Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Licence en Informatique

Thème

Réalisation d'un système de recherche d'information

Réalisé par :

- **M elle. Bouabane samia**
- **Mr. Benghelima mohamed amine**

Présenté le 9 Juin 2014 devant la commission d'examination composée de MM.

- *Mr. BENTAALLAH MOHAMED AMINE* *(Encadreur)*
- *Mme. EL YEBDRI ZINEB* *(Examinateur)*
- *Mr. MESSABIHI MOHAMED* *(Examinateur)*

Année universitaire: 2013-2014

REMERCIEMENT

Enfin, cette fameuse page qui tient tellement à nos cœur ……..

Un grand plaisir que nous remercions Allah le tout puissant et *miséricordieux qui nous a donné la force de surmonté tous les moments difficile pour accomplir et achever ce modeste travail.*

Permettez- nous tout d'abord d'exprimer notre profonde gratitude à tous ceux qui ont contribué directement ou indirectement à l'aboutissement de ce travail.

On tient à remercier tout particulièrement Notre encadreur, Mr Mohamed BENTAALLAH qui a acceptés de nous encadrer et guidées et soutenues toute la période de notre projet et pour tous ses précieux conseils.

On remercie aussi Mr Messabihi mohamed et Mme El yebdri zineb pour avoir fait l'insigne honneur d'avoir accepté d'être membre du jury de notre mémoire et de l'enrichir par leurs propositions.

Nos plus vifs remerciements vont également aux nos parent pour leur soutien, leur contribution et leur patience, ainsi que tous les membres de notre famille.

DEDICACE

Merci Allah (mon dieu) de m'avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller sur le droit chemin tout au long de nos années d'étude jusqu'au bout du rêve et le bonheur de lever mes mains vers le ciel et de dire" Ya Kayoum ", Sans sa miséricorde ce travail n'aura pas abouti ;

Je dédie ce modeste travail à celle qui m'a donné la vie, le symbole de tendresse et d'amour, Sont les moindres sentiments que je puisse vous témoigner. Quoi que je fasse, je ne pourrais jamais vous récompenser pour les grands sacrifices que vous avez faits et continuez de faire pour moi, pour mon bonheur et ma réussite ;

Aucune dédicace ne saurait exprimer mes grandes admirations, mes considérations et mes sincères affections pour mes chers parents ;

MA Mère, qui a toujours cru en moi et encouragée ;

Mon père, école de mon enfance, qui a été mon ombre durant toutes les années des études, et qui a veillé tout au long de ma vie à m'encourager, à me donner l'aide et à me protéger ;

Que dieu les gardes et les protèges ;

A ma très chère sœur fatima et petite princesse bochra ;

A mon seule et unique frère mohamed que j'aime beaucoup ;

A mes oncles, mes tantes, à chaque cousins et cousines et tous les membres de ma famille, petits et grands qui ont sacrifié leurs droits et leurs temps pour que je puise mener ces études;

A toutes mes ami(e)s wassila, fatima, sara, sihem, meriem, faten, asma, naima, Seif Eddine, youcef et mon binôme amine. Je ne peux trouver les mots justes et sincères pour vous exprimer mon affection et mes pensées, vous êtes pour moi des frères, sœurs et des amis sur qui je peux compter ;

A mes enseignants de l'université de Tlemcen et surtout Mr Bentaallah notre encadreur. Un remerciement particulier et sincère pour tous vos efforts fournis. Vous avez toujours été présente. Que ce travail soit un témoignage de ma gratitude et mon profond respect, Vos qualités professionnelles et votre rigueur sont pour moi des exemples à suivre ;

A tous ceux qui me sont chères ;

A tous ceux qui m'aiment ;

A tous ceux que j'aime ;

Je dédie ce travail.

Samia

DEDICACE

J’aimerais en premier lieu remercier Allah qui m’a donné la volonté et le courage et la santé pour la réalisation de ce travail.

Je dédie ce modeste travail à mes parents qui ont cru en moi et qui m’ont soutenu pendant mon parcours, Aucune dédicace ne saurait être assez éloquente pour leur exprimer tous les sacrifices que n’ont cessé de me donner depuis ma naissance, mon éducation et ma formation, Que dieu les préserve une longue vie heureuse.

A ma grande mère pour toutes ces prières.

A mon très cher frère Slimane que j’aime beaucoup et je lui souhaite la réussite dans ces études

A mes cousines et mes chers cousins nadir et hamza.

A mes oncles et tantes et à toute la famille.

A tous ami(e)s surtout Saïd, Ilyés , Achraf , Hicham ,Youcef , Adel , Oussama Je vous dédie ce travail et je vous souhaite un avenir à la hauteur de vos ambitions. Que notre amitié dure

A tous ceux que j’aime et qui m’aiment et me comblez de conseils.

Mohamed Amine

Table des matières

Liste des figures

Liste des abréviations

B	Bruit
P	Précision
R	Rappel
RI	Recherche d'Information
S	Silence
SRI	Système de Recherche d'Information
TALN	Compréhension du texte

Introduction générale

Aujourd'hui, l'information joue un rôle primordial dans le quotidien des individus et dans l'essor des entreprises. Cependant, le développement de l'internet et la généralisation de l'informatique dans tous les domaines ont conduit à la production d'un volume d'information sans précédent. En effet, la quantité d'information disponible, particulièrement à travers le web, se mesure en milliards de pages. Ainsi que la quantité d'information disponible dans le monde explose. Des études statistiques ont estimé qu'elle double tous les 20 mois. Dans ce même temps, celle accessible via Internet quadruple : Le volume de l'information, la taille des collections, ainsi que le nombre d'utilisateurs sont toujours en croissance vertigineuse.

Il est par conséquence, de plus en plus difficile de localiser précisément ce que l'on recherche dans cette masse d'information. La recherche d'information (RI) est le domaine par excellence qui s'intéresse à répondre à ce type d'attente. En effet, l'objectif principale de la RI est de fournir des modèles, des techniques et des outils pour stocker et organiser des masses d'information et localiser celles qui seraient pertinents relativement à un besoin en information d'un utilisateur, souvent, exprimé à travers une requête. Ces outils sont appelés des systèmes de recherche d'information (SRI). De manière générale, le fonctionnement d'un SRI consiste à construire une représentation des documents et de 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 pondération des termes est la phase la plus essentielle dans le domaine de recherche d'information, elle consiste à donner un poids à chaque terme. Le calcul de la pertinence d'un document en réponse à une requête d'utilisateur dépend des poids attribués aux termes.

Notre objectif dans le cadre de ce mémoire est de réaliser un système de recherche d'information en revisitant tous les étapes de processus en U de recherche d'information.

INTRODUCTION GENERALE

Notre mémoire est organisé sous forme de deux chapitres. Le premier chapitre par définir la recherche d'information et ses concepts fondamentaux et plus précisément le

processus de recherche d'information avec toutes ces étapes à savoir l'étape d'indexation des termes, l'appariement_requête-documents, et la_reformulation de requêtes. Le reste de ce chapitre sera consacré à présenter brièvement les principaux modèles de recherche proposés dans la littérature.

Le deuxième chapitre présente en détails les étapes de notre système de recherche d'information. Ce système est censé rechercher les documents qui répondent à un besoin d'information exprimé à l'aide d'une requête, en fait, le but principal c'est de retrouver tous les documents pertinents à cette requête.

Enfin, dans la conclusion, nous présenterons les principaux points abordés dans ce rapport. Ainsi que les perceptives.

Chapitre I

La recherche d’information

L'objectif de ce chapitre est de présenter les concepts de base de la RI. Dans la première section, nous décrirons le processus global de la RI, communément connu sous le nom du processus en U. Nous donnerons l'utilité et l'importance des opérations qui composent ce processus.

Nous décrirons dans la seconde section trois modèles connus de la RI, savoir le modèle booléen, le modèle vectoriel et le modèle probabiliste. Nous présenterons ensuite les techniques utilisées pour l'évaluation des systèmes de recherche.

I. Introduction :

Un système de recherche d'information (RI) est un système qui permet de retrouver les documents pertinents à une requête d'utilisateur, à partir d'une base de documents volumineuse.

Dans cette définition, il y a quatre notions clés : documents, requête, pertinence, base de documents.

- **Un document :** peut-être un texte, un morceau de texte, une page web, une image, etc. On appelle document toute unité qui peut constituer une réponse à une requête d'utilisateur.
- **Une requête :** exprime le besoin d'information d'un utilisateur, elle est en générale de la forme suivante : « trouver les documents qui….. ».
- **La pertinence :** est une notion très importante qui détermine la correspondance ou le degré de relation entre le document et la requête.

Si c'est l'indexation qui choisit les termes pour représenter le contenu d'un document ou d'une requête, c'est au modèle de leur donner une interprétation. En effet, étant donné un ensemble de termes pondérés issus de l'indexation, le modèle remplit les deux rôles suivants :

- créer une représentation interne pour un document ou pour une requête basée sur ces termes ;

- définir une méthode de comparaison entre une représentation de document et une représentation de requête afin de déterminer leur degré de correspondance (ou similarité).

La qualité d'un système de recherche d'information doit être mesurée en comparant les réponses du système avec les réponses idéales que l'utilisateur espère recevoir. Plus les réponses du système correspondent à celles souhaitées par l'utilisateur, meilleure est la performance du système.

- **Base de documents :** c'est l'ensemble des documents disponibles.

II. Objectif de la recherche d'information

La recherche d'information a pour objectif :

- Identifier en vue d'exploiter de l'information contenue dans des documents et des bases de données (son, texte, image) par rapport à une requête formulée par un utilisateur.
- Le (**SRI**) devra nous retourner le moins possible de documents non pertinents
- Les contenus des documents peuvent être non structurés ou semi structurés.

III. Principe de la recherche d'information

Abrégée en **RI** (Recherche d'Information) ou **IR** (Information Retrieval), la recherche d'information est un domaine historiquement lié aux Sciences de l'information et à la bibliothéconomie qui ont toujours eu le souci d'établir des représentations des documents dans le but d'en récupérer des informations, à travers la construction d'index.

Le terme de « recherche d'information » a été utilisé pour la première fois dans les années cinquante par *Kelvin N. Mooers* quand il travaillait sur son mémoire de maîtrise [Mooers, 48].

Mr Salton a défini la recherche d'information comme étant l'opération qui permet à partir d'une expression des besoins en information d'un utilisateur de retrouver l'ensemble des documents contenant l'information recherchée [Salton et al. 83].

Un système de recherche d'information (**RI**) est un ensemble de programmes informatiques ayant pour but de satisfaire le besoin en information d'une requête utilisateur. Le rôle principal est donc de sélectionner les informations ou les documents les plus pertinents correspondant à ce besoin, et cela à partir d'une base de documents volumineuse. La (**RI)** concerne donc les mécanismes qui facilitent l'accès à une base d'informations. Il existe un grand nombre de modèles de recherche d'information, et ces

modèles diffèrent principalement sur la façon dont les informations disponibles sont représentées, et sur la façon de représentation du besoin de l'utilisateur.

Bref, la recherche d'information s'intéresse à la représentation, le stockage, l'organisation, l'acquisition, la recherche et la sélection d'information.

Dans ce domaine plusieurs projets ont vu le jour dont les plus intéressants sont :

- Projet Cranfield (dirigé par Cyril Cleverdon, 1957-1967) [CLE 67]
- Projet MEDLARS – MEDical Literature Analysis and Retrieval System (F. Wilfrid Lancaster, complete en 1968) [LAN 68]
- SMART (Gerard Salton, 1ière version 1961-1965) [SAL 71]
- Projet STAIRS - STorage And Information Retrieval System (Blair et Maron) [BLA 85]
- TREC - Text Retrieval Conference, (D. Harman, 1992) [HAR 92]

IV. Le processus de recherche d'information :

Le processus de Recherche d'Information a pour but de mettre en correspondance les représentations des informations contenues dans un fond documentaire d'une part avec celle des besoins de l'utilisateur d'autre part, ou par d'autres termes, de correspondre au mieux la pertinence système avec la pertinence utilisateur

Ce processus est composé de trois fonctions principales :

- ***l'indexation*** des documents de la collection et des requêtes utilisateur.
- ***l'appariement*** des représentations requête-documents pour le calcul de la pertinence des documents en réponse un besoin utilisateur.
- ***la reformulation de requêtes*** qui permet de réécrire autrement la requête utilisateur puisqu'il est quasi-impossible aujourd'hui, de retrouver des informations pertinentes en utilisant la seule requête initiale de l'utilisateur, et ce à cause du volume croissant des bases documentaires.

Pour résumer ces fonctions, nous pouvons représenter schématiquement certaines fonctionnalités d'un **(SRI)** par ce que l'on appelle communément le processus « en **U** » tel que dans la figure 1.1

L'utilisateur exprime son besoin sous forme d'une requête, cette dernière sera analysée (*indexée*) par le système en parallèle ou préalablement. L'ensemble des documents est également indexé par l'index (collection des documents et requête indexés).

L'étape *d'appariement document requête* consiste à mettre en correspondance les représentations de la requête avec les représentations des documents de la collection afin de donner une liste de documents considérés comme pertinents.

La dernière étape est consacrée à *la reformulation de la requête* dans le cas ou les documents retrouvés ne répondent pas aux besoins de l'utilisateur.

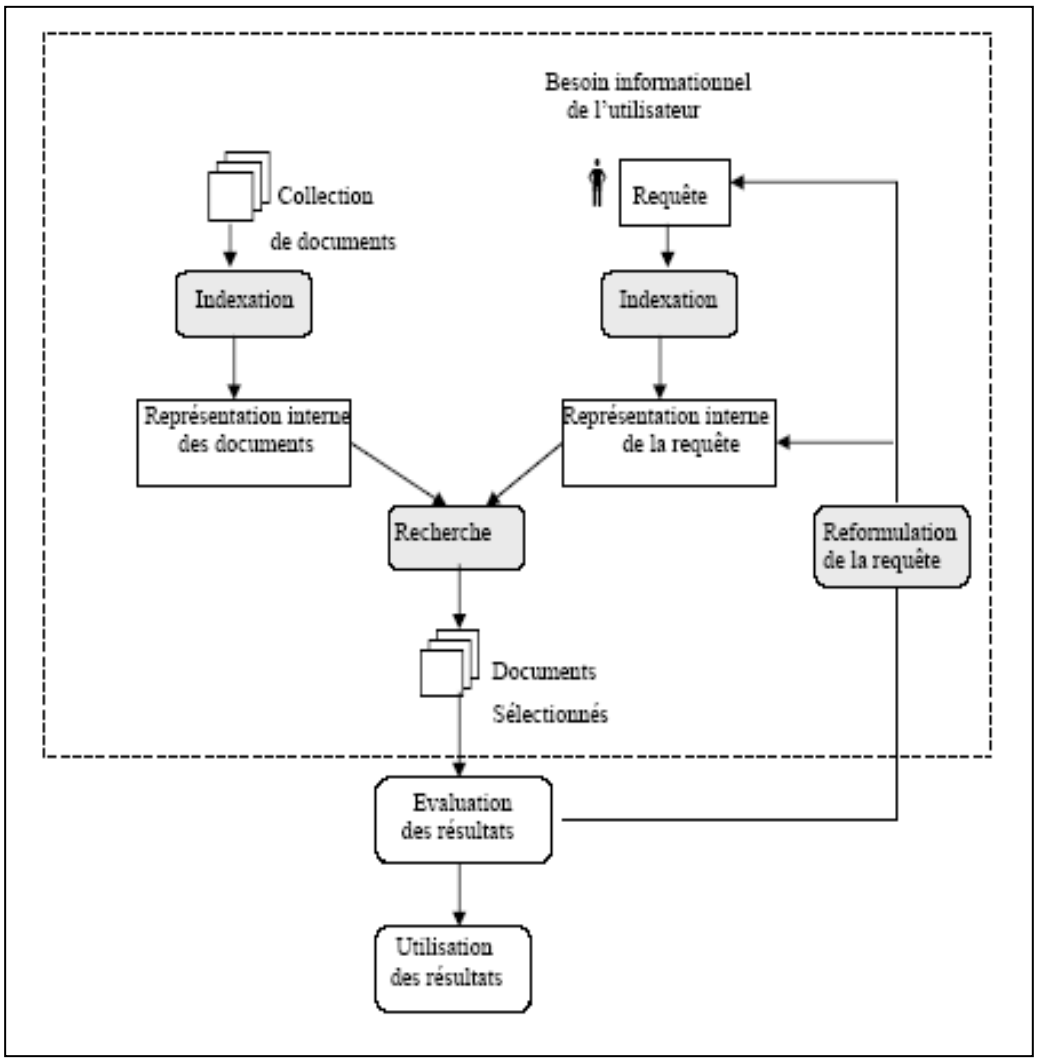

FIGURE 1.1 : Processus en U de la RI

IV.1 Processus d'indexation :

L'indexation est l'opération qui vise à construire une structure d'indexe qui permet de retrouver très rapidement les documents incluant des mots demandés.

Cette étape consiste à analyser chaque document de la collection afin de créer un ensemble de mots-clés. Son objectif est de trouver les concepts les plus importants du document (ou de la requête), qui formeront le descripteur du document.

Ainsi, en pratique on cherche plutôt des *représentants* des concepts. Ces représentants peuvent être de formes différentes : des mots simples, ou de groupes de mots (mots composés).

Les descripteurs des documents (mots, groupe de mots) sont rangés dans une structure appelée dictionnaire constituant le langage d'indexation.

Ce langage peut être de deux types :

- ***Langage libre :*** est construit à partir des termes extraits du document analysé.
- ***Langage contrôlé :*** est construit à partir d'un ensemble des termes préalablement définis et organisés généralement dans un thésaurus.

Lorsqu'un document est analysé, on ne garde que les mots clés qui appartiennent à ce thésaurus.

L'indexation peut être :

- ✓ ***Manuelle***

Chaque document est analysé par un spécialiste du domaine ou par un documentaliste, mais elle nécessite assez du temps pour sa réalisation, en plus, des termes différents peuvent être présentés par deux documentalistes différents pour représenter un même document, et un indexeur, à deux moments différents peut présenter deux termes distincts pour représenter le même concept.

- ✓ ***Automatique***

L'indexation automatique consiste à simuler par une machine cette opération d'indexation, que ce soit dans sa méthode ou surtout dans ses résultats.

En effet, si un ordinateur ne peut trouver un terme réellement descripteur d'un concept d'un document, il pourra caractériser celui-ci de façon à le retrouver. Le processus d'indexation est dans ce cas entièrement informatisé

L'indexation automatique repose sur un ensemble des méthodes automatisées sur un document comme l'extraction automatique des mots des documents, l'élimination des mots vides, la lemmatisation (radicalisation), la pondération des termes et la création de l'index.

- ✓ ***Semi-automatique***

Le choix final revient au spécialiste ou au documentaliste, qui intervient souvent pour choisir d'autres termes significatifs. Cette méthode est une combinaison des deux méthodes précédentes, elle appelé aussi indexation supervisée.

Qu'elle soit manuelle, semi-automatique ou automatique, l'indexation répond aux deux problèmes suivant : le choix des mots qui représente chaque document et l'évaluation de leur pouvoir de représentation.

Enfin l'indexation peut être caractérisée par sa fonction de pondération.

Voici la suite des opérations traditionnellement effectuées sur les documents textuelles lors de l'indexation, avec l'illustration des étapes d'indexation dans la figure 1.2 :

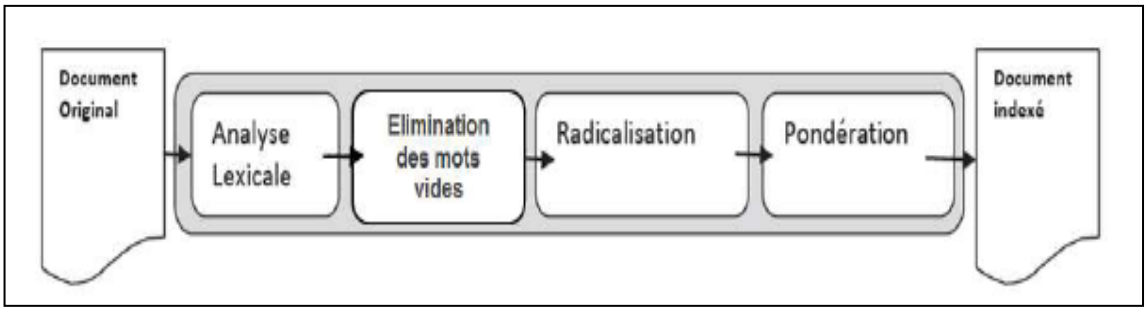

FIGURE 1.2 suite des traitements lors de l'indexation

IV.1.1 analyse lexicale

C'est un processus qui convertit le texte d'un document en un ensemble de termes ou un terme est un radical ou une unité lexicale. Cette analyse permet de reconnaître les espaces de séparation des chiffres, des mots, des ponctuations, etc.

IV.1.2 Elimination des mots vides

Un problème majeur de l'indexation est d'éviter les mots vides (pronom personnel, prépositions, articles, mots mathématiques (appartenir, contenir, inclure…) et choisir seulement les termes significatifs qui représentent au mieux un document donné.

Afin d'éliminer ces mots de force, on utilise une liste, appelée stoplist (ou parfois anti-dictionnaire) qui contient tous les mots qu'on ne veut pas garder. Une autre méthode consiste à éliminer les mots qui dépassent un certain nombre d'occurrences dans la collection.

IV.1.3 Lemmatisation

L'idée qui conduit à utiliser la lemmatisation est de pouvoir indexer un ensemble de mots par un seul mot qui représente le même concept.

En effet, on remarque que beaucoup de mots ont des formes différentes, mais leur sens reste le même ou très similaire et notamment dans le cas des mots conjugués. Ces mots ont la même racine (lemme). Ainsi, on arrive à éliminer les terminaisons des mots, et garder seulement la racine, on a donc une forme identique pour eux. Plusieurs méthodes sont utilisées : algo de porter, la troncature, variétés de successeurs, méthode de n-gramme

IV.1.4 La pondération

La pondération consiste à donner aux termes de l'index un poids mesurant leur importance dans les documents qui les contiennent. En effet, la pondération permet d'affecter à chaque terme d'indexation une valeur qui mesure son importance dans le document ou il apparait c'est-à-dire le mot est pondéré en fonction de sa rareté sur la toile, plus un mot est rare plus l'importance qui lui est accordé sur le site analysé sera grande et inversement. Ainsi, l'objectif est de trouver les termes qui représentent le mieux le contenu d'un document.

Plusieurs méthodes de pondération ont été proposées pour calculer les poids des termes de façon automatique, Les méthodes les plus utilisées dans ces domaines sont :

TF (term frequency) :

C'est un facteur de pondération local, cette mesure est proportionnelle a la fréquence du terme dans le document, l'idée est que plus un terme est fréquent dans un document, plus il est important dans la description.

IDF (inverse of document frequency):

« Log (n/df) » c'est un facteur de pondération globale, cette mesure permet de mesurer l'importance d'un terme dans toute la collection, l'idée est que les termes qui apparaissent dans peu de document de la collection sont plus représentatifs du contenu de ces documents, que ceux qui apparaissent dans tous les documents de la collection.

✓ **TF*IDF** (term frequency *inverse of document frequency):

On combinant les deux techniques précédentes, elle donne une bonne approximation de l'importance du terme dans le document relativement à une collection selon cette pondération, pour qu'on mot soit important dans un document il ne suffit pas qu'il soit fréquent dans le document mais aussi absent dans les autres documents.

La formule de **TF*IDF** est la suivante :

- **TFIDF** (**T**, **D**) : $Tf(t, d) * log\frac{N}{df(N)}$ (1)

Ainsi un terme qui a une valeur de **TF*IDF** élevé doit être à la fois important dans le document et aussi il doit apparaitre peu dans les autres documents.

- **TFC** (**T**, **D**) : $\frac{TFIDF(T,D)}{\sqrt{\sum_{k=1}^{M}(TFIDF(\mathrm{k,D}))^2}}$ (2)

La pondération **TFC** est semblable au **TF*IDF** à la différence que la **TFC** emploie la longueur du document.

- **LTC (T, D)** : $\frac{\log(fij+1)*dfi}{\sqrt{\sum_{k=1}^{M}(\log(\mathrm{f\,k\,j}+1)*\mathrm{df\,k})^2}}$ (3)

La pondération **LTC** est une approche légèrement différente, qui emploie le logarithme de la fréquence d'un terme, de ce fait elle réduit les effets de grandes différences dans les fréquences.

- **Okapi-BM25**: $\frac{tf(t,d)*(k+1)}{tf(t,d)+k*(1-b+b*\frac{dl(d)}{dl\,avg})} * log\frac{N-df(t)+0.5}{df(t)+0.5}$ (4)

La pondération **Okapi** peut être vue comme un **TF-IDF** prenant mieux en compte la longueur des documents. Sa définition est donnée dans l'équation (4) qui indique le poids du terme **t** dans le document **d** (**k** = 2 et **b** = 0.75 sont des constantes, **dl** la longueur du document, $\mathbf{dl_{avg}}$ la longueur moyenne des documents).

IV.2 Pertinence et appariement document-requête

La pertinence concerne d'une manière générale la sélection des documents susceptibles d'être pertinents à une requête donnée. Elle est basée sur une fonction d'appariement (matching) qui effectue une comparaison entre les représentants des documents et des requêtes construits lors de la phase d'indexation.

La comparaison revient à calculer un score représentant la pertinence du document vis-à-vis de la requête. Cette valeur est calculée à partir d'une fonction ou d'une probabilité de similarité notée **RSV (Q, d)** (Retrieval Status Value), où « **Q »** est une requête et « **d »** un document et elle tient compte du poids des termes dans les documents déterminé en fonction d'analyses statistiques et probabilistes. On parle souvent de la ***pertinence utilisateur*** et **la *pertinence système***. La première correspond au jugement de l'utilisateur sur la réponse en document rendu par le système et la deuxième est une mesure d'évaluation de la similarité entre le document et la requête.

Il existe deux types d'appariement :

- ✓ ***Appariement exact***

Le résultat est une liste de documents respectant exactement la requête spécifiée avec des Critères précis. Les documents retournés ne sont pas triés.

- ✓ ***Appariement approché***

Le résultat est une liste de documents censés être pertinents pour la requête. Les documents retournés sont triés selon un ordre de mesure. Cet ordre reflète le degré de pertinence document/requête.

Un problème fondamental est que la pertinence système est indépendante de l'utilisateur et du contexte de la recherche.

IV.3 Reformulation des requêtes (relevance feedback, expansion …)

L'utilisateur exprime son besoin en information sous forme d'une requête afin de trouver des résultats qui l'intéressent. Cependant, le (**SRI)** lui rend parfois des résultats qui ne lui conviennent pas. L'augmentation continue du volume des bases

Documentaires rend quasi-impossible le fait de retrouver des informations pertinentes en utilisant la requête initiale de l'utilisateur. Pour cela, une étape de reformulation de la requête est souvent utilisée dans l'espoir de retrouver plus de documents pertinents.

Ce processus permet de générer une requête plus adéquate que celle initialement formulée par l'utilisateur.

- ***Types de la reformulation***

❖ **Automatique** : Utilisation de thésaurus qui Permet de fournir un vocabulaire contrôlé de termes, ayant entre eux des relations d'ordre hiérarchique (par exemple du terme générique vers le terme spécifique), et qui s'applique à un ou plusieurs domaines del a connaissance. Les relations entre les termes représentent le corpus sémantique d'un domaine et tiennent compte de l'évolution du domaine concerné. Le thésaurus est donc un outil en construction permanente. Il se doit d'être un instrument de travail éminemment flexible et adaptable.

❖ **Manuelle** : sélectionner les termes importants appartenant aux documents jugés pertinents par l'utilisateur, et renforcer l'importance de ces termes dans la nouvelle formulation de la requête.

V. Modélisation des systèmes de recherche d'information

Le modèle joue un rôle central dans la (**RI)**, c'est lui qui détermine le comportement clé d'un système de recherche d'information (**SRI**).

Il existe un grand nombre de modèles de recherche d'information, et ces modèles diffèrent principalement, par la façon dont les informations disponibles sont représentées, et par la façon d'interroger la base documentaire.

V.1 Principaux modèles de recherche d'information

Comme illustré dans la figure 1.3, on distingue 3 principaux modèles dans le domaine de recherche d'information :

✓ les modèles booléens basés sur la théorie des ensembles comme le *modèle booléen* (booléen model), le *modèle booléen étendu* (extended boolean model) et le modèle basé sur les *ensembles flous* (fuzzy set model).

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

✓ les modèles algébriques (vectoriels) comme le *modèle vectoriel* (vector model), le *modèle vectoriel généralisé* (generalized vector model), *Latent Semantic model* (LSI) et le *modèle connexionniste*.

Dans ces modèles, la pertinence d'un document vis-à-vis d'une requête est définie par des mesures de distance dans un espace vectoriel

✓ les modèles probabilistes tels que le modèle probabiliste général, le modèle de réseau de document ou d'inférence (Document Network), et le modèle de langage.

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

Nous présentons dans la suite, les principaux modèles issus de chacun de ces trois groupes et qui sont largement exploités en (**RI)** à savoir le modèle booléen, le modèle vectoriel et le modèle probabiliste.

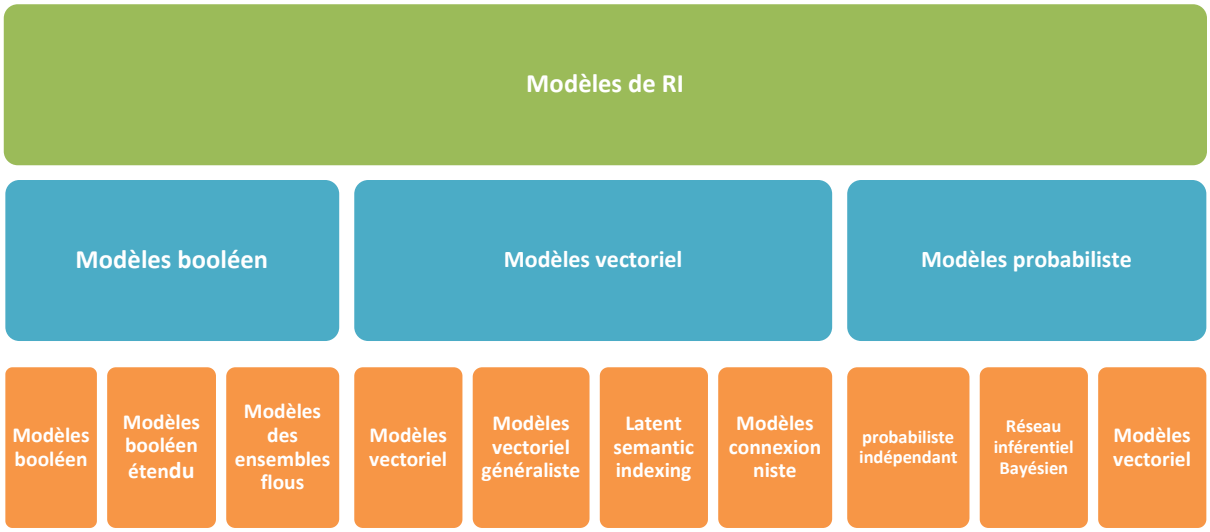

Figure 1.3 taxonomie des modèles en RI

- ✓ **Modèle Booléen**

Le plus simple des modèles de (**RI)**, le premier imposé dans le monde de la (**RI)**, reconnu pour sa force pour faire une recherche très restrictive et obtenir pour un utilisateur expérimenté, une information exacte et spécifique, il considère que les termes de l'index sont présents ou absents d'un document, en conséquence les poids des termes dans l'index sont binaires **c'est-à-dire** $W_{ij}= \{1,0\}$.

Dans ce modèle, un document « **d** » est représenté comme une conjonction logique des termes non pondérés.

Exemple :

$d= t_1 \cap t_2 \cap t_3 ..... \cap t_n$

Une requête « **q »** est composée de termes liés par 3 connecteurs logiques **ET, OU, NON. Exemple** :

$q= (t_1 \cap t_2) \cup (t_3 \cap \neg t_4)$

La correspondance entre une requête et un document (notée par rsv (d, q)) est déterminée de la façon suivante :

RSV (d, ti) = 1 si ti ∈ d ; 0 sinon.

RSV (d, $q_1 \cap q_2$) = 1 si RSV (d, q_1) = 1 et RSV (d, q_2) = 1 ; 0 sinon.

RSV (d, $q_1 \cup q_2$)= 1 si RSV (d, q_1)= 1 ou RSV (d, q_2)= 1 ; 0 sinon.

RSV (d, $\neg q_1$)= 1 si RSV (d, q_1)= 0 ; 0 sinon.

Avec :

RSV (d, $\neg q_1$) = $\neg$RSV (d, q_1).

Où :

q : la requête

$\mathbf{t_i}$: i^{ieme} terme d'indexation.

$\mathbf{w_{ij}}$: poids du terme $\mathbf{t_i}$ dans le document $\mathbf{d_j}$.

rsv (**q**, $\mathbf{d_j}$) : valeur de pertinence associée aux documents $\mathbf{d_j}$ relativement à la requête **q.**

Exemple :

La recherche des documents qui parlent de système d'exploitation ou de PHP mais pas de réseaux informatiques, s'exprime par la requête :

$$q = (se \cup PHP) \cap \neg réseau$$

- Le modèle booléen est plus facile à implémenter et nécessite relativement peu de ressources [SALT90].
- Le langage de requête booléen est plus expressif que celui des autres modèles [CROF87].

Ce modèle convient aux utilisateurs sachant exactement leurs besoins et en mesure de les formuler précisément avec le vocabulaire qu'ils maîtrisent parfaitement.

- Par contre, Il est difficile aux novices de formuler une requête combinant plusieurs opérateurs logiques, notamment pour les questions complexes. L'importance relative des mots clés ne peut pas être exprimée.

Le classement des documents extraits par ordre de pertinence est difficile.

En plus, la reformulation automatique des requêtes par la technique du « Relevance Feedback» est plus ardue.

✓ **Modèle vectoriel**

Le modèle vectoriel standard est un modèle de recherche d'information très connu. Il intègre dans un espace vectoriel une représentation qui symbolise les documents ou les requêtes en fonction des termes d'indexation qui les composent.

La forme d'implémentation la plus connue du modèle vectoriel est le système de recherche documentaire SMART [Salton et al., 1971], [Salton 1983].

Ce modèle représente les requêtes et les documents sous forme de vecteurs qui sont placés dans un espace vectoriel spécifique. L'espace est de dimension **N** (**N** étant le nombre de termes d'indexation de la collection de documents). Cet espace vectoriel est définit par l'ensemble de termes que le système a rencontré durant l'indexation.

Soit l'espace vectoriel suivant $< t_1, t_2, t_3, \ldots . t_n >$ Chaque document et requête sont respectivement représentés par un vecteur document et un vecteur requête :

- $d_j = (w_{1j}, w2_j, \ldots, w_{nj})$

w_{ij} =poids du terme $\mathbf{t_i}$ dans le document $\mathbf{d_j}$

- $Q= (w_{1Q}, w_{2Q},\ldots, w_{nQ})$

w_{iQ} =poids du terme $\mathbf{t_i}$ dans la requête **Q**.

Dans l'exemple de la figure1.4 on a une représentation de deux documents ($d_1$et d_2) et d'une requête (q) dans un espace vectoriel. La proximité de la requête aux documents est représentée par les angles α et θ entre les vecteurs.

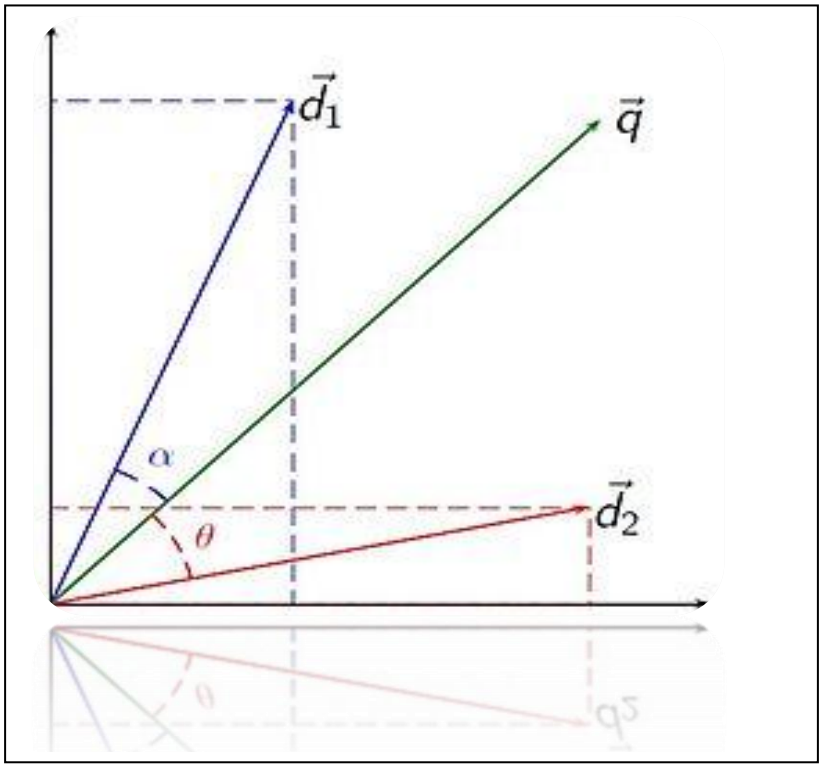

FIGURE 1.4 : Représentation vectorielle de deux documents ($d_1$et d_2) d'une requête(q) dans un espace composé de deux termes.

Étant donné ces deux vecteurs, leur degré de correspondance est déterminé par leur similarité. Cette similarité entre ces deux vecteurs peut être exprimée par des différents méthodes les plus utilisée sont :

Mesures	Formules
Le produit scalaire	$RSV(q,d_i)=\sum_{j=1}^{\lvert T\rvert} w_{qj}.w_{ij}$
La mesure de cosinus	$RSV(q,d_i)=\frac{q.d_i}{\lVert q\rVert.\lVert d_i\rVert}=\frac{\sum_{j=1}^{\lvert T\rvert} w_{qj}.w_{ij}}{\sqrt{\sum_{j=1}^{\lvert T\rvert} {w_{qj}}^2 \sum_{j=1}^{\lvert T\rvert} {w_{ij}}^2}}$
La mesure de Dice	$RSV(q,d_i)=\frac{2\times\sum_{j=1}^{\lvert T\rvert} w_{qj}.w_{ij}}{\sum_{j=1}^{\lvert T\rvert} {w_{qj}}^2+\sum_{j=1}^{\lvert T\rvert} {w_{ij}}^2}$
La mesure de Jaccard	$RSV(q,d_i)=\frac{\sum_{j=1}^{\lvert T\rvert} w_{qj}.w_{ij}}{\sum_{j=1}^{\lvert T\rvert} {w_{qj}}^2+\sum_{j=1}^{\lvert T\rvert} {w_{ij}}^2-\sum_{j=1}^{\lvert T\rvert} w_{qj}.w_{ij}}$

Le but final est d'arriver à retourner une liste ordonnée de documents selon ce degré. L'avantage de ce modèle réside dans l'expression des besoins de l'utilisateur, contrairement au modèle booléen où les termes de la requête doivent être reliés par des connecteurs logiques, l'utilisateur peut ici exprimer son besoin en information en langage naturel ou sous forme d'une liste de mots-clés. La pondération des termes augmente les performances des systèmes, le modèle permet de renvoyer des documents qui répondent approximativement à la requête, et la fonction d'appariement permet de trier les documents selon leur degré de similarité avec la requête.

Théoriquement, le modèle vectoriel a l'inconvénient de considérer que les termes de l'index sont tous indépendants.

Les requêtes et documents sont essentiellement similaires alors que certains résultats produits par le calcul de similarité requête - document ne reflètent pas la réalité

Cependant, en pratique, la prise en compte globale de la dépendance des termes peut faire baisser les performances d'un système (puisque les dépendances sont généralement locales).

- ✓ **Modèle probabiliste**

Le modèle probabiliste utilise un modèle mathématique fondé sur la théorie de la probabilité. Le processus de recherche se traduit par calcul de proche en proche, du degré ou probabilité de pertinence d'un document relativement à une requête. Pour ce

faire, le processus de décision complète le procédé d'indexation probabiliste en utilisant deux probabilités conditionnelles :

- P (w_{ji}/pert) : Probabilité que le terme $\mathbf{t_i}$ occurre dans le document $\mathbf{D_j}$ sachant que ce dernier est pertinent pour la requête.

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

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

Soit $D_j (t_1, t_2, \ldots, t_N)$ où

t_i =1 si $\mathbf{t_i}$ indexe le document $\mathbf{d_j}$

t_i =0 sinon

P (pert /D_j)= $\boldsymbol{P\left(\frac{Dj}{pert}\right) * P(pert)/P(Dj)}$

P (nonpert /D_j)= $\boldsymbol{P\left(\frac{Dj}{nonpert}\right) * P(nonpert)/P(Dj)}$

Avec :

P (pert /D_j) est la probabilité de pertinence du document $\boldsymbol{D_j}$ sachant sa description.

P(D_j)=P (D_j /pert) * P (pert) + P (D_j /Nonpert) * P(Nonpert)

P (D_j /pert) (respectivement P *(D_j /Nonpert*) est la probabilité d'observer le document D_j sachant qu'il est pertinent (respectivement non pertinent).

Si l'on considère l'indépendance des termes

P (pert/D_j) = P (t_1/pert) * P (t1)…P (t_N/pert) * P (t_N)

P (Nonpert/D_j) = P (t_1/Nonpert) * P (t_1)…P (t_N/Nonpert) * P (t_N)

Où :

P (t_i/pert)= $\frac{ri}{R}$

P (t_i/Nonpert)= $\frac{mi-ri}{M-R}$

M : nombre total de documents dans la collection

R : nombre de documents pertinents pour une requête

r_i : nombre de documents pertinents dans lesquels le terme $\mathbf{t_i}$ apparaît.

m_i : nombre total de documents dans lesquels le terme $\mathbf{t_i}$ apparaît.

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

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

Selon Savoy **[SAV094].**Le modèle de recherche probabiliste est plus efficace que le modèle de recherche booléen, mais **moins** performant que le modèle de recherche vectoriel.
Par contre, Il n'existe pas de méthode d'estimation de la pertinence des termes avant toute extraction de document pertinent. Cette estimation se fait à posteriori.

VI. Evaluation des SRI

L'évaluation des (**SRI)** constitue une étape importante dans l'élaboration d'un modèle de (**RI).** En effet, elle permet de caractériser le modèle et de fournir des éléments de comparaison entre modèles.

En général, tout système de recherche d'information a deux objectifs principaux :

Le premier est de retrouver tous les documents pertinents pour une requête utilisateur.

Le deuxième est de rejeter les documents jugés non pertinents.

L'évaluation des systèmes peut être abordée selon deux angles : l'efficience et l'efficacité.

- L'efficience regroupe le temps et l'espace :

Un système est considéré meilleur lorsque le temps entre la formulation de la requête et la réponse du système est court et l'espace occupé par le système est faible.

- L'efficacité d'un système peut être mesurée par les critères suivants :

• L'effort, intellectuel ou physique, nécessaire aux utilisateurs pour formuler les requêtes, conduire leur recherche, voir les documents résultats

• La présentation du résultat (capacité de l'utilisateur à utiliser les documents retrouvés)

• La qualité du corpus vis à vis du besoin de l'utilisateur (dans quelle mesure tous les documents pertinents sont dans le corpus)

• La capacité du système à retrouver des documents intéressants et à éliminer les autres. Cette caractéristique semble être la plus importante.

VI.1. Mesures d'évaluation :

a. **Rappel et précision :**

❖ **Précision** *:* La précision mesure la proportion de documents pertinents relativement à l'ensemble des documents restitués par le système. Elle mesure la capacité du système à rejeter tous les documents non pertinents à une requête donnée par le rapport entre l'ensemble des documents sélectionnés pertinents et l'ensemble des documents sélectionnés.

$$Précision = \frac{\text{nombre total de documents pertinents trouvés par le systéme}}{\text{nombre total de documents retrouvés par le systéme}}$$

❖ Rappel : Le rappel mesure la proportion des documents pertinents restitués par le système relativement à l'ensemble des documents pertinents contenus dans la base documentaire. Elle mesure la capacité du système à retrouver tous les documents pertinents répondant à une requête.

$$Rappel = \frac{\text{nombre total de documents pertinents retrouvés par le systéme}}{\text{nombre total de documents pertinents dans la collection}}$$

❖ Rappel/Précision : La précision mesurée indépendamment du rappel et inversement est peu significative. Pour pouvoir examiner les résultats efficacement, on calcule la paire des mesures (taux de rappel (silence), taux de précision (bruit)) à chaque document restitué. Pour un système idéal, le taux de précision est égal à celui de rappel.

$$S = 1 - R \qquad \text{et} \qquad B = 1 - P$$

Le système parfait trouverait seulement les documents pertinents, avec une précision et un rappel de 100%. En pratique, les mesures de rappel et précision évoluent inversement, ce qui signifie que la courbe interpolée de précision en fonction du rappel est décroissante. Plus la courbe est élevée, plus le système est performant.

VII. Domaines d'application :

La RI est un domaine vaste qui se situe dans les frontières de plusieurs disciplines tel que :

I. Recherche adhoc.
II. Classification /catégorisation (*clustering*), Question-réponses (*Query answering*).
III. Filtrage d'information (*filtering/recommendation*)
IV. Méta-moteurs (*data-fusion*, *Meta-search*)
V. Résumé automatique (*Summarization*)
VI. Croisement de langues (*cross language*)
VII. Fouille de textes *(Text mining*)

VIII. CONCLUSION :

Dans ce chapitre nous avons décrit les principes fondamentaux des systèmes de recherche d'information et les principaux concepts de base.

Nous avons définit l'architecture des SRI, le processus de recherche d'information (processus en U), les principaux modèles de recherche, les méthodes d'évaluations des (**SRI)** et enfin les critiques et les limites de ces méthodes.

Le chapitre suivant sera consacré à la présentation de notre S.R.I en détaillent chaque étapes, ainsi que les différents mesures utiliser (mesure de pondération, mesure de similarité).

Chapitre II

Réalisation de système de recherche d'information

Introduction

Les systèmes de recherche d'information (**SRI**), servent d'interface entre une collection contenant des quantités considérables de documents et des utilisateurs cherchant des informations susceptibles de se trouver dans cette collection, en utilisant des requêtes. Le besoin d'avoir des S.R.I vient du fait de la nécessité d'avoir les documents du plus bref délai.

Ainsi, notre travail consiste à implémenter un système de recherche d'information en se basent sur le processus en U de recherche d'information décrit dans le chapitre précédent.

La réalisation d'un tel système exige de faire des choix. Dans ce chapitre, nous allons présenter les techniques implémentées dans notre système à savoir : Le choix de méthode de représentation, le choix de la méthode de pondération et le choix de la méthode d'appariement.

I. Notre S.R.I

Comme tout S.R.I, notre système suit le processus en U avec toutes ces étapes à savoir : l'indexation, pondération, l'appariement document -requête.

II.1. l'indexation

L'objectif de l'indexation est de transformer l'ensemble des documents de la base documentaire en une matrice M dont les longueurs représentent les documents et les colonnes représentent les termes. Le problème majeur de cette étape est de savoir comment transformer un texte sous une forme exploitable par la machine. Plusieurs techniques de représentations existent. Dans notre S.R.I, on a choisi d'utiliser deux techniques pour la représentation de textes à savoir : la représentation par mot et la représentation par lemme.

- Représentation par mot :

Cette technique consiste à représenter chaque document sous forme d'un vecteur de mot (unité lexicale). Malgré sa simplicité, cette méthode présente les inconvénients suivants :

1- Pour certaines langues, il est difficile délimiter les mots ; par exemple, dans la langue allemande la chaine de caractères

« Levensversicherungsgesellschaftsangestellter » signifie « employé d'une société d'assurance vie ».

2- le chinois et le japonais ne séparent pas les mots par des espaces, ce qui peut mener à plusieurs segmentations

3- l'arabe et l'hébreu sont écrits de droite à gauche, mais certains éléments tels que les nombres sont écrits de gauche à droite.

4- Il est important aussi de reconnaître les mots composés car ce sont des unités de sens. C'est d'autant plus important lorsque ce sens ne se déduit pas des mots composants. Par exemple : « arbre à cames » ou « pomme de terre ».

5- Les entités nommées sont des mots ou des groupes de mots qui désignent des personnes, des organisations, des dates, des lieux, etc. Par exemple, si un texte contient l'expression : « 14 juillet 1789 » il est plus intéressant de l'indexer globalement par cette date plutôt que les trois termes : « 14 », « juillet » et « 1789 ».

Cette représentation nécessite les prétraitements suivants :

- Tokénisation des documents.
- Elimination des majuscules.
- Elimination des mots vides.

a- Tokénisation des documents

Cette étape consiste a transformé un texte en un ensemble de termes, on a choisi de considérer un mot comme une suite de caractères situés entre deux séparateurs. L'algorithme suivant illustre cette étape.

```
Entrées : un document, liste de séparateurs.
Sortie : tableau de terme.
Variable caractère : C, tableau de mot : TAB
Début
    Ouvrir le fichier
    C= lire caractère
      Tant que le caractère lu est différent  EOF faire
                Si le caractère n'est pas un séparateur Alors
                          Ajouter le caractère à l'unité
                 Sinon insère l'unité dans TAB
                Finsi
                C = lire caractère
     Fin.
```

La figure II.1 présente la liste des séparateurs utilisée dans notre système

‘\n’ , ‘ ‘
‘ ,’ , ‘ ;’ , ‘ :’ , ‘ .’ , ‘ ?’ , ‘ ! ‘ : La ponctuation
‘=’ , ‘+’ , ‘-‘ ,’*’ , ‘<’ , ‘>’ : les opérateurs arithmétiques et de comparaison
‘(‘ , ‘)’ , ‘[‘ , ‘]’ , ‘{‘ , ‘}’ : Les parenthèses
‘ « ’ , ‘\’ , ‘/’ , ‘ , ‘%’
‘&’ , ‘~’ , ‘#’ , ‘|’ , ‘ _’ : les opérateurs logiques
‘@’ , ‘§’ , ‘$’ , ‘£’
‘0’ , ‘1’ , ‘2’ , ‘3’ , ‘4’ , ‘5’ , ‘6’ , ‘7’ , ‘8’ , ‘9’ : les chiffres.

Figure II.1 : la liste des séparateurs

A la fin de cette opération chaque document sera représenté sans séparateurs (espaces de séparation des chiffres, des mots, des ponctuations, etc.) la figure II.2 présente un exemple de cette tokénisation.

La pondération ; est la Phase la Plus Importante dans la (SRI), le {SRI} 20 répond à un@ Besoin.

la	pondération	est	la	Phase	la	Plus	Importante	dans
la	SRI	le	SRI	répond	à	un	Besoin	

Figure II.2 : tokénisation de document

b- <u>Elimination des majuscules :</u>

Afin de pouvoir réduire la taille du tableau il est nécessaire de formater les mots avec majuscules et les mots minuscules comme étant un seul mot. Par exemple les mots « phase, Phase, pHAse, PHASE » considèrent comme un seul mot.

Voilà la figure II.3 illustre l’exemple précédent en éliminant les majuscules.

la	pondération	est	la	Phase	la	Plus	Importante	dans
la	SRI	le	SRI	répond	à	un	Besoin	

la	pondération	est	la	phase	la	plus	importante	dans
la	sri	le	sri	répond	à	un	besoin	

Figure II.3 : élimination des majuscules

c- Elimination des mots vides :

Les mots vides ou mots outils sont les mots non significatifs trouvés dans les documents.

En effet, ces mots ne traitent pas le sujet du document mais ils permettent de lier entre les mots d'une phrase pour la structurer comme les articles, les conjonctions de coordination, les verbes auxiliaires, etc.

Chaque langue a sa propre liste des mots vides. Dans notre application nous avons utilisé un fichier qui comporte 124 mots vides de la langue française et 582 mots de la langue anglaise.

Ces mots ne portent pas de sens**.**

Voila un extrait des mots vides de la langue française et anglaise :

Alors, au, aucuns, aussi, autre, avant, avec, avoir, bon, car, ce, cela, ces, ceux, chaque, ci, comme, comment, dans, des, du, dehors, depuis, deux, devrait, doit, donc, dos, droite, elle, elles, en, Il, ils, la, le …….	a, about, above, after, again, against, all, am, an, and, any, are, as, at, be, because, been, before, being, below, between, both, but, by, can, did, do, does, doing, down, during, each, few, for, from, further, ……..

```
Entrée : tableau de mots, la liste des mots vides.

Sortie : tableau de mot sans mots vides.

Début

  Pour chaque mot de tableau faire

    Si le mot figure dans la liste des mots vides
alors

          Supprimer le mot du tableau

    Finsi

  Finpour

Fin
```

Cette étape permet d'analyser et de réduire la taille de l'index.la figure II.4 montre l'exemple précédent après élimination les mots vides.

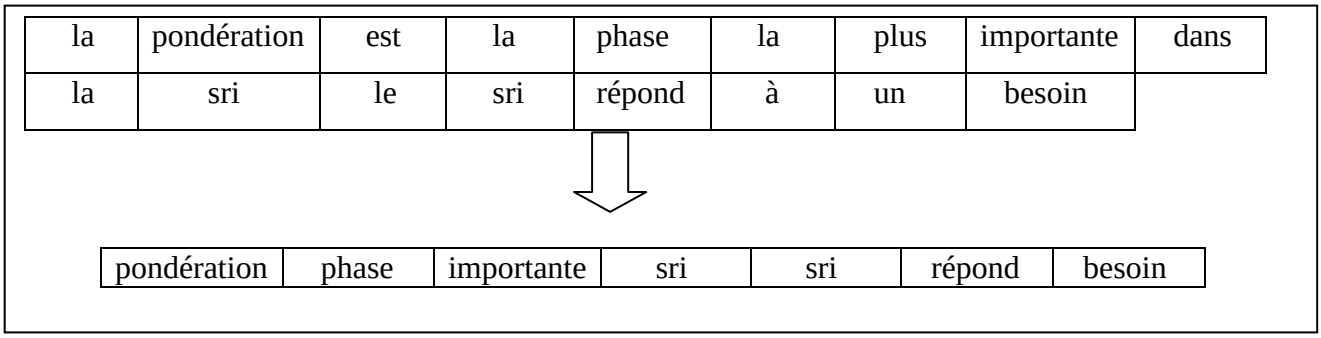

Figure II.4 : élimination des mots vides.

- Représentation par lemme :

La technique de lemmatisation consiste à remplacer chaque mot par sa forme canonique (lemme), en effet elle consiste à remplacer les verbes par leur forme infinitive et les noms par leur forme au masculin singulier. Voici quelques exemples de lemmatisation :

- écologie, écologiste, écologique ------» écolog.
- Informatique ------» informat.
- Petits, petite, petites ------» petit.
- Joue, jouer ------» jou.
- malade, malades, maladie, maladies, maladive ------» malad.

Cette phase consiste à indexer un ensemble de mots par un seul mot qui représente le même concept.

Il y a plusieurs algorithmes de lemmatisation telle que « l'algorithme de carry », « algorithme de paice/husk ».Dans notre système on a choisi d'utiliser « l'algorithme de porter », ce dernier est un algorithme de normalisation des mots. Il permet de supprimer les affixes des mots pour obtenir une forme canonique du mot. Cet algorithme a été proposé par Martin Porter en 1980, il est utilisé pour la langue anglaise, mais son efficacité est limitée pour la langue française où les flexions sont plus importants et plus diverses. Il se présente comme un ensemble de règles dont l'application successive à un mot de l'anglais produit la racine de ce mot. Il reste toutefois un algorithme fondamental couramment enseigné en TALN (compréhension du texte)

```
Entrée : tableau de mot.
Sortie : tableau des lemmes
Début
  Pour chaque mot du tableau faire
    Lem ← porter (mot)
      Insère lem dans le tableau des lemmes
  Finpour
Fin
```

Cette phase est utilisable car elle est consacré à réduire le nombre de terme dans le tableau et permet de représenter par un même descripteur des mots qui ont le même sens. Enfin le cas de notre exemple comme indiqué dans la figure II.5, on remarque que la taille du tableau a été réduite.

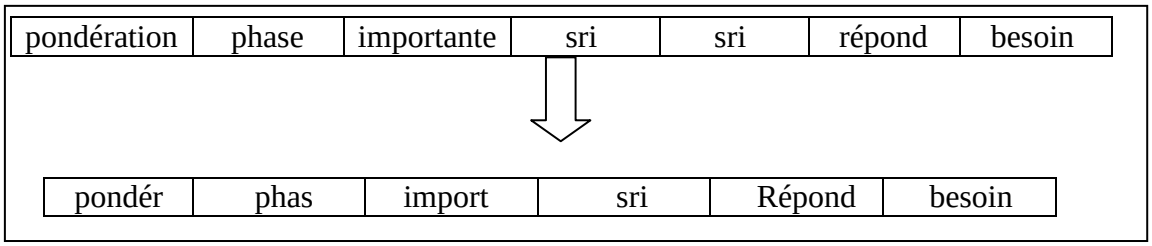

Figure II.5 : lemmatisation des mots

II.2. la pondération :

Une fois la liste des mots rencontrés dans les documents (formant l'index) est établit, la pondération consiste à donner un poids à chaque terme dans chaque document de la base documentaire.

Il existe plusieurs fonctions de pondération comme décrit dans le chapitre précédent, dans ce qui suit nous allons présenter les différentes pondérations qu'on a utilisées dans notre application :

- ✓ le facteur Tf (Term Frequency) est basé sur la fréquence d'un terme dans le document, Plus un terme est fréquent dans un document plus il est important dans ce dernier.
- ✓ Le facteur IDF (Inverse Document Frequency) se base sur le nombre de documents contenant un terme donné. En effet, un terme apparaissant dans tous les documents n'est pas important. Sa formule est la suivante :

  Idf =Log (N/n_i), ou :

- N est la taille de la collection

- n_i le nombre de documents contenant le terme t_i

✓ Tf*idf **:** cette pondération est une combinaison des deux pondérations précédentes « *Tf* » *et* « *idf* ». Ainsi, pour qu'un terme soit important dans un document il ne suffit pas qu'il soit fréquent dans le document mais aussi absent dans les autres documents. (Voir chapitre I, formule 1).

✓ Les Tf_{ij} doivent être normalisées en divisant chaque Tf_{ij} par le maximum des fréquences pour un même document.

✓ Tfc : cette technique est une variante de la méthode de tf*idf, elle prend en considération la taille des documents (Voir chapitre I, formule 2).

✓ Ltc : c'est une approche légèrement différente, qui emploie le logarithme de la fréquence d'un terme. (Voir chapitre I, formule 3).

```
Entrée : tableau des documents
Sortie : matrice (documents * termes)
Début
  Choisir la technique de pondération
  Pour chaque document i faire
    Pour chaque terme j faire
    Matrice [i] [j] ← pondération (terme j, document i)
    Finpour
  Finpour
Fin
```

La figure II.6 illustre un exemple de pondération en utilisant la mesure "TFIDF "

Voici notre corpus :

D_1 :" Une organisation retenue pour nos travaux dans une organisation ".

D_2 :" Un document renvoie à un ensemble formé par un support et une information ".

D_3: " Une approche subjective élabore une distinction entre « document par attribution» et « document par intention», document ".

*Calculons la pertinence du terme t_1=" document " pour les 3 documents en utilisant tf*idf **:**

Tf ("document", D_1) = 0 ; Tf ("document", D_2) = 1 ; Tf ("document", D_3) = 3

Idf ("document") = $\log_{10}$ (n/df ("document")) = $\log_{10}$ (3/2) = 0.1761

Ce qui fait:

Tf Idf ("document", D_1) = 0

Tf Idf ("document", D_2) = 0.1761

Tf Idf ("document", D_3) = 0.5283

Figure II.6 : la pondération par la mesure « TFIDF »

Pour la requête de l'utilisateur, la pondération de la requête consiste à affecter le poids '1'au mot de la requête s'il existe dans le tableau des mots, sinon on lui affecte le poids '0'.

II. L'appariement document –requête :

Les SRIs intègrent un processus de recherche/décision qui permet de sélectionner l'information jugée pertinente pour l'utilisateur. En effet, une mesure de similitude (correspondance) entre la requête indexée et les descripteurs des documents de la collection est calculée. Seuls les documents dont la similitude dépasse un seuil prédéfini sont sélectionnés par le SRI.

La fonction de correspondance est un élément clé d'un SRI, car la qualité des résultats dépend de l'aptitude du système à calculer une pertinence des documents la plus proche possible du jugement de pertinence de l'utilisateur.

Étant donné le vecteur du document et le vecteur de la requête, leur degré de correspondance est déterminé par leur similarité.

Il y a plusieurs mesures pour calculer la similarité entre ces deux vecteurs. Dans notre système on a utilisé les trois mesures suivantes :

- **Produit scalaire :**

Cette mesure permet de calculer le nombre de correspondances entre le vecteur document et le vecteur requête

$$RSV(Q_i,D_{ij})=\sum_{i=1}^{n} qi * dij$$

Ou :

q : la requête

d : la matrice

n : le nombre de mot

- **Mesure du cosinus :**

Elle mesure le cosinus de l'angle formé entre le vecteur document et le vecteur requête

$$RSV\ (Q_i, D_{ij}) = \frac{\sum_{i=1}^{n} qi * dij}{\sqrt{\sum_{i=1}^{n} qi^2} * \sqrt{\sum_{i=1}^{n} dij^2}}$$

- **Mesure de Jaccard :**

C'est une variante de la mesure de cosinus, elle consiste à comparer la similitude des termes entre les documents et la requête.

$$RSV\ (Q_i, D_{ij}) = \frac{\sum_{i=1}^{n} qi * dij}{\sum_{i=1}^{n}(qi^2) + \sum_{i=1}^{n}(dij^2) - \sum_{i=1}^{n} qi*dij}$$

La figure II.7 illustre un exemple de similarité en utilisant la mesure "cosinus "

Voici nos 3 documents (D_1, D_2, D_3) et la requête (Q)

2	1	0	1	D_1 :
0	0	2	0	D_2 :
1	2	1	1	D_3 :

Et

1	0	2	1	Q :

*Calculons la similitude pour les 3 documents en utilisant la mesure « cosinus » **:**

D1 et Q :

$$\textbf{Cos (Q, D}_1\textbf{)} = \frac{\sum_{i=1}^{4} qi * dij}{\sqrt{\sum_{i=1}^{4} qi^2} * \sqrt{\sum_{i=1}^{4} dij^2}} = \frac{\sum_{i=1}^{4}(1*2+0*1+2*0+1*1)}{\sqrt{\sum_{i=1}^{4}(1^2+0^2+2^2+1^2)} * \sqrt{\sum_{i=1}^{4}(2^2+1^2+0^2+1^2)}}$$

$$= \frac{3}{\sqrt{6} * \sqrt{6}} = \frac{3}{6} = \mathbf{0.5}$$

D2 et Q :

$$\textbf{Cos (Q, D}_2\textbf{)} = \frac{\sum_{i=1}^{4}(1*0+0*0+2*2+1*0)}{\sqrt{\sum_{i=1}^{4}(1^2+0^2+2^2+1^2)} * \sqrt{\sum_{i=1}^{4}(0^2+0^2+2^2+0^2)}} = \frac{4}{\sqrt{6} * \sqrt{4}} = \frac{4}{4.09} = \mathbf{0.81}$$

D3 et Q :

$$\textbf{Cos (Q, D}_3\textbf{)} = \frac{\sum_{i=1}^{4}(1*1+0*2+2*1+1*1)}{\sqrt{\sum_{i=1}^{4}(1^2+0^2+2^2+1^2)} * \sqrt{\sum_{i=1}^{4}(1^2+2^2+1^2+1^2)}} = \frac{4}{\sqrt{6} * \sqrt{7}} = \frac{4}{6.49} = \mathbf{0.62}$$

Figure II.7 : la similarité par la mesure « cosinus »

Le but final est d'arriver à retourner une liste ordonnée de documents selon ce degré. L'avantage du modèle utilisé qui est bien le modèle vectoriel c'est qu'il permet de renvoyer des documents qui répondent approximativement à la requête. La fonction d'appariement permet de trier les documents selon leur degré de similarité avec la requête.

Entrée : tableau des documents

Sortie : tableau mesure

Début

Choisir la technique de similarité

Pour chaque document i faire

Mesure [i] ← concaténer (requête i)

Finpour

Trier le tableau mesure par ordre décroissant ;

Afficher la liste des documents triés ;

III. l'environnement utilisé dans notre S.R.I :

Pour l'implémentation de notre S.R.I, nous avons utilisé « python » comme un langage de programmation orienté objet, multi-paradigme. Il est disponible sous une licence libre et fonctionne sur la plupart des plates-formes informatiques, des supercalculateurs aux ordinateurs centraux, de Windows à Unix en passant par GNU/Linux, Mac OS, ou encore Android, iOS, et aussi avec Java ou encore NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser. il est utilisé dans de nombreux contextes et s'adapte à tout type d'utilisation grâce à des bibliothèques spécialisées comme le module libre « PyQt » qui permet de lier le langage Python avec la bibliothèque Qt distribué sous deux licences : une commerciale et la GNU GPL. Il permet ainsi de créer des interfaces graphiques. Une extension de QtDesigner (utilitaire graphique de création d'interfaces Qt) permet de générer le code Python d'interfaces graphiques.

VI. Description de notre application :

Notre SRI propose à l'utilisateur plusieurs interfaces. La figure II.6 illustre L'interface principale.

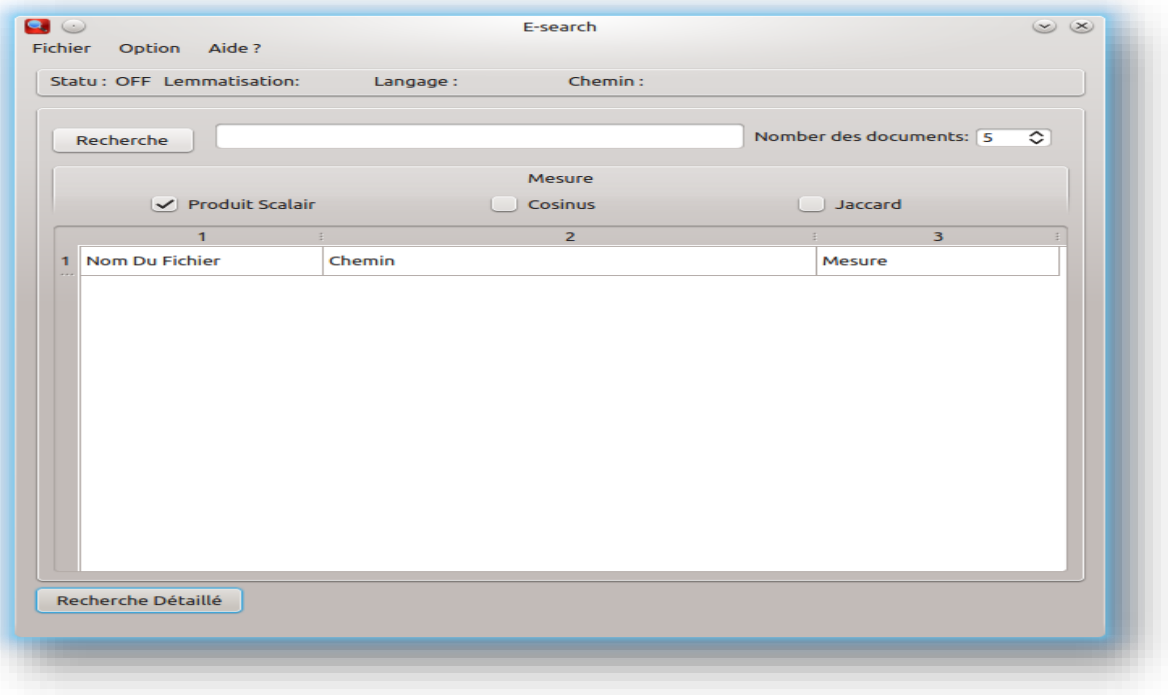

Figure II.6 : l'interface principale de l'application

Notre SRI doit effectuer en premier lieu l'indexation de la base documentaire. La barre d'information montre les propriétés de l'indexation comme indiqué dans la figure II.7 :

- Statu : indique si l'indexation est effectué (statu=on) ou non (statu=off).
- Lemmatisation : indique si la lemmatisation est pris en considération lors d'indexation (lemmatisation=on) ou non (lemmatisation=off).
- Langage : il indique le langage de la base documentaire.
- Chemin : il indique le chemin de la base documentaire.

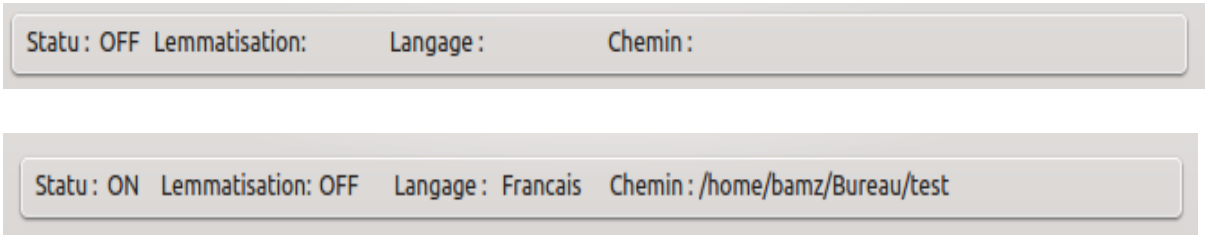

Figure II.7 : la barre d'information de l'index

Pour notre SRI, l'utilisateur peut choisir entre deux façons d'indexation :

- Effectuer une nouvelle indexation : pour cela l'utilisateur doit cliquer sur le bouton «option -> indexation» illustré dans la figure II.8, il doit choisi une des méthodes de pondération (indexation), on suppose que la méthode choisit est « TF » (tel que illustré la figure), il doit aussi sélectionner la langue utiliser et cocher la case de lemmatisation si il veut lemmatiser la requête et finalement, il doit choisi la base documentaire.

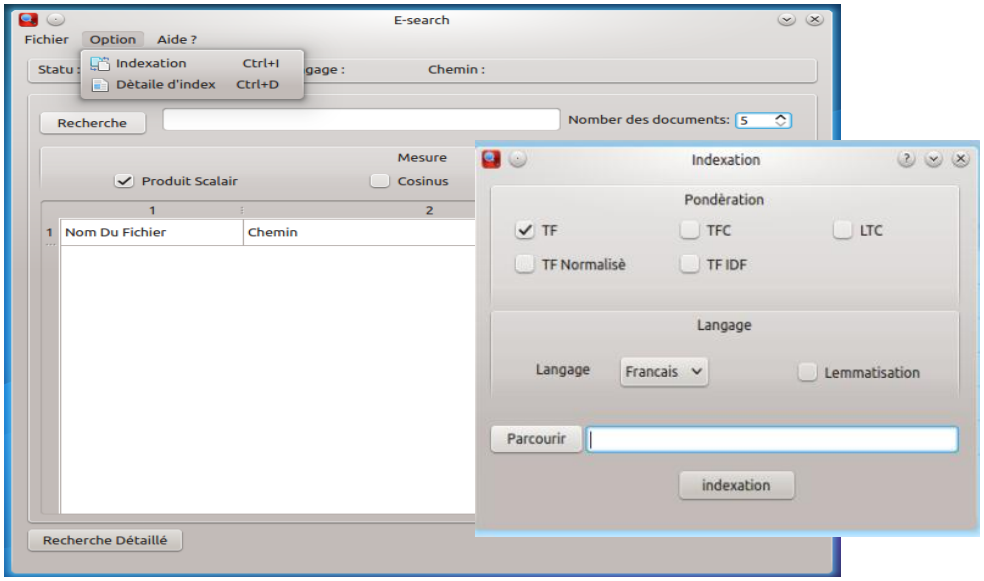

Figure II.8 : les propriétés d'indexation

- Charger une indexation existante : l'utilisateur peut aussi charger une indexation existante comme indiqué figure II.9 (fichier- ouvrir l'index), dans ce cas la, il doit donner le chemin de l'index choisi, le système va afficher la fenêtre (tel que illustré la figure) pour sélectionner l'index.

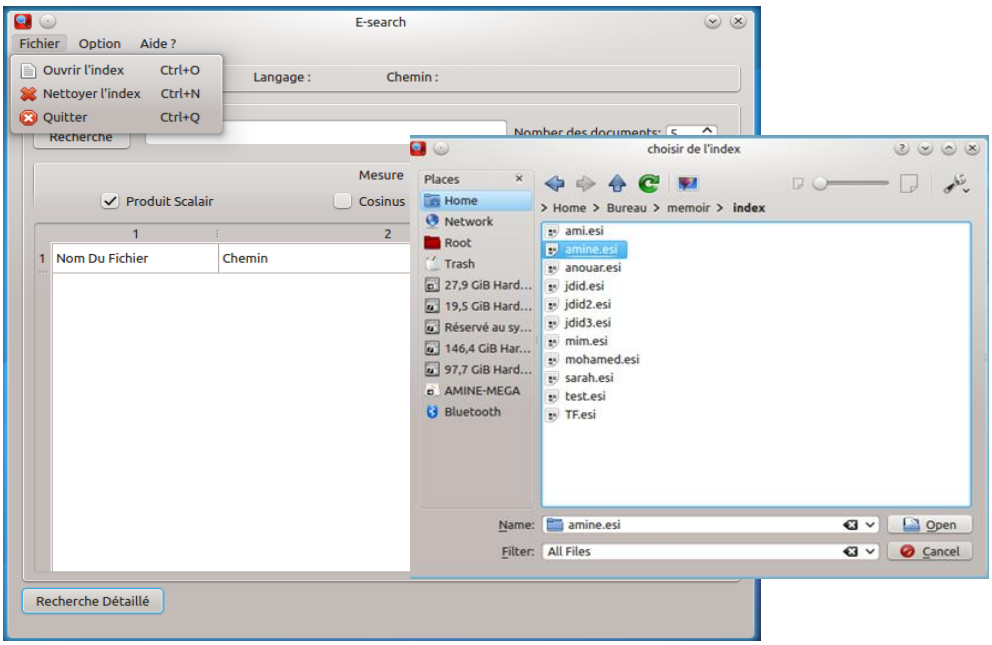

Figure II.9 : chargement d'un index existant

Une fois l'indexation achevée, l'utilisateur peut visualiser les résultats de cette dernière à travers l'interface « indexation détaillée » à partir du bouton (option -> détaille d'index) tel qu'illustré dans la figure II.10. Cette interface contient :

- Une matrice de pondération montrant le poids des termes dans chaque document.
- Les documents indexés.
- Dictionnaire des mots.
- L'état de lemmatisation.
- La méthode de pondération choisie.

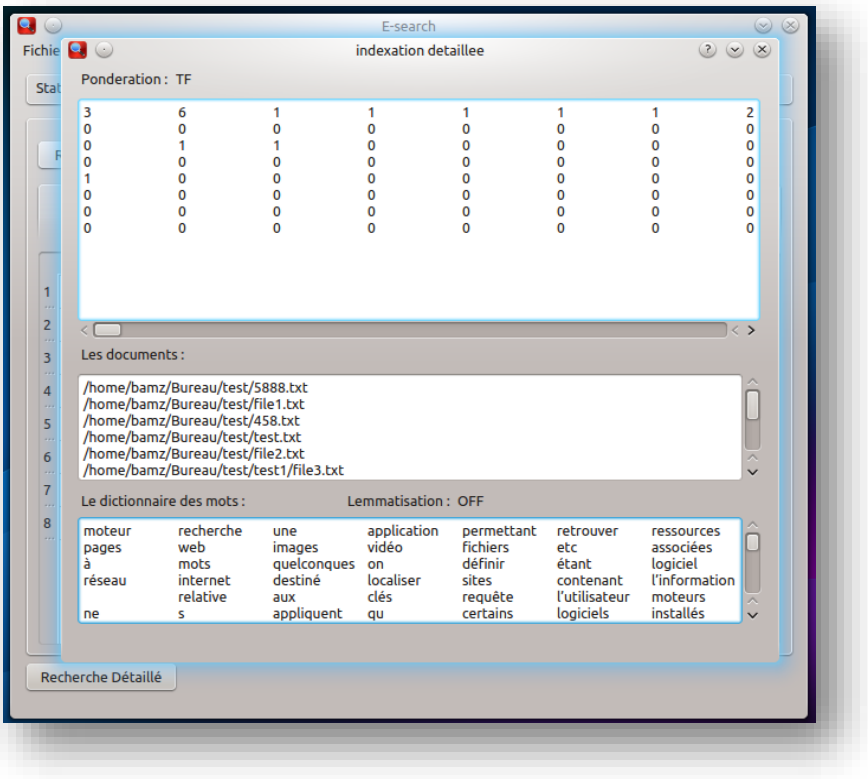

Figure II.10 : Indexation détaillée

Après avoir effectué l'indexation, l'utilisateur doit saisir sa requête et choisir une des mesures de similarités implémentées dans notre SRI à savoir : le produit scalaire, le cosinus et jaccard, il doit aussi choisir le nombre des documents que doit retourner le système comme illustré dans la figure II.11

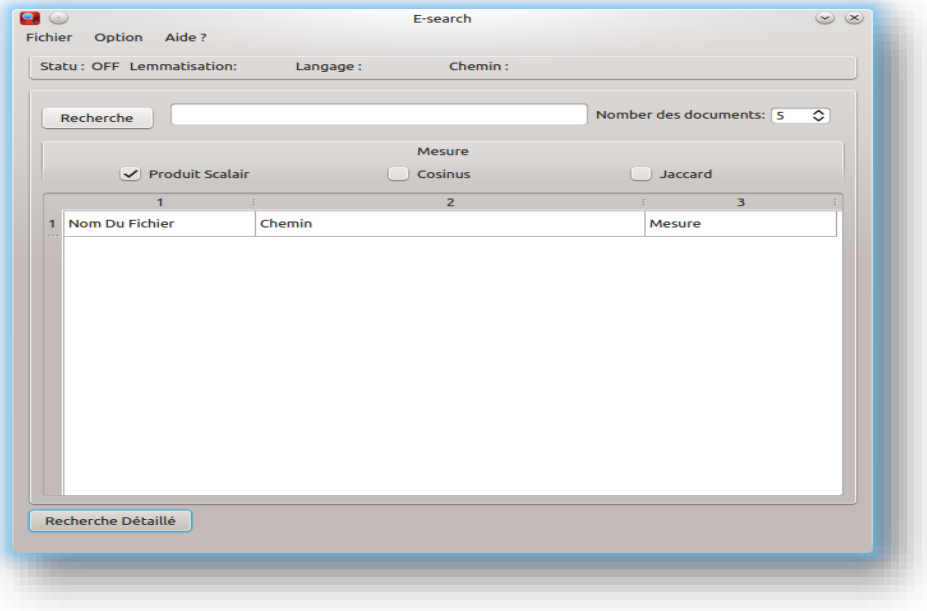

Figure II.11 : la saisie de la requête

En cliquant sur le bouton « recherche », le système va afficher les documents triés en ordre décroissant selon les similitudes avec la requête d'utilisateur indiqué dans la figure II.12

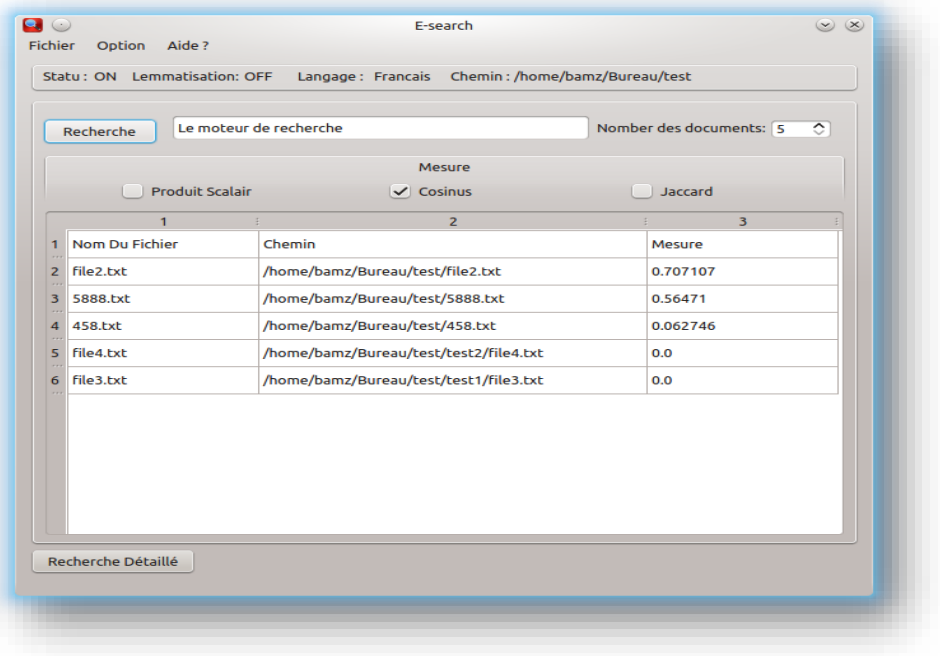

Figure II.12 : le résultat de la recherche

L'utilisateur peut visualiser les résultats des similitudes ainsi que le contenu des documents sélectionnés par le système. De plus, les mots de la requête figureront colorés(à travers le bouton de recherche détaillée). La figure II.13 illustre cette interface.

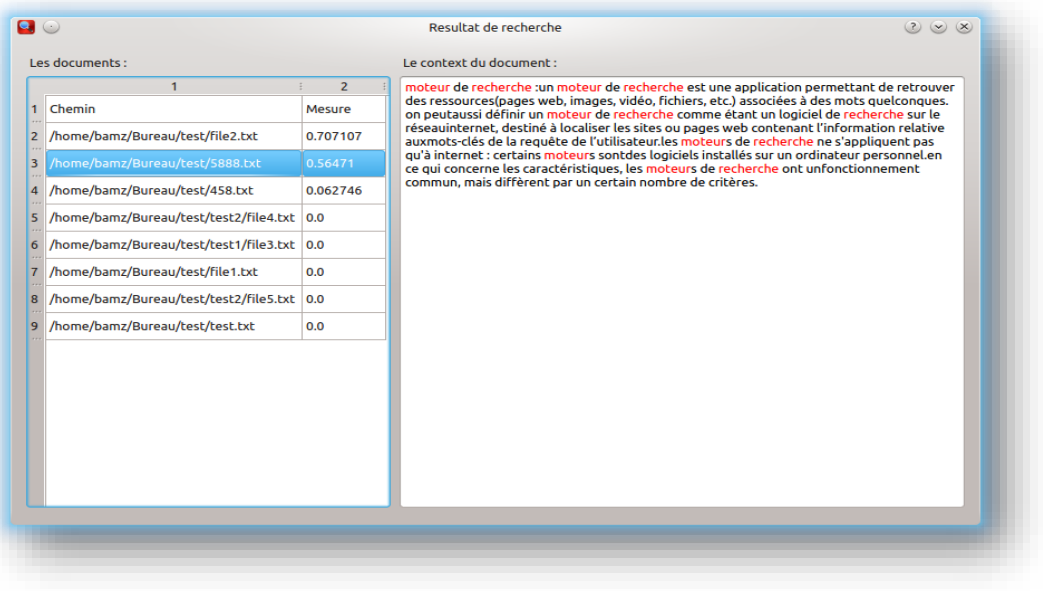

Figure II.13 : le resultat de la recherche détaillé

VII.Conclusion

Dans notre projet, nous avons implémenté les étapes du processus de recherche d'information. En effet, le but de notre travail était de retrouver tous les documents qui répondent au besoin d'utilisateur. Plus précisément nous avons étudié et discuté quelque techniques de pondération en RI, et aussi définir en détail les différèrent mesure de similarité dont l'utilisateur peut choisir une de ces méthodes.

CONCLUSION GENERALE

Le travail développé dans ce mémoire, s'inscrit dans le cadre de traitement de grands volumes d'informations que l'on désigne souvent par l'expression « passage à l'échelle ».

L'objectif principal de ce papier se situe dans le cadre de la conception et la réalisation d'un outil pour la recherche d'information. Il s'agit du développement d'un Système de Recherche d'Information (SRI). Le but de ces SRI est de récupérer des documents pertinents répondants à un besoin d'utilisateur exprimé dans une requête avec un temps d'accès optimum.

Comme perspective nous envisageons améliorer notre SRI en incorporant les points suivant :

- Appliquer d'autre modèle hors que le modèle booléen.
- Incorporer une phase sémantique à notre S.R.I
- Ajouter des modules pour le calcul des différents paramètres permettant l'évaluation des SRI comme la précision et le rappel.

Bibliographie

1. **[Mooers, 48]** Mooers C.N. Application of Random Codes to the Gathering of Statistical Information, MIT Master's Thesis, 1948.

2. **[Salton, 1983]** SALTON G, MACGILL M.J, « Introduction to modern information retrieval », McGraw Hill International Book Company, ISBN 0-07-Y66526-5, 1983.

3. **[Salton et al., 1971**] G. Salton, a Comparison between manual and automatic indexing methods. Journal of the American Documentation, 20(1), pp. 6171, 1971.

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

5. **[SALT, 71]** G. Salton, *The SMART Retrieval System: Experiments in Automatic Document Processing*, Prentice-Hall Inc, NJ.1971.

6. **[SALT 90]** G. Salton & C. Buckley. *Improving Retrieval Performance By Relevance Feedback*, Journal of the American Society for Information Science. 1990. 41(4) : 288-297

7. **[Coret , 92]** Coret, A. "Accès à l'information textuelle en français: le cycle exploratoire Amaryllis", Avignon : s.n, Premières Journées FRANCIL, pp. 5-8, 1992.

8. **[Chaudiron, 00]** S. Chaudiron et L. Schmitt, "AMARYLLIS: an evaluation-based program for Text Retrieval", Athens ELRA: s.n, Worshop Proceedings of LREC, pp. 65-68, 2000.
9. **[BLA 85]** Blair, D.C., Maron, M.E., An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Commun. of the ACM*, 28, 1985, pp. 289-299.

10. **[HAR 92]** Harman, D. K. (ed.), NIST Special Publication 500-207: The First Text REtrieval Conference (TREC-1), 1992.

11. **[LAN 68]** Lancaster, F.W., *Evaluation of the MEDLARS Demand Search Service*, National Library of Medicine, Bethesda, Maryland, 1968.

12. **[CLE 67]** Cleverdon, C.W., The Cranfield tests on index language devices. *Aslib Proceedings* 19(6), 173-193, 1967.

13. **[SAL 71]** Salton, G., The SMART Retrieval System. Prentice Hall, Englewood Cliffs, NJ, 1971.

14. **[CROF87]** W.B. Croft, R.H. Thomson, **UR** : A **new** Approach to the design of document retrieval systems, **J.** Am. Soc. Inf. Sci. **38(6)** 1987 **383-404**

Résume

Notre projet de fin d'études consiste à réaliser et développer un système de recherche d'information qui a pour but de faciliter à l'utilisateur de trouver un résultat selon son besoin, de ce fait la recherche d'information est devenu donc un domaine indispensable pour obtenir des données désirées, afin d'automatiser la gestion et la recherche des informations documentaires et se concentre plus précisément dans les différent techniques de pondérations et de similarités

Nous avons adopté le langage orienté objet python pour la réalisation de notre application.

Abstract

Our graduation project aims to produce and develop a system for information retrieval which facilitate to the user to find a result according to his need, therefore research aims to information has become an area necessary to obtain the desired data to automate the management and research of documentary information and focuses more specifically in different of weighing and technical similarities.

We adopted the object-oriented language Python to realize our application.