations exploitables et accessibles appelé aussi base documentaire ou corpus. Elle est constituée d'un ensemble de documents. Dans le cas général et pour un souci d'optimalité, la base constitue des représentations simplifiées mais suffisantes pour ces documents. Ces représentations sont étudiées de telles sortes que la gestion (ajout suppression d'un document) ou l'interrogation (recherche) de la base se font dans les meilleures conditions de coût.

Un document : représente le conteneur élémentaire d'information, exploitable et accessible par le SRI. En effet, un document peut être un texte, une page WEB, une image, une bande vidéo, Il désigne toute unité qui peut constituer une réponse à un besoin en information exprimée par un utilisateur.

Une requête : exprime le besoin d'information d'un utilisateur. Elle peut être exprimée selon différents langages, une liste de mots clés ou un langage booléen. Le langage le plus utilisé est le langage naturel.

La pertinence : est une notion fondamentale en RI. Elle est l'objet de tout système de recherche d'information. Elle peut être définie comme la correspondance entre un document et une requête selon le système ou l'utilisateur. Essentiellement, deux types de pertinence sont définis : la pertinence système et la pertinence utilisateur.

La pertinence Système : c'est l'évaluation par le système de recherche d'information, de l'adéquation entre des documents et une requête **[11].**

La pertinence Utilisateur : c'est l'évaluation par l'utilisateur, de la pertinence, vis-à-vis de son besoin en information, des documents retrouvés par le SRI.

I.4.2. Les processus de base

I.4.2.1. Le processus d'indexation

Pour que le coût de la recherche soit acceptable, il convient d'effectuer une étape primordiale sur la collection de documents. Cette étape consiste à analyser les documents afin de créer un ensemble de mots-clés : on parle de l'étape d'indexation. Ces mots-clés seront plus facilement exploitables par le système lors du processus ultérieur de recherche. L'indexation permet de créer une vue logique du document. On entend par vue logique la représentation des documents dans le système. L'indexation peut être :

a. **Manuelle :** chaque document de la collection est analysé par un spécialiste du domaine ou un documentaliste. L'indexation manuelle assure une meilleure précision dans les documents restitués par le SRI en réponse aux requêtes des utilisateurs. Néanmoins, cette indexation présente un certain nombre d'inconvénient liés notamment à l'effort et le prix qu'elle exige (en temps et en nombres de personnes).De plus, cette indexation est subjective, qui est liée au facteur humain, différents spécialistes peuvent indexer un document avec des termes différents. Il se peut même arriver qu'un spécialiste indexe différemment un document, à différent moment.
b. **Semi- automatique :** la tâche d'indexation est réalisée ici conjointement par un programme informatique et un spécialiste du domaine. Le choix final des descripteurs revient à l'indexeur humain. Dans ce type d'indexation un langage d'indexation contrôlé est généralement utilisé.
c. **Automatique :** Dans ce cas, l'indexation est entièrement automatisée. Elle est réalisée par un programme informatique et elle passe par un ensemble d'étapes pour créer d'une façon automatique l'index. Ces étapes sont : l'analyse lexicale, l'élimination des mots vides, la normalisation (lemmatisation ou radicalisation), la sélection des descripteurs, le calcul de statistiques sur les descripteurs et les documents (fréquence d'apparition d'un descripteur dans un document et dans la collection, la taille de chaque document, etc.) et enfin la création de l'index et éventuellement sa compression. Nous détaillons ces différentes étapes ci-dessous.

I.4.2.1.1. L'analyse lexicale

Elle consiste à découper le document en un ensemble d'unités lexicales (mots ou token).chaque unité lexicale ou un radical est une séquence de caractères entourée par des séparateurs d'unités. Elle permet alors de reconnaitre les espaces des séparations des mots, les chiffres, les ponctuations, etc.

I.4.2.1.2. L'élimination des mots vides

Les documents contiennent souvent des termes non significatifs appelés mots vides (pronoms personnels, prépositions, etc.), car ils ne traitent pas le sujet du document.
On distingue deux techniques pour éliminer les mots vides :

– L'utilisation d'une liste préétablie de mots vides (aussi appelée anti-dictionnaire, Stop-List en anglais),

– L'élimination des mots dépassant un certain nombre d'occurrences dans la collection.

L'élimination de ces termes peut réduire de manière considérable la taille de l'index, ce qui améliore le temps de réponse du système. Malgré ces avantages potentiels, il peut être difficile de décider du nombre de mot à inclure dans la liste des mots vides.

I.4.2.1.3. La normalisation

La normalisation consiste à représenter les différents variantes d'un terme par un format unique appelé lemme ou racine .ce qui a pour effet de réduire la taille de l'index. Plusieurs stratégies de normalisation sont utilisé :la table de correspondance, l'élimination des suffixes (on peut citer le très connu algorithme de Porter)**[12]**, la troncature, la méthode des n-grammes.

I.4.2.1.4. Le choix des descripteurs

Elle consiste à déterminer le type d'unité élémentaire pour présenter les documents. On parle aussi de descripteur. L'objectif est d'avoir une présentation des documents permettant une moindre perte d'information sémantique possible. On distingue plusieurs types du descripteur **[13]** :

- **Les mots simples :** les mots simples du texte de document en éliminant les mots vides.
- **Les lemmes** ou les racines des mots extraits.
- **Les N-grammes** : qui sont une représentation originale d'un texte en séquence de N caractères consécutifs. On trouve des utilisations des bigrammes et trigramme dans la recherche d'information.
- **Les mots composés :** groupe de mots ou expression sont souvent plus riche sémantiquement que les mots qui les composent pris séparément.
- **Les concepts :** qui sont des expressions pris généralement dans la structure conceptuelle, tels que le thésaurus et ontologie.

I.4.2.1.5.La création d'un index

Au terme de processus d'indexation, un ensemble de structure de données sont créé. Ces dernières permettent un accès efficace à la représentation des documents pour mémoriser les informations sélectionnées lors du processus d'indexation afin d'accélérer la réponse à une requête. Les moyens de stockage les plus utilisé sont :

Le fichier direct (maître) : c'est le fichier de base dans lequel sont stockées les données. L'opération peut durer quelque secondes sur un fichier maître de quelques centaines d'enregistrements, cependant, elle peut se révéler très lente si la base atteint des milliers de documents.

Le fichier inverse : Il est créé autour du fichier maître. Ce fichier comme son nom l'indique, est le résultat de l'invention du fichier maître. Plus exactement, au lieu de donner pour document les mots et les fréquences qui le constituent, on donne pour chaque mot les documents qui le contiennent et sa fréquence dans chacun des documents.

- TF (*Term Fréquence*) : cette mesure est proportionnelle à la fréquence d'un terme dans le document. Elle peut être utilisée telle quelle ou selon plusieurs déclinaisons (log(TF),...)
- IDF (*Inverse of Document Frequency*) : ce facteur mesure l'importance d'un terme dans toute la collection. Un terme qui apparait souvent dans la base documentaire ne doit pas avoir le même impact qu'un terme moins fréquent. Il est généralement

exprimé comme suit : *log* (*N/df*), où *df* est le nombre de documents contenant le terme et *N* est le nombre total de documents de la base documentaire.

La mesure $tf \times idf$ donne une bonne approximation de l'importance du terme dans le document, particulièrement dans les corpus de documents de taille homogène.

Le tableau I.2 illustre un exemple de fichier inverse construit à partir de la collection illustré par le tableau I.1 **[14] :**

Document	**Contenu**
d_1	La recherche d'information gère des textes 1 2 3 4 5 6 7
d_2	Un Système de recherche d'information doit restituer 1 2 3 4 5 6 7 8 L'information pertinente à l' utilisateur 9 10 11 12 13 14 15
d_3	Une information est pertinente si elle satisfait l'utilisateur 1 2 3 4 5 6 7 8 9

Tab I.1 : Exemple d'un fichier direct

Terme	d_1	d_2	d_3
recherche	2	4	
Information	4	6,10	2
Gère	5		
Textes	7		
Système		2	
Restituer		8	
Pertinence		11	4
Utilisateur		14	9
satisfait			7

Tab I.2 : Exemple d'un fichier inverse.

I.4.3. Appariement requête-document

La fonction d'appariement document-requête permet de mesurer la valeur de pertinence d'un document vis-à-vis d'une requête. Afin de réaliser cela, le SRI représente le document et la requête avec un même formalisme, puis il compare les deux représentations. Le résultat de cette comparaison ce traduit par un score qui détermine la probabilité de pertinence (degré de similarité ou degré de ressemblance) du document vis-à-vis de la requête **[15]**.Cette fonction d'appariement est notés RSV (d,q) (Retrieval Statut Value), où d représente un document de la collection et q la requête. Cette valeur permet ensuite au SRI d'ordonner les documents renvoyés à l'utilisateur.

I.4.4. La reformulation de la requête

Compte tenu des volumes croissants des bases d'information, retrouver celles qui sont pertinentes en utilisant seulement la requête initiale de l'utilisateur est une tâche quasi impossible.

Les documents retrouvés en réponse pourraient être analysés du point de vue pertinence puis utilisés pour améliorer la requête initiale.

La reformulation de requête est un processus permettant la construction d'une nouvelle requête, plus à même de représenter les besoins en information de l'utilisateur. Pratiquement, la reformulation de la requête consiste à modifier la requête de l'utilisateur par ajout de termes significatifs et/ou ré estimation de leurs poids.

Le processus de reformulation de requête est communément appelé : réinjection de pertinence (ou *relevance feedback*), lorsque l'information sur la pertinence des documents retournés en réponse à la requête initiale est utilisée pour l'améliorer ou expansion automatique de la requête (ou *query expansion*) lorsque l'information liée à la requête est utilisée pour l'étendre.

I.4.4.1. Reformulation par réinjection de la pertinence

L'idée est de faire participer l'utilisateur dans le processus de recherche de sorte à améliorer l'ensemble final de résultats. Le procédé de base est le suivant :

- l'utilisateur formule sa requête.
- le système lui renvoie un premier ensemble de résultats de recherche.
- l'utilisateur marque quelques documents retournés comme pertinents ou non pertinents.
- En pratique, seuls les 10 (ou 20) premiers documents classés sont examinés et ensuite ils sont exploités pour la reformulation de la requête initiale modifiant les poids des termes qu'elle contient et/ou en ajoutant de nouveaux termes considérés utiles pour retrouver des documents pertinents.

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

I.4.4.2. Pseudo-Réinjection de pertinence

L'idée est d'utiliser les résultats de la recherche en vue d'améliorer l'ensemble final de résultats sans l'intervention de l'utilisateur. On suppose uniquement que les documents les

mieux classées « les premiers » retournés sont pertinents, et on les utilise pour reformuler la requête **[16].**

Le procédé de base consiste à étendre la requête initiale avec de nouveaux termes corrélés aux termes de la requête initiale.

I.5. Les modèles de recherche d'information

Les modèles de recherche d'information ont pour rôle de fournir une formalisation du processus de recherche d'information et un théorique pour la modélisation de la mesure de pertinence. Il existe un grand nombre de modèles de RI textuelle développé dans la littérature. Ces modèles ont en commun le vocabulaire d'indexation basé sur le formalisme mots clés et diffèrent principalement par le modèle d'appariement requête-document. Le vocabulaire d'indexation V = $\{t_i\}$, i $\in$ {1, ..., n} est constitué de n mots ou racines de mots qui apparaissent dans les documents. Selon [Baeza, 1999], un modèle de RI est défini par un quadruplet (D, Q, F, R (q, d)) où :

- D est l'ensemble de documents
- Q est l'ensemble de requêtes
- F est le schéma du modèle théorique de représentation des documents et des requêtes
- R(q,d) est la fonction de pertinence du document d à la requête q

Nous présentons dans la suite les principaux modèles de RI : le modèle booléen, le modèle vectoriel et le modèle probabiliste.

I.5.1. Le modèle booléen

Les premiers SRI développés sont basé sur le modèle booléen **[17]**, même aujourd'hui beaucoup de systèmes commerciaux (moteur de recherche) utilisent le modèle booléen. Cela est dû à la simplicité et à la rapidité de sa mise en œuvre.

Le modèle booléen propose la représentation d'une requête sous forme d'une équation logique. Les termes d'indexation sont reliés par des connecteurs logiques ET, OU et NON **[4]**. L'appariement (RSV) entre une requête et un document est strict, autrement dit si un document implique au sens logique la requête alors le document est pertinent. Sinon, il est considéré non pertinent. La correspondance entre document et requête est déterminée comme suit :

RSV (d,q) = $\begin{cases} 1 \text{ si d appartient à l'ensemble décrit par q} \\ 0 \text{ Sinon} \end{cases}$ (I.1

Malgré la large utilisation de ce modèle, il présente certain inconvénients :

- La nécessité de savoir utiliser et interpréter les formulations booléennes.
- Les documents ne sont pas présentés par ordre de pertinence, tous les documents retournés ont la même mesure de similarité envers la requête soumise.
- Ce modèle ne supporte pas la réinjection de pertinence.
- Les tests effectuées sur des collections d'évaluation standards de RI ont montrés que les systèmes booléennes sont d'une efficacité de recherche inferieure.

Plusieurs extension ont été proposées dont :

- Le modèle booléen flou **[5][6].**
- Le modèle booléen étendu **[7][8].**

I.5.2. Modèle vectoriel

Le modèle vectoriel de base a été introduit par SALTON **[9]**, c'est le modèle le plus populaire en RI, concrétisé dans le cadre du système SMART. La représentation de l'index et la requête est considéré en tant que vecteurs incorporés dans un espace euclidien de *M* dimensions, ces dimensions étant les termes du vocabulaire d'indexation, ou chaque terme est attribué à une dimension indépendante. Chaque document est représenté par un vecteur : $dj = (w_{1j}, w_{2j}, ..., w_{Mj})$. De même chaque requête est représentée par un vecteur : $qi = (w_{1i}, w_{1i}, ..., w_{Mi})$. Avec : *w* correspond au poids d'un terme dans le documentd_j *ou* dans la requêteq_i .La pondération des composantes de la requête est soit la même que celle utilisée pour les documents, soit donnée par l'utilisateur lors de sa formulation. Dans les schémas de pondération qui ont été proposés, la pondération locale et globale sont pris en compte **[18]**.

La pondération locale permet de mesurer l'importance de terme dans le document. Elle prend en compte les informations locales du terme qui ne dépendent que du document. Elle correspond en générale à une fonction de la fréquence d'occurrence du terme dans le document, exprimée ainsi :

$$tf_{ij} = 1 + \log(f(t_i, d_j)) \quad \text{(I.2)}$$

Où f(t_i ,d_j)) est la fréquence du terme t_i dans le document d_j

Quant à la pondération globale, elle prend en compte des informations concernant le terme dans la collection. Un poids plus important doit être assigné aux termes qui apparaissent moins fréquemment dans la collection. Car les termes qui apparaissent dans de nombreux documents de la collection ne permettent pas de distinguer les documents pertinents des documents non pertinents. . Autrement dit, le pouvoir de discrimination d'un terme est proportionnel à sa fréquence documentaire inverse (notéeidf_t) exprimé comme suit :

$$idf = \log\left(\frac{N}{n_i}\right) \quad \text{(I.3)}$$

Où n_i est la fréquence en document du terme considéré, N est le nombre total de document dans la collection.

Les fonctions de pondération combinant la pondération locale et globale sont référencées sous le nom de la mesure $tf \times idf$. Cette mesure donne une bonne approximation de l'importance de terme dans les collections de documents de taille homogène. Cependant, un facteur important est ignoré, la taille de document. En effet, la mesure $tf \times idf$ ainsi définie favorise les documents longs, car ils ont tendance à répéter le même terme, ce qui accroit leur fréquence, par conséquent augmente la similarité de ces documents vis-à-vis de la requête.

Pour remédier à ce problème, des travaux ont proposé d'intégrer la taille du document dans les formules de pondérations comme facteur de normalisation. L'appariement document-requête dans le modèle vectoriel, consiste à trouver les vecteurs documents qui s'approchent le plus de vecteur de la requête. Cet appariement est obtenu par l'évaluation de la distance entre les deux vecteurs. Plusieurs mesures de similarité ont été définies **[19]**, dont les plus courantes sont décrites dans le tableau ci-dessous :

Mesure	Formule
Le produit scalaire	RSV (q,d_i) = $\sum_{j=1}^{\|T\|} W_{qj} \times W_{ij}$
La mesure de cosinus	RSV (q,d_i) = $\frac{q \times d_i}{\|\|q\|\| \times \|\|d_{i\|\|}} = \frac{\sum_{j=1}^{\|T\|} W_{qj} \times W_{ij}}{\sqrt{\sum_{j=1}^{\|T\|} W_{qj^2} \times \sum_{j=1}^{[T\|} W_{ij^2}}}$
La mesure de Dice	RSV (q,d_i) = $\frac{2 \times \sum_{j=1}^{\|T\|} W_{qj} \times W_{ij}}{\sqrt{\sum_{j=1}^{\|T\|} W_{qj^2} + \sum_{j=1}^{[T\|} W_{ij^2}}}$
La mesure de Jaccard	RSV(q,d_i) = $\frac{\sum_{j=1}^{\|T\|} W_{qj} \times W_{ij}}{\sqrt{\sum_{j=1}^{\|T\|} W_{qj^2} + \sum_{j=1}^{[T\|} W_{ij^2}} - \sum_{j=1}^{\|T\|} W_{qj} \times W_{ij}}$

Tab I.3 : Les mesures de similarité utilisées dans les modèles vectoriels.

Les avantages de modèle vectoriel

- Simplicité de mise en œuvre.
- Le langage de requête est plus simple (liste de mot-clés).
- Les performances sont meilleures grâce à la pondération des termes.
- Evaluation flexible (non stricte) des requêtes permettant d'ordonner les documents par degré de pertinence.

Les inconvénients de modèle vectoriel

- Le langage de requête est moins expressif.
- Ce modèle ne permet pas de modéliser les associations entre les termes d'indexation, chacun des termes est considéré comme indépendant des autres.

Plusieurs variantes du modèle vectoriel ont été proposées pour remédier à cette limitation. C'est-à-dire prendre en compte la dépendance entre terme d'indexation. Parmi elles, on trouve le modèle vectoriel généralisé, le modèle LSI (Latent Semantic Indexing) **[20]** et le modèle connexionniste **[21]**.

I.5.3. Le modèle probabiliste

Le modèle probabiliste aborde le problème de la recherche d'information dans un cadre probabiliste. Le premier modèle probabiliste a été proposé par Maron et Kuhns au début des

années 1960 **[22]**.Plusieurs variantes ont proposé pour le modèle probabiliste, parmi elles, on trouve le modèle probabiliste de base, le modèle de langue.

I.5.3.1. Le modèle probabiliste de base

Le principe de base consiste à présenter les résultats d'un SRI dans un ordre basé sur la probabilité de pertinence d'un document vis-à-vis d'une requête.

Etant donné une requête utilisateur notée q et un document d, le modèle probabiliste tente d'estimer la probabilité que le document d appartienne à la classe des documents pertinents (ou non pertinents).

Un document est sélectionné si la probabilité qu'il soit pertinent pour q, notée P(R/d), est supérieure à la probabilité qu'il soit non pertinent pour q, notée P (NR/d). La fonction de classement (tri) est exprimée ainsi :

$$\text{RSV (q,d)} = \frac{P(Per|q,d_i)}{P(NPer|q,d_i)} \qquad \text{(I.4)}$$

En appliquant de bayes pour les deux probabilités on obtient :

$$P(Per|q,d_i) = \frac{P(Per|q)\times P(d_i|Per,q)}{P(d_i)} \qquad \text{(I.5)}$$

$$P(NPer|q,d_i) = \frac{P(NPer|q)\times P(d_i|NPer,q)}{P(d_i)} \qquad \text{(I.6)}$$

Où :

$P(d_i)$ Est la probabilité de choisir le document d_i, on considère qu'elle est constante.

$P(d_i|Per,q)$ Indique la probabilité que d_i fait partie des documents pertinents pour la requête q (respectivement $P(d_i|NPer,q)$ pour les documents non pertinents).

$P(Per|q)$ et$P(NPer|q)$ indiquent respectivement la probabilité de pertinence et de non pertinence d'un document quelconque qui sont constantes.

Après remplacement dans la fonction de tri, nous aurons la formule suivante :

$$\text{RSV (q, d)}=\frac{P(d_i|Per,q)}{P(d_i|NPer,q)} \quad \text{(I.7)}$$

Différentes méthodes sont utilisées pour estimer ces probabilités. La plus connue celle du modèle BIR (Binary Independance Retrieval) .On considère dans ce modèle que la variable document d (t1 = x1, t2 =x2, .., tn = xn) est représentée par un ensemble d'événements indépendants qui dénotent la présence (xi = 1) ou l'absence (xi = 0) d'un terme dans d. Les probabilités de pertinence (resp. de non pertinence) d'un document, notées $P(d_i|Per,\ q)$ (resp. $P(d_i|NPer,\ q)$) ,sont données par :

$$P(d_i|Per,q)=\prod_{t_j\in d_i} P(t_j|Per,q) \times \prod_{t_j\notin d_i} 1-P(t_j|Per,q) \quad \text{(I.8)}$$

$$P(d_i|NPer,q)=\prod_{t_j\in d_i} P(t_j|NPer,q) \times \prod_{t_j\notin d_i} 1-P(t_j|NPer,q) \quad \text{(I.9)}$$

Où $P(t_j|Per,q)$indique la probabilité d'apparition du terme t_j sachant que le document appartient à l'ensemble des documents pertinents et $P(t_j|NPer,q)$ indique la probabilité d'apparition du terme t_j sachant que le document appartient à l'ensemble des documents non pertinents.

En posant $p_i = P(t_j|Per,q)$, $q_i= P(t_j|NPer,q)$ et $p_i = q_i$ pour les termes q_i n'apparaissent pas dans la requête, et après simplification, le calcul du score de correspondance entre un document et une requête peut être exprimé ainsi :

$$\text{RSV } (d_i, \text{q})=\sum_{t_i\in q} \log \frac{p_i(1-q_i)}{q_i(1-p_i)} \quad \text{(I.10)}$$

Afin de classer les documents avec cette formule, il faut estimer les valeurs des deux probabilités p_i et q_i. En l'absence de collection d'apprentissage, on peut attribuer la valeur fixe à $p_i$comme par exemple 0.5, comme elles peuvent être estimées à l'aide de l'avis de l'utilisateur sur les résultats d'une première recherche (réinjection de pertinence).

Inconvénients du modèle BIR

Impossibilité d'estimer ses paramètres si des collections de test ne sont pas disponibles. Pour pallier cet inconvénient, S. Roberston **[23]** a proposé le modèle 2-poisson basé sur la notion de termes élites. Le résultat de ses travaux est la formule BM25, largement utilisée dans les travaux actuels de RI.

I.5.3.2. Le modèle de langue

Le modèle de langue est l'un des premiers modèles en recherche d'information a été proposé par (Ponte et Croft, 1998).Dans les modèles de recherche classique, on cherche à mesurer la similarité entre un document d_j et une requête q ou à estimer la probabilité que le document réponde à la requête (P (d_j /q)). L'hypothèse de base dans ces modèles est qu'un document n'est pertinent que s'il ressemble à la requête. Les modèles de langage sont basés sur une hypothèse différente : un utilisateur en interaction avec un système de recherche fournit une requête en pensant à un ou plusieurs documents qu'il souhaite retrouver. La requête est alors inférée par l'utilisateur à partir de ces documents **[24].**

Le modèle de langue considère que la pertinence d'un document pour une requête est en rapport avec la probabilité que la requête puisse être générée par le document. Soit M_d , le modèle de langue du document d, la pertinence de vis-à-vis d'une requête Q, notée P (Q/M_d), est la probabilité que la requête Q soit générée par M_d**[25].**

Cette pertinence est mesurée par :

$$\text{RSV (d, q)} = \text{P}\,(q = t_1, t_1, \dots, t_n)/M_d\,) = \prod_i P(t_1/M_d\,) \qquad \text{(I.11)}$$

$P(t_1/M_d\,)$ Peut être estimée en se basant sur l'estimation maximale de vraisemblance (maximum likelihood estimation). Elle est donnée par :

$$P\left(\frac{t_1}{M_d}\right) = \frac{tf(t,d)}{|d|} \qquad \text{(I.12)}$$

$tf(t,d)$ Est la fréquence du terme t_i dans le document d. Pour remédier au problème posé par les mots de la requête absents dans les documents, qui ont pour effet d'avoir la probabilité $P(t_1/M_d\,)$ nulle, des techniques de lissage (smoothing) sont utilisées, dont le lissage de laplace (ajouter-un), le lissage de Good-Turing, le lissage Bachoff, Le lissage par

interpolation, etc. **[30]** Leur principe consiste à assigner des probabilités non nulles aux termes, qui n'apparaissent pas dans les documents.

a. Lissage de la place : cette méthode consiste à ajouter la fréquence un (1) à tous les n-grammes, appelé aussi ajouter-un. La probabilité du n-gramme α est estimée ainsi :

$$P_{ajouter\_un}(\alpha|C) = \frac{|\alpha|+1}{\sum_{\alpha_j \epsilon C}|\alpha_j|+1} = \frac{|\alpha|+1}{|C|+N} \quad \text{(I.13)}$$

Où N est le nombre de n-gramme (distinct) et |C| est la taille du corpus.

L'inconvénient de cette méthode est qu'une grande masse de probabilité est distribuée sur les n-gramme non observés dans le corpus.

b. Lissage de Good-Turing : cette méthode permet l'ajustement de la fréquence « r » d'un n-gramme (α) en une fréquence dite corrigée « r^* », exprimée ainsi :

$$r^* = (r+1) \times \frac{n_r + 1}{n_r} \quad \text{(I.14)}$$

Où n_r est le nombre de n-gramme de fréquence « r » dans la collection d'apprentissage. Ainsi, pour tout n-gramme (α) l'estimation de sa probabilité devient alors :

$$P_{GT}(\alpha) = \frac{r^*}{\sum_{\alpha_j \epsilon C} r_j^*} \quad \text{(I.15)}$$

Dans cette méthode la fréquence d'ordre $\frac{r^*}{r}$ pour un n-gramme vu sera redistribuée sur les n-grammes non vus dans le corpus. La méthode de Good-Turing est recommandée pour les n-grammes de faibles fréquences, car elle n'effectue pas de grandes modifications comme c'est cas pour les n-grammes de grandes fréquences.

c. Lissage de Backoff : le principe de cette méthode consiste à utiliser un modèle de langue spécifique d'ordre inférieur, lorsqu'un n-gramme n'est pas observé dans le corpus. C'est le cas de lissage de Kats qui combine le modèle uni-gramme et le modèle bi-gramme, comme suit :

$$P_{Katz}(m_i|m_{i-1}) = \begin{cases} P_{GT}(m_i|m_{i-1}) \text{ si } |m_{i-1}m_i| > 0 \\ \alpha(m_{i-1})P_{Katz}(m_i) \text{ Sinon} \end{cases} \quad \text{(I.16)}$$

Dans cette méthode, la diminution de la fréquence utilisée dans P_{GT} est redistribuée au modèle d'ordre inférieur (uni-gramme). $\alpha\,(m_{i-1})$ Est un paramètre qui détermine la part de cette redistribution à m_i, déterminée comme suit :

$$\alpha\,(m_{i-1}) = \frac{1-\sum_{m_i:|m_{i-1}m_i|>0} P_{GT}\,(m_i|m_{i-1})}{1-\sum_{m_i:|m_{i-1}m_i|>0} P_{ML}\,(m_i)} \qquad \text{(I.17)}$$

Le lissage de Katz est proposé pour palier au problème posé par les n-gramme de hautes fréquences.

d. Lissage par interpolation (Jelinek-Mercer) : Ce type de lissage consiste à combiner le modèle de langue considéré avec un ou plusieurs modèles de références estimés sur d'autres corpus d'apprentissage. Typiquement, dans le cas de collection de documents, on pourrait par exemple estimer le modèle de document en le combinant avec le modèle de la collection. Dans ce ca, le modèle de document est exprimé ainsi :

$$P_{JM}(m_i|d) = (1-\lambda)P_{ML}\,(m_i|\mathrm{d}) + \lambda P_{ML}\,(m_i|\mathrm{C}) \qquad \text{(I.18)}$$

Les modèles $P_{JM}(m_i|d)$ et $P_{ML}\,(m_i|\mathrm{C})$ sont estimés selon le maximum de vraisemblance.

e. Lissage de Dirichlet : le lissage précèdent ne tient pas compte de la taille des échantillons pour remédier à cela, le lissage de Dirichlet exploite les valeurs de λ (formule 30) en fonction de la taille de l'échantillon. Dans ce cas cette formule s'écrit comme suit :

$$P_{Dir}(m_i|d) = \frac{|d|}{|d|+\mu} P_{ML}\,(m_i|\mathrm{d}) + \frac{|d|}{|d|+\mu} P_{ML}\,(m_i|\mathrm{C})$$

$$= \frac{|d|\times P_{ML}(m_i|\mathrm{d}) + \mu P_{ML}\,(m_i|\mathrm{C})}{|d|+\mu} = \frac{tf(m_i,d)+\mu P_{ML}\,(m_i|\mathrm{C})}{|d|+\mu} \qquad \text{(I.19)}$$

Avec $P_{ML}\,(m_i|\mathrm{d}) = \frac{tf(m_i,d)}{|d|}$

Où |d| est la taille du document (le nombre d'occurrence de mots), $tf(m_i,d)\ est\ la\ fréquence\ du\ mot\ m_i$ dans d et μ est un paramètre appelé pseudo fréquence.

Plusieurs études en recherche d'information ont montrés que le choix de la méthode de lissage a un grand impact sur les performances du Système de recherche d'information.

I.6. Au-delà des mots simples

La majorité des approches développés en RI se basent sur l'utilisation des mots simples comme unités de représentations des documents et des requêtes, souvent représentation au sac de mots, ces approches posent deux problèmes : l'ambiguïté des mots et leurs disparité.

L'ambigüité des mots

Dites ambigüité lexicale, se rapporte à des mots lexicalement identique et portant des sens différents. Il est généralement deviser en deux types : ambigüité syntaxique, ambigüité sémantique

- L'ambigüité syntaxique se rapporte à des différences dans la catégorie syntaxique par exemple : « nous avions des avions », avions peut apparaitre en tant que nom ou verbe.
- L'ambigüité sémantique se rapporte à des différences dans la signification, et est décomposé en homonymie et polysémie selon que les sens sont liés ou non.

L'homonymie est une relation entre plusieurs forme linguistique ayant le même signifiant graphique et /ou phonique et des signifiés totalement différents c-à-d, ont la même forme orale ou écrite mais des sens différents. Par exemple, le mot « aids » en anglais désigne les aides et autrement le sida.

La polysémie est une propriété d'un terme qui présente plusieurs sens. Par exemple le mot « terre » désigne la matière sol, et désigne autrement notre planète.

Le problème d'ambigüité implique que des documents non pertinents, contenant les mêmes mots que la requête sont retournés (Bruit Documentaire).

La disparité

Se refaire à des mots lexicalement différents mais portants un même sens. Ceci implique que des documents, portant pertinents ne partagent pas de mot avec la requête, ne sont pas

retournés (Silence documentaire).Par exemple, des documents contenants le terme « tablette tactile » peuvent ne pas être retrouvés en réponse à une requête « I-pad ».

La solution permet de répondre à ces deux problèmes, trois alternatives peuvent être distingués : l'indexation par des mots composés, l'indexation sémantique et l'indexation conceptuelles.

I.6.1. L'indexation par des mots composés

Est une technique qui permet l'utilisation des mots composés comme unités d'indexation. Ceci a pour objectif une représentation plus précise du contenu sémantique des documents et des requêtes **[15]**.

L'idée d'utilisé les mots composé comme unité d'indexation est que ces derniers sont moins ambigus et plus précis que les mots simples. Par exemple, le terme˝java ″ est ambigu, par contre les mots composés˝l'ile de java″ et˝langage de java″ sont non ambigus.

L'intuition est claire, les mots composés aident à construire des unités d'indexation non ambiguës et plus précises et peuvent par conséquent améliorer la précision dans la RI.

Cinq paramètres sont généralement considérés dans l'exploitation des mots composés comme unités d'indexation **[15]**.

1. **La directionnalité**

C'est-à-dire l'ordre des termes. Dans certains cas la préservation de l'ordre est importante pour préserver les sens de l'unité d'indexation.par exemple,˝recherche d'information ″, dans d'autre cas l'ordre n'est pas important,˝Recherche et développement ″.peu de travaux existe en recherche d'information où sont utilisés les mots composés directionnels. La plupart des travaux exploitant les mots composés sont basés sur la non directionnalité de ces derniers **[15].**

2. **La distance**

La distance entre les termes formant le mot composés (l'adjacence ou le non-adjacence des termes), l'intensité de liens entre termes opérationnalisée à travers le distance reflète la proximité sémantique entre termes.la capture de cette proximité est importante pour la recherche d'information **[15].**

Les études effectuées en RI sur l'extraction des mots composés suppose que la cooccurrence des mots dans les éléments moins structurés (c.-à-d., des paragraphes ou des sections).

Ainsi, la recherche sur l'extraction des mots composés a été dominée par l'analyse de phrase .l'analyse empirique justifie de limiter l »extraction des mots composés aux combinaisons des termes apparaissant dans la même phrase **[15]**.Martin et al **[89]** ont constaté que 98% de combinaison syntaxique associent les termes qui sont dans la même phrase et sont séparés par cinq mots au plus.

Fagan **[47]** a constaté que la restriction de l'extraction des mots composés à une fenêtre de distance de cinq termes est presque aussi efficace que des mots composés extraits dans une phrase sans une telle restriction, soutenant ainsi les résultats de Martin et al **[89].**

3. La taille des mots composés

En principe la taille d'un mot composé peut être de n'importe quelle longueur (supérieur ou égale 2).Dans la pratique les mots composés longs conduisent à des index très spécifiques qui sont généralement moins utiles pour la RI **[15].**

4. La pondération des mots composés

Les différents schémas de pondération proposés pour l'attribution d'une poids à un mot simple dans un document, prennent générale ment en considération trois facteurs : le facteur de pondération local (tf), qui mesure l'importance du terme dans le document ;un facteur de pondération globale, mesurant le représentativité globale du terme dans la collection (idf) et un facteur de normalisation qui pend en compte la longueur de document**[15].**

Cependant, pour les mots composés, il n'y pas de schéma de pondération bien accepté.

En générale, trois approches sont proposées pour la pondération des mots composés :

- L'utilisation de fréquence (tf) du mot composé dans le document **[92]**;en se basant sur le fait que la fréquence d'un terme est corrélé avec son importance **[93]**.
- L'adaptation de schéma de pondération ($tf \times idf$) appliqué pour les mots simples. Comme c'est le cas dans **[46]**.
- L'utilisation des mesures d'association, telle que l'information mutuelle **[94]**.

5. Repérage des mots composés

Trois approches principales existent dans la littérature pour le repérage et l'extraction des mots composés.

a. **Approches linguistique :** ces approches se basent sur une analyse syntaxique partielle ou l'utilisation de patrons(Templates) syntaxiques pour détecter les mots composés. Le plus souvent, un ensemble de patrons syntaxiques comme (NOM NOM) ou (NOM PREP NOM) est utilisé pour l'identification. Malgré les nombreuses études consacrées à ce problème, il n'existe pas encore, à notre connaissance, une méthode effective qui permette de distinguer les termes des non termes d'un point de vue syntaxique **[15]**.Deux exemples d'outils issus de ces approches sont TreeTagger et AZNOUN PHRASER de l'université de l'Arizona.
Cependant, ces approches souffrent d'un inconvénient majeur puisque elles sont basées sur des règles, et ces règles sont dépendantes de la langue.

b. **Approche statistiques :** Les approches se basent sur la cooccurrence des termes dans le corpus pour extraire les mots composés et cela en partant de l'hypothèse que des termes (souvent réduits à deux ou trois mots) qui apparaissent ensemble dans le texte sont susceptibles de représenter un concept **[8].**
Les mots composés sont extraits ici soit en se basant sur leurs fréquences observées dans le corpus soit par l'utilisation des mesures d'association qui déterminent le degré d'association entre les mots composants.

c. **Les mesures d'association :** Les mesures d'association permettent de calculer « un score d'association »pour chaque paire de termes candidat dans le corpus ; ce score indique le potentiel de ce candidat d'être reconnu comme un mot composé **[15]**.
Plusieurs mesures d'association ont été proposées dans la littérature, telles que l'information mutuelle et le coefficient de Dice **[13].**Toutes ces métriques adoptent le postulat suivant : « les mots composés sont ceux dont les composants apparaissent ensembles plus souvent que par hasard », cela est obtenu en comparant la fréquence observée dans le corpus et la fréquence attendue (qui se base sur l'hypothèse d'indépendance des termes) **[15].**

Les approches statistiques ont un avantage considérable puisqu'elles ne nécessitent aucune autre information ou ressource pour l'extraction des mots composés. Elles

exploitent seulement les informations apparaissent dans le corpus, d'ou leurs flexibilité et portabilité (i.e. : elles ne dépendent ni de la langue ni du domaine traité par le corpus) **[15]**.

d. **Approches mixtes :** ces approches se basent sur les régularités statistiques et les patrons syntaxiques pour l'extraction des mots composés. Fagan**[47]**a comparé l'apport pour la RI des mots composés extraits statistiquement et des mots composés extraits linguistiquement, en utilisant l'analyse syntaxique, la troncature et la normalisation.

   L'évaluation a montré que les mots composés extraits linguistiquement est donné des résultats semblables ou plus faibles que les résultats obtenus avec les mots composés extraits statistiquement.les gains de performance constaté en utilisent les mots composés extraits statistiquement dans son expérience étaient de l'ordre de 17% à 39%.L'exploitation des mots composés dans le contexte du modèle de langue est présentée en détails dans le chapitre trois.

I.6.2. L'indexation sémantique

Tente d'apporter des solutions au niveau de la représentation des documents et des requêtes. L'objectif est d'indexer par les sens des mots plutôt que par les mots. Dans un contexte où l'ambigüité est présente, l'indexation sémantique est sensée à améliorer les performances de SRI, et elle s'intéresse d'abord de retrouvé le sens correct de chaque mot dans le document (resp. de la requête), ensuite représente ce document (resp. cette requête).

Pour retrouver les sens corrects des mots dans un document (resp. dans une requête), l'indexation sémantique requiert des techniques de désambigüisation sémantique (Word Sense Désambiguïsions WSD). Ces techniques se basent sur la sélection du sens approprié pour un terme dans un contexte donné, elles impliquent donc l'association pour un terme donnée dans un texte avec un sens ou une définition qui est distinguable des autres sens ou définitions attribué à ce terme.

I.6.3. L'indexation conceptuelle

Elle est né des travaux de Woods en 1997, qui a été le premier à proposer ce concept qui se refaire alors à la construction de taxonomie à partir de texte. Dans des travaux plus récents le terme indexation conceptuelle a été utilisée pour designer de manière plus générale une indexation à base de concepts issus de terminologie. Ces peuvent être générique ou spécifique à un domaine, Les structures conceptuelles peuvent être construite manuellement, automatiquement ou semi-automatiquement. Plusieurs travaux en RI ont utilisés ce type d'indexation dans des domaines spécifiques comme le domaine du sport.

I.7. L'évaluation de recherche d'information

L'évaluation constitue une étape importante dans la mise en œuvre d'un Système de recherche d'information. Elle permet de mesurer les caractéristiques du système en termes de qualité de service et de facilité d'utilisation **[26].**

Sans une évaluation propre, on ne peut pas :

Déterminer le taux de performance d'un Système de recherche d'information. Objectivement comparer ces performances avec ceux d'autres systèmes.

L'évaluation des performances d'un système de recherche d'information peut porter sur plusieurs critères. Selon Cleverdon (1970) **[29]**, les principaux critères pour mesurer la qualité d'un Système de recherche d'information se résument par :

- le temps de réponse.
- la présentation des résultats.
- l'effort requis de l'utilisateur pour retrouver, parmi les documents retournés, ceux qui répondent à son besoin.
- le taux de rappel du système.
- la précision du système.

L'évaluation d'un SRI consiste à comparer le résultat retourné par ce dernier par rapport au jugement de pertinence. Des mesures d'évaluations décrites dans la section suivantes sont utilisées pour effectuer cette comparaison. Les collections de test sont les résultats des projets d'évaluation qui se sont multipliés depuis les années 1970, on peut citer la collection CACM, la collection CISI, la compagne CLEF et la compagne TREC.

I.7.1. Les mesure d'évaluation

Etant donnée une requête Q, les documents de la collection peuvent être globalement classifiés en fonction de leur rapport à la requête en:

- ensemble de documents pertinents DP,
- ensemble de documents pertinents retrouvés DPR,
- ensemble de documents pertinents non retrouvés DPNR,
- ensemble de documents non pertinents DNP,
- ensemble de documents non pertinents retrouvés DNPR
- ensemble de documents non pertinents non retrouvés DNPNR.

Le principal objectif d'un système de recherche d'information est de restituer à l'utilisateur tous les documents pertinents et de rejeter les documents non pertinents.
Pour évaluer cet objectif, des mesures d'évaluation ont été définit **[27]** :

- **La précision** : représente le taux de document pertinents sélectionnés, exprimé ainsi :

$$P=\frac{|DP(R|}{|DPR\cup DNPR|} \qquad (I.20)$$

Une précision égale à 1 signifie que le système n'a retrouvé que des documents pertinents.

- **Le rappel :** représente le taux de documents pertinents sélectionnés parmi l'ensemble des documents pertinents, exprimé ainsi :

$$R=\frac{|DPR|}{|DP|} \qquad (I.21)$$

Un rappel égal à 1 signifie que tous les documents pertinents ont été retrouvés.
L'idéal serait d'avoir une précision et un rappel égaux à 1, signifiant que tous les documents pertinents sont retrouvés et qu'aucun document non pertinent n'a été retrouvé. En pratique, cet idéal n'est jamais atteint puisque ces deux quantités évoluent en sens inverse. Intuitivement, si on augmente le rappel en retrouvant plus de documents pertinents, on diminue la précision en retrouvant aussi plus de documents non pertinents. Inversement, une

plus grande précision risque de rejeter des documents pertinents diminuant ainsi le rappel.Les différentes valeurs de précision aux différents points de rappel servent àtracer la courbe Rappel/précision.

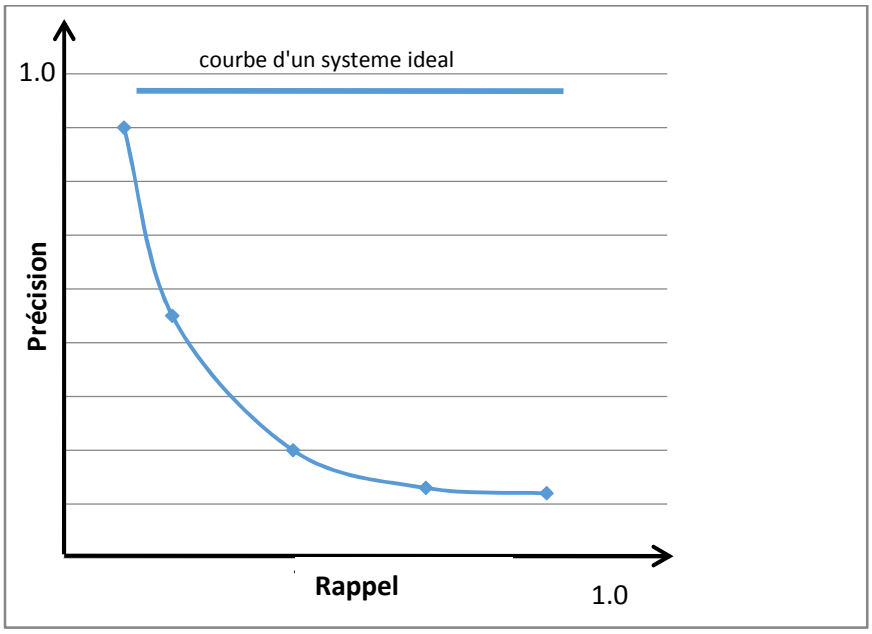

Figure I.2: courbe précision rappel

Des mesures complémentaires du rappel et de la précision sont respectivement : le silence et le bruit.

Le silence : la mesure dévaluation silence est une notion complémentaire au rappel, elle est définie par :

$$1- \text{R} = \frac{|DNPR|}{|DP|} \qquad \text{(I.22)}$$

Le bruit : la mesure dévaluation silence est une notion complémentaire à la précision, elle est définie par :

$$1-P = \frac{|DNPR|}{|DPR \cup DNPR|} \qquad \text{(I.23)}$$

Les mesures de rappel et de précision utilisées seules ne sont pas de bons Indicateurs de la performance d'un SRI. Pour remédier ce problème Plusieurs approches ont été proposées **[28]**, parmi elles :

- Agrégation du rappel et de la précision dans une seule mesure (F-score).

La F-mesure permet d'agréger le rappel et la précision dans une mesure unique.
Ce dernier est définit par défaut comme suit :

$$\text{F-score} = \frac{2 \times P \times R}{P + R} \qquad \text{(I.24)}$$

Courbes interpolées rappel/précision :

Cette courbe permet l'évaluation de la performance du système pour chaque requête. Pour avoir une évaluation de la performance du système sur toutes les requêtes et non pas sur une seule, on calcule une précision moyenne à chaque niveau de rappel appelé MAP (Mean Average Precision). Pour ce faire, Il faut unifier les niveaux de rappel pour l'ensemble des requêtes. On retient généralement 11 points de rappel standards, de 0 à 1 avec un pas de 0.1. Les valeurs de précision non obtenues à partir des valeurs de rappel sont calculées comme suit, par interpolation linéaire.

Pour deux points de rappel, i et j, i < j, si la précision au point i est inférieure à celle au point j, on dit que la précision interpolée à i égale la précision à j. Formellement :

$$P(r_j) = max_{r_{j>r\geq r_i}} P(r) \qquad \text{(I.25)}$$

Où $P(r_j)$est la précision interpolée au niveau standard de rappel r.

Vu que tous les systèmes sont évalues avec une courbe interpolée, l'interpolation ne donne pas plus d'avantage à un système qu'à un autre, et les courbes interpolées sont une base équitable pour comparer des systèmes.

I.7.2. Les mesures alternatives

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

La R-précision d'un SRI pour une requête « q » est la précision calculée au niveau de rang R, où R est le nombre de documents pertinents pour la requête « q ».cette précision alors mesure la proportion des documents pertinents retrouvés parmi les R premiers documents retournés.
Cette mesure est calculée comme suit :

$$R{-}prec = P@R = \frac{|DPR|}{R} \quad \text{(I.26)}$$

La R-précision est un bon paramètre pour observer le comportement d'un système pour chaque requête individuellement. La R-précision moyenne calculée sur toutes les requêtes n'a pas d'intérêt.

- **La précision moyenne non interpolée (Mean Average Precision: MAP)**

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

Première étape

L'idée est de calculer la moyenne de toutes les précisions obtenues après chaque document pertinent observé. Par exemple, on a R_q= {d3, d5, d9, d25, d39, d44, d56, d71, d89, d123}.Les valeurs de précision obtenues à chaque document observé sont: 1 ; 0,66 ; 0,5 ; 0,4 ; 0,33. Donc, La précision moyenne est calculée comme suit:

$$AveP(q) = P_{moy}(q) = \frac{1+ 0{,}66+ 0{,}5+ 0{,}4+ 0{,}3}{5} = 0{,}57$$

D'où la formule de précision moyenne est exprimée comme suit :

$$P_{moy}(q) = \frac{1}{N}\sum_{i=1}^{N} pr(d_i) \quad \text{(I.27)}$$

Avec :

$$pr(d_i) = \frac{r_{n_i}}{n_i} \quad \begin{cases} \text{Si } d_{ij}\ est\ retrouvé \\ 0 \quad \text{sinon} \end{cases} \quad \text{(I.28)}$$

Où :

n_i Dénote le rang du document d_i qui été retrouvé et qui est pertinent pour la requête.

r_{n_i}Représente le nombre de document pertinent retrouvé au rang (n_i).

N représente le nombre total de documents pertinents pour la requête (q).

Deuxième étape

Nous calculons la précision moyenne pour un ensemble de requête, en effectuant la moyenne de la précision moyenne de chaque requête, elle est exprimée ainsi :

$$MAP = \frac{\sum_{q \in Q} P_{moy}(q)}{|Q|} \qquad \text{(I.29)}$$

Q étant l'ensemble des requêtes.

I.8. Conclusion

Dans ce chapitre nous avons passé en revue les principaux concepts de la RI. Nous avons, particulièrement introduit des notions de base, telles que le besoin en information, la requête, le document et la pertinence. Nous avons aussi décrit les processus de base de la RI, à savoir l'indexation, l'appariement requête-document et reformulation de la requête. Ensuite, nous avons étudié les différents modèle de la RI. Enfin, l'évaluation des systèmes de recherche d'information est traitée.

A travers les différentes sections que nous avons présentées, nous ne concluons que la recherche d'information, s'attache à définir des modèles et des systèmes afin de faciliter l'accès à un ensemble de documents se trouvant dans des bases documentaires. Le but est de permettre aux utilisateurs de retrouver les documents dont le contenu répond à leur besoin en information.

Chapitre II: Expansion de la requète

II.1. Introduction

Les performances d'un Système de Recherche d'Information, mesurées en général par la double mesure rappel-précision, dépendent d'une part de l'efficacité du modèle de recherche mise en œuvre pour l'appariement entre requête documents, et d'autre part des requêtes formulées par l'utilisateur.

La requête initiale de l'utilisateur est souvent représentée par une liste de termes très réduite. Cette liste manque souvent des termes intéressants pouvant exprimer effectivement le besoin en information de l'utilisateur. Ceci a plusieurs raisons, la plus importante vient de la diversité du vocabulaire de la collection de documents.

Pour pallier ce problème, les Systèmes de Recherche d'Information (SRI) proposent des techniques, appelées expansion de requête, basée sur les caractéristiques de celles-ci pour améliorer automatiquement la requête initiale de l'utilisateur.

Dans ce chapitre en présente la reformulation de la requête plus claire expansion de requête automatique (les fonctionnalités et les différentes approches).

II.2.Expansion de requête

II.2.1. Définition

L'utilisateur formule son besoin en information par une requête composée de ses propres mots clés et le choix de chaque terme a une influence directe sur l'ensemble des documents restitués par le système. La requête initiale seule est souvent insuffisante pour permettre la sélection de document répondant au besoin de l'utilisateur. Pour cela, une étape d'expansion de la requête est souvent utilisée. Donc l'expansion de requête peut être définie comme un processus de transformation d'une requête d'un utilisateur dans le but de lui apporter des réponses les plus pertinentes possibles. La requête initiale donnée par l'utilisateur est modifiée si le système de recherche estime que les réponses qu'elle apporte ne sont pas satisfaisantes et les nouveaux résultats sont présentés à l'utilisateur **[31]**.

Plusieurs autres définitions ont été attribuées à l'expansion :

–Abberley**[32]** : définit l'expansion de requêtes comme un moyen qui permet de reformuler les requêtes et d'améliorer le processus de Recherche d'information.

– Elle est considérée selon **[33]** : 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**[34]**, en plus de proposer des classifications des méthodes d'expansion de requête, il a aussi donné la définition suivante: " l'expansion de requête 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.2.2. Les approches de l'expansion de la requête

Les systèmes de recherche d'information proposent différentes approches d'expansion de requête, pour affiner et améliorer la requête initiale de l'utilisateur d'une façon totalement ou partiellement automatique.

L'expansion de requête a été traitée selon deux (2) classes :

- Approche basée sur le contexte global ;
- Approche basée sur le contexte local ;

II.2.2.1. Approche basée sur le contexte global

Les techniques d'expansion globales correspondent à une analyse complète du corpus pour créer une source d'information qui sera utilisée pour l'expansion de la requête. Le but de ces approches (Saint Réquier et al. 2010) **[35]** est de compléter la requête initiale en utilisant de l'information globale provenant de la collection entière ou éventuellement de ressources sémantiques.

II.2.2.1.1.Utilisation de la collection

L'idée générale de cette approche est d'utiliser toute la collection de documents pour créer une source d'information qui sera utilisée pour l'expansion de la requête. Ce genre d'approche est souvent évalué avec des collections de tests de taille relativement petite par rapport aux collections actuelles, pour cela, on trouve de moins en moins d'études sur ce

genre d'approche, surtout dans un contexte Web **[36]**. Les premières techniques employées dans ce domaine **[37] [38]** sont basées sur la classification automatique de mots (term clustering), où sont ajoutés, pour chaque terme de la requête, tous les termes du même groupe que celui-ci. Ces groupes sont auparavant construits selon les statistiques de cooccurrence de termes dans toute la collection. L'efficacité de cette méthode n'a pas été confirmée par **[39]**. Ces auteurs ont argumenté que les termes qui ont un degré élevé de cooccurrence sont des termes très fréquents dans la collection de documents et ils sont donc peu discriminants et mauvais pour l'expansion de la requête. L'effet de la cooccurrence entre les termes sur la performance (en précision et en rappel) d'un modèle de recherche d'information n'a pas été étudié. Plus tard, **[40]** ont introduit le thésaurus de similarité pour l'expansion des requêtes. Un thésaurus de similarité est une matrice, construite en considérant les similarités terme à terme plutôt que les simples données de cooccurrence. Contrairement à une matrice de cooccurrence, ce thésaurus **[40]** est basé sur la façon dont les termes de la collection "sont indexés" par les documents.

Dans la construction d'un thésaurus de similarité, Chaque terme t_i de la collection est représenté comme un vecteur dans l'espace vectoriel de document **[42]** :$\vec{t_i}$= $(d_{i1}, d_{i2}, \dots, d_{in})$ où n est le nombre de document dans la collection et d_{ik} est le poids du document D_k dans la présentation du termet_i.

La formule de pondération utilisée par **[42]** pour calculer le poids d_{ik} est la suivante :

$$d_{ik}=\frac{(0.5+0.5\frac{tf(d_k,t_i)}{maxff(t_i)}\times iif(d_k)}{\sqrt{\sum_{j=1}^{n}\left(\left(0.5+0.5\frac{tf(j,t_i)}{maxtf(t_i)}\times iif(d_k)\right)\right)}} \qquad \text{(II.1)}$$

Où :

$tf(d_k, t_i)$: La fréquence du terme ti dans le document d_k ;

$idf(d_k)$=log($\frac{m}{|d_k|}$) : La fréquence inverse de d_k , avec :

m : Le nombre de termes dans la collection.

$|d_k|$: Le nombre de termes dans le document d_k.

$maxff(t_i)$: La fréquence maximale du terme t_i dans la collection.

Le thésaurus de similarité est construit en calculant la similarité entre toutes les paires de termes (t_i,t_j) de l'indexation. La similarité entre deux termes est exprimée par le produit scalaire suivant :

$$sim(t_i, t_j) = \vec{t_i} . \vec{t_j} = \sum_{k=1}^{n} d_{ik} . d_{jk} \quad \text{(II.2)}$$

En utilisant ce thésaurus, l'expansion d'une requête˝ q″ consiste à calculer une similarité, $sim_{qt}(q, t_i)$entre chaque terme du thésaurus et la requête q.

La formule de cette similarité est la suivante :

$$sim_{qt}(q, t_i) = \sum_{t_i \in q} q_i . sim(t_i, t_j) \quad \text{(II.3)}$$

Où

q_i : est le poids du terme t_i dans la requête q.

$sim(t_i, t_j)$: La similarité entre deux termes.

Ainsi, tous les termes de la matrice associés à la requête˝ q″, peuvent être ordonnés par ordre décroissant de leur valeur de similarité sim_{qt} .

Les poids des termes, rajoutés à la requête˝ q″ dans cette dernière, sont donnés par la fonction de pondération suivante :

$$w(q, t_j) = \frac{sim_{qt}(q,t_j)}{\sum_{t_j \in q} q_j} \quad \text{(II.4)}$$

II.2.2.1.2. Utilisation d'une ressource sémantique

L'utilisation d'une ressource sémantique pour l'expansion de la requête permet d'éviter la complexité élevée des approches basées sur toute la collection de documents **[36]**.

Les approches proposées dans ce domaine, ont fait l'objet de plusieurs études, comme celles de **[43] [44]**, **[45]**. Nous présentons dans cette section uniquement les approches qui utilisent une ressource sémantique générique comme Wordnet.

WordNet est une base de données lexicale **[46]**, conçue manuellement par des experts dans le but de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise. Malgré le fait que, des versions de WordNet existent pour d'autre langues, mais la version anglaise reste la plus complète pour l'instant.

Le contenu de la base de données WordNet couvre la majorité des noms, verbes, adjectifs et adverbes de la langue Anglaise structurés en un réseau de nœuds et de liens. Chaque nœud, appelé synset (set of synonyms), est constitué d'un ensemble de termes synonymes. Cela signifie que les synonymes ayant le même sens sont groupés ensemble dans un nœud pour former un synset. Chaque synset représente un sens unique d'un mot particulier. Un terme peut être un mot simple ou une collocation (i.e. deux mots ou plusieurs mots reliés par des soulignés pour constituer un terme complexe correspondant).

Les synsets de WordNet sont reliés par des liens ou relations sémantiques. La relation de base entre les termes dans WordNet est la Synonymie. Les différents synsets sont autrement liés par diverses relations sémantiques telles que la relation de subsomption (hyperonymie-hyponymie), et la relation décomposition (Meronymie-Holonymie).

II.2.2.2. Approche basée sur le contexte local

On parle de méthode locale si les termes ajoutes proviennent des documents résultants de la première recherche. Dans ce cas on trouve deux méthodes sont:

- La réinjection de pertinence.
- Le pseudo réinjection de pertinence.

II.2.2.2.1.La réinjection de pertinence

La réinjection de pertinence est l'une des méthodes de l'expansion de la requête, connue aussi sous le nom de Relevance Feedback (RF), est une technique utilisée pour améliorer la performance de la recherche d'information **[41] [47]**. Au cours de ce processus, l'utilisateur utilise une requête initiale, puis fournit un retour sur la pertinence des documents. Les termes de ces documents (jugés pertinents) sont donc ajoutés à la requête initiale. L'expansion par réinjection de la pertinence est une technique qui vise à enrichir la qualité de recherche

lorsque la seule évaluation de la similarité entre les requêtes et les documents n'est plus suffisante. Le principe de la reformulation par réinjection de pertinence se résume en quatre étapes principales **[48]**, à savoir :

a. Les utilisateurs effectuent une première requête;
b. Des documents sont retournés en fonction de cette première interrogation;
c. Les utilisateurs doivent ensuite indiquer parmi les documents retournés, lesquels sont pertinents, et/ou lesquels ne le sont pas;
d. La requête de départ est alors modifiée automatiquement pour tenir compte des jugements des utilisateurs.

La réinjection de pertinence peut passer par une ou plusieurs itérations pour une même séance de recherche : nous parlons alors de la réinjection de pertinence à itérations multiples **[24]**.

La technique de réinjection de pertinence a été mise en place à l'origine dans le modèle vectoriel. Rocchio **[49]** a proposé le modèle de reformulation de requête suivant :

$$Q_N = \alpha . Q_O + \beta . \frac{1}{|R|} \sum_{d_p \in R} d_n - \gamma \frac{1}{|NR|} \sum_{d_{np} \in NR} d_{np} \qquad \text{(II.5)}$$

Où :

Q_N: Le vecteur de la nouvelle requête ;

Q_O : Le vecteur de la requête originale ;

R: L'ensemble de documents pertinents ;

NR: L'ensemble de documents non-pertinents ;

d_p : Le vecteur associé à un document pertinent ;

d_{np} : Le vecteur associé à un document non-pertinent ;

α, β, γ: représentent les paramètres de la reformulation.

Nous pouvons remarquer que cette formule permet d'obtenir une nouvelle requête dont le vecteur se rapproche des vecteurs des documents jugés pertinents et s'éloigne des vecteurs des documents jugés non pertinents.

Dans le modèle probabiliste, la réinjection de pertinence est mise en place directement dans le modèle de mesure de pertinence. Elle consiste à revoir les poids des termes de la requête **[50],** comme suit :

$$w_{q_j}=\log\left[\frac{r_i+0{,}5/(R-r_i+0{,}5)}{(n_i-r_i+0{,}5)/(N-df_j-R+r_i+0{,}5)}\right] \qquad \text{(II.6)}$$

Où :

R: représente le nombre de documents pertinents ;

r_i: représente le nombre de documents pertinents contenant le terme t_i;

n_i : représente le nombre de documents contenant le termet_i ;

N: représente le nombre total de documents dans la collection.

Cependant, La réinjection de pertinence peut réaliser de très bonnes performances si les utilisateurs fournissent des jugements de pertinence suffisants et corrects (Cui et al. 2002) **[51]**. Mais malheureusement, cette méthode s'est montrée peu populaire pour les utilisateurs, et dans un contexte de recherche réel, ils sont souvent peu disposés à fournir ce genre d'informations de pertinences explicites, qui sont généralement ressenties comme une charge supplémentaire lors de leurs interactions avec le SRI **[52]**.

Pour surmonter les difficultés**[51]**, dues au manque de jugements de pertinences suffisants, la réinjection de pertinence est remplacée par la pseudo-réinjection de pertinence (qui est appelée également réinjection de pertinence aveugle ou réinjection locale).

II.2.2.2.2. La Pseudo réinjection de pertinence

La Pseudo-Réinjection de Pertinence (PRP) est une approche alternative à la réinjection de pertinence**[24]**, qui est appelé également la réinjection de pertinence implicite ou Aveugle, (En anglais, « Blind Relevance Feedback ») car elle utilise les techniques de réinjection automatique à l'aveugle pour construire une nouvelle requête. Elles se basent sur l'hypothèse que les documents les mieux classés (les premiers) sont considérés comme pertinents. Le système utilise alors les premiers documents pour reformuler la requête **[15]**.

La variante de la formule de Rocchio pour la réinjection automatique de la requête est exprimée par la formule suivante :

$$Q_N = \alpha . Q_O + \beta \frac{1}{|R|} \sum_{d_p \in R} d_p \qquad \text{(II.7)}$$

Nous voyons dans cette formule que l'expansion de la requête est uniquement positive, car on ne peut faire aucune hypothèse sur les documents non pertinents, mais rien ne nous empêche de prendre les derniers documents de la liste comme non pertinents **[13]**.

Cette technique est utilisée dans plusieurs systèmes de reformulation de requête et apporte de nombreux avantages aux SRI **[52]**. Néanmoins, cette méthode a un inconvénient évident **[53]**. Si une grande partie des documents les mieux classés par le système contiennent peu d'informations pertinentes ou aucune, alors les termes ajoutés par la réinjection pour l'extension de la requête sont susceptibles de causer une dégradation des performances. Ainsi, les effets de la réinjection de pertinence implicite dépendent fortement de la qualité de la recherche initiale.

Plusieurs travaux **[54]** ,**[55]** ont tenté d'évaluer l'impact de la pseudo-réinjection, en variant le nombre de termes à rajouter à la requête. Ils montrent que la performance du système est obtenue lorsque la requête est construite entre 20 et 40 termes.

II.2.3. Le processus d'expansion automatique de la requête

L'expansion de la requête comprend quatre étapes principales que nous allons définir par la suite **[56]**, présentées dans la figure (II.3)

- Traitement de données ;
- Génération et classement des termes ;
- Sélection des termes ;
- La reformulation de la requête initiale.

L'entrée de ce processus c'est la requête initiale de l'utilisateur et la source de données, et en sortie on aura la requête reformulée avec les termes d'expansion.

Nous schématisons ce processus avec la figure suivante :

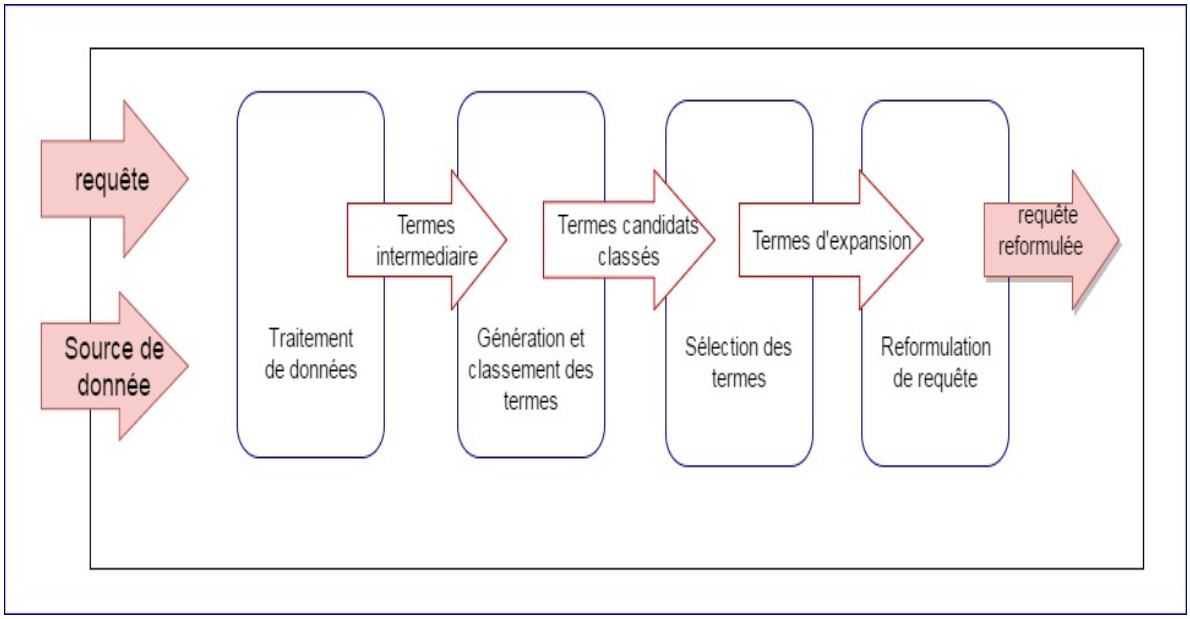

Figure II.1: Processus d'expansion automatique de la requête.

II.2.3.1. Traitement de données

Cette étape transforme la première source de donnée utilisée pour étendre la requête de l'utilisateur dans un format qui sera traité plus efficacement par les étapes suivantes. Elle consiste habituellement d'une phase d'extraction de termes pour faciliter l'accès et la manipulation des fonctions de traitement.

Prétraitement de la source de données (document ou autre) est généralement indépendant de la requête de l'utilisateur qui doit être étendue, mais elle est spécifique au type de source de données.

De nombreuses techniques d'extension de requête sont basées sur les informations contenues dans les tops documents, extraits en réponse à la requête de l'utilisateur initiale à partir d'une collection de documents. Dans cette étape de traitement de données, il est nécessaire d'indexer la collection et d'exécuter la requête.

En conséquence, chaque document est représenté comme un ensemble de termes pondérés, avec un fichier complémentaire inversé de l'indice qui associe les termes du document aux termes de la requête. Le système d'indexation peut également stocker les positions des termes afin de fournir la recherche basée sur la proximité. Lorsque la collection utilisée pour l'expansion de la requête est la même que celle en cours de recherche, le système de classement à lequel la requête étendue sera soumise est généralement utilisé pour effectuer aussi une première recherche de classement. Si un corpus externe est utilisé (par exemple, des données Web pour les recherches sur l'Internet), comme dans **[53]** ,**[57],[58],[59].**

D'autres techniques d'expansion de requêtes, basées sur l'analyse du corpus, nécessitent l'extraction des termes particuliers de la collection manuellement, qui sont généralement différents de ceux utilisés à des fins d'indexation par un Système de Recherche d'Information (SRI) classique. Une approche bien connue de Qiu et Frei **[60],** où chaque terme est représenté comme un vecteur de documents pondéré en utilisant les statistiques d'une collection non-standard. Un autre exemple de Crouch et Yang **[61]**, qui construit un thésaurus de statistique en regroupant d'abord l'ensemble de la collection de documents via l'algorithme complet de clustering.

Il existe peu de travaux réalisés pour la sélection des documents comparativement à la sélection des termes, la méthode présentée dans **[62]** considère le document pertinent comme un bon représentatif d'un ensemble de D_{rel} (les documents pertinents dans le corpus) dans la mesure où il peut aider efficacement pour trouver les documents pertinents à partir de D_{rel}par l'intermédiaire d'une recherche effectuée sur le corpus. L'objectif de cette méthode est de sélectionner un ensemble de K documents pertinents représentatifs qui aide à trouver les documents pertinents lors de l'utilisation de retour de pertinence basée sur la recherche pour classer tous le corpus, les performances de recherche qui en résultent seront optimales en ce qui concerne tous les sous-ensembles des k documents dans D_{rel}.

Deux (2) grandes familles de méthodes sont présentées pour la sélection de documents :

La première :

Elle estime la représentativité d'un document indépendamment des autres documents dans D_{rel}, en sélectionnant les k-tops documents classés, cette méthodes assigne au documents d (appartient àD_{rel}) une pertinence, RSV(d) qui reflète la représentativité de d dans D_{rel}.

Les k-tops documents dans D_{rel} sont ensuite utilisés comme entrée à la méthode de sélection des termes d'expansion. Parmi les caractéristiques utilisées pour la sélection des documents :

– La méthodes Random :

Elle affecte une pertinence à chaque document selon la fonction suivante :

$$\mathrm{RSV}_{(\mathrm{random})}(\mathrm{d})^{\mathrm{def}} = \frac{1}{\mathrm{n}} \qquad \text{(II.8)}$$

Puis choisir les k-documents à partir de D_{rel} au hasard.

Où

n : représente le nombre de documents dans la collection.

– La méthode QuerySim :

Elle considère le document (d) qui a une similarité élevée avec la requête comme un bon représentant et le calcul du la pertinence (RSV) pour chaque document se fait selon la fonction suivante :

$$RSV_{QuerySim}(d)^{def} = sim(q, d) \qquad \text{(II.9)}$$

– La méthode basée sur la longueur d'un document :

Elle suppose que les documents pertinents courts sont les meilleurs représentants que les documents longs pertinents :

$$RSV_{length}(d)^{def} = -|d| \qquad \text{(II.10)}$$

Où

$|d|$: Représente la taille de document d.

La deuxième :

Elle est basée sur l'hypothèse suivante: les documents représentatifs sont semblables les une des autres en exploitant les relations (similitude) entre les documents dans D_{rel}. Cette

méthode nécessite une connaissance de tout l'ensemble de documents pertinents. A titre d'exemple :

– La méthode centroïde:

En termes de modèle de langage, le centroïde est une probabilité de distribution de tout le vocabulaire.

$$p(w\backslash Cent(D_{rel}))^{def} = \frac{1}{n}\sum_{d\in D_{rel}} p(w\backslash d) \qquad \text{(II.11)}$$

Ensuite, la méthode centroïde permet d'estimer les documents représentatifs en utilisant le KL-divergence du modèle de langue induit à partir du centroïde :

$$score_{centroid}(d)^{def} = (-|d| * \left(p.\backslash Cent(D_{rel})\right)||p(.\backslash d)) \qquad \text{(II.12)}$$

– Les méthodes basées sur le graphe:

Certains travaux sur le reclassement de la première liste de documents qui sont très semblables aux autres documents dans la liste ont une forte probabilité de pertinence. L'idée est que ces documents représentent la liste entière, par la vertu de la façon dont la liste a été créée, c'est-à-dire en réponse à la requête, ils pourraient être pertinents au besoin fondamental de l'information. Toutefois, la liste est composée de plusieurs documents à la fois pertinents et non pertinents.

II.2.3.2.Génération et classement des termes candidats d'expansion

Dans la deuxième étape de l'expansion automatique de la requête, le système génère et classe les termes candidats d'expansion. La raison pour laquelle le classement est important, c'est que la plus part des méthodes d'expansion de requête, ne pourront choisir qu'un petit nombre de termes candidats d'expansion à ajouter à la requête initiale.

L'entrée de cette phase est la requête d'origine et la source de données transformée; le résultat est un ensemble de termes d'expansion, généralement avec des pertinences associées. La requête initiale peut être prétraitée pour supprimer des mots communs et/ou extraire des termes importants pertinents.

Nous classons les techniques utilisées pour exécuter la génération et le classement des termes candidats selon :

- le type de relation entre les termes d'expansion générés;
- les termes de la requête initiale.

Il existe deux (2) types d'association:

– Association un-à-un.

– Association un-à-plusieurs.

A. Association un à un

La forme la plus simple de génération et de classement des termes candidats est basée sur les associations un-à-un entre les termes d'expansion et les termes de la requête initiale, c'est-à-dire, 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 pour chaque terme de la requête à l'aide d'une variété de techniques.

L'une des ces techniques consiste à s'appuyer sur les associations linguistiques, comme l'utilisation d'un algorithme de lemmatisation qui consiste à regrouper les mots d'une même famille et les réduire en une entité appelée lemme.

Les approches statistiques consistent à analyser un document, en évaluant les éléments d'un document par leur fréquence d'occurrence dans un document. Ces statistiques peuvent être utilisées pour créer des index ou extraire les concepts d'un domaine en vue de sa modélisation.

Par contre, l'approche linguistique consiste à calculer automatiquement la similarité terme-à-terme dans une collection de documents.

L'idée générale est que les deux termes sont sémantiquement liés lorsqu'ils apparaissent dans les même documents, tout comme deux documents sont considérés comme similaires s'ils contiennent les mêmes termes. Deux mesures de similarité utilisées dans le calcul sont :

- Le coefficient Dice (D) ;
- l'indice de Jaccard (J).

Compte tenu des termes u et v, le coefficient Dice (D) est définit comme suit :

$$D = \frac{2.df_{u\cap v}}{df_u + df_v} \qquad \text{(II.13)}$$

Où

$df_{u\cap v}$: représente le nombre de documents qui contiennent à la fois les termes u et v;

df_u,df_v; Représentent les nombres de documents contenants les termes u et v respectivement.

L'indice de Jaccard (J) est définit comme suit :

$$J = \frac{df_{u\cap v}}{df_{u\cup v}} \qquad \text{(II.14)}$$

Où

$df_{u\cup v}$: représente le nombre de documents contenants le terme u ou le terme$v$.

Une approche plus générale est définit dans ce cas. Si on considère une matrice ⱡAⱡ terme-document où chaque cellule $A_{t,d}$ représente le poids $W_{t,d}$ pour le terme (t) dans le document (d), et si considère une matrice C de similarité terme-à-terme alors est calculée ainsi :

$C = AA^t$

Où

$C_{u,v}$: représente le degré de similarité entre le terme$u$ et le terme v .

Alors $C_{u,v}$ est données comme suit :

$$C_{u,v} = \sum d_j \, W_{u,j}.W_{v,j} \qquad \text{(II.15)}$$

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 :

$$\frac{C_{u,v}}{\sqrt{\sum d_j W^2{}_{u,u} \sum d_j W^2{}_{v,v}}} \qquad \text{(II.16)}$$

La formule (II.16) 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 **[63]** 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 terme 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. Une mesure plus complète pour l'association de mot, qui intègre la dépendance entre termes c'est l'information mutuelle **[64]**, **[65]**.Elle est définit ainsi :

$$I_{u,v} = log_2\left[\frac{P(u,v)}{P(u).P(v)} + 1\right] \qquad \text{(II.17)}$$

Où :

$P(u,v)$: représente la probabilité conjointe que le terme u et le terme $v$apparaissent dans un certains contexte (généralement un document);

$d(u), d(v)$:Represente 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 :

$$P(v,u) = \frac{P(u,v)}{P(u)} \qquad \text{(II.18)}$$

De ce fait, les règles d'association ont été utilisées afin de trouver les termes d'expansion en corrélation avec les termes de la requête **[66], [67].**

B. Association un-à-plusieurs

Association un-à-un à tendance à ajouter un terme quand il est fortement lié à l'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 par**[68],** par exemple, si le terme « 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'application », mais pas pour d'autres requêtes comme : « programme TV », « programme spécial ». Ici encore, nous rencontrons la question de l'ambiguïté de la langue.

Le principe de l'approche association un-à-plusieurs est d'étendre l'association un-à-un pour les autres termes dans la requête. L'idée est que si un terme d'expansion est corrélé à plusieurs termes de la requête, donc un terme est corrélé à la requête dans son ensemble.

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 pertinences trouvés pour trouver la corrélation de la requête (q) globale :

$$C_{q,v} = \frac{1}{|q|}\sum_{u\in q} C_{u,v} \qquad \text{(II.19)}$$

Une approche similaire a été proposé dans **[69]** et **[70],** et plusieurs autres travaux de recherche ont suivi **[71],[72],[73]** et **[74].**

La formule (II.15), dans **[75]** 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.

II.2.3.3.Sélection des termes

Après avoir classé les termes candidats. Les principaux éléments (termes) sont sélectionnés pour l'expansion de la requête.

Plusieurs techniques pour la sélection des termes ont été proposées, elles utilisent des 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.

Dans **[77]** 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'information est créés dans lequel les termes simples sont étiquetés comme bons ou mauvais selon leurs influences sur les résultats de la recherche.

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

- **Salton et Buckley [78]** : 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 [79] et Haines [80] :** 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.
- **Harman [81]** : propose les fonctions suivantes :

$$SV(i) = \frac{RT_j * df_i}{N} \qquad \text{(II.20)}$$

Où

RT_j : représente le nombre total de documents retrouvés par la requête ;

df_i : représente la fréquence d'occurrence du terme t_i dans la collection ;

N : représente le nombre total de documents dans la collection.

$$SV(i) = \frac{r_i}{R} - \frac{df_i}{N} \qquad \text{(II.21)}$$

Avec

r_i : représente le nombre de documents pertinents contenant t_i;

R : représente le nombre de documents pertinents.

$$SV(i) = log_2 \frac{p_i(1-q_i)}{(1-p_i)} \quad \text{(II.22)}$$

Avec :

p_i: représente la probabilité que t_i appartienne aux documents pertinents ;

q_i: représente la probabilité que t_i appartienne aux documents non pertinents.

Les expérimentations réalisées sur différentes collections standards, ont révélé que la fonction (II.22) est la meilleure.

II.2.3.4. La reformulation de la requête

La reformulation de la requête 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 (SRI) ce qui revient généralement à affecter un poids à chaque terme qui décrit la requête étendue. Il existe plusieurs techniques de pondérations de requêtes, la plus populaire est reproduite à partir de la formule Rocchio pour la réinjection de pertinence **[82].** La formule générale est donnée comme suit :

$$W_{t,q'}^{'} = (1 - \lambda) \times W_{t,q} + \lambda \times RSV_t \quad \text{(II.23)}$$

Où

$q^{'}$: représente la requête étendue.

Q : représente la requête originale.

λ : représente le paramètre de pondération.

RSV_t : représente le poids attribué au terme d'expansion t.

Plusieurs techniques de normalisation simple, discutées dans **[83],** ont été proposées, elles produisent des résultats comparables en général. Afin d'optimiser les performances, la valeur λ peut être ajustée si les données sont disponibles. Donner plus d'importance par exemple au terme de la requête initiale deux (2) fois plus que les termes d'expansion. Ou autres possibilités comme la technique suggérée dans **[84]** et qui consiste à utiliser une formule de pondération de requêtes sans paramètres.

II.2.4. L'expansion de la requête dans le modèle de langue

Nous présentons ci-dessous les méthodes, les plus en vue, d'expansion de la requête dans le cadre du modèle de langue.

II.2.4.1. Modèle de pertinence [88]

Au lieu de modéliser la recherche d'Information comme processus de génération de la requête, Lavrenko et Croft ont proposé de modéliser explicitement le modèle de pertinence. Ils ont en effet, proposé d'estimer ce modèle à partir du modèle de la requête sans utiliser les 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 (θ_R).

Le but est alors d'estimer la probabilité P (t|θ_R), de générer un terme à partir du modèle de pertinence. Comme le modèle de pertinence n'est pas connu, les auteurs ont suggéré d'exploiter les documents les mieux classé (retour de pertinence) en assurant qu'ils sont générés du modèle de pertinence. Ce modèle est formalisé comme suit :

$$P(t|\theta_R)=\sum_{d\ \epsilon R} p(d) \times p(t|d)\ \times \prod_{i=1}^{k} p(q_i, d) \qquad \text{(II.24)}$$

Où

R : l'ensemble de documents de retour de pertinence,

P(d) : représente la probabilité de choisir un document d des documents de retour de pertinence.

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 $p(q_i, d)$.

Les résultats des expérimentations ont montré que cette approche améliore sensiblement les performances de la recherche d'information, de 10% à 20% d'amélioration de précision moyenne par rapport au modèle de langue de base **[89], [90].**

II.2.4.2. Modèle model-based feedback [91]

Dans la même optique que **[88],** Zhai et Lafferty ont proposé un modèle nommé model-based feedback, où le nouveau de la requête est obtenu par l'interprétation du modèle original de la requête avec le modèle de matière θ_T(Topic model), obtenu en utilisant les documents les mieux classés (retour de pertinence). La construction du modèle $\theta_T$consiste en l'extension d'une partie des documents retournés pertinents qui est distincte de l'ensemble des documents de la collection. Comme les documents les mieux classés sont susceptibles de contenir à la fois des informations pertinentes génériques(ou même non pertinentes), ils peuvent être représentés par un modèle génératif mixte qui combine le modèle θ_T (à estimer) et le modèle de langue de la collection. Le logarithme de la probabilité des documents les mieux classées est donné comme suit :

$$\log p(r|\theta_T) = \sum_{d \in R} \sum_t c(t,d) \log((1-\lambda)\, p(t|\theta_T) + \lambda\, p(t|c)) \quad \text{(II.25)}$$

Où

R : représente l'ensemble des documents les mieux classés ;

C(t,d) : représente le nombre d'occurrence du terme t dans le document d ;

λ: représente le poids d'interpolation.

L'algorithme EM (Expectation-Maximisation) est ensuite utilisé pour extraire le modèle θ_T.

II. 3. Conclusion

Dans ce chapitre, nous avons présenté deux aspects, dans la première section nous avons détaillé l'expansion de requête, dans laquelle, nous avons énuméré et expliqué les principes approches de l'expansion. Ensuite nous avons cité les étapes de processus d'expansion

automatique de la requête, dont nous avons expliqué le prétraitement des données, la génération et le classement des termes candidats d'expansion, la sélection des termes et la reformulation de la requête, et enfin, nous avons abordé l'expansion de requête .

Chapitre III: Evaluation et expérimentation

Chapitre III: Evaluation et expérimentation

III.1. introduction

Un système de recherche d'information doit faire deux opérations l'indexation de documents-requêtes et la recherche. Un problème dans la RI est l'ambigüité des termes et l'utilisation des termes simple est la disparité des termes pour résoudre ce problème divers approches sont proposées.

Le chapitre est organisé comme suit : dans la première section, nous présentons notre environnement de développement. Dans la seconde section nous décrivons notre approche. Le dernier point concerne la présentation des résultats d'expérimentation obtenus.

III.2. L'environnement de développement

Dans ce qui suit nous allons présenter notre environnement technique et préciser les différents outils utilisés : la plateforme Terrier, le langage de programmation JAVA et Netbeans.

III.2.1.plateforme Terrier

Terrier (Terabyte Retriever) est un outil de recherche gratuit, très flexible, efficace et effectif, facilement déployable sur de grandes masses de collections de documents.

Terrier implémente plusieurs fonctionnalités de recherche et fourni une plate forme idéal pour le développement rapide et l'évaluation pour les applications de recherche à grande échelle. C'est une plate forme, complète et transparente pour la recherche et l'expérimentation dans la recherche de texte, cette recherche peut être effectuée facilement sur les collections de tests standard TREC et CLEF. Il est développé par l'université de Glasgow, Ecoles des sciences informatiques débuté en l'an 2000. C'est un projet open source écrit en Java, fonctionne sous différentes plateformes: Windows, Mac OS X, Linux, Unix.

Comme tous moteurs de recherche, Terrier permet :

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

III.2.1.1. Architecture de Terrier

La figure ci-dessous montre l'architecture générale de Terrier :

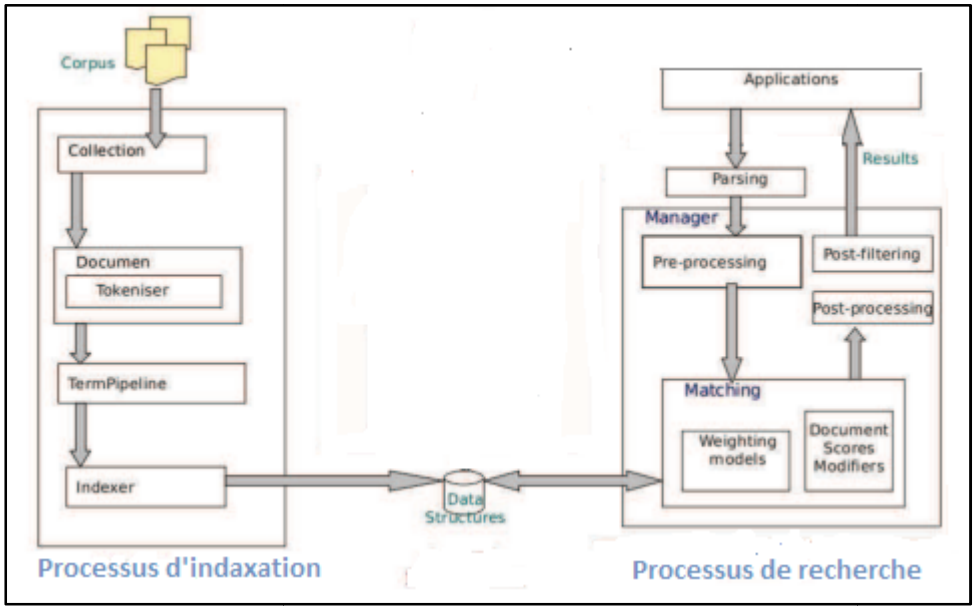

Figure III.1 : Vue d'ensemble de l'architecture Terrier

III.2.1.1.1.Processus d'indexation

L'indexation dans Terrier est divisée en quatre procédures et à chaque procédure, des classes java peuvent être ajoutées pour la personnalisation du système.

Les quatre procédures sont :

1. Splitter la collection de documents : consiste à parcourir l'ensemble du corpus reçu en entrée par Terrier et envoyer chaque document à l'étape suivante.
2. Extraction des termes (Tokenize document) : qui consiste à parser chaque document reçu et extraire les différents termes.

3. Traitement des termes extraits avec Term-Pipeline : consiste à l'élimination des mots vides et la lemmatisation des termes.
4. La construction de l'index.

La figure ci-dessous donne une vue d'ensemble d'interaction des composants principaux impliqués dans le processus d'indexation.

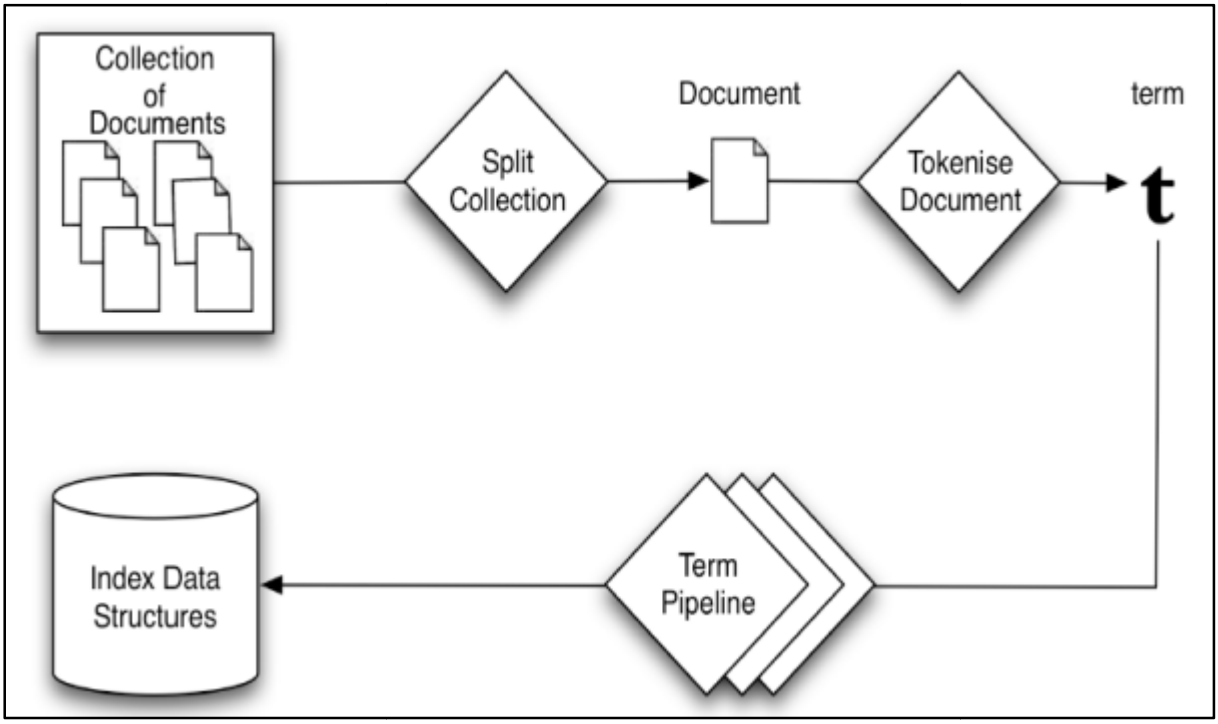

Figure III.2 : Le processus d'indexation dans Terrier

Les différentes classes associées au processus d'indexation sont organisées dans un ensemble de package, on cite :

Org.terrier.indexing : ce package contient les différentes classes permettant de réaliser un ensemble d'opérations sur la collection des documents, dans le but d'extraire les termes de tous les documents de la collection.

Org.terrier.terms : les classes de ce package permettent d'effectuer un ensemble de traitements sur les termes extraits. Parmi ces traitements, l'élimination des mots vides, lemmatisation des termes, …etc.

Org.terrier.structures : les classes de ce package permettent la construction d'un ensemble de structures ou un ensemble de données stockées. Parmi ces structures, on a :

- **Lexicon :** contient les informations sur chaque terme de la collection (Terme, Id terme, nombre de documents qui contiennent le terme, fréquence du terme dans la collection, Offset dans le fichier inverse).
- **Direct index :** il enregistre pour un document les termes qui apparaissent dans ce dernier. Il est souvent utilisé pour la reformulation de la requête, la classification et la comparaison des documents.
- **Inverted index :** contrairement à l'index direct, il enregistre pour un terme les documents dans lesquels il apparait, il contient aussi la position de chaque terme **et sa fréquence dans ces documents.**
- **Document index :** contient des informations sur les différents documents de la collection(Id Terme, Fréquence terme, #Fields)

III.2.1.1.2. Le processus de recherche

Durant le processus de recherche, chaque requête doit passer par les étapes suivantes :

1. **Query :** est une classe abstraite qui représente la requête.

   Terrier supporte trois modèles de requête :

   - *Single Term Query* : désigne la requête qui contient un seul terme.
   - *Multi Term Query* : désigne la requête qui contient plusieurs termes.
   - *Field Query* : terme qualifié par un champ.
2. **Parsing :** est l'opération qui se charge de tokenizer la requête.
3. **Pré-processing :** est l'opération qui applique le TermPipeline à la requête. Elimine les mots vides et les lemmatises.
4. **Matching :** est l'opération responsable de l'initialisation du Weighting Model et du calcul des scores entre la requête et les documents.
   - **Weighting Models :** est une interface qui assigne un score pour chaque terme de la requête dans le document.

     Terrier offre plusieurs classes qui implémentent cette interface, parmi eux (TF-IDF, BM25).

- **Document Score Modifiers :** permet de modifier le score des documents en fonction du langage de la requête.
- **Term Score Modifiers :** permet de modifier le score des documents en fonction de la position des termes.

5. **Post-processing :** est l'opération appropriée pour implémenter des fonctionnalités qui apportent un changement à la requête originale. Un exemple de Post-processing est l'expansion de requête, puis exécute une autre fois le Matching avec cette nouvelle requête. Un autre exemple de Post-processing est l'application de clustering.
6. **Post-filtering :** est l'étape finale du processus de recherche de Terrier où une série de filtres peut enlever les documents déjà recherchés, qui ne satisfont pas une condition donnée.

A la suite des étapes précédentes, Terrier s'occupe de retourner un ensemble de documents triés selon l'ordre décroissant de leurs scores.

La figure ci-dessous donne une vue d'ensemble d'interaction des composants de Terrier dans la phase de recherche.

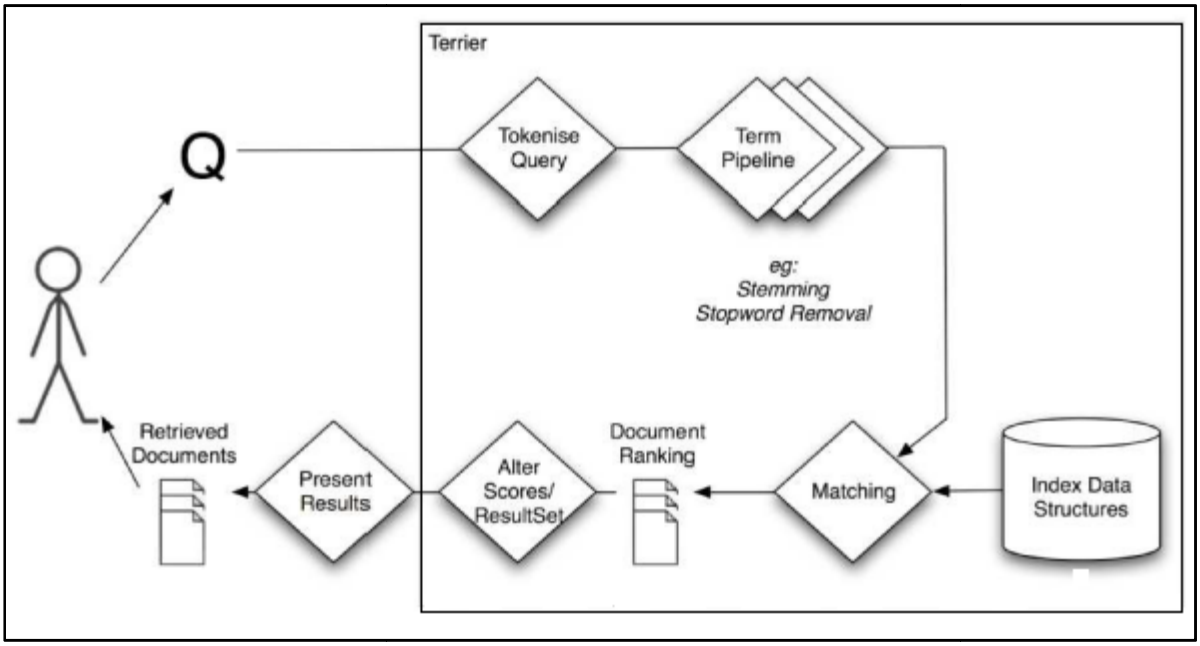

Figure III.3 : Processus de recherche dans Terrier.

III.2.2. Langage java

Java est un langage de programmation moderne développé par Sun Microsystems (aujourd'hui racheté par Oracle). Il ne faut surtout pas le confondre avec JavaScript (langage de scripts utilisé principalement sur les sites web), car Java n'a rien à voir. Une de ses plus grandes forces est son excellente portabilité :une fois votre programme créé,il fonctionnera automatiquement sous Windows, Mac, Linux, etc.

III.2.3.Netbeans

Netbeans IDE (souvent appelé Netbeans) est un environnement de développement gratuit et intégré écrit entièrement dans le langage de programmation Java et fonctionnant sur la plate-forme Netbeans, placé en open source par Sun en juin 2000.L'IDE Netbeans fonctionne sur de nombreuses plates-formes, telles que Windows , Linux , Solaris et MacOs . Netbeans IDE offre aux développeurs les outils dont ils ont besoin pour créer des applications professionnelles bureautique, commerciale, Web et multiplateforme. Un environnement Java développement Kit JDK est requis pour les développements Java.

La figure suivante illustre l'environnement de développement Netbeans :

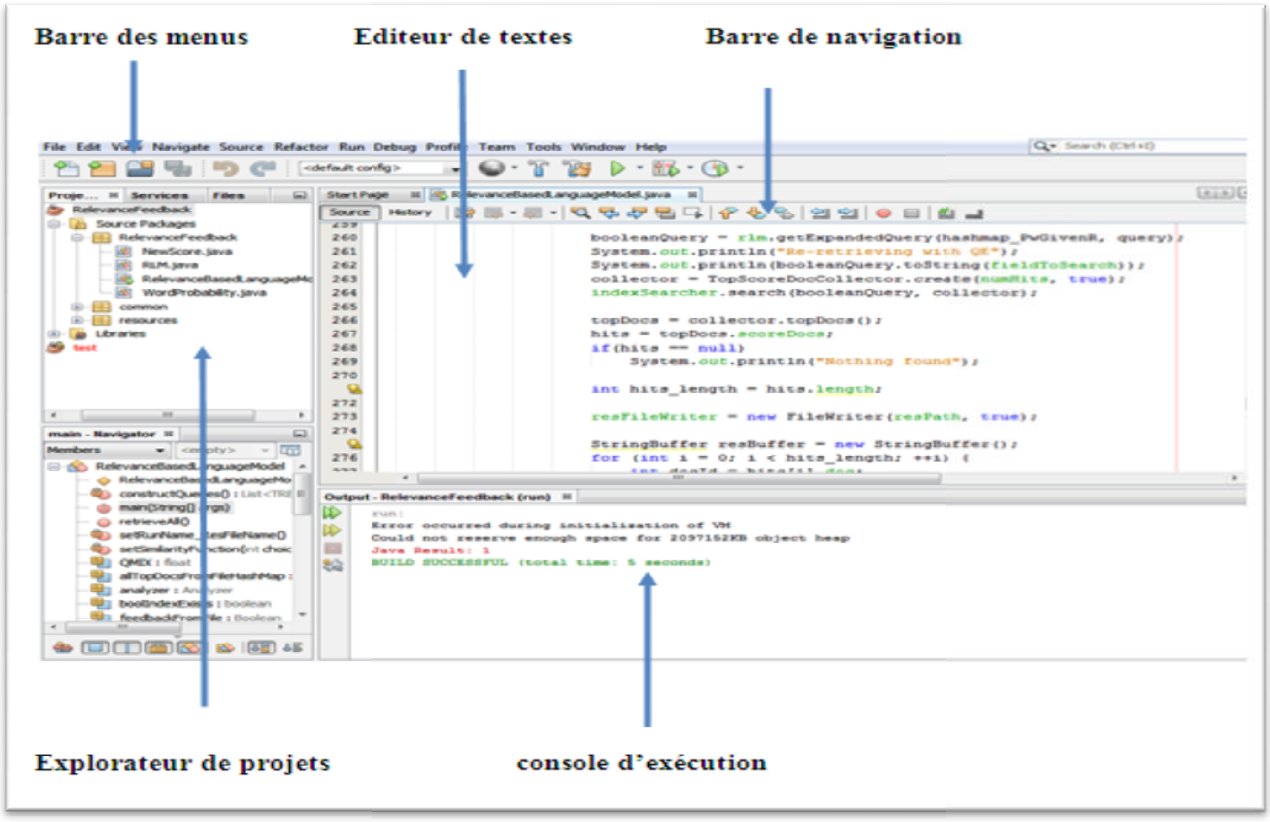

Figure III.4 : Environnement de développement de NetBeans

III.2.4.Outil TEXT-NSP

Le paquet de statistiques de Ngram (NSP) est une collection de modules de Perl qui facilitent en analysant Ngrams dans des fichiers texte. Nous définissons un Ngram comme ordre `n" tokens (terme) ; marqués qui se produisent dans une fenêtre au moins de `n" tokens marqués dans le texte ; se qui constitue un`n" tokens ; peut être définis par l'utilisateur. Les modules sous le texte :: NSP :: Mesure d'instrument de l'association qui sont employées pour évaluer si la cooccurrence des mots dans Ngram est purement par hasard ou statistiquement significatif.

III.3.Architecture générale de notre approche

La figure suivante illustre l'Architecture de notre approche, plus précisément :

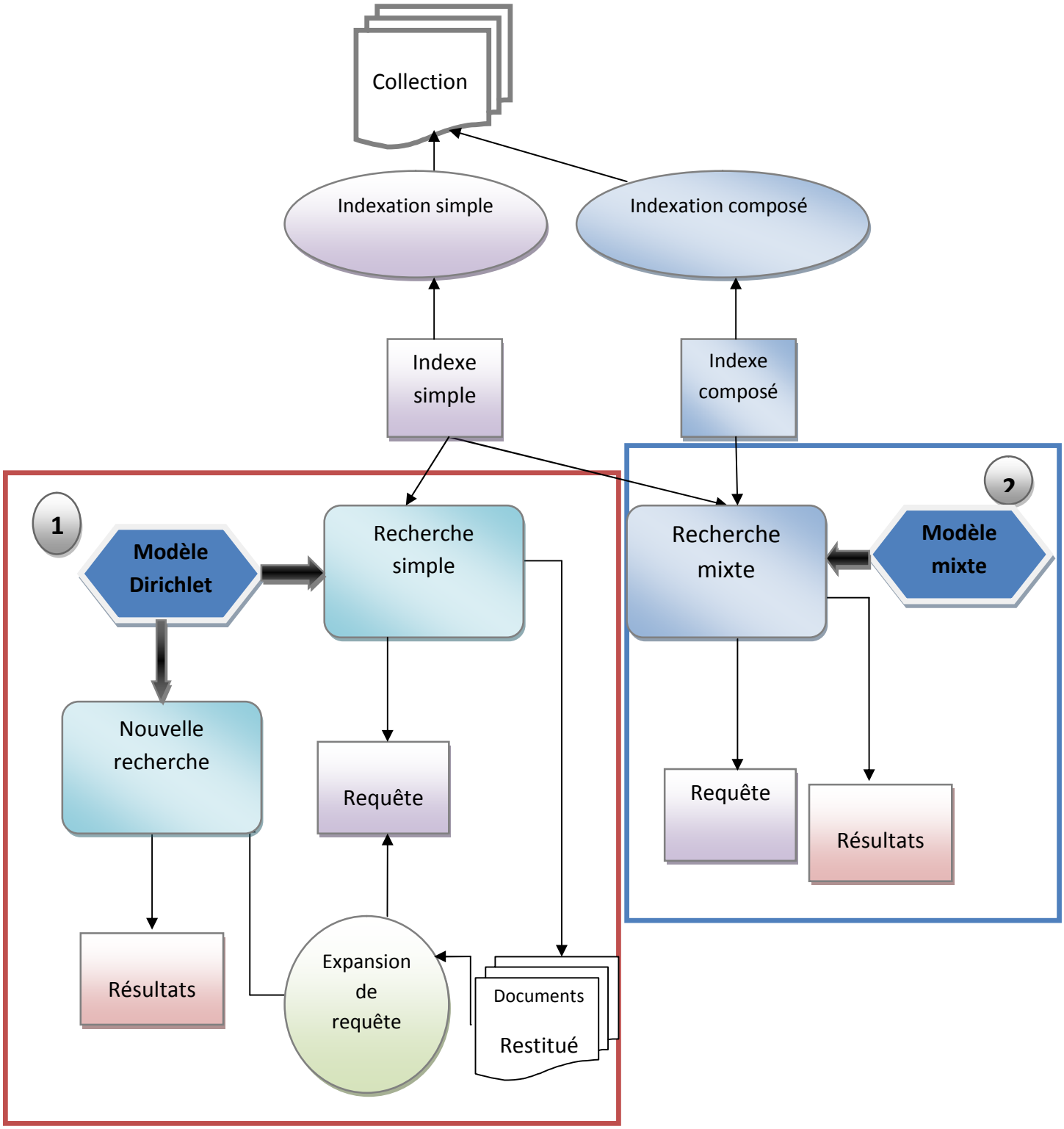

Figure III.5 : Architecture générale de notre approche

III.4. Présentation de notre approche

Notre travail se base sur une approche qui tente de réduire l'ambigüité des termes. Dans le processus d'indexation l'unité d'indexe est un terme, cette approche d'indexation rencontre le problème d'ambigüité (les phrase perdent leur sens).dans le cadre de notre travail s'intéresse à se problème. Nous proposons l'approche d'indexation avec les mots composés, cette approche se base sur l'unité d'indexation, un mot composé définit comme tout terme adjacent non vide.

De plus de mot composé, l'expansion de requête est une approche qui remédie le problème d'ambigüité et la disparation des termes. Nous avons implémenté cette approche.

Dans le premier point, nous allons présenter notre travail qui consiste en l'implémentation d'une nouvelle approche d'indexation et de recherche des documents sous la plateforme terrier.

Dans le deuxième point, l'implémentation de la recherche et l'expansion de requête simple sous terrier.

III.4.1.le processus d'indexation

A. **Le prétraitement de la collection:** en premier lieu, nous procédons au prétraitement de la collection. Nous Parsons les documents, nous éliminons les mots vides et nous appliquons l'algorithme de Porter [152]. Les documents traités obtenus sont ensuite utilisés comme entrées par l'outil Text-NSP [13] pour l'extraction des mots composés**.**
B. **L'extraction des mots composés :** pour l'extraction des mots composés nous avons utilisé l'outil Text-NSP. Le package Text-NSP est un outil permettant l'identification et la sélection de n-grammes ou séquence de mots dans une collection de texte. Dans le processus d'extraction des mots composés, nous tenons compte des paramètres suivants:

   La directionalité entre mots simples : dans certains cas la préservation de l'ordre des mots est important pour garder le sens de l'unité d'indexation, ceci est vrai par exemple pour le terme «système d'exploitation », dans d'autres cas l'ordre n'est pas important, par exemple le terme «système et organisation».

La distance : la distance entre les termes formant le mot composé (ou l'adjacence ou la non-adjacence des termes) : l'intensité de liens entre termes – opérationnalisée à travers la distance- reflète la proximité sémantique entre termes. La capture de cette proximité est importante pour la recherche d'information.

La taille des mots composés : touts N-gramme (N supérieur ou égale à 2),forment les mot composés. Dans notre cas, nous s'intéressons à la taille égale à deux, un mot composé de deux mots simples. Pour former la liste des termes nous utilisons l'outil Text-NSP, ce dernier est composé de deux processus principale pour former la liste des mots composés **: (1)** « count.pl » est un module qui permet de sélectionner les bi-grammes avec une fréquence minimale précisé, une fréquence de terme supérieure à un seuil donné noté « seuil_freq ».une fois la liste des mots composés formé en sortie de « count.pl », cette liste passe en entrée de deuxième module. (2) « statistic.pl », donne en sortie une liste des bi-grammes avec leur différente fréquence calculée avec des mesures statique. Dans notre cas, Nous avons utilisé la mesure de Pointwise Mutual Information (PMI).L'étude menée par Petrovic et al [150] a montré que la mesure PMI permet l'identification de mots composés pertinents pour la RI. Nous ne gardons dans la liste finale que les bi-grammes ayant un score supérieur à un seuil noté « seuil_PMI ». Cette liste est ensuite utilisée dans les étapes d'indexation et de recherche.

III.4.2. Recherche simple

Dans notre approche la base des indexes produite par le processus d'indexation simple sous terrier, la recherche se base sur le modèle Dirichlet. Ce dernier est une technique de lissage de modèle de langue, son principe de ne pas avoir une probabilité nulle.

III.4.3. Recherche mixte

En se basant sur la nouvelle structure de donnée index produite par le processus d'indexation précédent, le modèle de recherche assigne un score pour chaque terme de la requête (indexé avec les composées) dans le document, avec le modèle mixte.

A. Le modèle mixte

En suivant la logique du modèle de langue et en considérant que le contenu d'un document comporte à la fois des mots simples et mots composés. Chacun produisant un type de terme.

Nous supposons don que le modèle de document peut être estimé à l'aide de deux modèles : un modèle des mots simples (M_{D_t}) et un modèles des mots composés (M_{D_t}).Ainsi ‟n", étant donné une requête q, exprimée par des mots simples et des mots composés le modèle d'appariement document-requête que nous proposons combine les deux modèles de la manière suivante :

$$P(q|D) = \prod_{t_i \in q} P(t_i|D) \times \prod_{T_j \in q} P\left(T_j \middle| D\right) \quad \text{(III.1)}$$

Avec

$$P(t_i|D) = \lambda\, P\left(t_i \middle| M_{D_t}\right) \times (1-\lambda) P\left(t_i \middle| M_{D_t}\right) \quad \text{(III.2)}$$

$$P(T_i|D) = \alpha\, P\left(T_j \middle| M_{D_t}\right) \times (1-\alpha) \prod_{t_k \in T_j} P\left(t_k \middle| M_{D_t}\right) \quad \text{(III.3)}$$

Où :

λ et $\alpha \in [0,1]$ sont des paramètre de lissage

$P\left(t_i \middle| M_{D_t}\right)$ et $P\left(T_j \middle| M_{D_t}\right)$ peuvent être évalué en utilisant n'importe que modèle de langue uni-gramme. Nous avons pour notre part opté pour lissage Dirichlet :

$$P_{Dir}\left(t_i \middle| M_{D_t}\right) = \frac{F(t_i, D_t) + \mu P(t_i|C_t)}{|D_t| + \mu} \quad \text{(III.4)}$$

Où :

$F(t_i, D_t)$: Est la fréquence du mot simple t_i dans le document D;

$P(t_i|C_t)$: est le modèle de langue de la collection (la fréquence globale du terme est utilisée) ;

$|D_t|$: est la longueur du document exprimée avec des mots simples ;

μ : est le paramètre de lissage

III.4.4. expansion de requêtes dans la recherche simple (EQ_TS)

Dans notre approche nous utilisons la plateforme terrier pour modèle d'expansion de requête. Terrier mise en œuvre un modèle d'expansion automatique. Le modèle d'expansion de requête utilisée est modèle par défaut Bol.

III.5. Expérimentation et résultats

III.5.1.Collection de test

Différentes collections de tests sont utilisées en recherche d'information. La collection que nous avons utilisée pour nos expérimentations est: TREC AP88 (Associated Press newswire, 1988) et WT10g.

Pour la recherche nous avons utilisé 50 requêtes issues des topics numérotées « 101-150 » de la collection TREC.

III.5.2.Evaluation de notre approche

Cette section est consacrée à la présentation l'évaluation des performances de notre approche.

Nous avons évalué les trois points suivants :

- La recherche simple
- Expansion de requêtes basées sur les termes simples (EQ_TS)
- La recherche basée sur les mots composés(TC).

III.5.2.1. Résultats de la recherche simple

L'objectif de l'ensemble des testes d'évaluation basé sur la recherche simple est de déterminer la meilleure valeur de paramètre **μ** du modèle de recherche d'information.

Nous avons varié la valeur de ce dernier (μ) de 100 à 5000 avec un pas de 500, les résultats obtenus sont représentés dans le tableau suivant :

Les valeurs de µ	MAP
100	0,1918
500	0,2168
1000	0,2227
1500	0,2211
2000	0,2190
2500	0,2124
3000	0,2098
3500	0,2078
4000	0,2058
4500	0,2032
5000	0,2018

Tab III.1 : Résultats obtenu avec la recherche simple

III.5.2.2. Résultats obtenu avec l'expansion de requêtes basées sur les termes simples

Dans ce point nous intéressons à ajouté des termes simple ou reformulation de la requête d'utilisateur. Pour cela en a besoin des informations (données) suivant :

µ	Nombre de document	nombre de terme	saut
1000	20	30	5

µ fixé à 1000 déduit dans TS, le nombre de documents et de termes.ces deux dernière nous les avons variés de 5 à 20 le saut 5.pour chaque document nous varions le nombre de termes dans le but de trouver la meilleure précision d'expansion de requête, pour quelle nombre document et terme.les résultats sont représentés dans le tableau en dessous.

Nombre de document	Nombre de terme	MAP
5	**5**	0,2223
	10	0,2187
	15	0,2180
	20	0,2183
	25	0,2183
	30	0,2178
10	**5**	0,2254
	10	0,2260
	15	0,2267
	20	0,2299
	25	0,2299
	30	0,2297
15	**5**	0,2216
	10	0,2207
	15	0,2218
	20	0,2224
	25	0,2224
	30	0,2164
20	**5**	0,2199
	10	0,2168
	15	0,2174
	20	0,2188
	25	0,2140
	30	0,2138

Tab III .2 : Résultat obtenu l'expansion de requêtes basées sur les termes simples

Selon le tableau III .2 l'expansion de la requête améliore les résultats obtenu tel que ; le nombre de document égal à 10 et le nombre de terme égal à 20 donne la meilleure

amélioration avec la précision égale à 0.2299,nous remarquons qu'il y'a une amélioration par rapport à la recherche simple.

III.5.2.3. Résultats obtenu avec la recherche terme composée

L'objectif de la recherche avec des termes composée est de réduire l'ambigüité des termes, c'est l'approche proposé**.**les paramètre exigés sont : μ = 1000 et nous varions la valeur de **α** de la formule (III.3) entre 0 et 1 avec un pas de 0,1 pour trouver la meilleure précision moyenn

μ	Les valeurs d'Alpha	MAP
1000	0	0,2431
	0,1	0,2458
	0,2	0,2487
	0,3	0,2493
	0,4	0,2487
	0,5	0,2449
	0,6	0,2403
	0,7	0,2381
	0,8	0,2359
	0,9	0,2289
	1	0,2196

Tab III.3 : Résultats obtenu avec la recherche terme composée

A l'observation des valeurs de α, nous déduisons que la valeur idéal de la MAP est calculé pour α=0.3, MAP=0.2493.

Nous remarquons qu'il y'a une amélioration visible en passant de recherche simple(TS) à recherche avec mot composé (TC).

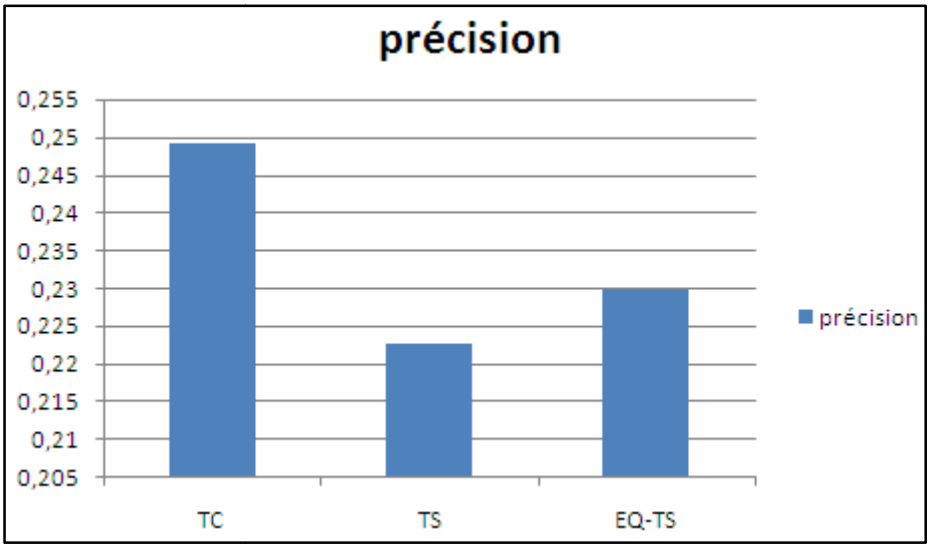

Figure III.6 : comparaison les résultats de la recherche TS, TC et EQ-TS

Comme pouvons le remarquer, notre approche améliore les résultats du recherche comme suit :

1. Premièrement, le modèle utilisant les mots composée donne une amélioration par rapport à la recherche simple ce qui implique que l'utilisation des mots composées apporte un plus pour la RI.

2. Deuxièmes, le modèle utilisant l'expansion de requête améliore le modèle de base sur les mots ; ce qui indique que l'expansion de requête peut apporter une amélioration pour la RI.

III.6.Conclusion

Dans ce chapitre, nous avons présenté le principe de notre approche et son fonctionnement, avons aussi présenté l'environnement d'implémentation et les testes de notre approche. A partir de résultats d'expérimentations obtenus nous déduisons les points suivants :

- L'utilisation des mots composés améliore les résultats de la recherche d4information d'une manière substantielle
- L'utilisation de l'expansion de requête améliore aussi les résultats de la RI.

Conclusion générale

Conclusion générale

Notre approche dans ce mémoire s'inscrit dans les travaux qui améliorent la performance de recherche d'information, nous étudions d'ambigüité et de disparité des termes de recherche d'information.les solution donnes sont : l'expansion de requêtes qui permit de reformulée la requête de l'utilisateur et le terme composé comme unité d'index dans le modèle de langue mixte.

Pour mener à terme notre travail, nous avons donné un aperçu général sur la recherche d'information ainsi les systèmes de recherche d'information et nous avons traité l'expansion de requêtes et les mots composés , ce qui nous a permis d'enrichir nos connaissances pour le bon déroulement de notre travail. De même nous avons défini et suivi, la plate-forme de recherche d'information Terrier, le langage de programmation "Java" et l'environnement "Netbeans" afin d'implémenter notre approche.

Ce travail nous a permis d'aborder le domaine de la recherche d'information, d'enrichir nos connaissances et plus précisément :

- Approfondir nos connaissances sur la recherche d'information.
- Mettre l'accent sur la manière dont les systèmes de recherche d'information fonctionnent dans la plate-forme Terrier.

.

Bibliographique

Bibliographique

[1] : Gerard Salton. Automatic information organization and retrieval. 1968.

[2] : Ounnaci Idir : ‘’recherche d’information dans les documents pédagogique structuré adaptée aux besoins spécifiques des apprenants’’. Magister en informatique, Université Mouloud Mammeri de Tizi Ouzou.

[3]: Introduction de Jian-Yun Nie (Université de Montréal).

[4] : Lynda Tamine : “Optimisation de requêtes dans un système de recherche d’information“ .Thèse de doctorat en informatique, université PAUL SABATIER DE TOULOUSE, 2000.

[5] : Kraft, D.H, and Buell, D.A.Fuzzy sets and generalized Boolean retrieval systems. International Journal on Man-Machine Studies, 19:pp. 49-56,1983

[6]: Radecki, T.Fuzzy set theoretical approach to document retrieval. *Information Processing and management*, 15:pp.247-259, 1979.

[7] : Salton, G., E.A. Fox, H. Wu. Extended Boolean information retrieval system.

CACM 26(11), pp. 1022-1036, 1983.

[8] : Salton, G., Fox, E., and Wu, H. Extended Boolean information retrieval. Communications of the ACM, 26(12), 1983.

[9]: Salton, G.The smart Retrieval System: Experiments in Automatic Document Processing. *Prentice-Hall*,1971.

[10]: N.J. Belkin and W.B. Croft. Information retrieval and information filte-ring : two sides of the same coin ? Communications of the ACM, 35(12),December 1992.

[11]: M. Boughanem, J. Savoy, editors; *Recherche d’information états des lieux et perspectives.* Hermès Science Publications,2008.

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

[13] : Manning,D.,Schutze,H.Foundation if Statistical Natural Language processing.MIT Press,2000.

[14] :Ben Aouicha, Mohamed (2009). *Une approche algébrique pour la recherche d'information structurée.* Thèse de doctorat en informatique, Université Paul Sabattier de Toulouse.

[15] : Hammache Arezki : "recherche d'information : modèle de langue combinant mots simple et mots composés". Thèse de doctorat en informatique, Université Mouloud Mammeri de Tizi Ouzou, 2013.

[16] : Donna Harman : Relevance Feedback Revisited, in the Proceedings of the ACM SIGIR Conference On Research and Development in Information Retrieval (SIGIR), pp 1-10, 1992.

[17]: Salton, G. (1971). A comparison between manual and automatic indexing methods. Journal of American Documentation, 20(1) :61{71.

[18] :Lv,Y., Zhai.C Position Relevance Model for Pseudo-Relevance Feedback. *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pp.579-586, 2010.

[19]: Salton G., Buckley C., "Term Weighting Approaches in Automatic Text Retrieval", Cornell University, Ithaca, NY, 1987. Information processing & Management Vol 24/5 pp:513-523.

[20]: Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas and Richard A. Harshman, 1990. "Indexing by Latent Semantic Analysis". In Journal of the American Society of Information Science, Vol. 41 :6, 391-407.

[21]: Mohand Boughanem, C. Soulé-Dupuy : A Connexionist Model for Information Retrieval. DEXA 1992 : 260-265.

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

[23]: ROBERTSON S. E., WALKER S., « Some Simple Effective Approximations to the2-Poisson Model for ProbabilisticWeighted Retrieval », Proceedings of SIGIR 1994, p. 232-241, 1994.

[24] : Lobna HLAOUA :'' Reformulation de Requêtes par Réinjection de Pertinence dans les Documents Semi-Structurés''. Thèse de doctorat en informatique, Universit´e Paul Sabatier de Toulouse,2007.

[25] :Boubekeur-Amirouche, Fatiha (2008). *Contribution à la définition de modèles de recherche d'information flexibles basés sur les CP-Nets.*

[26] : Lynda Tamine. OPTIMISATION DE REQUETES DANS UN SYSTEME DE RECHERCHE D'INFORMATIONAPPROCHE BASEE SUR L'EXPLOITATION DE TECHNIQUES AVANCEES DE L'ALGORITHMIQUE GENETIQUE. Informatique [cs]. Université Paul Sabatier - Toulouse III, 2000. Français.

[27]: Sanderson, M. Test collection based evaluation of information retrieval system. *Foundations and Trends in information retrieval 4, pp.247-375,2010.*

[28]: Van Rijsbergen, C. J. Information retrieval. London : Butterworth, (1979).

[29]: C. Cleverdon. *Progress in documentation : Evaluation of information retrieval system.* In Journal ok Documentation 26, p. 55-67,1970.

[30]: M. Boughanem, W. Kraaij, J.Y. Nie, Modèles de langue pour la recherche d'informations, dans Les systèmes de recherche d'informations - Modèles conceptuels, ed. M. Ihadjadene, Hermes, pp. 163-184, 2004.

[31]:" Expansion de requêtes spatio-thématiques dans un service de catalogage ". Mémoire de Stage de Master, sous la direction de Thérèse Libourel, Pierre Maurel(Cemagref), Jean-Christophe Desconnets (IRD), 2006.

[32] : Abberly, D, Kirby, S. Renals, and T. Robinsob. THISL broadcast news retrieval system, In Proc. ESCA Workshop on Accessing Information in Spoken Audio, pages 19 {24, Cambridge, 1999}.

[33]: Improving Retrieval Performance by Relevance Feedback. Journal of the America Society for Information Science 41, 4, 288-297.

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

[35]: Saint Réquier et al. 2010

[36]: Bissan Audeh. Reformulation sémantique des requêtes pour la recherche d'information ad hoc sur le Web. Autre. Ecole Nationale Supérieure des Mines de Saint-Etienne, 2014. Français.

[37]; J Minker, G.A Wilson, and B.H. Zimmerman. An evaluation of query expansionby the addition of clustered terms for a document retrieval system. InformationStorage and Retrieval. Information storage and retrieval, 8 :329–348, 1972.

[38] : M.E Lesk. Word-word associations in document retrieval systems. American Documentation, 1969.

[39] : Helen J. Peat and Peter Willett. The limitations of term co-occurrence data forquery expansion in document retrieval systems. Journal of the American Societyfor Information Science, 42 :378–383, June 1991.

[40]: Yonggang Qiu and H.P. Frei. Concept Based Query Expansion. In Proceedings of the international ACM SIGIR conference on Research and development in information retrieval, volume 11, page 212, NY, January 1993. ACM.

[41]: Imran Hazra et Sharan Aditi. Thesaurus et Query Expansion. International Journal of Computer science & Information Technology (IJCSIT), Vol 1, No 2, November 2009.

[42]: Qiu Yonggang et Frei Hans-Peter. Concept-based query expansion. SIGIR. Zurich, Switzerland, 1993.

[43]: J. Bhogal, a. Macfarlane, and P. Smith. A review of ontology based query expan-sion. Information Processing and Management, 43 :866–886, July 2007.

[44]: Jiewen Wu, Ihab Ilyas, and Grant Weddell. A study of ontology-based query ex-pansion. Technical report, Technical report CS-2011-04, University of Waterloo,2011.

[45]: Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. ACM Computing Surveys (CSUR), 44 :1, 2012.

[46]: Baziz Mustapha. Indexation conceptuelle guidée par ontologie pour la recherche d'information. Thèse de l'Institut De Recherche En Informatique De Toulouse, l'Université Paul Sabatier de Toulouse, 14 décembre 2005.

[47] :Fagan, J. L. Expriments in Automatic Phrase Indexing For Document Retrieval:A Comparison of Syntatic and Non-Static Methids.

[48] : El GHALI Btihal . “ Informatique et Télécommunications : Modèles Contextuels pour la Recommandation et l'Expansion de Requêtes en Recherche d'Information“. PES, Ecole Nationale Supérieure d'Informatique et d'Analyse des systèmes, Rabat ,2016.

[49] : Rocchio J.J., Relevance Feedback in Information Retrieval, SMART Retrieval System Experiments in Automatic Document Processing, 1971, Prentice Hall (Publisher), 1971.

[50]: Robertson, S.E, Sparck Jones, K. Relevance Weighting of Search Termes. Journal of the American Society for Information Science 27, pp.129-146, 1976.

[51]: Cui Hang, Wen Ji-Rong, Nie Jian-Yun et Ma Wei-Ying. Probabilistic Query Expansion Using Query Logs. WWW2002, Honolulu, Hawaii, USA, May 7-11, 2002.

[52]: Saint-Réquier Aurélien, Dupont Gérard, Adam Sébastien, Lecourtier Yves. Évaluation d’outils de reformulation interactive de requêtes. Conférence en Recherche d'Information et Applications, Tunisie. pp. 223-238, Mars 2010.

[53] : Xu Jinxi et Croft W. Bruce. Query Expansion Using Local et Global Document Analysis. SIGIR’96, Zurich, Switzerland, 1996.

[54] : Boughanem, M., Chrisment, C., Soule-Dupuy, C. Query modification based on relevance back-propagation in adhoc environment. Information Processing and Management, 35, pp. 121-139, 1999.

[55]: Harman, D. Relevance feedback and other query modification technique. In Information Retrieval: Data Structures and Algorithms, William B. Frakes and Ricardo Baeza-Yates, editors, Prentic Hall, Anglewood, Cliffs, NJ, pp. 241-263, 1992.

[56]:E; Voorheses,(using wordnet to disambiguate word senses for text retrieval), processing of the annual conference on research and development in information retrieval,SIGIR 93 , Pittsburgh, PA, 1993.

[57]: Voorhees, E, 2004. Overview of the trec 2004 robust track. In Processing of the 13 Gí Text Retrieval Conference (TREC-7), NIST Special Publication 500-261. National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA.

[58]: Diaz, F.and Metzler, D. Improving the estimation of relevance models using large external corpora. In proceedings of the 29 Gí Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press, Seattle, Washington, USA, pp. 154-161. 2006.

[59]: Chirita, P.-A., Firan, C. S., AND Nejdl, W. Personalized query expansion for the web. In proccedings of the 30 Gí Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press, Amsterdam, The Netherlands, pp. 7-14. 2007.

[60]: Qiu Y., Frei H. –P., << Concept-based query expansion>>, Proceedings of SIGIR-93, 16 Gí ACM International Conference on Research and Development in Information Retrieval, Pittsburg, US, 1993, pp. 160-169.

[61]: C. and Yang, B. Experiments in automatic statistical thesaurus construction. In Proceedings of the 15 Gí Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press, Copenhagen, Denmark, pp. 77-88, 1992.

[62]: Fianna Raiber, Oren Kurland, « On Identifying Representative Relevant Documents ». Faculty of Industrial Engineering and Management, 2000.

[63]: Local feedback in full-text retrieval systems.J. ACM 24, 3, 397- 417.

[64]: Word association norms, mutual information and lexicography.Computat.Linguist. 16, 1, 22-29.

[65]: Information Retrieval .Butterworths.

[66]: Query expansion using fuzzy association rules between terms. In Proceedings of the 4th International Conference Journ'ees de l'Information Messine(JIM'03).

[67]: Integration of association rules and ontologies for semantic query expansion. Data Knowl.Engin. 63, 1, 63-75.

[68]: Extending query translation the cross-language query expansion with markov chain models.In Proceedings of the 16th Conference on Information and Knowledgs Management (CIKM'07).ACM Press.

[69]: Concept-based query expansion.In Proccdings of the 16the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.ACM Press, 160-169.

[70]: 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.

[71]: Query expansion using term relationships in language models for information retrieval. In Proceedings of the 14th ACM International Conference on Information and Knowledgs Management.ACM Press, 688-695.

[72]: Query expansion by mining user logs.IEEE Trans.Knowl.Data Engin. 15,4,829-839.

[73]: improving retrieval performance by global analysis.In Proceedings of the 18th International Conference on Pattern Recognition.IEEE Computer Society, 703-706.

[74]: Mining dependency relations for query in passage retrieval.In Proceedings of the 29th Annual International ACM SIGIR Conference on Reseatch and Develoment in Information Retrieval.ACM Press, 382-389.

[75]: 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

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

[77]: Selecting good expansion terms for pseudorelevance feedback. In Proceedings of the 31st Annual International ACM SIGIR Conference on Researchaud Devlopment Information Retrieval. ACM Press, 243-250.

[78]: G.Salton & C.Buckley: Improving Retrieval Performance by Relevance Feedback, Journal of The Amercican Society for Information Science, Vol.41, N=4, pp 288-297, 1990.

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

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

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

[82]: Karthik Raman, Pannaga Shivaswamy, and Thorsten Joachims. Online learning to diversify from implicit feedback. In Proceedings of KDD, pages 705–713, Beijing, China, 2012.

[83]: Re-examining the effects of adding relevance information in a relevance feedback environment.Info.Process.Manage.44, 3, 1086-1116.

[84]: Probabilistic models for information retrieval based on divergence from randomness.Ph.D.thesis, Department of Comuting Science, University of Glasgow, UK.

[85]: Relevance based language models. In Proceedings of the 24th Annual International ACM SIGIR Conference on Reseaech and Development in information Retrieval.ACMPress, 120-127.

[86]: Zhai, C, Lafferty, J.Model-based feedback in the language modeling approach to information rretrieval.Proceedings of the 10th International Conference en Information and Knowledge ManagementACM Press, pp 403-410, 2001.

[87]: Query expansion using term relationship in language models for information retrieval in Language models for information retrieval. In Proceedings of the 14th ACM International Conference on Information and Knowledgs Management.ACM Press, 688-695.

[88]: Lavrenko, M. and Aslam, J. 2001. Relevance-based language models. In W.B Croft, D.J. Harper, D.H Kraft, & J. Zobel (Eds). Proceedings of the 24th Annual International ACMSIGIR Conference on Research and Developpement in information retrieval, New Orleans, Louisiana, pp. 120-127,2001.

[89]: Martin,W.J.R., AI, B. P? F.,and van Strenkenbung, P.J. G.On the processing of Test corpus:Froni Textual Data to Lexicographical Information.

[90]: Chkrabarti, S. Dom, B., Raghavan, P., Rajagopalan, S., Gibson, D., Kleingerg, J.Automatic resource list compilation by analyzing hyperlink structure and associated text. Proceedings of the 7th international World Wide Web Conference, pp. 65-74, 1998.

[91]: Zhai, C. and Lafferty, J. 2001. Model-based feedback in the language modeling approach to information retrieval .In Proceeding of the tenth international conference on Information and Knowleds management ACM Press, Atlanta Georgia, USA, 403-410.

[92]: Khoo, C., Myaeng,S., and Oddy ,R.Using Cause-Effet Relation in Text to Improve Infirmation Retrieval Precising.

[93]: Luhn, H. P .A Business Intelligence System.IBM Journal Research and Development (2:4),pp.314-319,1958.

[94]:Bartell, B.T.,COTTRELL?G.W.,Belew ,R.K.Automatic combination of multiple ranked retrieval systems.